

Editor: Dana Simian

Proceedings of
The Sixth International Students
Conference on Informatics

ICDD

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

Romania, Sibiu, May 19-21, 2016

Editor DANA SIMIAN

**IMAGINATION, CREATIVITY,
DESIGN, DEVELOPMENT**

Proceedings of International Students
Conference on Informatics
ICDD

**May 19th – 21th, 2016
Sibiu, Romania**

“Lucian Blaga” University of Sibiu

“Lucian Blaga” University, Sibiu, 2016

Editor Dana Simian

All papers in this volume were peer review by two independent reviewers

ISSN 2069-964X

Associated Editor Laura Florentina Stoica

Proceedings of International Students Conference on
Informatics ICDD
May 19th – 21th, 2016, Sibiu, Romania

Copyright @ 2016 All rights reserved to editors and authors

Preface

This volume contains refereed papers presented within the Sixth International Students' Conference on Informatics - ICDD – 2016, which was held between May 19th – 21th, at the Faculty of Sciences, University "Lucian Blaga" of Sibiu, Romania.

The conference is addressed to bachelor and master level students. Conference aim is to bring together students from different universities from all over the world to discuss and present their researches on informatics and related fields (like computational algebra, numerical calculus, bioinformatics, etc) and their original results. Specific topics of the conference included but were not restricted to: Algorithms and data structures, Graph theory and applications, Formal languages and compilers, Cryptography, Modelling and simulation, Computer programming, Computer vision, Computer graphics, Game design, Data mining, Distributed computing, Artificial Intelligence, Service oriented applications, Networking, Grid computing, Mobile operating systems, Scientific computing, Software engineering, Bioinformatics, Robotics, Computer Architecture, Evolutionary Computing, Multimedia Systems, Internet Communication and Technologies, Web Applications.

The talks were delivered by students from 5 countries (Bulgaria, Estonia, Romania, Russia and Serbia).

We thank all the participants for their interesting talks and discussions. We also thank the members of the scientific committee for their help in reviewing the submitted papers and for their contributions to the scientific success of the conference.

The conference was organized with the support of the following sponsors (in alphabetical order): CEPROCS, COSSOFTWARE, EBS, GSD, IQUEST, MARQUARDT, OMERON, PAN FOOD, ROPARDO, TOP TECH, VISMA.

May 2016

Dana Simian
Conference chairman

Motto:

“There are no limits, only your imagination”

Proceedings of International Students Conference on Informatics, ICDD – 2016
IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT
Sibiu, Romania

Editor *Prof. PhD Dana Simian*

Scientific committee

- Prof. PhD. Grigore Albeanu - University of Bucharest, Romania
- Prof. PhD. Kiril Alexiev - Bulgarian Academy of Sciences, Bulgaria
- Prof. PhD. Florian Boian - "Babes-Bolyai" University of Cluj-Napoca, Romania
- Prof. PhD. Alina Barbulescu - Higher College of Technology, Sharjah, UAE
- Prof. PhD. Dan Cristea - "Alexandru Ioan Cuza" University of Iasi, Romania
- Prof. PhD. Ioan Dzitac - "Aurel Vlaicu" University from Arad, Romania
- Prof. PhD. Oleksandr Dorokhov - Kharkiv National University of Economics, Ukraine
- Prof. PhD. Heiner Gonska - Duissburg-Essen University, Germany
- Prof. PhD. Gheorghe Grigoras - "Alexandru Ioan Cuza" University of Iasi, Romania
- Prof. PhD. Katalina Grigorova - University of Ruse, Bulgaria
- Prof. PhD. Ion Iancu - University of Craiova, Romania
- Prof. PhD. János Karsai - University of Szeged, Hungary
- Prof. PhD. Milena Lazarova - Technical University of Sofia, Bulgaria
- Prof. PhD. Mariana Nagy - "Aurel Vlaicu" University from Arad, Romania
- Prof. PhD. Ing. Cristian Oară - Polytechnic University of Bucharest, Romania
- Prof. PhD. Bazil Parv - "Babes-Bolyai" University of Cluj-Napoca, Romania
- Prof. PhD. Petrica Pop-Sitar - North University of Baia Mare, Romania
- Prof. PhD. Klaus Bruno Schebesch - "Vasile Goldis" University, Arad, Romania
- Prof. PhD. Ing. Ioan Silea - Polytechnic University of Timisoara, Romania
- Prof. PhD. Dana Simian - "Lucian Blaga" University of Sibiu, Romania
- Prof. PhD. Gheorghe Stefanescu - University of Bucharest, Romania
- Prof. PhD. Mihai Talmaciu - University of Bacau, Romania
- Prof. PhD. Milan Tuba - Megatrend University of Belgrade, Serbia
- Prof. PhD. Dan Eugen Ulmet - University of Applied Sciences Esslingen, Germany
- Assoc. Prof. PhD. Vsevolod Arnaut - Moldova State University, Republic of Moldova
- Assoc. Prof. PhD. Vasile Aurel Căus - University of Oradea, Romania
- Assoc. Prof. PhD. Ioana Chiorean - "Babes-Bolyai" University of Cluj-Napoca, Romania
- Assoc. Prof. PhD Daniela Danciulescu - University of Craiova, Romania
- Assoc. Prof. PhD. Stefka Fidanova - Bulgarian Academy of Sciences, Bulgaria
- Assoc. Prof. PhD. Victoria Iordan - West University of Timisoara, Romania
- Assoc. Prof. PhD. Gabriela Moise - Petroleum-Gas University of Ploiesti, Romania
- Assoc. Prof. PhD. Antoanela Naaji - "Vasile Goldis" Western University of Arad, Romania
- Assoc. Prof. PhD. Sorin Noaghi - University of Petrosani, Romania
- Assoc. Prof. PhD. Corina Rotar - "1 December 1918" University of Alba Iulia, Romania
- Assoc. Prof. PhD. Livia Sangeorzan - Transilvania University of Brasov, Romania
- Assoc. Prof. PhD. Sofia Visa - The College of Wooster, United States

- Lecturer PhD. Alexandru Bobes - Ovidius University of Constanta, Romania
- Lecturer PhD. Mihaela Ciortea - "1 December 1918" University of Alba Iulia, Romania
- Lecturer PhD. Ralf Fabian - "Lucian Blaga" University of Sibiu, Romania
- Lecturer PhD. Daniel Hunyadi - "Lucian Blaga" University of Sibiu, Romania
- Lecturer PhD. Alina Pitic - "Lucian Blaga" University of Sibiu, Romania
- Lecturer PhD. Florin Stoica - "Lucian Blaga" University of Sibiu, Romania
- Lecturer PhD. Laura Florentina Stoica - "Lucian Blaga" University of Sibiu, Romania
- Lecturer PhD. Anca Vasilescu - Transilvania University of Brasov, Romania

Contents:

HoloMat - View 3D Graphs in MatLab into a hologram	13
Petru Amortoae, Maria – Alexandra Craiu	
Coordinator: Laura Stoica	
An experiment on running surveillance software as a unikernel application	21
Paul – Gheorghe Barbu	
Scytale encryption in modern days	27
István Bialkó	
Coordinator: Dan Dumitru	
Foody, you are what you eat	35
Cristian Bota, Adrian A. Bărbos	
Coordinator: Camelia-M. Pinte, Hajdu Măcelaru Mara	
Arcadia: Socialize and Learn in Virtual Reality	42
Petrică Bota, Robert Săndică	
Coordinator: Dana Simian	
Remote monitoring applications through Windows APIs	47
Momcilo Brajic	
Smart city and smart transportation. Bucharest case study	58
Crina Cristea, Alexandru Daniel, Darko Shuleski, Alexandru Birsan	
Coordinator: Ioan Radu	
Structuring business process models	68
Mykhailo Dorokhov	
Coordinator: Marlon Dumas	
Mark It	76
Andrei Dragomir, Ioan Grozea	
Coordinator: Delilah Florea	
Fireball	83
Bogdan Georgescu, Ciprian Petrean	
Coordinator: Ramona Humeniuc	
PhotoMaster	89
Radu Handolescu, Alexandru-Nicolae Motoc	
Coordinators: Nicolae Steavu, Cristina-Elena Steavu	

What's What android application	95
Darius Hațegan Coordinator: Dana Simian	
3D Mobile cross platform game	99
Dragos Hodina Coordinator: Dana Simian	
Dodgy Ninja	109
Flavius Holerga, Andrei Milik, Inocențiu Moga Coordinator: Gabriela Florea	
OpenGL Game Engine	113
Svetlozar Iliev Coordinator: Katalina Grigorova	
Smart Backpack	118
Vlad Ionescu, Emanuel Giurgiu, Vlad Murzea, Sebastian Vidrea Coordinators: Livia Sangeorzean	
Angry Driver	124
Marius Daniel Marinescu, Mihail Irinel Preda Coordinators: Martin Elena Liliana, Dinu Alina, Raicu Viorica	
Test Generator Tool for Embedded Testing	132
Fineas - Iacob Mureșan Coordinator: Dana Simian	
StarDust	141
Alexandru Oltean, Alin Gabriel Prăvariu Coordinators: Carmen Popescu, Georgeta Preda	
Roev - Event Planner	145
Theodor-Ovidiu Palamaru Coordinator: Dana Simian	
Pseudocode Learning Companion	154
Cătălin Popa Coordinator: Cornelia Ignat	

Multicriteria alternative choice in the conditions of uncertainty on the basis of the fuzzy preference relations	161
Maria Sokolova Coordinator: Vladimir Chernov	
Bluetooth Rover	166
Radu Stoican Coordinator: Oana Georgescu	
Embedded application controlled by a mobile device	173
Marius Toma Coordinators: Dana Simian	
ChessTime – a simple chess clock for Android	181
Dimitar Trifonov Coordinators: Katalina Grigorova	
Parameters tuning for support vector machines	186
Eva Tuba Coordinators: Milan Tuba	
Optimization of Otsu's method for multilevel image thresholding by grid search	195
Una Tuba Coordinators: Milan Tuba	
List of authors	204
Sponsors	210

HoloMat - View 3D Graphs in MatLab into a hologram

Petru Amortoae, Maria – Alexandra Craiu
Coordinator Professor: Laura Stoica

Abstract

HoloMat is an application for desktop developed in Matlab. The aim of the article is to present an original application in order to project a 3D hologram through a transparent pyramid using a smartphone. In this article we described our application that allows us to transform a 3D graphic of Matlab to be transformed in order to be designed as a hologram. The application has a user- friendly interface with the capacity to customize the graphics created in Matlab changing the colormap, shading, delay between frames and number of faces. Also this application has 5 different functions and some predefined animations.

1 Introduction

1.1 Short history about holograms

Authentic three dimensional visualization technologies have been fantasied about over the past half century. This introduction talks about the tantalizing prospects of finally realizing holographic video screens and the history.

1.1.1 Looking Back

Dennis Gabor, a Hungarian scientist working in the UK, presented the foundational concepts of holography in a patent and a series of papers written between 1948 and 1951 that were aimed at microscopy [1]. These introduced the notion of saving the 3D info related as a 2D interference pattern, including amplitude and period information, which may afterwards be reconstructed through illumination. Holography stayed somewhat obscure owing to its dependence on coherent light sources. Nonetheless, things changed radically after the creation of the laser in 1960 when the field had a revival with the upsurge in research and development. Lieth and Upatnieks reached a crucial milestone with the off-axis transmission hologram. At about exactly the same time, Denisjuk initiated the reflection hologram. Dennis Gabor was awarded the Physics Nobel Prize in 1971.

Today, the principal applications of holograms that are static are for branding, bank notes, passports etc. in authentication and security Holographic imaging is mostly consigned to specialist optical elements or artistic exhibitions.

1.1.2 Underground Research

Unfortunately, holography hasn't lived up to expectations, which were largely driven by the realms of science fiction fantasy following the iconic Star Wars film in 1977. While inactive holograms were popular in the 70s and 80s, it soon became clear that stationary holograms needed to go beyond the lab whereas dynamic holographic displays were "in a land far, far away." Holographic screen research was mostly performed by military contractors or high-end academic research facilities.

Regarding 3D displays, 3D TV and theatre have come and gone in recent years. This is because conventional stereo 3D is based on glasses or 'glasses-free' lenticular arrays or parallax barriers, leading to a poor 3D experience because it is really just an illusion created by twin-2D images aimed at either eye. This leads to all kinds of problems including dizziness and nausea and is simply unacceptable to the human brain.

1.1.3 Data explosion and 3D content

The recent proliferation of 3D content has created a need for such screens because of the explosion of 3D data from many sources including geographic data, medical scans, computer design, simulation, low cost depth scanners and video gaming. It has proven quite hard to view this content in real 3D, whereas it is simple to generate 3D content.

A true 3D display should have the scene or object in real space (floating in mid-air), enabling the observer to look around items and see them from a somewhat different outlook, as they'd in actual life. This contributes to a more comfortable and naturalistic viewing experience without all the problems of stereo 3D. According to the scientific literature, such a display can only be realised with holographic or similar light field techniques.

1.2 Motivation

The idea for our application appeared some time ago after we were watching at the movie Avatar by James Cameron [2]. The next sequence from the movie was the starting point for the application.



Fig. 1: Avatar movie caption by James Cameron [2]

In this image we see a fantastic 3D hologram which we thought we can reproduce them in reality in some technical fields like medical or industrial. After some research we have found on internet some blueprints on how to create a pyramid to project any hologram.

1.3 Structure of the paper

The paper contents four sections. In introduction we present a short history about holograms. In the second section we describe how it's necessary to be prepared the image to its design as hologram. In the third section we describe our application interface creating in Matlab and then for an example we explain the process of transform Matlab graph into an image for projection. In the last section there are conclusions and future works.

2 General description

2.1 Creating the projection structure

First of all, to create the projection we need to define a black background to make the hologram show up correctly. For this we defined the Matlab figure background to be black. The entire area need to be divided in four sections like on the next example.

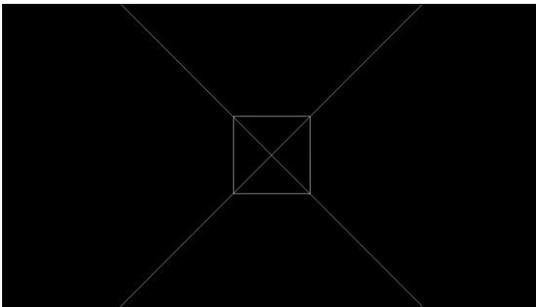


Fig. 2a: Black background

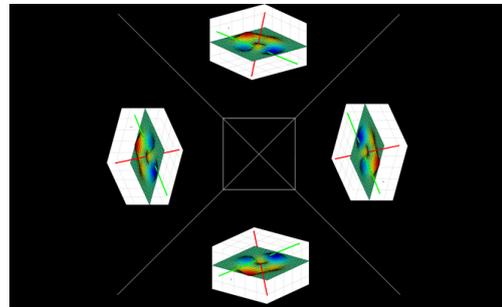


Fig. 2b: Image rotation inside subplots

To define in Matlab these four sections we have created a 3x3 matrix from subplots. We represented in the subplots 2, 4, 6 and 8 the forms we will animate.

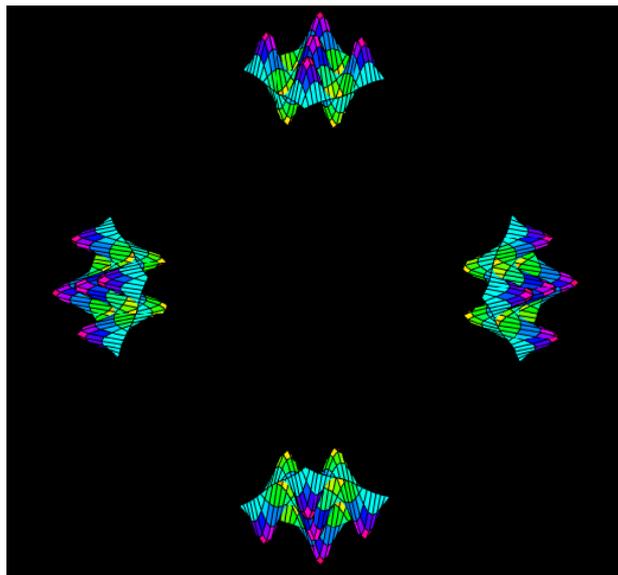


Fig. 3: Gif file with an animated surface

The form from the plot number 6 will be rotated 90 degrees to the right, from the plot number 8 will be rotated 180 degrees to the right and the one from the subplot number 4 will be rotated 270 degrees to the right. In figure 4 we noted all the subplots with Arabic number and we chose to show the image rotation using the 4 geographical coordinates. The red line means how the image rotates toward the side marked in red.

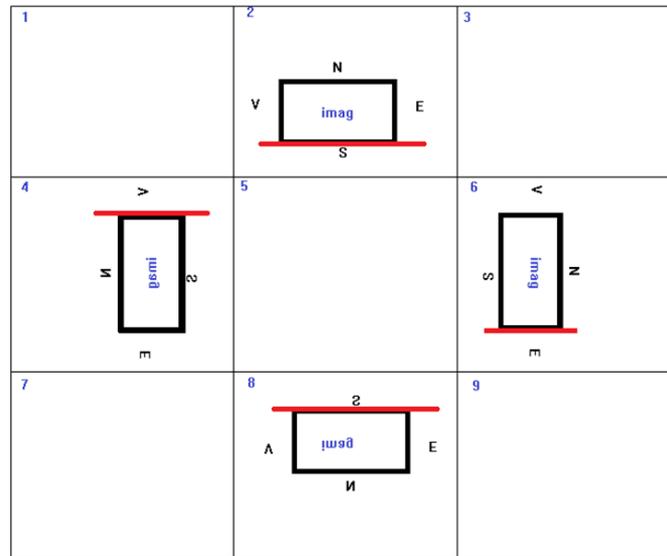


Fig. 4: Image rotation inside the 4 subplots

2.2 Hologram projector

There are three key types of projector used to create a hologram; single plane, pyramid and multi-planar, each one creating a very different effect.

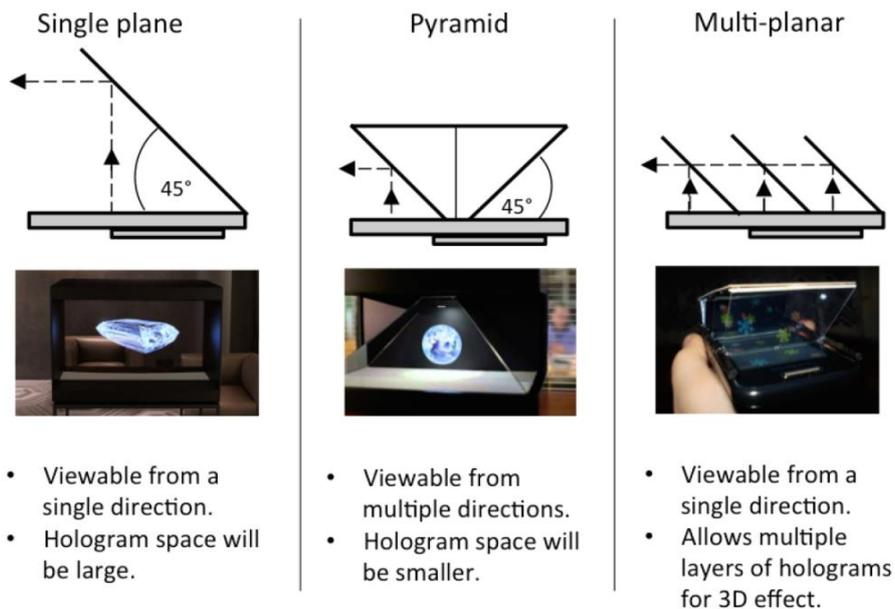


Fig. 5: Types of projectors [3]

For this project we have chosen to use the pyramid because we can see the animation at 360 degrees offering a much entertaining experience.

It's one of those concepts straight out of a science fiction movie like Avatar - holographic projectors that show images and video in 3D. But what if I told you that in only 3 simple steps, you could make your very own holographic imaging? Does it sound too good to be true??? Well, it's not! To display correctly the hologram we need to follow some guidelines [4]. First, the pyramid needs to have a 45 degree with the horizontal plane. Secondly, to create the pyramid, we need four trapezoid shapes with these dimensions: for iPads or any other tablet with the diagonal between 7 and 12 inch they need to have 2cm by 7 cm by 12 cm and for smartphones with the diagonal between 3.5 inch and 6.5 inch they need to have 1cm by 3.5cm by 6 cm.

The concept of a hologram projector may sound quite complex, but using some simple optical principles, we can make one with the following supplies: a graph paper, a CD case, tape or super glue, a pen, scissors and a glass cutter.

Cut 4 identical trapezoid shapes from the clear plastic. To do this, cut out a basic trapezoid shape from my graph paper (1cm by 3.5cm by 6cm or 2cm by 7cm by 12cm). Trace the shape on the CD case, and then cut the four pieces with the glass cutter. After that tape the four plastic cut-outs together, place the 3D construction over your smartphone, load up one of our gifs and press play.

3 Application description HoloMat

3.1 GUI interface for HoloMat

The GUI interface for HoloMat is very user-friendly, attractive, clear and concise. In this section we will present every module of our application.

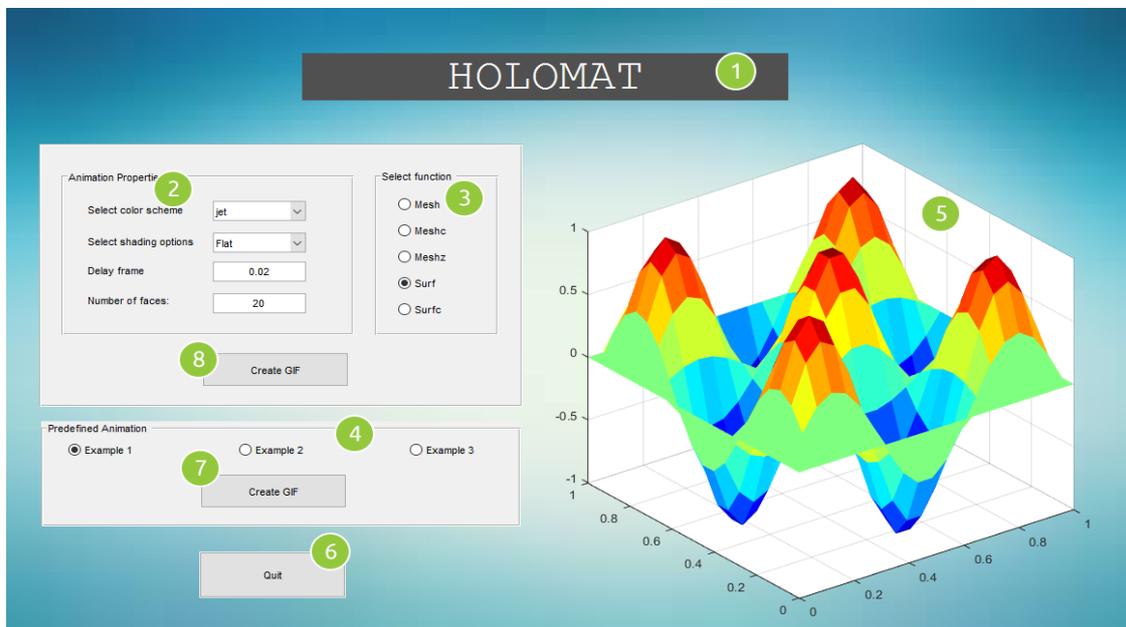


Fig. 6: Holomat interface

We noted with (1) the *Application's* title.

The *Animation Properties panel* noted with (2) contains:

- a) **color scheme**: parula, jet, hsv, hot, cool, spring, summer, autumn, lines, colorcube, prism, flag, white
- b) **Shading option** (just for surf and surfc functions): faceted, flat, interp
- c) **Delay between frames** (can be any number in seconds)
- d) **Number of faces**.

The *Select Function panel* noted with (3) contains 5 functions:

- Mesh, Meshc, Meshz, Surf, Surfz

The *Predefined Animation* noted with (4) contains the 3 example:

- Example 1 – we draw a sphere and make sure that the axis box keeps its proportions correctly and is rotate from our view angle [5].
- Example 2 – we draw an animation using 3d Polar Coordinates [6].
- Example 3 – we draw a “snow cone” [7].

The *Axes system* noted with (5) show us the 3D graph with the selected options.

The *Quit Button* noted with (6) close the application window.

The *Create gif* buttons noted with (7) and (8) create the animate image with the selection options.

Further we present the steps of transforming a 3D graphics in Matlab to be designed as a hologram.

3.2 Example of hologram projection with a Matlab generated gif file

The first step which we perform before representation the Matlab graphic, it's to select the properties for drawing image.

From the *Animation Properties panel* we select the color scheme, shading options (just for *surf* and *surfc* functions) and after we can introduce a value for the delay between frames and for the number of faces. After that we select the function (*mesh*, *meshc*, *meshz*, *surf*, *surfc*) from the *Select function panel* (and it will be created and displayed automatically the animation on the axes system from the right).

For begin we select the type of colormap *hsv* from *Color scheme* popup menu, after that we select the faceted from shading options, then we set delay frame to 0.02 and number of faces to 15. From the *Select function* panel we choose for graphic representation the radio button *surf* and finally just click on *Create gif* button. In this moment begins the process of creating a gif file. Within several seconds the gif file will be created, saved in *google drive* and shared with your phone connected to the same google drive account.

Steps for creating the gif file are:

- 1) Hide the window until the gif is ready;
 - 2) Changing the color background in black for all subplots and hide axis and grid;
 - 3) Image rotation for the 4 subplots desired
- Here we have 2 situations: First, when we save all frames to a static camera view; second, when we save all frames to a dynamic rotate object view (rotate y-axes or z-axes)

```
% example for the subplot with number 8 (static camera view)
h8 = subplot(3,3,8);
% code for 3D graphic representation in Matlab
set(h8, 'color', 'k'); % background color
axis off; grid off; % disable axis and grid
```

```
% Captures all frames for animation (step 4)
imshow(flipud(rgb));% rotate image for the 8 subplot

% example for the subplot with number 8 (rotate object towards z-Axes)
% code for 3D graphic representation in Matlab
hs=surf(x,y,z);rotate(hs, [0 0 1], j, [0 0 0]);
% Captures all frames for animation (step 4)
imshow(flipud(rgb));% rotate image for the 8 subplot
```

4) Captures all frames for animation

```
f = getframe; % Screen shot captures
[data_im,map] = frame2im(f); % Return the image datas

if isempty(map) %Truecolor system
    rgb = data_im;
else %Indexed system
    rgb = ind2rgb(data_im,map); %Convert image data
end
imshow(rgb); % displays image rgb in a Handle Graphics figure
```

5) Saving the gif file by capturing the all subplots

```
frame = getframe(gcf);
im = frame2im(frame);
[imind,cm] = rgb2ind(im,256);

if j == 1;
    imwrite(imind,cm, 'surf.gif', 'gif', 'Loopcount', inf, 'DelayTime', sec);
else
    imwrite(imind,cm, 'surf.gif', 'gif', 'WriteMode', ...
        'append', 'DelayTime', sec);
end
```

6) When the image is saved in the screen we see a message box which tells us that the gif file was saved.

Now we can send the created gif file and we can open from smartphone. For projection, it needs to hold the pyramid over the center of the screen and now we can watch through the side of the pyramid. This is the final result:

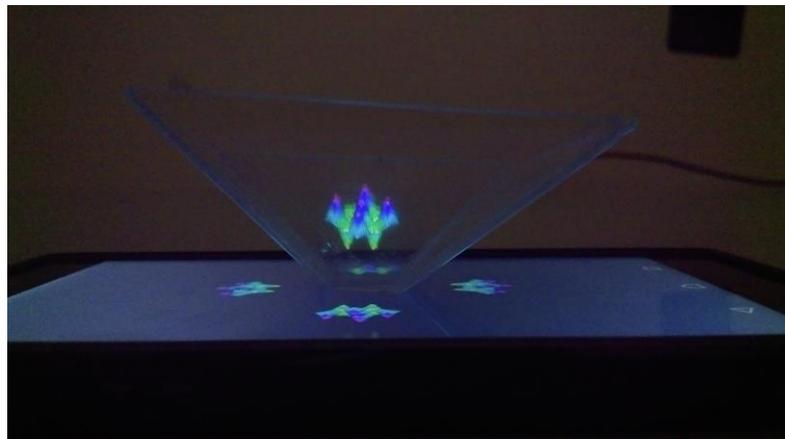


Fig. 7: Hologram example

4 Conclusions

In our work we create GUI application which allow us to create a gif image with the 3D graphs from Matlab and then to project the graph to design a hologram graph using the pyramid. Since Matlab is currently used in various fields such as engineering, medicine, biology, we believe that holographic projection would make it easier to view and interpret 3D images on the four surfaces of the pyramid than in two dimensions. For moment our application have the all graph predefined in Matlab code but for future work we intend to build a GUI interface which will allow reading images from the user.

References

- [1] History of holographic technology - <http://www.holoxica.com/Blog/a-brief-history-of-holographic-technology.html>
- [2] Avatar movie image - http://sciencefictionstarsystem.wikia.com/wiki/HUD_Touchscreen
- [3] Pepper's Ghost - <http://www.barques.co.uk/peppers-ghost-and-the-hunt-for-the-hologram/>
- [4] Create hologram projector - <http://smartphones.wonderhowto.com/how-to/create-easy-hologram-projector-for-your-smartphone-just-ten-minutes-0163589/>
- [5] 3D simple animation - <http://www.matrixlab-examples.com/simple-animation-3d.html>
- [6] 3D polar coordinates - <http://matlab.ly/3d-polar-coordinates/>
- [7] Snow cone: - <http://math.umn.edu/~rober005/math5385/matlabinfo/surfplot.html>

Petru Amortoae
Lucian Blaga University
The Faculty of Sciences
5-7, Ion Ratiu Street, Sibiu, 55001
ROMANIA
E-mail: petru.amortoae@gmail.com

Maria – Alexandra Craiu
Lucian Blaga University
The Faculty of Sciences
5-7, Ion Ratiu Street, Sibiu, 55001
ROMANIA
E-mail: alexandra_mariaa27@yahoo.com

An experiment on running surveillance software as a unikernel application

Barbu Paul - Gheorghe

Abstract

The aim of this paper is to explore the recent shift from monolithic applications to using micro-services in a virtualized environment for serving clients' needs. During the experiment I created an application that runs as a unikernel and manages video streams from an IP camera, thus running IP camera surveillance software on bare metal hardware and on type one hypervisors.

1 Introduction

Since software complexity is increasing with time, teams of software developers and maintainers are trying to find new ways of keeping this complexity at manageable levels. As a result of this effort are applications that can be run as the only entity on a system, the resulting unnecessary parts of the operating systems (eg. virtual memory, process schedulers) are removed and made optional, only the drivers remain, these are necessary in order to still be able to use devices such as disks (be them Solid State Drives or Hard Disk Drives), network interface cards, different input and output modules (USB, video, etc.). Once such effort is the *rumpkernel*¹ whose aim is to run the NetBSD² kernel in an userspace process. This is done with the help of anykernels³, which are developed by the NetBSD team, a project started in order to develop device drivers directly in an userspace process⁴, without needing a virtual machine such as VirtualBox⁵ or QEMU⁶. Later, with the emergence of the unikernels⁷ this project has evolved into *rumprun*, which allows the programmer to design an application as if it were running directly on hardware. Based on this, I fiddled with the idea of creating surveillance software that runs on bare-metal hardware. In this paper I present the successful materialization of this idea. The paper presents and compares related work in the second section, then proceeds to list the requirements that drove the implementation of my experiment together with its details and its architecture in the third section. Lastly the conclusions and possible improvements of the application are shown.

¹<http://rumpkernel.org/>

²<https://www.netbsd.org/>

³<https://wiki.netbsd.org/rumpkernel/>

⁴<http://rumpkernel.org/misc/usenix-login-2014/login.1410-03-kantee.pdf>

⁵<https://www.virtualbox.org/>

⁶http://wiki.qemu.org/Main_Page

⁷HaLVM, MirageOS and others: <https://en.wikipedia.org/wiki/Unikernel>

2 Related work & software

As far as surveillance software goes, there are plenty of alternatives ⁸, but no such alternative running as a unikernel exists ⁹. So this section will proceed with a comparison of unikernel projects.

Unikernels are single address space systems which bundle up an application and a selection of system components relevant for that particular application into a single image. The lightweight image can then be run on the cloud or on hardware, depending on what driver components are available for that particular unikernel system.

One approach for unikernel implementations is to use a single language such as Erlang, Haskell or OCaml and a clean-slate implementation of everything (drivers, network layer, etc).[1]

2.1 HaLVM

The Haskell Lightweight Virtual Machine describes itself as "a port of the Glasgow Haskell Compiler tool suite to enable developers to write high-level, lightweight virtual machines that can run directly on the Xen hypervisor". ¹⁰ This essentially means that one can write code in Haskell, compile it, and run it in a hypervised environment such as Xen. When targeted at the proper applications, the combination of Haskell and the type-safety it provides together with the isolation provided by Xen can yield profitable results.¹¹

2.2 MirageOS

"MirageOS is a library operating system¹² that constructs unikernels for secure, high-performance network applications across a variety of cloud computing and mobile platforms. Code can be developed on a normal operating system such as Linux or MacOS X, and then compiled into a fully-standalone, specialised unikernel that runs under the Xen hypervisor."¹³ This project is similar to the HaLVM, the main difference is the choice of language used for developing the application.

2.3 Rumprun

While the other unikernel alternatives require writing the software in a specific language like Haskell or OCaml, the Rumprun unikernel places no such requirement on the programmer, it is language agnostic. The only requirement of the Rumprun platform is that the code should run on POSIX-like systems, generally speaking this would include Linux, the BSD family and OS X ¹⁴. Another advantage of the Rumprun platform would be the fact that over 99% of the code in the Rumprun stack is used unmodified from NetBSD, hence providing high quality software and code reuse (including drivers, networking stack and security updates).[1]

3 Rump Camera

Keeping in mind that software should run in the real world, solve real problems and should meet users' needs I designed the *Rump Camera* project with some realistic requirements in mind:

1. The video format should be easily recognized by video-players
2. The total size of recorded videos should be limited, in order to avoid filling the storage medium

⁸https://en.wikipedia.org/wiki/IP_camera#List_of_IP_camera_software

⁹as far as my knowledge holds

¹⁰<https://github.com/GaloisInc/HaLVM#1-what-is-the-halvm>

¹¹<http://galois.com/>

¹²Another common name for "unikernel"

¹³<https://mirage.io/>

¹⁴<https://github.com/rumpkernel/rumprun-packages>

3. Each video should have a fixed specified length, in order to be able to create, for example, daily videos
4. The end user should have access to the surveillance videos via an easy-to-use interface

The first requirement is satisfied by using the MJPEG¹⁵ format, which is what the majority of IP cameras support, so I chose this format for storing the videos, too. This choice involves some trade offs in terms of memory, since the MJPEG format features only intra-frame compression, not inter-frame compression seen in the more advanced video codecs. Also the simplicity of the codec is chosen over its lack of standardization.

The POSIX-like nature of the applications written for the Rumprun unikernel makes it easy to achieve the second requirement, while encoding the video on disk, the software also computes its size and by storing meta-data about all created videos, a total size is easily calculated. When this total size is reached¹⁶ the oldest videos are deleted to make space for the new one.

In code, this is achieved by using a simple FIFO queue¹⁷ in the `manager.c` and `manager.h` files, for more details, the responsible functions are:

- `_manager_remove_oldest`: which removes the oldest video in the queue from its tail, thus making place for a new one at the head of the list
- `_manager_clean_size`: responsible for calling the `_manager_remove_oldest` function when appropriate, i.e: when the remaining space would not suffice for storing another video

The length of each video is configurable at compile time in the `config.h` file, this comes useful for different clients since some of them may want to have hourly videos and some may want to have daily ones.

Configuration at compile time was chosen for two reasons. The first was to keep the software complexity at a minimum and the second, a practical one, because after deployment, a system like this is unlikely to be changed, it needs to be robust and long-running, hence introducing run-time configuration makes little sense. Also this approach to configuration is not new, the `dwm`¹⁸ project in the Linux world is well known for being configurable exclusively at compile time.

The `config.h` file also provides other configuration options besides the maximum length of the video, such as the file path, file name of the created videos and the maximum number of videos, which determines the queue's maximum length.

The most important requirement of all is the last one. This last requirement is the most important because it allows the user to have access to the video recordings, without such a feature the whole application would be rendered useless. As such I've chosen to use a third party library, thus illustrating the flexibility provided by the Rumprun unikernel and the fact that we can have multi-threaded applications running in an unikernel.

The interface with which the user interacts is a web-based interface (see figure 1), so the *Rump Camera* project also acts as a web server with the help of the `libmicrohttpd`¹⁹ library. Running as a unikernel, the application cannot link to a dynamically loaded library since filesystems may not be available, thus there's always a need for statically linking libraries used in unikernel applications, this is also a practical constraint since the environment should be self-contained when booting of off bare-metal hardware and should not be dependent on run-time components.

So with the help of this library the application can serve web requests to users, they should only connect to it (using the IP configured on the Xen or Virtual Machine where the unikernel will run) in order to get a listing of the available video files, then by clicking on a file, the download will start. This type of interface is inspired from how the FTP²⁰ URLs are rendered in the browser and was chosen

¹⁵https://en.wikipedia.org/wiki/Motion_JPEG#M-JPEG_over_HTTP

¹⁶in reality the algorithm makes sure another video can be stored on the remaining space by estimating its size

¹⁷with its length configurable by the programmer when compiling the application in the `config.h` file

¹⁸dynamic window manager: <http://dwm.suckless.org/>

¹⁹<https://www.gnu.org/software/libmicrohttpd/>

²⁰File Transfer Protocol

because of the omni-presence of web-browsers despite the fact that I have to add third party code to the application. Still, if it weren't for this approach, I'd have to implement a file transfer protocol in order to be able to get access to the surveillance videos, thus increasing code size, reinventing the wheel and gaining nothing in terms of code complexity.

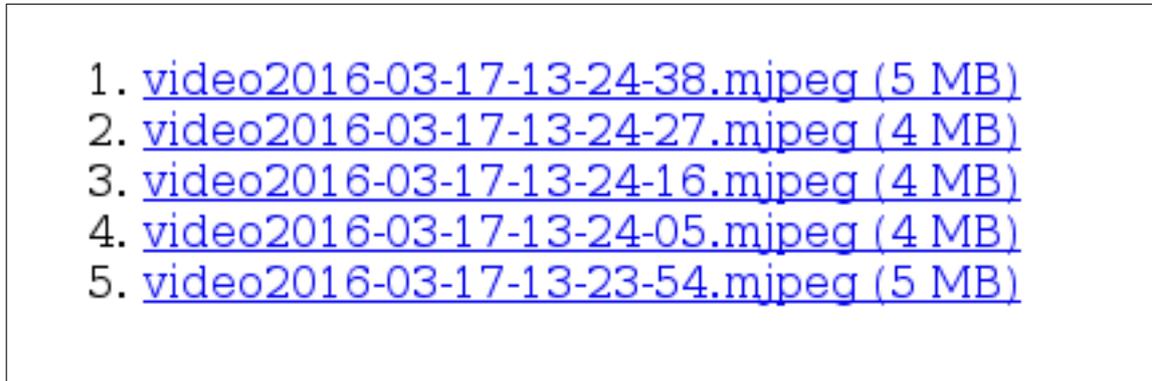


Figure 1: A screenshot of the application's simple interface

The user interface is coded in `httpd.c` and `httpd.h`:

- `httpd_serve`: This is the request dispatcher, or router, if the request path matches a video file name, then that file is sent to the client for downloading, otherwise the file listing seen in figure 1 is rendered
- `_httpd_list_files` together with `_sprint_video_list` creates and renders the set of files available for download, the rendering settings can be configured in the `config.h` file
- `_httpd_try_download` uses the `sendfile`²¹ syscall to optimize the transfer of files between two end-points.

I shall note that the video capture part of the application is independent of the `libmicrohttpd` library (this can be seen in figure 2) and is manually coded in `video.c` and `video.h`, without having any dependencies on a third party library.

This module of the application handles the connection, authentication, retrieval and storage of data from an IP camera. It can be considered the central piece of the experiment. The credentials used for authentication to the camera should be stored in the `credentials.h` file (so they should be made available at compile time), similarly to the other parts of the application, the address and port on which to connect to the video device can be configured in the `config.h` file.

Figure 2 describes the architecture of the system as a diagram.

²¹via the `libmicrohttpd` library

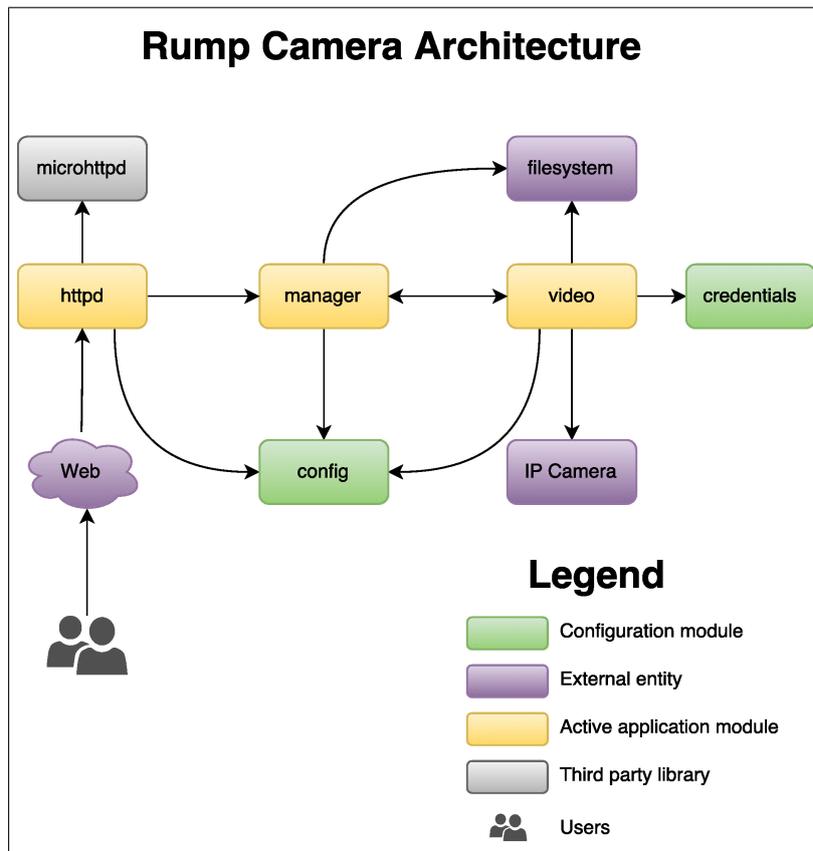


Figure 2: The Rump Camera Architecture

It can be seen that the users have access to the application solely through the web interface, which is implemented by the `httpd` module. In turn, this module is supported by the third party library `libmicrohttpd` which abstracts away the details of creating, sending and replying to HTTP requests. This library was chosen for its compactness and because it has an extended test suite. The next module, the `manager`, is the layer that provides the order in which the videos were created (for the interface) and decides what video to delete in the event that the disk becomes full or the maximum number of videos created has been reached. The `manager` module is of course in strong relation with the `video` module, which has the responsibility to connect to the IP camera (possibly by using authentication) and to store the captured videos on disk.

4 Future work and conclusions

Up to this point the main focus of the experiment was simplicity, both in code and in user experience. So as far as improvements are concerned, the interface of the application can be enhanced by making it more user friendly, less "brutal". Also the resulting video may be post-processed in order to include a time-stamp of the capture, while this post-processing may be made in a parallel fashion (with multiple unikernels working on different sets of images, similarly to processes in a normal operating system) it would certainly add complexity into the system.

A downside of static linkage, required by the nature of unikernel applications, are the more stringent rules that apply when linking a library provided under the GPL license [6], a discussion that quickly escalates into a legal one ²². Consequently, the programmer has to carefully choose his libraries when developing for such an environment where dynamic linking is not available.

²²https://en.wikipedia.org/wiki/GPL_linking_exception

Having a single application that runs on a single (low-power) device, with the unnecessary parts removed out of the kernel (such as the scheduler) and with only the strictly necessary parts included - such as the drivers), I believe programmers can write applications that are more environment friendly, more modular and more robust, being focused on a single, specific task. These modular applications can then be chained together into a system that resembles software pipe lines or functional composition.

Running these unikernel applications in a virtual environment (like Xen) would also be beneficial to our surrounding medium since the applications would be started when needed and stopped at the proper times, thus consuming less energy. [4] [5]

This paper successfully demonstrates that surveillance software for IP cameras (build out of real requirements) can be written to run as a unikernel with minimal footprint and dependencies. Such an approach is well suited to this kind of task since the boot-up times are much faster due to the reduced code size and, as a consequence, the down-times are small, too. Also when security is of concern, the unikernel approach to developing applications can help since the attack surface is greatly reduced.

Of course this approach is not well suited for fast-paced mediums where the software configuration changes often, since this would mean too many re-compilations, also this applies for more complex and feature-full software, because adding features to a single unikernel application doesn't scale well, multiple applications chained together in a computational pipe-line shall be used instead.

The source code of the application can be freely accessed at: <https://github.com/paulbarbu/rump-camera>

References

- [1] [github.com/rumpkernel](https://github.com/rumpkernel/wiki/wiki/Info%3A-Comparison-of-rump-kernels-with-similar-technologies), Info: Comparison of rump kernels with similar technologies, <https://github.com/rumpkernel/wiki/wiki/Info%3A-Comparison-of-rump-kernels-with-similar-technologies>, Accessed 15th of March 2016
- [2] A. Kantee, J. Cormac, Rump Kernels No OS? No Problem!, http://rumpkernel.org/misc/usenix-login-2014/login_1410_03_kantee.pdf, October 2014, Accessed 15th of March 2016
- [3] [wikipedia.org](https://en.wikipedia.org/wiki/Unikernel), Unikernel, <https://en.wikipedia.org/wiki/Unikernel>, Accessed 14th of March 2016
- [4] Silicon Valley Leadership Group, Accenture, Data Centre Energy Forecast Report-Final Report, July 2008
- [5] Y. Jin, Y. Wen, Q. Chen, Energy Efficiency and Server Virtualization in Data Centers: An Empirical Investigation, March 2012
- [6] GNU General Public License, <http://www.gnu.org/licenses/gpl-3.0.en.html>, Version 3, 29 June 2007, Accessed 17th of March 2016

Barbu Paul - Gheorghe "Lucian Blaga"
University of Sibiu
Department of Computer Science and
Electrical Engineering 10, Victoriei Bd.,
Sibiu, 550024, Romania
ROMANIA
E-mail: barbu.paul.gheorghe@gmail.com

Scytale encryption in modern days

István Bialkó
Coordinator Professor: Dan Dumitru

Abstract

Reviving the ancient Greek encryption method (using a baton and a strip of parchment), with a slight modification. This article searches the answers to the questions: “How safe is it?”, “How easily can it be cracked?” and “How can we make it more reliable?”

1 Introduction

This encryption method is very easy to implement and use, but can be cracked just as easily. On a computer you can simulate a very large baton, but using the wrong size can lead to an instant decryption.

To use the scytale ('skɪtəli:), you roll the parchment on the baton, and write your message horizontally, therefore on the parchment the message will be mixed up. To decrypt the message, you need the same sized baton.

In a program it can be represented as a matrix (or simulated), and the message is written on lines, and read on columns. The decryption works inverse, writing the crypted text on columns and read on lines.



Figure 1.: A scytale

After an implementation of the original scytale, I present a modified program and offer some possible extensions. Finally, I will conclude the article with my observations and answering the questions stated above.

2 Original Scytale

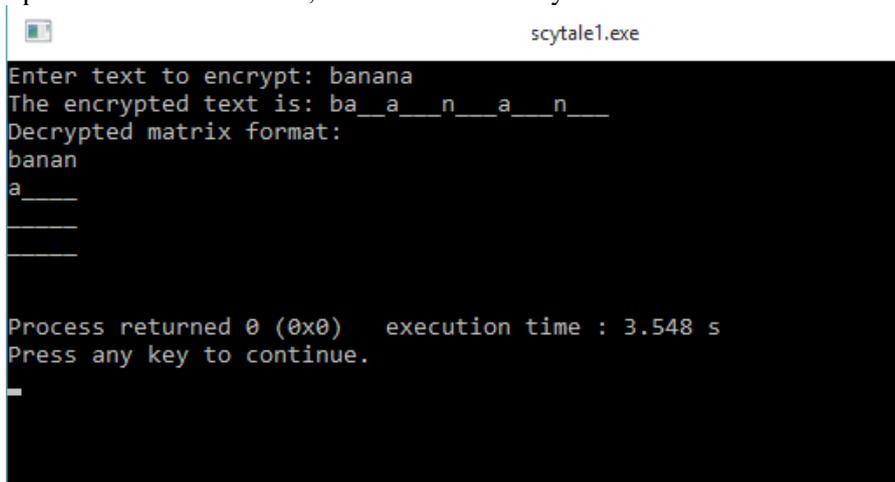
```
int R = 4; //radius, letters around rod - lines of the matrix
int H = 5; // length of the rod - columns of the matrix
```

```
string encrypt(string& input)
{
    string nrypted("");
    for(int i=0; i<H; i++)
    {
        for(int j=0; j<R; j++)
        {
            int location = i+j*H;
            if(location >= input.length())
                nrypted += '_';
            else
            {
                if(input[location] == ' ')
                    nrypted += '_';
                else
                    nrypted += input[location];
            }
        }
    }

    return nrypted;
}

void decrypt(string& crypted)
{
    for(int i=0; i<R; i++)
    {
        for(int j=0; j<H; j++)
        {
            int location = i+j*R;
            cout << crypted[location];
        }
        cout << endl;
    }
}
```

As you can see, for a maximum length of 20 ($R \cdot H$), encrypt will calculate the position of each letter of the input text in a matrix, and adds padding if needed (random character). The space character is replaced with underscore, but is not necessary.



```
scytale1.exe
Enter text to encrypt: banana
The encrypted text is: ba_a_n_a_n_
Decrypted matrix format:
banan
a
_
_
_

Process returned 0 (0x0)   execution time : 3.548 s
Press any key to continue.
-
```

Figure 2.: example for simple scytale

The algorithm is simple and it has some flows. Firstly, knowing the length, you can find out the combinations of R and H, then try to decrypt the text until you see a clear message. Secondly, R and H values are fixed, and that is not the best option in all cases.

2.1 Iteration

I noticed after a little fiddling with the encryption method, that repeating it will reproduce the plain text after a while. That means, for the example above that number is 9. Knowing the length, the code breaker doesn't have to try many combinations of R and H. If the text is encrypted just once, the decryption also reveals the plain text after one turn. If the text is encrypted the maximum amount of times (before it turns again to plain text), it needs just one encryption turn to get the plain text.

2.2 Maximum iterations

But how can we know, what is the best combination of (R, H, I)?

The algorithm searches this triplet, by repeating the encryption, saving these numbers, grouped by H, and then sort by I. The result can be exported to a CSV format and represented with a diagram in Excel.

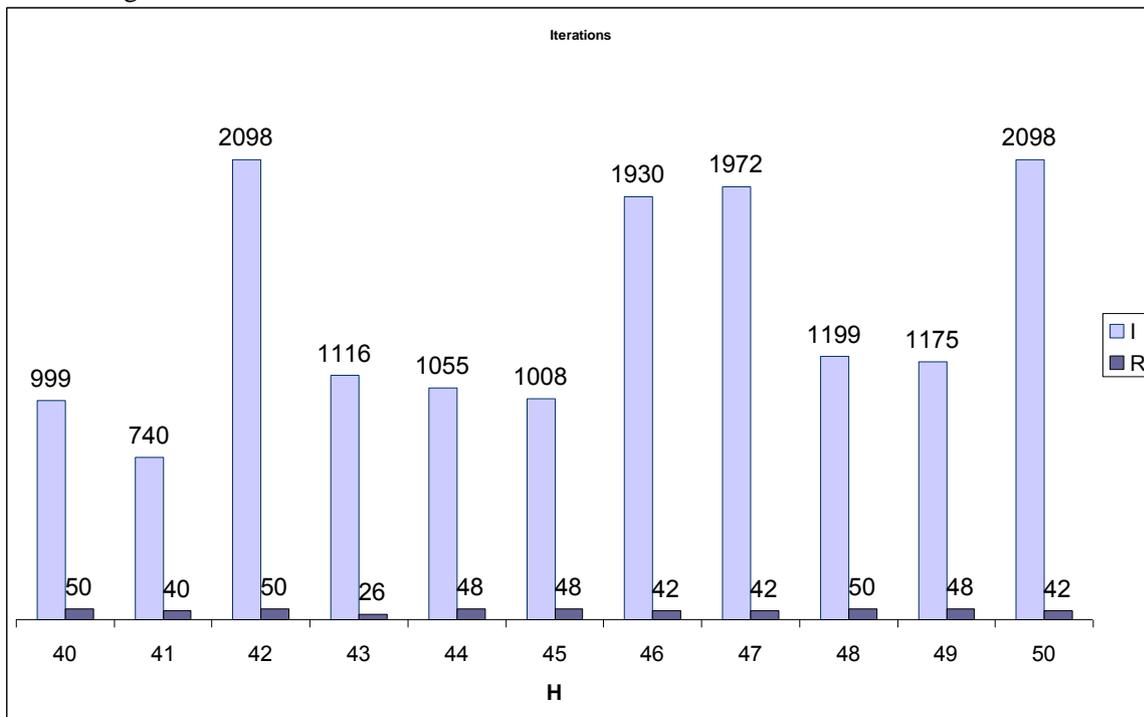


Figure 3.: diagram representation, H:[40, 50]

This diagram shows the maximum iterations needed for each (R, H) pair. The search went through all possible R values for each H, in range of [2, 50], and then only saved the biggest iteration.

It is visible, that the transposed “matrix” gives the same iteration value. Surely, the text will be different, but it won't effect the calculations. Also, if R and H are equal, the iteration will be 1!

There are two ways to express the iteration number (but these are not proven):

$$I = H*(R-2)-3 \quad (1)$$

or

$$I = (H*R)/2-2 \quad (2)$$

The number of iterations is linear to the length of the text.

3 Improved scytale

For testing and improving, I created a class for encryption. It contains the necessary functions to determine the best solution.

The program reads in the plain text and maximum length, and generates two crypted texts (with different alphabets).

3.1 Helping functions

These functions are not taking part in the encryption, and some of them can be left out of a search implementation.

3.1.1 Padding

If the text is shorter than $R \cdot H$, an underscore character will be added to it. This function is called for each combination in the search, to see when the encrypted text reappears as the plain text.

```
string padding(int h, int r)
{
    string nstr=input; //private variable of class

    for(int i=0; i<r*h; i++)
        if(i < input.length())
        {
            if(input[i] == ' ')
                nstr[i] = '_';
            }else
                nstr+='_';

    return nstr;
}
```

3.1.2 Sorting, grouping

While these functions are not obligatory, it is helpful to deal with larger amount of data, which is stored in static arrays. These can be replaced in the search by a local variable containing the maximum iteration value.

3.1.3 Printing

Two functions display information on the elements of the grouped array. An element of the array contains the values of R, H, I, the crypted text, and the last character of this text which is not an underscore.

```
void print_iteration(iteration* iter, int n)
{
    for(int i=0; i<n; i++)
    {
        cout << "I=" << iter[i].i << " R=" << iter[i].r << " H=" << iter[i].h <<
            " Length: " << iter[i].crypted.length() << " Last letter at: " << iter[i].lastNonPad << endl;
        printMatrixed(iter[i].crypted, iter[i].h, iter[i].r);
    }
}
```

3.2 Phase 1

The function goes through all available R, H pairs, determining the maximum iterations, completing the data arrays.

```
void encryptPhase1()
{
    iteration iter[1000];
    bool usedH[1000] = {false};
    int k=0;

    for(int i=H; i <= maxLength/2; i++)
    {
        usedH[i] = true;
        for(int j=R; j <= maxLength/i; j++)
        {
            if(i*j <= maxLength && i*j > input.length() && !usedH[j])
            {
                string padded = padding(i, j);
                string crypted = input;

                //find the iteration and shortest text
                int smallestLastNonPad = i*j;
                int it=0;

                while(crypted != padded)
                {
                    crypted = encrypt1(crypted, i, j);

                    int lnp = lastNonPaddingLetter(crypted);
                    if(lnp < smallestLastNonPad && crypted != padded)
                    {
                        smallestLastNonPad = lnp;
                        it=n;
                    }
                    n++;
                }

                //do encryption for shortest text
                crypted = input;
                for(int x=0; x < it; x++)
                {
                    crypted = encrypt1(crypted, i, j);
                }

                iter[k].r = j;
                iter[k].h = i;
                iter[k].crypted = crypted;
                iter[k].lastNonPad = smallestLastNonPad;
                iter[k].itLNP = it;
                iter[k++].i = n-2;
            }
        }
    }

    iteration grouped[100];

    int groupedNum = 0;
    groupIterations(grouped, iter, k, groupedNum);

    //sort iterations
    sort_iteration(grouped, groupedNum);
}
```

```
print_iteration(grouped, groupedNum);

cout << "Ideal iteration is " << grouped[0].i << " with R=" << grouped[0].r << " and H=" <<
grouped[0].h << endl;

I = grouped[0].i;
R = grouped[0].r;
H = grouped[0].h;
}
```

The initial H and R values are 2 and 3. After the search and sorting, the first element of the array will be selected for application. This function dumps a lot of information on the screen, to help understanding the nature of the encryption.

It works in two different directions:

- Searches the maximum iterations for each pair, stores them and groups them by H.
- Also finds the iteration, for which the encrypted text is the shortest (last character which is not underscore). My idea was to reduce the length of the text, if possible, to make it harder to crack. What I noticed is, for the test text “banana”, which is 6 letters long, for most of the triplets, the shortest iteration is 11 letters long. That results in “b_a_n_a_n_a”. The problem is, even if the padding a random letter, it can be figured out. It might not be that easy for a length of hundreds of characters, but a program can take every second, third... letter to see if it gives any result. This approach will need more research.

3.3 Phase 2

Applies the previously found triplet.

```
void encryptPhase2()
{
    string crypted = input;
    for(int i=0; i <= l/2; i++)
        crypted = encrypt2(crypted);

    output = crypted;

    crypted = input;
    for(int i=0; i <= l/2; i++)
        crypted = encrypt3(crypted);

    output2 = crypted;
}
```

As mentioned earlier, using the maximum number of iterations needs one more turn to reveal the plain text. To avoid this, the method goes only halfway. This will increase the security, but also the time needed to decrypt the message.

4 Extending

Implemented on a computer, this crypting method is not really safe. Even with some variations, it can be cracked in acceptable time.

4.1 Changed length

For example, if maxLength is specified at 100, the ideal iteration will be a (14, 7, 95) triplet, with the length of 98. The cracker can find the divisors of 98, then the plain text. But let's

say the length is changed to 100, or even less (but still containing all the characters of the plain text), so it will be harder to find the combinations. An even better choice would be the closest prime number.

This won't affect the decryption, because it will work on the length of $R \cdot H$.

4.2 Layering

The longer the maximum length, the more layers can we apply. A layer is an applied triplet. The first layer generates an encrypted text, then the second layer uses this as plain text, increases length, generates a second encrypted text and so on. The decryption has to know these parameters to successfully retrieve the original message.

4.3 Combining with other crypting methods

There are many similar (easy to implement) methods, with similar aspects. However, the combinations might result in a much more robust encryption.

Before using the scytale, the plain text can be modified in a Caesar encryption, or any other replacing method. Even if the cracker deals with the scytale, this would mean a new obstacle.

5 Target implementation

Since the cryption itself cannot be named the safest, I wouldn't recommend it to be used for highly sensitive data. However, I can imagine implementing it (in a much more complex composition) to store passwords for a website, or for a crypted messenger. Certainly the length should be maximized for more safety.

An exact code should use any of the suggested extensions, and have a fixed triplet as a private key.

6 Conclusion

As a modern encryption, scytale is not as strong as many other, professionally developed methods (AES-256), or any mathematical methods. For a computer, a simple scytale is not a problem, because it is just a permutation of the letters. However, it can be expanded to be a solid solution to use in school projects, or to introduce students to crypting.

6.1 How safe is it?

Short texts can be recovered very fast, but it becomes more reliable as the text grows. Also, at least one of points from section 3 should be used.

6.2 How easily can it be cracked?

If the length is not a prime number, or has very few divisors, and is not a big number, the basic algorithm doesn't provide much against attacks. The necessary time to unveil the plain text grows with the complexity.

6.3 How can we make it more reliable?

By combining with other methods, repeating, expanding the alphabet.

7 References

- [1] <https://en.wikipedia.org/wiki/Scytale>

István Bialkó
Spiru Haret University, Bucharest
Faculty of Mathematics and Informatics
str. Ion Ghica, nr. 13, cod postal 030045, Sector 3, Bucharest
Romania
Email: bialko.istvan@gmail.com

Foody, you are what you eat

Cristian Bota, Adrian A. Bărbos
Coordinator Professors: Camelia-M. Pinteau, Hajdu Măcelaru Mara

Abstract

The current paper illustrates a new application meant to innovate the way you track your meals and also to help you discover new ones. Meet Foody, an application inspired by the need to track and to keep in order the most essential routine of our daily basis: the food we eat. They say “You are what you eat“, but hence we live in an alert system we are constantly neglecting the importance of eating healthy, and eating regular. We seek to ease and improve the way we organize our daily meals. The application looks promising for developing strong knowledge of what a normal diet should look like, thus maintaining a healthy lifestyle and a fit body.

1 Introduction

The desire to eat healthy generated an idea that could change lives. It started as a simple idea but then it turned out to be something bigger, something we believe in. We decided to develop this app because we believe it's time to create something new that innovates the way we track our daily meal routine by integrating what recently became a habit and a need in people's lives - sharing with others.

Sometimes we just want to experience new things. Why not do this in the kitchen? Foody gives you access to original ideas while you can share your own brand new ideas.

2 Foody, you are what you eat

We started by doing research and we studied many solutions developed in order to help people in this manner, but we believe that our original and easy approach is capable of changing people's menu, thus slowly changing their lives.

Some of the other apps have a more technical approach by making the user add the calories from each ingredient that compose every meal, which can be a time consuming process. Regarding this, we wanted to keep the adding process as simple as possible, each entry having just one required characteristic: a photo of it.

Foody offers the possibility of keep in check the meals journal for every day, so regardless if you're a personal trainer and you want to monitor one's eating habits, or you just want to share with your friends the diet you kept in order to lose weight, or you want to track how a certain kind of alimentation affects you on the long term, now it's easier than ever. With the new **Channel** function in which all entries are arranged in a chronological order you have a an intuitive overview of your meals. Even more, we developed the awesome feature of exporting one or more entries from the journal in a mobile and web friendly format for easy sharing with any of your friends.

We are thinking of constant improving of our app keeping in mind the following aspects:

- **Simplicity** - Foody is easy to grasp and the flow is intuitive
- **Connectivity** - be connected with any person or any type of food you're interested in
- **Calendar** - easy to check any entries
- **Flexibility** - it does not limit you to the amount of details you may add to a certain meal
- **Timeline** - the discover process is simplified by having suggestions one touch away
- **Push notifications** - alert the user with a message when new data is published

3 Core and API development

3.1 The tough call

When we first thought about building the app's core functionalities we knew we needed to sync all the user data and furthermore, to scale it and make it available on multiple platforms. In order to achieve this we decided to create an **REST API** coded in **PHP**, one of the most popular web development programming languages.

API (application program interface) is a set of routines, protocols, and tools for building software applications. The API specifies how software components should interact and APIs are used when programming graphical user interface (GUI) components. But what about REST? REST (REpresentational State Transfer) is an architectural style and an approach to communications that is often used in the development of Web services. The use of REST is often preferred over the more heavyweight SOAP (Simple Object Access Protocol) style because REST does not leverage as much bandwidth, which makes it a better fit for use over the Internet. The SOAP approach requires writing or using a provided server program (to serve data) and a client program (to request data).

But then we had a harder decision to make, should we make the API in vanilla PHP from scratch or use a PHP Framework ? After days with many searches on Google and seeing what other developers decided when they were facing a similar problem, we decided to go with a framework rather than building it from scratch.

3.2 The advantages of using a framework

So it was a tough call but we learned that using a framework has many advantages and it makes the development faster and easier than building it from scratch.

Efficiency - tasks that usually would take hours and hundreds of lines of code to write can now be done in minutes with pre-built functions. Development becomes a lot easier. If it's easier it's faster, and consequently efficient.

Security - a widely used framework has big security implementations. The big advantage is the community behind it, where users become long-term testers.

Cost - most popular frameworks are free and since it also helps the developer to code faster, the cost for the final client will be smaller.

Support - as any other distributed tool, a framework usually comes with documentation, a support team, or big community forums where you can obtain quick answers.

We also had to keep in mind that using a framework will also come with some limitation and strict rules that you need to learn to be able to get your application up and running but we were ready to do so. The framework that better suited our needs was **Laravel 5**.

3.3 Laravel 5

Although Laravel is a relatively new PHP framework (it was released in 2011), according to Sitepoint's recent online survey it is the most popular framework among developers. Laravel has a huge ecosystem with an instant hosting and deployment platform, and its official website offers many screencast tutorials called Laracasts.

Laravel has many features that make rapid application development possible. Laravel has its own lightweight templating engine called "Blade", elegant syntax that facilitates tasks you frequently need to do, such as authentication, sessions, queueing, caching and RESTful routing. Laravel also includes a local development environment called Homestead that is a packaged Vagrant box.

3.3 Back to the drawing board

After choosing how we want to do our application and the technologies we want to use we needed to go back to the drawing board and define the system architecture and how the application will communicate with the API. Because we wanted a Restful API service we needed to use HTTP methods:

- POST - create a new resource
- GET - read/retrieve a representation of a resource
- PUT - update a resource
- PATCH - similar to PUT, it contains the changes to the resource, not the complete resource
- DELETE - self-explanatory, used to delete a resource

To a better understanding of how this method would be used on a resource, let's say we have a product resource and we need to do CRUD (Create, Read, Update, Delete) operations on this resource. We assume that the root api url is <https://foody-api.dev/v1>, V1 since it's best practice to version an API for long term usage.

- **Create a new product** - we need to do a **POST** request to <https://foody-api.dev/v1/product> with all the required field that make up one product.
- **Get all products** - we make a **GET** request to <https://foody-api.dev/v1/product> but this time without any parameters. What about one single product? Well it's fairly easy, we just add the product's identifier to the url like this <https://foody-api.dev/v1/product/1> and we will receive only the one products with the identifier we just passed into the url as an **wildcard** (a symbol used to replace or represent one or more characters), the url would be defined as https://foody-api.dev/v1/product/{product_id}.
- **Update a product** - we make a **PUT** request to https://foody-api.dev/v1/product/{product_id} with all the fields that make up a product. What about **PATCH** ? We make the request to the same url but with only one of the fields that make up a product, let's say price and this time we only made a partial update.
- **Delete a product** - make a **DELETE** request to https://foody-api.dev/v1/product/{product_id} without any fields. In our case a best practice is to do only a soft delete, which means we don't actually delete a resource from the database, we just make it unavailable for the end user. This helps to recover sensitive data that was accidentally deleted.

These are the advantages of creating a REST API and also the reasons we choose to do so; it's much easier to make full CRUD operation on a certain resource.

3.4 Foody Api

The first core functionality we needed to implement was authentication. In order to succeed we knew that implementing a normal email and password authentication system was not enough because users are used to authenticate into application using a third party system like Google, Twitter and Facebook. At this early stage of development we decided to go only with Facebook for the moment, since it's the most used social network and add more methods later on.

Authentication related methods:

- **Login with Facebook** - **POST** request <https://foody-api.dev/auth/facebook> also passing the Facebook session id token so the server side can retrieve and store the name, surname and email.
- **Normal Login** - **POST** request <http://foody-api.dev/auth/login> with required parameters: email and password.
- **Sign Up via email** - **POST** request <http://foody-api.dev/auth/sign-up> with required parameters: name, surname, password and email.
- **Forgot password** - this api is used in case the user forgot his password but for security reasons we don't store the user's password in a readable way, passwords being encrypted using MD5. So the app will do a **POST** request to <https://foody-api.dev/auth/forgot> sending the user's email address and the user will receive an email that will contain a temporary link (available 24h) that will redirect to a change password webpage where the user will be able to change his password.

All the above sign up/login methods described above, if successfully, will generate an API Token that will be required to be passed on to every other post login API Method in the request header. The API

token will be unique to every user and the API Token will only be changed after the user recovered/changed his password.

All the post login APIs don't need to be explained here since the API is only an REST interface for the client apps and all the functionalities will be explained in the mobile application development below.

4. Mobile application development

The mobile application was developed on two of the most popular mobile operating systems, Google's Android and Apple's iOS. The Android mobile application and its modules will be presented further on, while the iOS implementation and development will only be discussed.

Both Android and iOS applications contain the following main modules:

- **Login** is the first module and the first screen a user will see once the application is opened. It contains the logo of the application and a login form with a name field, password field and a submit button to perform the login action. It also contains a button for the creating of a new account.
- **Register** is the option to choose for creating a new account. It contains a register form with a name field, an email field, a password field and a submit button to perform the registration. Once registered, a **Channel** is created and assigned to the new account. This is where the user will have all his information regarding meals and related activity.
- **User profile** is a common module of all app users. Basic information like the name, surname, email and profile picture will be shown and it will also contain the possibility of changing the password.
- **Add meal** module makes it easier than ever to add a new meal-event, having first a image button where you will snap a photo of your meal, and having many other optional fields such as title, description, ingredients, time of cooking, etc. Posting is done by simply clicking the **Done** button after the desired fields have been filled.
- **Calendar** module is like any other normal calendar module and allows you the see or export the timeline of each day/month.
- **Discover** module selects the best meals from every user and lists them randomly in a grid-like view which allows you to browse and see details for any of selected entry.

The **discover** mechanic is further split into more sections to make browsing much easier, such as:

- Breakfast
- Brunch
- Desert

Channels module lists the top subscribed channels to make it easier for the new users to find the content they like.

4.1 Android data request method development

Android application development is done using the official Android SDK in Android Studio. The programming is done in Java and the layouts are created in XML. **Java** is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. **Extensible Markup Language (XML)** is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. It is defined by the W3C's XML 1.0 Specification and by several other related specifications, all of which are free open standards. The design goals of XML emphasize simplicity, generality and usability across the Internet.

The Parser class makes an HTTP request via post and requires implementation of the "onPreExecute" and "onPostExecute" methods.

```
Parser parser = new Parser(context, Parser.OBJECT){
    @Override
    protected void onPreExecute() {}

    @Override
    protected void onPostExecute(Void result) {
        processData(getObject());
    }
};
parser.execute();
```

The Parser class takes the context as a parameter and a predefined constant, which is either OBJECT or ARRAY that specifies the type of the response. Via the "getObject()" or "getArray()" methods, the response is returned and the resulting data can be processed.

5. Conclusion

The application can be further enlarged with much more functionalities based on user feedback and our next steps, not necessarily in this order:

- Scale the application and develop iOS and Web Based Applications
- Go public with the API and let other developers access the core functionalities to be able to build third party application
- Smart Watch/Band integration to monitor how active a user is and generate in app analytics

References

- [1] PHP NET-complete and hyperlinked manual documents and explains all elements of PHP.
<http://php.net/manual/>
- [2] Android Developer suite, <http://developer.android.com>
- [3] Laravel docs, <https://laravel.com/docs/5.2>
- [4] Laracast, <https://laracast.com>
- [5] Stack Overflow - question and answer site for programmers *<http://stackoverflow.com>*.
- [6] Wikipedia

Adrian BĂRBOS
Technical University
Cluj-Napoca
Faculty of Science
76 Victoriei
ROMÂNIA
ady.barbos@gmail.com

Cristian Bota
Technical University
Cluj-Napoca
Faculty of Science
76 Victoriei
ROMÂNIA
cristianbota94@gmail.com

Arcadia: Socialize and Learn in Virtual Reality

Petrică Bota, Robert Săndică
Coordinator Professor: Dana Simian

Abstract

Recently, the popularity of virtual reality headsets (VR) has increased dramatically. This is a direct result of the advancements made in processing power and screen resolutions driven especially by the smartphone market. At the moment, there are many VR apps that immerse the user in a simulated world. We consider that most of the apps currently existing on the market are lacking interactivity, thus we have created Arcadia. Arcadia is an online virtual platform where people from around the globe can meet with virtual representations of themselves for a variety of purposes, e.g. a meeting with a class of students attending a course in distance learning system.

1. Introduction

In the history of distance learning there have been three generations. The first generation was correspondence learning, debuting in 1840 in England. The second generation started on a more technological route with the use of the television and telephone between teachers and students. The third generation is the generation that has computers and the Internet at their disposal. With the creation of the personal computer and of the World Wide Web the amount of information available to us has increased exponentially and so has the ability to share information.

Recently the rise of virtual reality and augmented reality [5] creates prerequisites for development of new classrooms' type: virtual classroom. In a virtual classroom students from around the world could attend a course taught by a teacher in a virtual setup.

The concept is not new; it began in the 1950 [1] when the first computers capable of running a virtual world came to existence. The first device capable of rendering a virtual reality application was in the form of a flight simulator created in the year 1966 [2]. At the moment virtual reality applications are used in a variety of fields from medical to entertainment or even military use.

The aim of this article is to present Arcadia as a virtual reality application that intends to create an environment capable of shortening the distance between people and offering new ways of teaching. Arcadia is able to draw the user's attention by immersing him in a virtual world where he can see and experience famous landmarks or visit well known museums without him actually having to leave the comfort of his home.

2. Description

Arcadia was initially created as a simple platform where multiple users can meet. Originally we created a simple map that contained a few buildings and landscapes that the users could explore together. We continue to expand on that original world by adding other in-game objects or features that we believe to be of service to future users. Such a feature is the teaching system that we have implemented. Students are able to learn by actually experiencing different scenarios from a first person perspective. For instance, you could learn about the pyramids by taking a virtual tour of Giza while having a teacher explain the major points to you.

In our opinion virtual reality will be a major outlet for distance teaching in the nearby future. Many fields can benefit from the use of this technology. For example a psychiatrist could use Arcadia to try to help a patient get over his fear of heights. By simply creating a scenario where the patient is on top of a skyscraper we can build an environment in which the patient can be treated without putting his life in any real danger.

2.1 App structure

The application has been created in the Unity game engine. The main functionality of Arcadia has been written with C# scripts that contain the logic behind each object or action inside the app. Next, we have a snippet of code that dictates the character's movements inside Arcadia:

```
void Start()
{
    anim = GetComponent<Animator>();
}
void Update()
{
    float translation = Input.GetAxis("Vertical") * speed;
    float rotation = Input.GetAxis("Horizontal") * rotationSpeed;
    translation *= Time.deltaTime; rotation *= Time.deltaTime;
    transform.Translate(0, 0, translation); transform.Rotate(0, rotation, 0);
    if (Input.GetKeyDown(KeyCode.Joystick1Button17) || Input.GetKeyDown(KeyCode.Space))
    {
        anim.SetBool("isJumping", true);
        au_isWalking.Stop();
    }
}
```

Most of the buildings that you will see inside the app have been imported from the Unity's asset store. You can either import your objects in Unity or you can create your own 3D assets (Fig. 1).

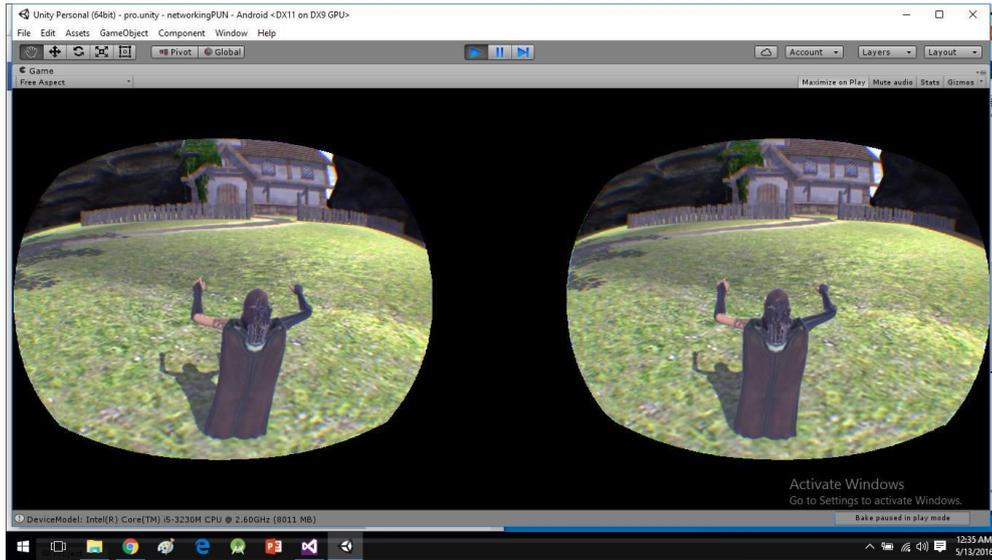


Fig. 1 Character Looking at a 3D House

2.2 Devices

The basic principle required to create a virtual reality setup is to render a view of the virtual environment from two separated angles simulating the human eyesight. By doing so, the brain is tricked into thinking that what it sees is real and that the world he is in had a layer of depth to it.

Devices that are used to experience a reality created digitally create the depth by displaying two images of the same point of view in front a set of lenses specially designed to give the field of view effect [3](as presented in Fig. 2).

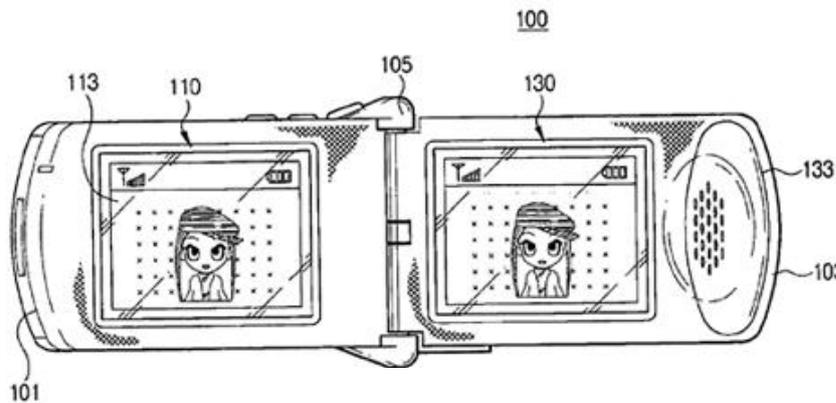


Fig. 2 Image Positioning in the Headset [8]

There are two types of devices that allow the user to enter a virtual setup. The first, is the device that has the screen, on which the apps will be displayed, incorporated into the headset. Such devices include Oculus Rift (Fig. 2) or Vibe [4]. Another way to create a functioning VR headset is to use your smartphone as a display screen. This is the type of headset that we use due to the fact that we consider it as a cheaper alternative. The devices that we have used for the creation and testing of these apps are the Zeiss VR One [8] and Baofeng Mojing 3.

The devices that are only for the purpose of rendering a virtual environment are composed from a headset, display, a pair of lenses, head tracking sensor, cables, optionally a camera for other purposes



Fig. 3 Oculus Rift Composition [9]

In order to enter Arcadia the user has to download and install our app from the Play Store. The user also has to have in his possession a VR headset, a game controller and an OTG cable (easily bought from most electronics stores). Though, at the moment many people have smartphones few have the other components required to run our app. We believe that with time this will change and that in a couple of years most houses will have a VR headset laying around.

Consider that the user has everything set up, including an Internet connection, he needs to plug the controller to the smartphone, next slide the smartphone into the headset and put the headset on while being careful not to lose the controller

Once inside Arcadia, the user can explore the “campus”, interact with other persons or simply go to one of the courses. The Advantage of having course in a virtual setup is that we can change the classroom with the press of a few buttons. We do not have to be limited to the boring old chairs, tables and/or projector.

Think of studying ancient history and being able to see a “real-life” warrior; or better yet, being able to see a full fight between the Romans and the Persians. Such a historic battle could be programmed as another room or course inside Arcadia.

2.3 Technologies

Our application has been created with the help of the game engine Unity. Unity is a flexible, easy to use environment that allows programmers to code complex setups relative easily. We also chose the Zeiss VR One headset due to the fact that it has a standard development kit [7] that permits programmers to create virtual reality applications. That SDK is compatible with Unity [6] and functions as the gateway between normal apps and VR apps. It basically splits the screen of the smartphone in the two points of view mentioned earlier. Another program that we have used is Photon. Photon is a networking programs that allows for the integration of multiple apps from multiple platforms. We have also selected this program due to the fact that it enable Unity written apps to communicate with each other. As a result multiple users are able to use our platform at the same time.

3. Conclusion and future developments

To sum up, Arcadia is an ongoing project to which we can add features as we go along. At the moment we are implementing the backend required to host online courses. The most important feature that we desire to implement is our own built-in chat system. Now we are limited to using Skype in order to have conversation in Arcadia, but that will change in the near future.

Though, at the moment VR technology is relatively cheaper than it was thirty years ago people are still hesitant to buy the headsets required. We hope that by working on projects such as Arcadia we will help with the spread of this technology.

References

- [1] Roy S. Kalawsky, *Science of Virtual Reality and Virtual Environments*, Addison-Wesley Pub (Sd); 2nd edition, 2004.
- [2] Matjaz Mihelj, Domen Novak, Samo Beguš, *Virtual Reality Technology and Applications*, Springer Science + Bussiness Media, Dordrecht, 2014.
- [3] Virtual Reality headset with review - www.aliexpress.com/
- [4] Virtual Reality Software, Augmented Reality, 3D Visualization, Knowledge Transfer, Training – www.eonreality.com/
- [5] Virtual Reality Software - www.worldviz.com/virtual-reality
- [6] VR One SDK - <https://bitbucket.org/vrone/unity3d/src/3c7395473a371ff755f89715a36ee532427c780d#markdown-header-what-is-the-vr-one-sdk>
- [7] Zeiss VR One homepage - <http://zeissvrone.tumblr.com/>
- [8] Fig. 2 - <https://news.samsung.com/global/gear-vr-how-samsung-makes-virtual-reality-a-reality>
- [9] Fig. 3 - <https://www.goglasses.fr/oculus-rift/oculus-rift-dk2-composition-characteristiques-techniques/attachment/oculus-rift-dk2-composition>

Robert Sandica
„Lucian Blaga” University
Faculty of Sciences
Str. Dr. I. Rațiu, Nr.5-7, Sibiu
Romania
Email: sandica_robert@yahoo.ro

Petrica Bota
„Lucian Blaga” University
Faculty of Sciences
Str. Dr. I. Rațiu, Nr.5-7, Sibiu
Romania
E-mail: petrica.bota2@yahoo.com

Remote monitoring applications through Windows APIs

Momcilo Brajic

Abstract

In software development, an application programming interface (API) is a set of routines, protocols, and tools which provides facilities to develop applications using a given programming language. In this paper, we will use several APIs of Windows operating systems class to develop a client/server application able to intercept and monitor events occurred on targeted computers.

1 Introduction

Using the Windows API, we can develop applications that run successfully on all versions of Windows while taking advantage of the features and capabilities unique to each version [1]. Although the original name was formerly the Win32 API, the name Windows API more accurately reflects its roots in 16-bit Windows and its support on 64-bit Windows. In this paper, we will use a low-level programming style to access services provided by Windows APIs, in order to:

- operate in resource-constrained environments;
- have little runtime overhead;
- obtain direct and "raw" control over memory access and other system resources.

The aim of our approach is to develop a client/server application for monitoring events occurred on certain computers which have the client component installed. The events are sent over the network to the server component for further investigation.

Our application is based on the following Windows APIs:

- User Interface (Windows UI API) for building the user interface of the client;
- User Input and Messaging to implement a Hook mechanism;
- System Services for Dynamic-Link Libraries, Memory Management, Processes and Threads;
- Networking and Internet for sending information using Windows Sockets.

The paper is organized as follows. In the next section, we provide some background material related to Win API Programming. In section 3 we present an overview of Windows Hooks and details for their use to intercept and monitor events. The Section 4 is dedicated for network communications through Windows sockets. The SysMonitor application is presented in Section 5. Finally, are presented some conclusions and opportunities for future developments in Section 6.

2 Windows API

The Windows API, informally WinAPI, is Microsoft's core set of application programming interfaces (APIs) available in the Microsoft Windows operating systems. The name Windows API collectively refers to several different platform implementations that are often referred to by their own names (for example, Win32 API). Almost all Windows programs interact with the Windows API. On the Windows NT line of operating systems, a small number (such as programs started early in the Windows startup process) use the Native API.

Developer support is available in the form of a software development kit, Microsoft Windows SDK, providing documentation and tools needed to build software based on the Windows API and associated Windows interfaces.

The Windows API (Win32) is focused mainly on the programming language C in that its exposed functions and data structures are described in that language in recent versions of its documentation. However, the API may be used by any programming language compiler or assembler able to handle the (well defined) low level data structures along with the prescribed calling conventions for calls and callbacks. Similarly, the internal implementation of the API's function has been developed in several languages, historically. Despite C lacking any traits of object-oriented programming, the Windows API, and Windows, has sometimes been described as object-oriented. There have also been many wrapper classes and extensions (from Microsoft or other sources) for object-oriented languages that makes this object-oriented structure more explicit (Microsoft Foundation Class Library (MFC), Visual Component Library (VCL), GDI+, etc.). For instance, Windows 8 provides the Windows API, and the WinRT API which is implemented in C++ and is object-oriented by design.

2.1 Windows API Index

The functions provided by the Windows API can be grouped in the following categories [3]:

2.1.1 Base Services

Provide access to the basic resources available to a Windows system. Included are things like file systems, devices, processes, threads, and error handling. These functions reside in kernel.exe, krnl286.exe or krnl386.exe files on 16-bit Windows, and kernel32.dll on 32-bit Windows.

2.1.2 Advanced Services

Provide access to functions beyond the kernel. Included are things like the Windows registry, shutdown/restart the system (or abort), start/stop/create a Windows service, manage user accounts. These functions reside in advapi32.dll on 32-bit Windows.

2.1.3 Graphics Device Interface

Provides functions to output graphics content to monitors, printers, and other output devices. It resides in gdi.exe on 16-bit Windows, and gdi32.dll on 32-bit Windows in user-mode. Kernel-mode GDI support is provided by win32k.sys which communicates directly with the graphics driver.

2.1.4 User Interface

Provides the functions to create and manage screen windows and most basic controls, such as buttons and scrollbars, receive mouse and keyboard input, and other functions associated with the graphical user interface (GUI) part of Windows. This functional unit resides in user.exe on 16-bit Windows, and user32.dll on 32-bit Windows. Since Windows XP versions, the basic controls reside in comctl32.dll, together with the common controls (Common Control Library).

2.1.5 Common Dialog Box Library

Provides applications the standard dialog boxes to open and save files, choose color and font, etc. The library resides in a file called `commdlg.dll` on 16-bit Windows, and `comdlg32.dll` on 32-bit Windows. It is grouped under the *User Interface* category of the API.

2.1.6 Common Control Library

Gives applications access to some advanced controls provided by the operating system. These include things like status bars, progress bars, toolbars and tabs. The library resides in a dynamic-link library (DLL) file called `commctrl.dll` on 16-bit Windows, and `comctl32.dll` on 32-bit Windows. It is grouped under the *User Interface* category of the API.

2.1.7 Windows Shell

Component of the Windows API allows applications to access functions provided by the operating system shell, and to change and enhance it. The component resides in `shell.dll` on 16-bit Windows, and `shell32.dll` on 32-bit Windows. The Shell Lightweight Utility Functions are in `shlwapi.dll`. It is grouped under the *User Interface* category of the API.

2.1.8 Network Services

Give access to the various networking abilities of the operating system. Its subcomponents include NetBIOS, Winsock, NetDDE, remote procedure call (RPC) and many more. This component resides in `netapi32.dll` on 32-bit Windows.

2.1.9 Web

The Internet Explorer (IE) web browser also exposes many APIs that are often used by applications, and as such could be considered a part of the Windows API. IE has been included with the operating system since Windows 95 OSR2, and has provided web related services to applications since Windows 98. Specifically, it is used to provide:

- An embeddable web browser control, contained in `shdocvw.dll` and `mshtml.dll`.
- The URL moniker service, held in `urlmon.dll`, which provides COM objects to applications for resolving URLs. Applications can also provide their own URL handlers for others to use.
- An HTTP client library which also takes into account system-wide Proxy settings (`wininet.dll`). However, Microsoft has added another HTTP client library called `winhttp.dll` which is smaller and more suitable for some applications.

2.2 Windows API Versions

Almost every new version of Microsoft Windows has introduced its own additions and changes to the Windows API. The name of the API, however, remained consistent between different Windows versions, and name changes were kept limited to major architectural and platform changes for Windows. Microsoft eventually changed the name of the then current Win32 API family into Windows API, and made it into a catch-all term for both past and future API versions.

Win16 is the API for the first, 16-bit versions of Microsoft Windows. These were initially referred to as simply the *Windows API*, but were later renamed to Win16 in an effort to distinguish them from the newer, 32-bit version of the Windows API. The functions of Win16 API reside in mainly the core files of the OS: `kernel.exe` (or `krnl286.exe` or `krnl386.exe`), `user.exe` and `gdi.exe`. Despite the file extension of `exe`, these actually are dynamic-link libraries.

Win32 is the 32-bit application programming interface (API) for modern versions of Windows. The API consists of functions implemented, as with Win16, in system DLLs. The core DLLs of Win32 are `kernel32.dll`, `user32.dll`, and `gdi32.dll`. Win32 was introduced with Windows NT. The version of Win32 shipped with Windows 95 was initially referred to as Win32c, with *c* meaning *compatibility*. This term was later abandoned by Microsoft in favor of Win32.

Win32s is an extension for the Windows 3.1x family of Microsoft Windows that implemented a subset of the Win32 API for these systems. The "s" stands for "subset".

Win64 is the variant of the API implemented on 64-bit platforms of the Windows architecture (as of 2011 x86-64 (AMD64) and IA-64). Both 32-bit and 64-bit versions of an application can be still compiled from one codebase, although some older APIs have been deprecated, and some of the APIs that were already deprecated in Win32 were removed. All memory pointers are 64-bit by default (the LLP64 model), so the source code must be checked for compatibility with 64-bit pointer arithmetic and rewritten as necessary.

WinCE is the implementation of the Windows API for the Windows CE operating system.

3 Windows Hooks Overview

A *hook* is a mechanism by which an application can intercept events, such as messages, mouse actions, and keystrokes. A function that intercepts a particular type of event is known as a *hook procedure*. A hook procedure can act on each event it receives, and then modify or discard the event.

The following are some examples suitable for hooks [2]:

- Monitor messages for debugging purposes;
- Provide support for recording and playback of macros;
- Provide support for a help key (F1);
- Simulate mouse and keyboard input;
- Implement a computer-based training (CBT) application.

3.1 Hook Chains

The system supports many different types of hooks; each type provides access to a different aspect of its message-handling mechanism. For example, an application can use the WH_MOUSE hook to monitor the message traffic for mouse messages.

The system maintains a separate hook chain for each type of hook. A *hook chain* is a list of pointers to special, application-defined callback functions called *hook procedures*. When a message occurs that is associated with a particular type of hook, the system passes the message to each hook procedure referenced in the hook chain, one after the other. The action a hook procedure can take depends on the type of hook involved. The hook procedures for some types of hooks can only monitor messages; others can modify messages or stop their progress through the chain, preventing them from reaching the next hook procedure or the destination window.

3.2 Hook Procedures

To take advantage of a particular type of hook, the developer provides a hook procedure and uses the *SetWindowsHookEx* function to install it into the chain associated with the hook. A hook procedure must have the following syntax [7]:

```
LRESULT CALLBACK HookProc(  
    int nCode,  
    WPARAM wParam,  
    LPARAM lParam  
)  
{  
    // process event  
    ...  
    return CallNextHookEx(NULL, nCode, wParam, lParam);  
}
```

HookProc is a placeholder for an application-defined name.

The *nCode* parameter is a hook code that the hook procedure uses to determine the action to perform. The value of the hook code depends on the type of the hook; each type has its own characteristic set of hook codes. The values of the *wParam* and *lParam* parameters depend on the hook code, but they typically contain information about a message that was sent or posted.

The *SetWindowsHookEx* function always installs a hook procedure at the beginning of a hook chain. When an event occurs that is monitored by a particular type of hook, the system calls the procedure at the beginning of the hook chain associated with the hook. Each hook procedure in the chain determines whether to pass the event to the next procedure. A hook procedure passes an event to the next procedure by calling the *CallNextHookEx* function.

Note that the hook procedures for some types of hooks can only monitor messages. The system passes messages to each hook procedure, regardless of whether a particular procedure calls *CallNextHookEx*.

A *global hook* monitors messages for all threads in the same desktop as the calling thread. A *thread-specific hook* monitors messages for only an individual thread. A global hook procedure can be called in the context of any application in the same desktop as the calling thread, so the procedure must be in a separate DLL module. A thread-specific hook procedure is called only in the context of the associated thread. If an application installs a hook procedure for one of its own threads, the hook procedure can be in either the same module as the rest of the application's code or in a DLL. If the application installs a hook procedure for a thread of a different application, the procedure must be in a DLL.

4 Network programming with threads, ports, and sockets

We can use the word-picture presented to us by the name "socket" in a similar fashion to illustrate what they are and how they work. In an actual mechanical socket, we may recall that it is the "receiving" end of a connection. A "thread" is a symbolic name for a connection between our computer and a remote computer, and a thread is connected to a socket.

We might think of a thread as an actual, physical, sewing-type thread stretched from one computer to the other, as the common analogy goes. In order for the threads to be attached to each computer, however, there must be a receiving object that attaches to the threads, and these are called sockets. A socket can be opened on any "port"; which is simply a unique number to distinguish it from other threads, because more than just one connection can be made on the same computer. A few of these ports have been set aside to serve a specific purpose. Beyond these ports, there are quite a large number of other ports that can be used for anything and everything: over 6,000, actually. A few commonly used ports are listed below with their corresponding services:

Port	Service
7	Ping
13	Time
15	Netstat
22	SSH
23	Telnet (default)
25	SMTP (Send mail)
43	Whois (Query information)
79	Finger (Query server information)
80	HTTP (Web pages)
110	POP (Receive mail)

Table 1: Commonly used ports and their corresponding services

There are many more ports used for specific purposes that are not shown here. Typically though, if we wish to use a port that has no specific assigned service, any port from 1,000 to 6,555 should be just fine. Of course, if instead we want to listen in on messages sent to and from service ports, we can do that too. It's not enough just to know what port number we're using, obviously; we need to know what remote computer/server we're connecting to (we have to know the "IP address" of the host we are connecting to, if we are connecting and not just listening). An IP address is an identification number that is assigned to each computer on the network, and consists of four sets of digits separated by periods.

The monitor system presented in this paper uses network services provided by the Winsock library in order to implement its communication mechanism.

We are using Windows Sockets 2 version of the library, which extends the functionality of Windows Sockets 1.1 in many areas [3]. Winsock defines a standard service provider interface (SPI) between the application programming interface (API), with its functions exported from WS2_32.dll and the protocol stacks. We are using the TCP/IP protocol stack to implement the communications between the client/server components of our monitoring system.

5 The SysMonitor Application

In this section we present an application with a client/server architecture used to log events occurred on certain monitored computers.

5.1 Overall functionality

The SysMonitor system is composed by two distinct applications:

- Monitor - the server part, use to log all events received from the monitored computers;
- AppLog – the client part, installed on monitored computers, used to intercept events such as application launching and send all information to the Monitor.

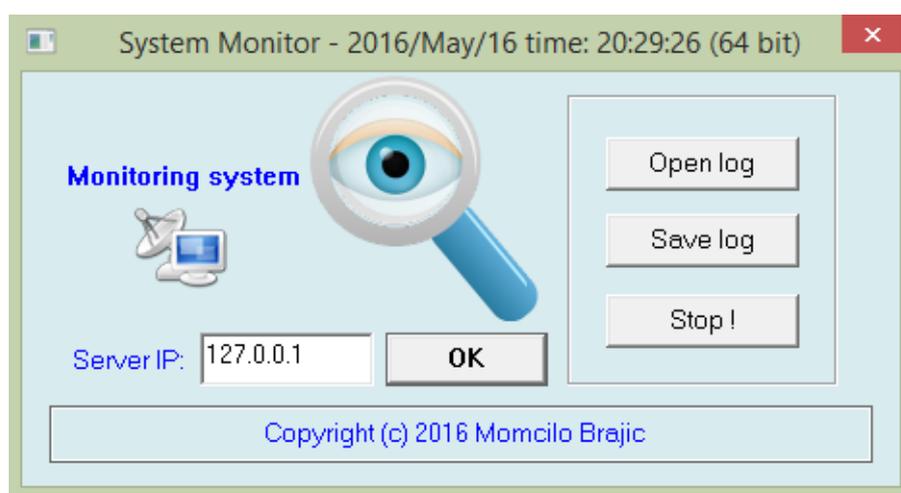


Fig. 1: The AppLog interface, activated using the taskbar icon

In the field Server IP, the address of the server where the Monitor is running must be introduced. If the server is not already launched, an error will be raised when the used select the *Ok* button, as we can see from the Fig. 2a.

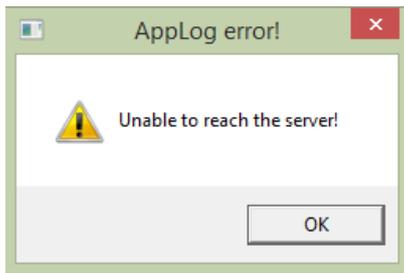


Fig. 2a: Server was not started

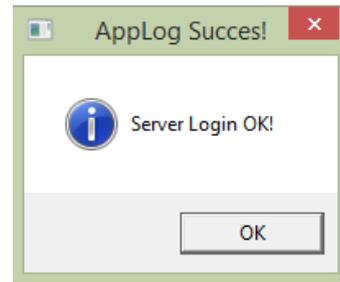


Fig. 2b: Successfully connecting to the server

If the Monitor is running and the IP address is correct, a successful connection to the server will be established (Fig. 2b). From this moment, all logged events are sent by the AppLog client to the Monitor, as can be seen from the Fig. 3. By selecting the *Stop* button, the user will stop monitoring of computer, and the AppLog application will be terminated. In addition to sending events to the Monitor, the client application builds a local encrypted log. Decryption and visualization of the local log is done by selecting the *Open* button. The local log can be copied (in the encrypted form) in a location selected by the user using the *Save log* button.

The AppLog client is minimized in taskbar after registration in server and its icon remain active in the Notification Area Icons, as is depicted in the Fig. 4.

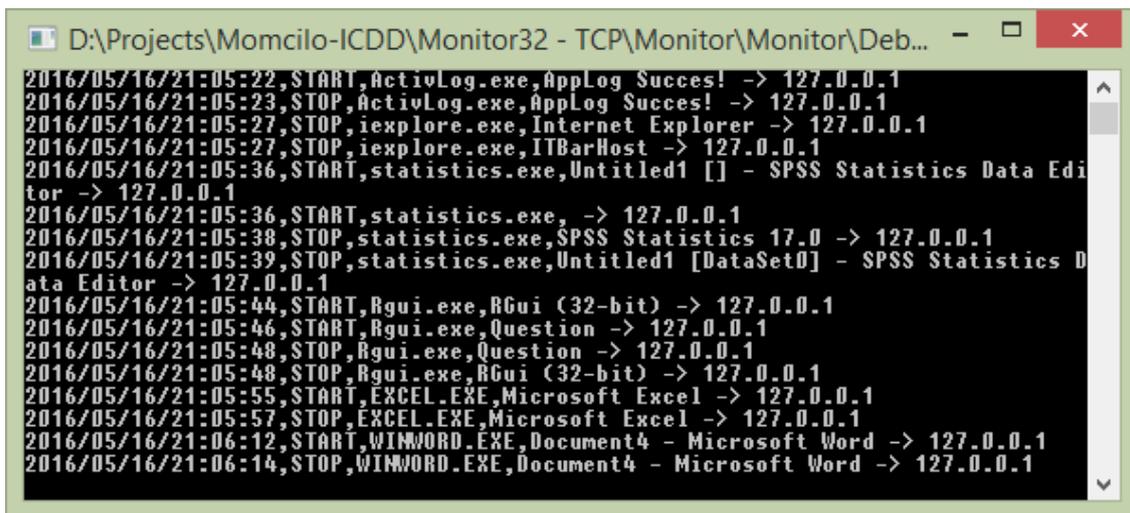


Fig. 3: Receiving events in the Monitor application

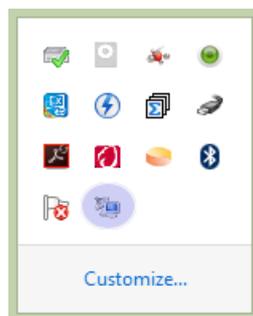


Fig. 4: The icon of the AppLog application

5.2 Technical details of implementation

The AppLog client installs a global hook through a custom dynamic-link library [5]:

```
DllExport BOOL InstallHook (void) {  
  
    g_hHook = SetWindowsHookEx(WH_SHELL, (HOOKPROC)ShellProc,  
                               g_hinstDll, 0);  
  
    if (g_hHook == NULL) {  
        MessageBox(NULL, __TEXT("Error occurred while installing monitoring!"),  
                  __TEXT("AppLog error!"), MB_OK | MB_ICONERROR | MB_SYSTEMMODAL);  
        return(FALSE);  
    }  
    return(TRUE);  
}
```

The *ShellProc* procedure is automatically called by the operating system when a window is created or destroyed [6]. The *wParam* parameter contains the handle of the respective window, and with *GetWindowThreadProcessId()* function we determine the process ID that contains that window [4]. Using the *Process32First()* and *Process32Next()* functions, the set of all active processes is scanned to find the respective process and extract further information about it.

```
LRESULT CALLBACK ShellProc (int nCode, WPARAM wParam,  
                            LPARAM lParam) {  
  
    DWORD processId;  
    PROCESSENTRY32 process;  
    HANDLE hSnap;  
    BOOL f;  
    HWND hwnd;  
    LRESULT retValue = 0;  
    char whatlog[2*_MAX_PATH];  
    char windowtext[100];  
  
    retValue = CallNextHookEx( g_hHook, nCode, wParam, lParam );  
    if (nCode < 0) return(retValue);  
  
    if (nCode == HSHELL_WINDOWCREATED ||  
        nCode == HSHELL_WINDOWDESTROYED) {  
  
        hwnd = (HWND)wParam;  
        if (!IsWindow(hwnd)) goto Ok;  
  
        time( &lt;time );  
        today = localtime( &lt;time );  
        strftime( timebuf, 128,"%Y/%m/%d/%H:%M:%S",today );  
  
        GetWindowThreadProcessId(hwnd,&processId);  
        hSnap=CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS,0);  
        if (hSnap==NULL) {  
            strcpy(whatlog,timebuf);  
            strcat(whatlog,"ERROR,");  
            strcat(whatlog,"<<<SYSTEM MONITOR>>>,Error in determining the launched process!");  
            LogFileName(whatlog);  
            SendFileName(whatlog);  
            goto Ok;  
        }  
        process.dwSize=sizeof(process);  
    }  
}
```

```
f=Process32First(hSnap,&process);
while (f)
    if (process.th32ProcessID==processId)
        break;
    else
        f=Process32Next(hSnap,&process);
CloseHandle(hSnap);
GetWindowText(hwnd,windowtext,100);
switch (nCode) {
    case HSHELL_WINDOWCREATED:
        strcpy(whatlog,timebuf);
        strcat(whatlog,"START,");
        strcat(whatlog, process.szExeFile);
        strcat(whatlog, ",");
        strcat(whatlog, windowtext);
        break;
    case HSHELL_WINDOWDESTROYED:
        strcpy(whatlog,timebuf);
        strcat(whatlog,"STOP,");
        strcat(whatlog, process.szExeFile);
        strcat(whatlog, ",");
        strcat(whatlog, windowtext);
        break;
}
LogFileName(whatlog);
SendFileName(whatlog);
}
Ok:
return(retValue);
}
```

Finally, information about the process launched / terminated is sent to the Monitor application using the *SendFileName()* function. The communication is achieved by calling functions from Winsock library:

```
BOOL SendFileName(char* message) {

    BOOL fResult;

    int sock;                /* socket descriptor*/
    struct sockaddr_in servAddr; /* server address */
    WSADATA wsaData;        /* The structure for setting communication via WinSock */
    int strMessageLen;

    if (!serverReady) return FALSE;
    if (!Connect()) return FALSE;

    if (WSAStartup(MAKEWORD(2, 0), &wsaData) != 0) /* Initialization of Winsock 2.0 DLL */
    {
        return FALSE;
    }

    /* Create a TCP socket */
    if ((sock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0)
        return FALSE;

    /* Building the structure for the server address */
    memset(&servAddr, 0, sizeof(servAddr)); /* reset */
    servAddr.sin_family = AF_INET; /* Internet address */
    servAddr.sin_addr.s_addr = inet_addr(server_ip); /* IP server address */
    servAddr.sin_port = htons(server_port); /* server port */
    /* establish connection to the server */
}
```

```
if (connect(sock, (struct sockaddr *) &servAddr, sizeof(servAddr)) < 0)
    fResult = FALSE;
else
{
    fResult = TRUE;
    strMessageLen = strlen(message);    /* determine the length of the string */

    /* send message, including the NULL character, to the server*/
    if (send(sock, message, strMessageLen, 0) != strMessageLen) fResult = FALSE;
}

closesocket(sock);
WSACleanup(); /* release resources of Winsock library */

if (!fResult) {
    // error message
    MessageBox(NULL, __TEXT("Socket communication error!"),
        __TEXT("AppLog Error!"), MB_OK | MB_ICONERROR | MB_SYSTEMMODAL);
}
return fResult;
}
```

The Monitor is a multi-threaded application, which can handle concurrent clients by creating a thread on each accepted connection:

```
for (;;) /* infinite cycle */
{
    /* length of structure containing client address */
    clntLen = sizeof(clntAddr);

    /* waiting client connection */
    if ((clntSock = accept(servSock, (struct sockaddr *) &clntAddr, &clntLen)) < 0)
        ExitWithError("accept() failed!");

    /* has been achieved a client connection */

    sprintf(clntIP,"%s", inet_ntoa(clntAddr.sin_addr));

    /* allocate memory for the arguments of thread */
    threadArgs = (PThreadArgs) malloc(sizeof(ThreadArgs));
    threadArgs->clntSock = clntSock;
    strcpy(threadArgs->clntIP, clntIP);

    if (CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE) ThreadMain, threadArgs,
        0, (LPDWORD) &threadID) == NULL)
        ExitWithError("CreateThread() failed!");
    //printf("Client connected %s; was assigned the thread: %ld\n", clntIP, threadID);
}
}
```

6 Conclusions and Future Work

In this paper we presented an application developed in C language which exploits intensively the Windows API services. Mainly, the SysMonitor application is based on Windows Hooks and Network Services. We intend to realize some future improvements: an automated connection of the client to the Monitor server, rewriting the server part as a Windows service, extending the class of monitored events, etc.

References

- [1] Mark E. Russinovich, David Solomon, Microsoft Windows Internals, Fourth Edition, Microsoft Press, 2005
- [2] Jeffrey Richter, Advanced Windows, Microsoft Press, 1998
- [3] Microsoft Developer Network (MSDN) - Windows API Index,
[https://msdn.microsoft.com/en-us/library/windows/desktop/ff818516\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ff818516(v=vs.85).aspx)
- [4] Charles Petzold, Programming Windows - Win32 Api 5Th Ed, Microsoft Press, 1998
- [5] Florin Stoica, Laura Florentina Cacovean, Programare API, Ed. Psihimedia Sibiu, ISBN 978-973-1753-75-1, 354 pages, 2009
- [6] Florin Stoica, Sisteme de operare, Ed. Universităţii „Lucian Blaga” Sibiu, ISBN 978-973-739-369-2, 331 pages, 2007
- [7] Microsoft Developer Network (MSDN) - Hooks,
[https://msdn.microsoft.com/en-us/library/windows/desktop/ms644959\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms644959(v=vs.85).aspx)

Momcilo Brajic
Faculty of Computer Science, Master
Bulevar umetnosti br. 29, 11070 Belgrade
SERBIA
E-mail: momcilob@live.com

Smart city and smart transportation. Bucharest case study

Crina Cristea, Alexandru Daniel, Darko Shuleski, Alexandru Birsan
Coordinator Professor: Ioan RADU

Abstract

“Non-smart” cities are particularly challenged today, if we consider transportation problems, meanwhile smart cities are able to find solutions to these problems. In spite of this, the concept of smart city is still undefined, even if there is abundant research on this concept and on the components of the smart city. This study aims to explain the manner in which a smart city offers a sustainable economic development and transportation, as well as a high quality life for its inhabitants. The perception of individuals on the current public transportation in Bucharest represents the main objective of the case study. According to our research, the inhabitants from Bucharest are highly unsatisfied with the public transportation, which results in their desire for better transportation and in solutions proposed for this reason.

1 Introduction

Bucharest is not a smart city. Consequently, its' transportation isn't smart either, which causes great problems to its inhabitants. These transportation problems represent the motivation for this article, especially since there are many smart cities in the world which have solved their transportation problems. This is the main reason why I firmly believe that, Bucharest too, can be converted into a smart city, and this research comes an original one which could help to improving public transportation in our capital and to transform Bucharest in a smart city. The paper is organized on two parts: literature review and case study. The first parte studies previous research on the concepts of Intelligent City, Knowledge City and Smart City focusing on the six-axes model that characterizes a Smart City, from the point of view of the European Cities Project. Also are being analyzed ITS and transportation systems together with problems related to transportation. The second part focuses on real time transportation problems in Bucharest and the way inhabitants perceive these problems.

2 Literature review

2.1 Intelligent City, Knowledge City, Smart City

Despite its similarity to the U-city concept, the concept of Smart City is more popular in the present. The difference between the two concepts is in the degree of intelligence and therefore the

Smart city is actually a more developed post-Ubiquitous city. U-city has an artificial intelligence and it is a city created from the conjugation of various basic facilities and information technology. As afore mentioned, the Smart city is a more developed version of the U-City and this development is due to the introduction of new technologies which make it possible for individuals to be interconnected like a human neural network. Hence, no longer dependent on a certain location, involvement and inter-communication are easier in a Smart city. In comparison with the digital or intelligent cities, the social and environmental capital draws a clear line between the Smart cities and the more technology-based digital or intelligent cities. Moreover, according to the literature on this subject, Smart cities are less dependent on the “physical capital” or what we can call the hard infrastructure and much more on the “intellectual capital and social capital”, the availability and quality of knowledge communication, thus the intellectual and social infrastructure [1].

However, the concept of Smart city remains flexible and continuously adapting to the ever-changing technologies. Undoubtedly, all areas on smart cities such as government facilities, health, traffic, electricity, water, buildings, and transport are covered by the information of communication technology [ICT].

Nowadays, SMART cities are called MESH cities [2]. MESH is an acronym for: Mobile [the mobile technologies and the informational networks offer the bottom-up, real-time information, feedback about a city, its systems and its citizens, the so-called users of the city], Efficient [sustainable cities that use effectively their resources and also monitor and manage public services like health, energy, traffic, etc], Subtle [discreet, non-intrusive, even invisible systems, user friendly modern city systems], Heuristics [heuristics-based constant development, self-improvement, continuous adapting and citizen-focused].

It is already predictable that the ICT will also enter the soft and warm techniques domain. Therefore, the Artificial Intelligence will no longer be stranger to human emotions and will be developed to act like human beings [3]. Today's Smart cities will be tomorrow's Sense, Soft and Warm Technology City. A classification of the definition of Smart cities from a subjective perspective is presented by Lee and Hancock [4]. The concept of Smart city has a wide variety of definitions and practical and academic views being nearly impossible to present an all-inclusive definition. Table 1 below presents the three categories of definitions. Given the numerous definitions and interpretations of the Smart City, the present study wishes to advance a comprehensive model to better explain the ideas and projects behind the Smart city. In the early years, the Digital cities were build on ICT, but nowadays the more developed and intelligent Smart cities have found new ways to engage technology and restructured in order to provide citizens more effective and efficient public services [5]. In the end, it all comes down to people because no matter the changes and transformations within a municipality, customer-centricity remains the true spirit of a Smart city. Technology is a means to an end, but it all depends on the customers experience with that technology; if the customer doesn't like it he won't use and that e-service will fail despite its innovative nature. Therefore, municipal employees must always focus on their relations with the customers especially if they directly face citizens every day, but even if they don't. On the other hand, citizens also must collaborate with their city authorities and contribute in order to help e-services to deliver its promises. Citizens that in this case become customers are the main focus of the Smart city and some even joke that there may even be an app to prove it.

Returning to the definition of the Smart city, it can be considered as the next step in the evolution of urbanization and Digital City Development, where the central point remains the ICT infrastructure but other factors become equally important and we must mention Customer Centricity and Governance, Civic Engagement, Human and Social Capital, and Environmental Interest.

Smart City Projects - Mechanisms and approaches to define the concept

The afore-mentioned definitions of the Smart city are only few of the attempts to better explain this concept. Starting from the volatile nature of the Smart city definition, some researchers consider best to methodize the features and characteristics, highlighting the most important parameters of the Smart city project. One of these models described by researchers will be presented further on in this paper.

2.2 Intelligent City, Knowledge City, Smart City

The smart city model presented by European Cities Project defines a Smart City as a city well performing in 6 main characteristics, built on the ‘smart’ combination of endowments and activities of self-decisive, independent and aware citizens.

SMART ECONOMY [Competitiveness] <ul style="list-style-type: none"> • Innovative spirit • Entrepreneurship • Economic image & trademarks • Productivity • Flexibility of labor market • Embedded Internationally • Ability to transform 	SMART PEOPLE [Social and Human Capital] <ul style="list-style-type: none"> • Level of qualification • Affinity to life long learning • Social and ethnic plurality • Flexibility • Creativity • Cosmopolitanism/ Open-mindedness • Participation in public life 	SMART GOVERNANCE [Participation] <ul style="list-style-type: none"> • Participation in decision making • Public and social services • Transparent governance • Political strategies & Perspectives
SMART MOBILITY [Transport and ICT] <ul style="list-style-type: none"> • Local accessibility [Inter-] national accessibility • Availability of ICT infra • Sustainable, innovative and safe transport systems 	SMART ENVIRONMENT [Natural resources] <ul style="list-style-type: none"> • Attractiveness of natural conditions • Pollution • Environmental protection • Sustainable resource management 	SMART LIVING [Quality of life] <ul style="list-style-type: none"> • Cultural facilities • Health conditions • Individual safety • Housing quality • Education facilities • Touristic attractiveness • Social cohesion

Table 1: *The six-axes model to characterize a Smart City, the view of the European Cities Project*

The table above illustrates a perspective on Smart cities that bring together different economic, social or environmental factors and components. The success of the project consists in optimally integrating these components having the innovative technologies as a binder. To summarize, the main features of a Smart city are:

1. access to enhanced information flow for citizens and services providers;
2. maximizing the use of key resources according to the data collected through high-tech sensors, real time data analytics and widespread communications;
3. systematization of heterogeneous data in order to provide productivity insights and augmentation to its citizens and service providers;
4. maximizing the economies on the different layers of infrastructure using a service delivery platform, for instance Urban Operating System [“Urban OS”];
5. exceeding economic targets by engaging innovative technology and innovation in order to offer its citizens, industry and business environment sustainable, quality of life improvements.

”Smart city” is a label or better yet, a status that many cities around the world aspire to have. In this context, we can draw a series of interesting observations. Firstly, across the world there are numerous acceptances of the concept of “Smart city” and while all of them are centered on the essential role of ICT and new technologies, these cities apply differently the technologies. For example, most of them use ICT for utilities, transportation, citizen engagement or environmental issues but few of these cities [e.g. Edinburgh, Boston, Nanjing, Syracuse, etc.] have considered urbanism or city planning to be adequate fields of application. Secondly, Smart cities are heterogeneous in terms of evolution, some being more developed than others. While some cities like Stockholm, Singapore, Rio de Janeiro or Amsterdam, have already implemented and are using new systems, others remain in a conceptual phase. Thirdly, approximately one in three cities is collaborating with some of top global technology providers like CISCO, Siemens and IBM. However, not all cities are organized and while some are working with regional or local technology providers and have partnerships with research institutions, others still have to determine such matters. Lastly, researchers claim that all these efforts to become smart have been intensified in the last decade. Even though city planning and management have been using technology for years, the present “smart” trend is focused on cities’ branding as “smart”. Therefore, technology providers are making efforts to establish partnerships with these cities.

Short review on Global Technology Providers

According to ABI Research [6] the market for technologies that feed into and support Smart City programs and projects will grow on a global basis from \$8 billion in 2010 to exceed \$39 billion in 2016”. On the other hand, Pike Research [7] is more optimistic and estimates that “investment in smart city technology infrastructure will total \$108 billion during the years from 2010 to 2020”. As research shows, global technology providers are very interested in establishing partnerships with cities. Precisely, top technology providers such as Cisco, IBM and Siemens have intensified their efforts to determine cities in adopting innovative technologies and ICT. The efforts of IBM, Cisco and Siemens focus on sustainable development, improving and enhancing city services, efficient use of energy and resources. Next will be outlined the premises of the partnerships and collaborations between the top technology providers and Smart cities.

The top technology providers: IBM, Cisco and Siemens. Viewing the city a "system of systems", IBM [8] highlights the role of ICTs in facilitating the understanding of these systems and they way they work, therefore enabling better decision-making for planners and policymakers. This idea is based on Harrison's and Donnelly research that argue that ICTs is an useful instrument for observing and measuring and then aggregating the “choices and construction” of city systems [9]. IBM has envisaged and customized solutions for cities in a variety of city sectors such as energy and utilities, transportation, healthcare, airports, communications, education, social services, economic development and public safety. IBM argues that actually “smart is a verb” and therefore encourages decision making processes “based on evidence, not on habit or opinion or gut”. They highlight that the successful cities, in particular city leaders, of the future are those “using even more powerful tools, sharpened by ever more precise information and insight,... are creating the big transformational shifts that reverberate through their organizations”, and who are "anticipating, rather than merely reacting to events”, and “seizing competitive advantage” [10].

Taking it a step further, IBM has created the Smarter Cities Challenge program, a \$50 million dollars program, where together with city leaders are finding more effective and efficient solutions for the cities’ stringent problems. An illustrative example is that of Syracuse, New York that was confronted with an outflow of jobs and population leading to an increase of vacant properties and IBM is trying to develop a method to observe, analyze and predict the vacancy phenomenon evolution. In long term, the system proposed by IBM will help city officials in neighborhood planning and development [11].

CISCO [12] presents the solutions cities find to their problems “by using networked information to transform urban centers into networked communities”, i.e. “smart + connected communities”. One method is with the help of a network “underlying services delivery platform” that “connects

everyone to everything”. Through this network they create and implement solutions that make citizens, businesses and governments “realize sustainable economic growth”, “enable environmental sustainability through resource management and operational efficiencies”, and “enhance quality of life”. The smart solutions envisioned by technology providers like Cisco refer to transportation, real estate, utilities, safety and security, housing and government sectors. In Amsterdam, Cisco’s “smart work centers” are conference rooms and hot-desk office spaces with high-tech, innovative networked telecommunications systems for working and collaboration, for instance TelePresence [13]. These work centers are accessible to companies and/or workers who can flexibly rent them and were created aiming to minimize and even eliminate commuting. If these work centers prove efficient they will positively impact transportation and land use planning.

The Smart + Connected Communities Institute is another program implemented by Cisco in order to offer a “central hub for city planners, developers, academic institutions, systems integrators and visionary leaders in which to collaborate on the issues involved in the [re]development of sustainable cities” [14]. This Institute brings together online forums and blogs and also releases interesting publications such as reports, case studies, white papers and presentations on subjects like the application of new technologies and the ICTs in city government.

Siemens concentrates its efforts on urban sustainability. Siemens works together with the Economist Intelligent Unit and publish “The Green City Index”, a series of studies on over 120 cities and their policies and best practices in achieving urban environmental sustainability. The Crystal is the world's largest exhibition set up by Siemens in London centered on urban sustainability, aiming to offer a “global knowledge hub that helps a diverse range of audiences learn and understand how we can all work to build better cities for ourselves and for future generations” [15]. Taking this into consideration, Siemens presents ICTs as a “fifth utility”, an essential component of “smart cities” that “can enable increased efficiency and flexibility to use new resources” to “enable sustainable behavior” [16]. Therefore, the main focus is on integrated technological solutions for sustainable urban infrastructure, in particular, in transportation [e.g. traffic management, electric vehicles infrastructure, high-speed trains], energy [e.g. energy storage, smart grids], environmental, water [e.g. water treatment], green building [intelligent building solutions], healthcare sectors.

2.2.1 ITS - Intelligent Transportation Systems and Services

ITS is an abbreviation for Intelligent transportation systems and services and is starting to be a very important part of a Smart city. In his paper, Salido and his collaborators argue that in the present the traffic is chaotic because of the frequent congestions and accidents, pollution and unpredictability. These are serious concerns worldwide and despite some recent improvements, the overall situation is aggravating [17]. In achieving smart, sustainable growth, intelligent transportation systems have been considered one viable alternative [18]. Mondale [19] also thinks that transportation is essential in the competitiveness of cities and the quality of life. Additionally, he highlights that it can also positively impact infrastructure expenses and also helps businesses and building flourishing neighborhoods. ITS-Finland claims that “Intelligent Transport Systems and Services [ITS] encompass a broad range of information and communications technologies that improve the safety, efficiency, and performance of the transportation system. When integrated into the nation's roadways, vehicles and public transportation network, ITS can help reduce congestion, improve mobility, save lives and optimize our existing infrastructure”. Taking this in consideration, it is a logical development that smart transportations’ importance is growing.

Similarly, in his study, Singh [20] argues that the new mega cities resulting from urbanization will radically impact transportation. He predicts that technology-push perspective will turn into a user-pull claiming that in the future cars will be designed around cities, not cities around cars. Therefore, companies and other service providers will rethink their new perspective from market share to “mobility share” because of the new types of business models and services that will

emerge [21]. Moreover, as urbanisation continues transport-related infrastructure will be the centre of all infrastructure development and spending [22]. Therefore, there is demand for innovation in transportation solutions and people's transportation routines will also change as the evolution of cities, technologies and smart environments continues.

Even though intelligent transportation has many advantages and benefits the vast majority of citizens, the new solutions aren't so appealing for everyone. For instance, a Finnish expert declared at some point that in respect to intelligent transportation solution plans the user perspective has been disregarded and the city officials as well as technology providers are ideating innovative solutions from a technology-push perspective. Therefore, a strong top-down approach is counterproductive because citizens are unlikely to accept the new solutions when those are almost forced by public administration. Hence, a bottom-up approach, from citizens to city officials, may prove to be more successful in adopting innovations and new technologies. Resistance may be minimum if users, thus the citizens, would be able to express their opinion regarding the design of new transportation solutions. However, if transportation would be design only from the users' perspective, the resulting solutions may not challenge the existing transportation habits or be actually innovative. In this case, a combination of the two approaches should be considered in order to meet both the interests of users and service providers. Nevertheless, users have not traditionally been included in the process of designing intelligent transportation solutions, but now it appears to be demand for user-inclusive approaches.

2.2.2 Transportation Systems and Problems related to Transportation

City centres are predisposed to significant transport problems mainly because mobility requirements exceed transport systems capacity. Productivity is very dependent on the efficiency of the transportation system for taking workers to their jobs and for transporting people and various goods between different destinations. Additionally, ports and airports are important transportation pivots that raise other specific problems. Congestion is one of the oldest problems but new ones appear such as the environmental effects. In the following I will present the most noticeable urban transport problems.

- **Lack of Parking Space and Traffic Congestion:** Congestion is one of the top issues when it comes to transportation in metropolitan areas. The continuous spread of motorized transport and increased number of private cars determined an even larger demand for transport infrastructure. Nonetheless, infrastructure is not able to meet the need generated by traffic growth. Congestion is also related to parking troubles because looking for parking and not being able to find a space creates additional delays and hinders local traffic circulation. Looking for a parking place may take drivers even 20 minutes in central areas of large cities.
- **More Time spent Traveling:** Nowadays, the transition between the place of residence and place of work is taking a lot more time. This trend is link to housing prices; the most affordable houses are located away from central areas.
- **Inadequate Public Transportation:** Public transport systems don't meet demand and create inconvenience for users during peak hours. Especially in suburban areas, a decrease in passengers' numbers makes public service financially unsustainable and thus incapable to generate sufficient income to cover its operating costs and capital.
- **Environmental Impact and Energy Consumption:** The quality of life and health of urban residents is seriously affected by the pollution including noise generated by the rotation. Another form of pollution is energy consumption by urban transport.
- **Accidents and Safety:** Increased traffic in urban areas generates an increased number of accidents and fatalities, especially in developing countries. Repeated delays are also linked to accidents and with the continuous increase in traffic, people feel unsecure in using the streets. [23].

3 Case Study

3.1 Research Objectives

The first general objective of this research aims to reveal if there are significant problems in the transportation system in Bucharest. The second general objective of this research aims to highlight significant solutions in improving the transportation system in Bucharest.

3.2 Participants to the study

Subjects on whom research was carried out are 160 individuals living in Bucharest, from either private or government organizations.

Organization	Hierarchical level	N valid
Public organization	Management	7
	Operational Employees	73
	Total	80
Private organization	Management	8
	Operational Employees	72
	Total	80

Table 2: Distribution of participants to the study after "the hierarchical level"

3.3 Questionnaire

The questionnaire is made up of 4 questions addressed to the demographic features of the participants to the research and 6 questions dealing with challenges of a Smart City and a smart transportation.

3.4 Analysis and interpretation of results

According to the research participants, the biggest problem of the transportation in Bucharest is that of traffic congestion, followed by the lack of parking space and the time dedicated to travelling in Bucharest, which is really high.

Traffic Congestion	68
Lack of Parking Space	47
More Time spent Traveling	34
Inadequate Public Transportation	11
Environmental Impact (Pollution)	17
Accidents and Safety	3

Table 3: The biggest problem of the transportation in Bucharest

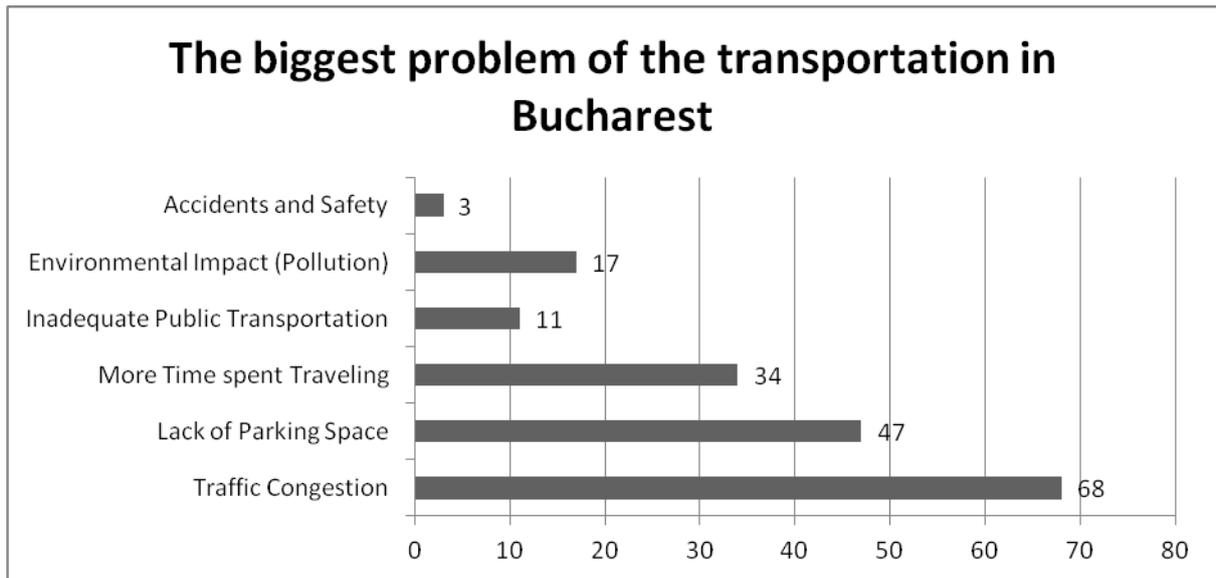


Fig. 1: The biggest problem of the transportation in Bucharest

In terms of solutions to the transportation problems in Bucharest, participants focused on sustainable transport and infrastructure necessary for the achievement of smart mobility, together with intelligent logistics, dynamic traffic management concepts, as well as a network of recharging points for electric vehicles throughout Bucharest. Carpooling, referring to sharing a car to a destination to reduce fuel use, pollution and travel costs is slightly considered by people living in Bucharest.

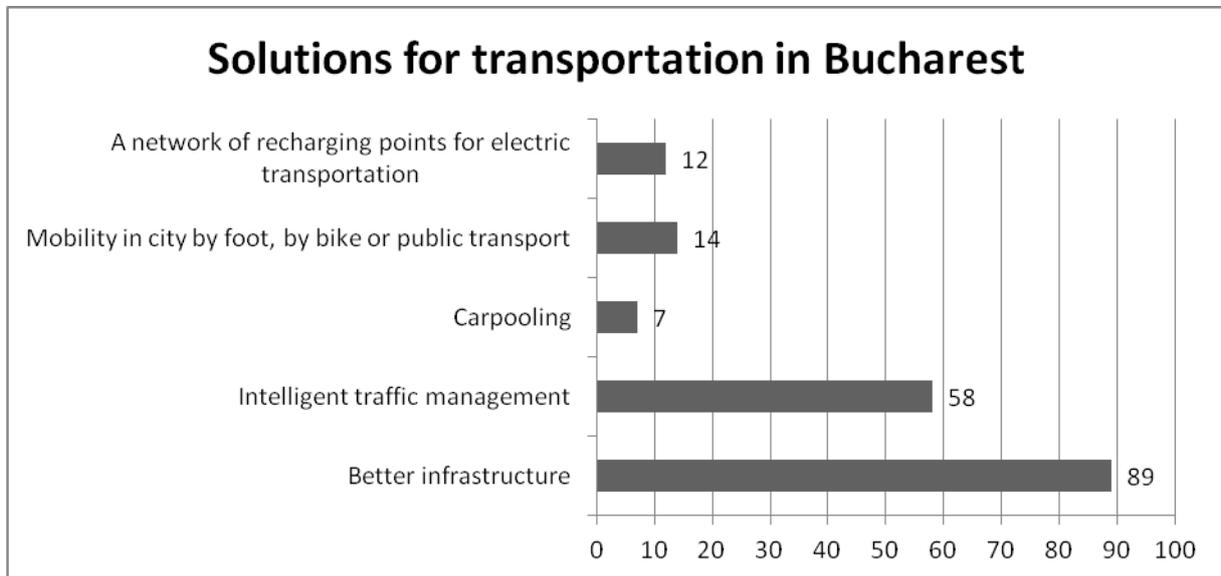


Fig. 2: Solutions to the transportation problems in Bucharest

4 Conclusions

From our research, Bucharest is far from being a smart city. The article focused mainly on transportation challenges of the capital, providing several solutions, which, if considered carefully could improve transportation in Bucharest. Among these challenges can be mentioned: infrastructure, time, road maintenance, congestion, bike community, digital softs, public transport, electrical vehicles, traffic management and others. As major solutions proposed by participants are smart mobility, intelligent logistics, recharging points for electric vehicles as well as carpooling. This research can represent a starting point for government authorities, urban planners, IT&C researchers and policy-makers in the development of cities. As main contributions can be mentioned the analysis of the different challenges in the transportation of Bucharest, as well as the proposal of several solutions, which if implemented could increase the quality of life in Bucharest and to a certain point, even turn our capital into a smart city.

References

- [1] Caragliu, A., Del Bo, C., Nijkamp, P., *Smart Cities in Europe*, 3rd Central European Conference in Regional Science – CERS. 2009.
- [2] Komninou N. *MESH Cities*. Research report on Digital Cities, Cyber Cities. www.urenio.org. 2011.
- [3] Shin J. *ICT Leadership Toward Human-centered Technology*. Korea IT Times, Sept. 24th, 2012.
- [4] Lee, J., H., Hancock, M. G., *Toward a framework for Smart Cities: A Comparison of Seoul*, Yonsei University. Seoul. Korea. 2012.
- [5] Foley P. *Defining Smart Cities*. Workshop 2: going smart and accessible in public services and cities, Digital Agenda for Europe. 2013.
- [6] ABI Research. *Smart City Technologies Will Grow Fivefold to Exceed \$39 Billion in 2016*. 2012. Available at: <http://www.abiresearch.com/press/smart-city-technologies-will-grow-fivefold-to-exce>
- [7] Pike Research. *Global Investment in Smart City Technology Infrastructure to Total \$108 Billion by 2020*. 2011. Available at: <http://www.pikeresearch.com/newsroom/global-investment-in-smart-citytechnology-infrastructure-to-total-108-billion-by-2020>
- [8] IBM. *Welcome to The Smarter City*. 2010. Available at: http://www-03.ibm.com/innovation/us/thesmartercity/index_flash.html.
- [9] Harrison, C. & Donnelly, I.A. *A Theory of Smart Cities* in Proceedings of the 55th Annual Meeting of the ISSS. 2011. pp. 1-15. Disponibil la: journals.issis.org/index.php/proceedings55th/article/view/1703.
- [10] IBM. *Changing conventions: city leaders*, 2012. Available at: <http://www.ibm.com/smarterplanet/global/files/us-enuscities-city-leaders-wsj.pdf>.
- [11] IBM. *Syracuse Summary Report*, 2011. Available at: smartercitieschallenge.org/...reports/SmarterCities-Syracuse.pdf.
- [12] Cisco. *Industry Solutions: Overview - Transform Physical Communities*. 2012. Available at: <http://www.cisco.com/web/strategy/smart-connectedcommunities/overview.html>
- [13] Cisco. *European City Connects Citizens and Businesses for Economic Growth*, 2011. Available at: www.cisco.com/web/strategy/docs/scc/cisco-amsterdam-cs.pdf.
- [14] Smart + Connected Communities Institute. *About Us - Smart + Connected Communities Institute*. Available at: <http://www.smartconnectedcommunities.org/community/partners>, 2012.
- [15] The Crystal. *About the Crystal*, 2012. Available at: <http://www.thecrystal.org/html/about/about.html>.
- [16] Siemens. *Smart Cities and Sustainable Technology*. Presentation. 2010. Available at: http://www.seai.ie/NewsEvents/PreviousSEAIevents/The_roleofSmartCities_/Liam%20Mulligan%20,%20Siemens.pdf.
- [17] Salido, M.A., Peinado, M. & Giret, A. *A holonic simulation environment for smart transportation systems*. International Journal of Production Research, vol. 49, no. 5, 2011. pp. 1425-1439.
- [18] Bochner, B.S. *Smart growth tools for transportation*. Institute of Transportation Engineers. ITE Journal. vol. 70. no. 11. 2000. pp. 26-29.

- [19] Mondale, T. *Transportation - A Major Player in Smart Growth*. Institute of Transportation Engineers 70(11).pp 39-43, 2000.
- [20] Singh. R. B. Abhineety G., *Sustainable Forestry in Mega-cities of India for Mitigating Carbon Sequestration: Case Study of Delhi*, Presented paper in International Workshop on Developing Asian Mega Cities toward a Sustainable World, at Zhangjiajie, China during April 25-27, 2012.
- [21] Singh. R. B. Abhineety G., *Sustainable Forestry in Mega-cities of India for Mitigating Carbon Sequestration: Case Study of Delhi*, Presented paper in International Workshop on Developing Asian Mega Cities toward a Sustainable World, at Zhangjiajie, China during April 25-27, 2012.
- [22] Singh. R. B. Abhineety G., *Sustainable Forestry in Mega-cities of India for Mitigating Carbon Sequestration: Case Study of Delhi*, Presented paper in International Workshop on Developing Asian Mega Cities toward a Sustainable World, at Zhangjiajie, China during April 25-27, 2012.
- [23] Jean-Paul, R., *The Geography of Transport Systems*. University of Denver USA and the Journal of Economic Geography. 2013.

Crina Cristea
Bucharest Academy of
Economic Studies
Management
Piata Romana 6, Bucuresti
010374
Romania
crina.cristea@gmail.com

Alexandru Daniel
Bucharest Academy of
Economic Studies
Management
Piata Romana 6, Bucuresti
010374
Romania
dan74a@yahoo.com

Darko Shuleski
Bucharest Academy of
Economic Studies
Management
Piata Romana 6, Bucuresti
010374
Romania
darko.shuleski@man.ase.ro

Alexandru Birsan
Bucharest Academy of
Economic Studies
Management
Piata Romana 6, Bucuresti
010374
Romania
birsan.alexandru@gmail.com

Structuring business process models

Mykhailo Dorokhov
Coordinator Professor: Marlon Dumas

Abstract

Nowadays, the trend is moving towards developing the systems that are not brand new, but created to substitute the existing ones. These systems are going to follow the requirements, which are partly inherited from the previous systems. That means, that analyst can not only create the behaviour model (in particular BPMN models), but also try to extract it from the event log, using the automated process discovery techniques [4]. However, making these results of a human-readable form is a significant issue, since the models on the output are not the best tuned for the better human comprehension, they are not only usually flat, but they also are not composed of a blocks that are easy to understand and read [3]. Thus, having an automated means of structuring the process models would significantly facilitate the whole process of the inheriting requirements and analysing them.

1 Introduction

In the current information technology era we are observing a significant raise of the complexity of the systems around us. It is not only a quantitative, but more importantly a qualitative change. Starting from the 60s, information technology was step-by-step reaching more and more fields it can support by automation, information storing and centralization, etc. As software engineering technology level increased, the fields that were before impossible to support becoming opportunities. We have seen the simple control systems, even semi-automated, then the databases, industrial complexes, Material Requirements Planning systems, and finally appearing of the their successors – Enterprise Resource Planning systems [11].

Currently, a lot of the systems that are functioning in the enterprises are getting older not only in terms of the age from its first deployment, but from the perspective of approaches to the business, business rules and processes that have been changed. Companies are trying to upgrade the systems they have to follow the changed happened, and to form a firm ground for the further development. To it is obvious to predict that today's trend in the information systems development will be a redeveloping, creating or enhancing the new solutions where existent ones were not enough to fill all the requirement business or government have as of now.

In Estonia, such a problem exists only partially. A lot of systems started to appear only after 1990s, and particularly the government's X-road was created in 2000s [10]. But for the companies from the western countries, this redeveloping will mean working not from the scratch, but taking into the consideration existing expertise, existing processes, rules, frameworks and flow that are taking place in the current systems, analyzing them, and tuning or updating according to the new needs and features. Business analyst will have to dig into the current system, which could have been developed a long ago, without any documentation left, or in the mainstream language of that time, which is reducing the effectiveness of white-box analyzing of the code to figure out all the

possible flows and cases. Thus usage of automated discovery techniques is the right choice in this situating. As the system was working for a significant time, having the system log from that time could be used to discover the process in a form of Petri Net or a BPNM model.

This idea suits to the vision of the future development of the process mining in the business process management field, introduced by Dumas [6], to define the operation in the process mining and its relation (figure 1).

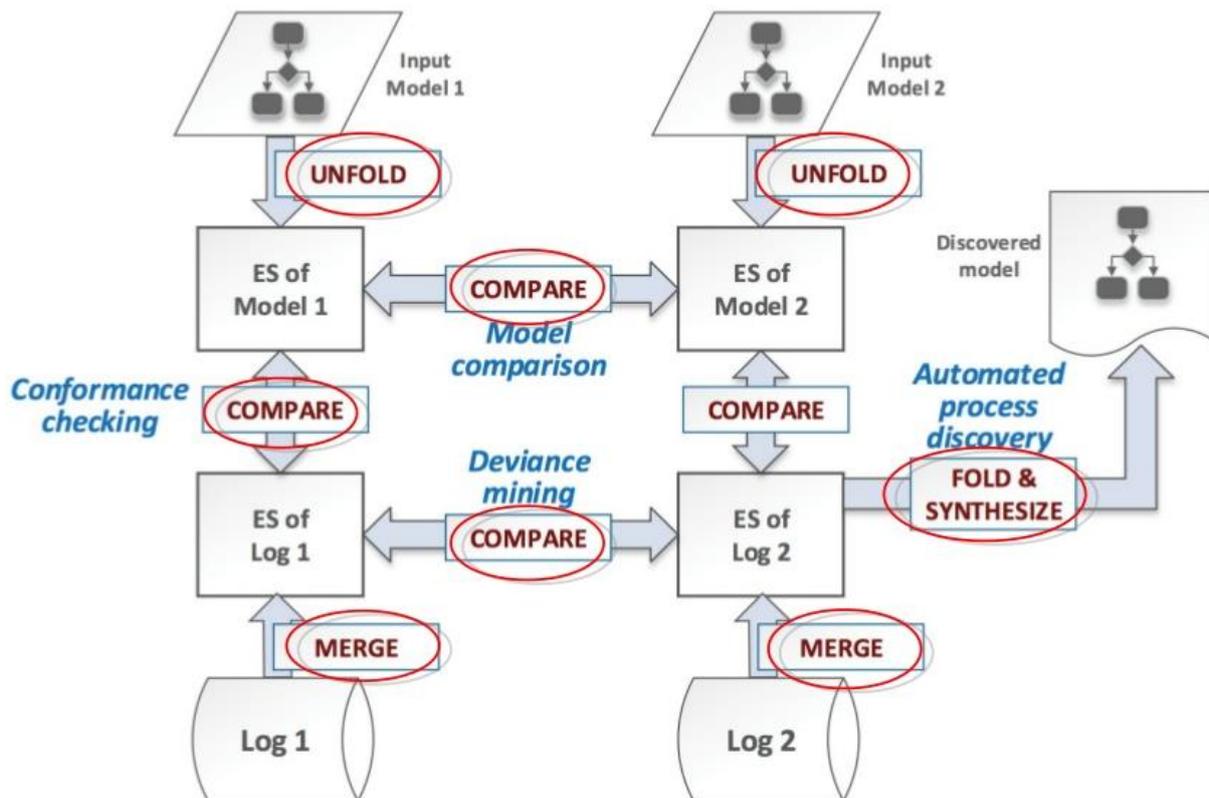


Fig. 1. Process Mining Reloaded by Dumas [6]

According to the Process Mining Reloaded set of operations, the automated process structuring is a perspective field of researches that can take its place right after the model discovery, to assure the human-readable result over the automated process discovery outputs, and can help in the comparison of the processes' event structures.

2 Automated process models structuring

The following research will be organized in the several parts. In the introduction to the question, motivation and research question were stated. In the main part, the literature on the topic is going to be reviewed, main ideas and issues are going to be identified and considered. The existing tools set on the issue of automated models structuring is going to be highlighted, as well as the real-life scenario on the use of such techniques. In the summary section conclusions are going to be given about the current state of the art in the field of structured process modelling, as well as existing gaps are going to be summarized together with the future development proposals.

Conducting the research was possible with the help of University of Tartu Library, and online databases, such as ScienceDirect and Springer, that contain a lot of the business process management, modelling, discovery, analysis and structuring literature.

2.1 BPMN as a process documentation notation

Both for Business Analyst and domain experts, the convenient way to illustrate the process is using the BPMN notation. It is intuitive, well-understandable, and if well-structured – it can easily present even a complicated processes to the people who haven't seen the notation before.

BPM is a behavioral modeling type of the process modelling. In the late 1980s it was a need to visualize and describe processes in enterprises, and companies started to dealing with BPM in order present the processes in the understandable manner to the management, analyze than, find inefficiencies that can cause losses, optimize business structure and processes. Using a business process model that illustrates the situation that is current - redesign and optimization could be performed to improve the business organization overall. This process is called business process reengineering.

Business Process Model and Notation (BPMN) is a graphical representation that allows to represent the business processes as models. The most current version of BPMN (Version 2.0) was released in January 2011 [13]. The name at that time was changed to Business Process Model and Notation, because besides the notational and diagramming elements there was execution semantics introduced.

A BPMN diagram consist of the following elements:

- 1) Flow objects: events, activities, gateways
- 2) Connecting objects: sequence flow, message flow, association
- 3) Swim lanes: pool, lane
- 4) Artifacts: data object, group, annotation

You can observe a simple BPMN diagram on the figure 2.

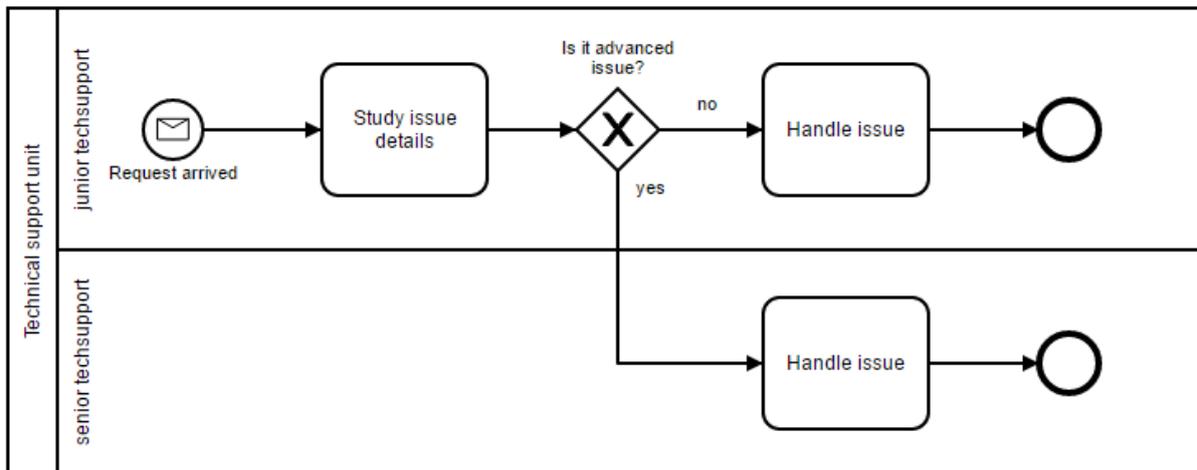


Fig. 2. Example of the BPMN diagram

The advantage of using BPMN as a part of documentation is the fact that it has a lot of applications and algorithms developed, having even automated means for the process discovery [5]. The latter plays an important role in the vision called Process Mining Reloaded that is going to be elaborated from the idea presented in the Introduction. Automated process discovery play an important role here, as they can make the process of automated documentation and it analysis possible alongside with other techniques, like automated feature extraction based on the natural language processing algorithms [1], and what is most important here it can save the time and resources as for the analyst work and improve the overall process of creating and analyzing requirements [3].

2.2 Necessity of structuring the process model

Since there are a lot of outdated systems, we already have some information about existing flows that there are implemented in the system. The problem of the white-box extraction from the code exists due to the big shift in the programming language usage. So using logs from the system seems to be a right idea, since from the time it is being used there should be a lot of them left. The big amount of the logs left is also a big advantage for the auto-mated discovery algorithms [4].

After the discovery is done, we have either a Petri Net or a BPMN model. The vast majority of the business process discovery techniques can restore the BPMN model in a „flat“ appearance, so that they cannot properly capture error handling and repetition constructs [5]. Usually, hierarchical processes discovery produces complex and unreadable models. So for the final stage of discovery – automated structuring is the right thing to develop and apply [9]. Having structured model is the naturally logical step to perform after the BPMN model restoration from the event log. The quality of this model is crucial, because it is the point when the improvements, changes and tuning are going to be added, and according to the butterfly effect, the more understandable and structured the model is before this trans-formation – the more successful its redesign will be.

To define the notion of the structuredness, let's use the following definition: the model is well-structured if it contain SESE block [7]. Single-entry-single-exit block (SESE block) is a block that have single entry point, so that splits and joins are always paired. The model that is well-structured is more readable and is much easier to understand and analyze. The following example (figure 3) shows that the second model will be more intuitively understood even for the person not aware of the BPMN notation, where the first model will require certain effort to understand even for the person working with BPMN.

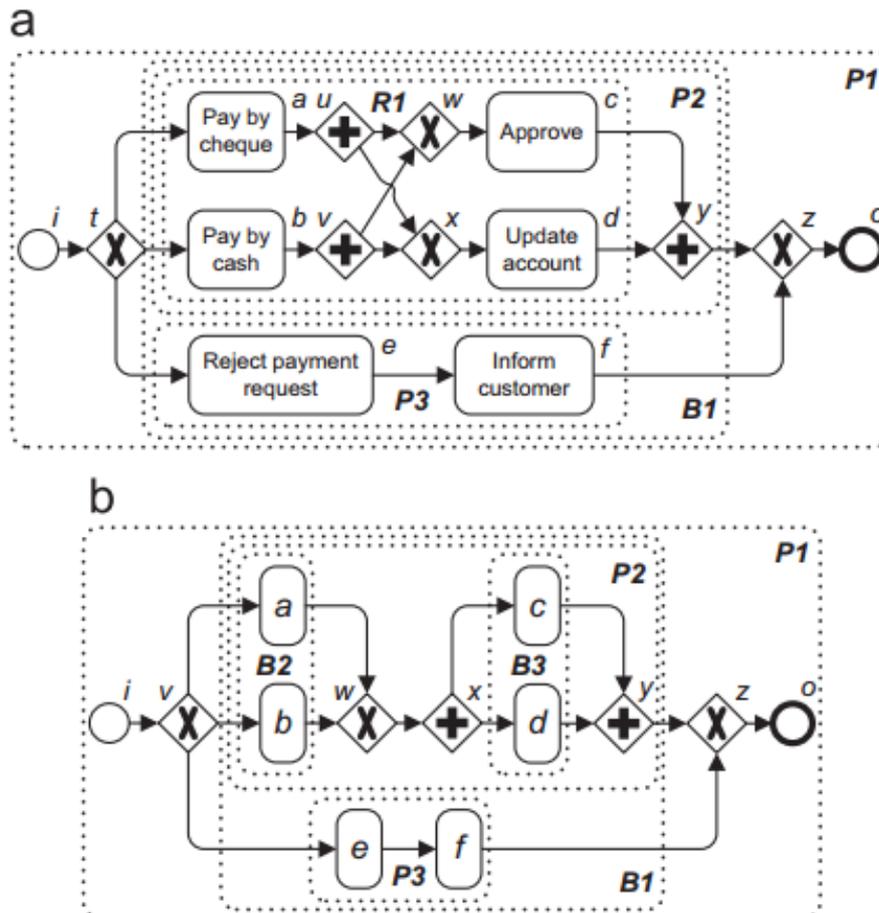


Fig. 3. Example of unstructured (a) and structured (b) process models from [14].

When all the advantages of having the model clearly structured are that obvious and understandable, why then the whole problem still exists? The issue is not only in the auto-mated process discovery methods, but also in a human factor. No one can force all the analyst or expert to follow some strict guidelines of creating model only in the allowed way, and even in this case it is not possible to control it then. So a reasonable solution is to have automated methods that can deal with unstructured models and convert them into structured ones.

The structuring process models problem has been studied quite extensively since the early 2000s [8]. It has its roots in compiler theory and code refactoring. It started with the first attempt at classification of unstructured process models and defining whether they could be transformed to structured equivalents. As a result it was concluded that it is not always possible to create the behavior-equivalent structured model from the unstructured one [12]. The main breakthrough was made with the paper of Polyvyanyy, García-Banuelos and Dumas [14], where a complete description on the algorithm of converting unstructured acyclic process models was provided, and all its applications and limitations were mentioned. It can deal with a majority of the acyclic process models, and can identify the cases it is not capable to create the well-structured process model.

2.3 Automated structuring algorithm

The algorithm provided in [14] can process only the acyclic process models, so that the model should have a single start element (e.g. a start event) and a single end element (e.g. an end event). Next, the process model is decomposed into SESE subgraphs (single-entry, single-exit) regions, as can be seen from the figure 3. The above is done with the RSPT (Refined process structure tree) decomposition developed by Vanhatalo, Völzer and Leymann [15]. There exist four types of SESE subgraphs in the RPST form presented as:

- 1) Trivial (a single node)
- 2) Polygon (a sequence)
- 3) Bond (diamond shape, when one entry point with all paths converging into the exit point without any crossing)
- 4) Rigid (any other subgraph with crossing edges)

If the model consists of SESE subgraphs of type 1, 2 or 3 only it is an indicator that the model is already structured. In other cases, the algorithm traverses the RSPT, processing every subgraph in a bottom up way to find a rigid. When it happens, it tries to restructure the rigid, and to replace it with the resulting graph, continuing over. If the rigid has only XOR gateways, the techniques developed for compilers, mentioned earlier, could be applied. In the other cases, the rigid is converted into a Petri net, which is then being unfolded. The set of ordering relations is computed in the unfolding, applying the modular decomposition.

The modular decomposition comes from the graph theory, it decomposes a graph into sub-sets of so-called modules. A module is an abstraction of a connected component of a graph. One module can be a proper subset of another, therefore, as opposite to just a partition, the module structure leads to a recursive decomposition of the graph. The modular decomposition results in also in a tree. The tree has 4 types of nodes:

- 1) Trivial (one single node)
- 2) Linear (the underlying relation is transitive)
- 3) Composite (blocks of concurrency/conflict)
- 4) Primitives (everything else)

A linear corresponds to a sequence (Polygon), a composite to a bond (Diamond shape), and a primitive maps into a Rigid. One should notice here that the modular decomposition is unique and there could be no several version of the resulting graph. This advantage of the modular decomposition appearance is used to determine if there is a possibility of conversion into the structured model. As an indication serves the primitive node – if it is found that means that there are no structured version exists, and algorithm cannot create a structured process model.

The algorithm uses a following decomposition hierarchy (figure 4).

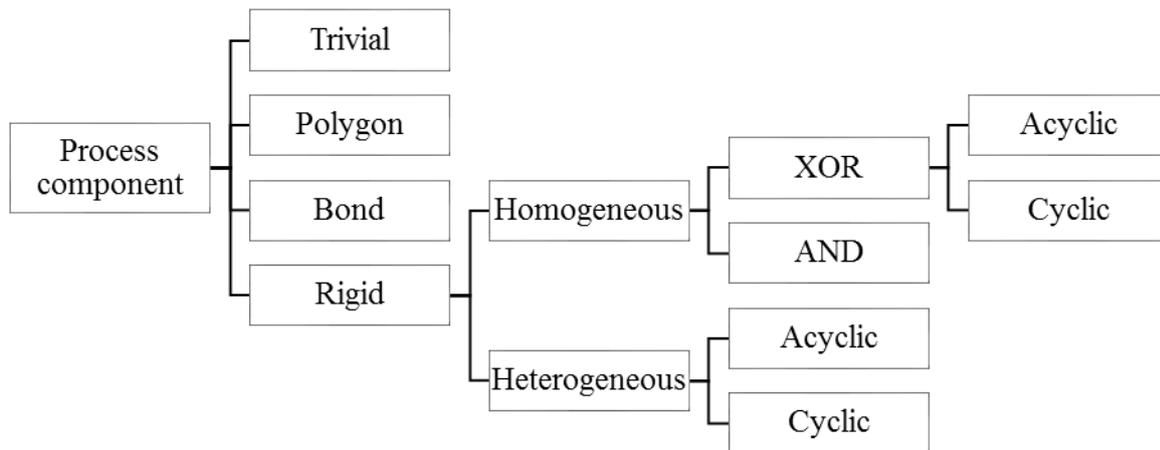


Fig. 4. System of components used in the algorithm

According to the algorithm described above and the outputs of the corresponding paper, the model is unstructured and cannot be converted into well-structured one only if its RPST contains a rigid component, having performed the modular decomposition on which its ordering relations graph contains a concurrent primitive. Otherwise, the algorithm is applicable to every process model not under the condition described previously, and thus can construct the equivalent well-structured process model.

The result of the work are available online, in the BPStruct tool, that is hosted in the Google Code archive (figure 5), which is contributed by the algorithm described [2].

The public RestAPI also can be used by everyone. The BPStruct tool accumulates the previous researches in the field, thus it can structure even the models containing cycles, just following the idea of creating not fully structured, but maximally structured model, skipping the nodes that cannot be handled.

BPStruct is also integrated in the Signavio software (figure 5).

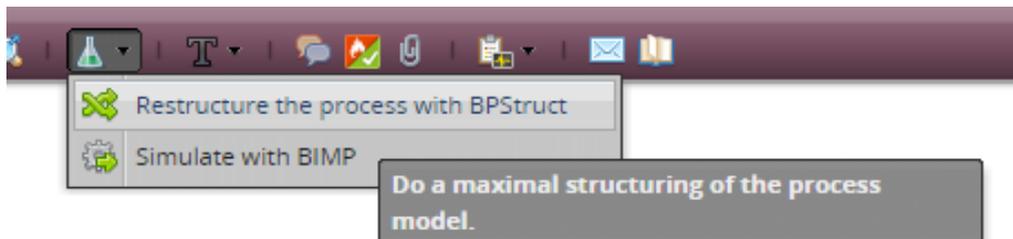


Fig. 5. BPStruct in the Signavio's toolbox

The tool can be tested by creating the process model that is known as unstructured. This process model will have two conflict tasks (meaning that only one could be executed at a time, so they connected through the XOR gateway) and two parallel tasks (splitted and joined through to AND gateway).

As you can see from the process model on the top from the figure 6, the model even look like it is a structured model, because it is visually composed as a block. Although one can observe that the split and join of the element is of the different type.

After running the BPStruct, we see that the process model is not only well-structured, but also has been simplified, thus is not only easier to understand nu the domain experts and analyst, but also take less time and resources to analyze it, and can serve as a good starting point for the redesign and feature completion

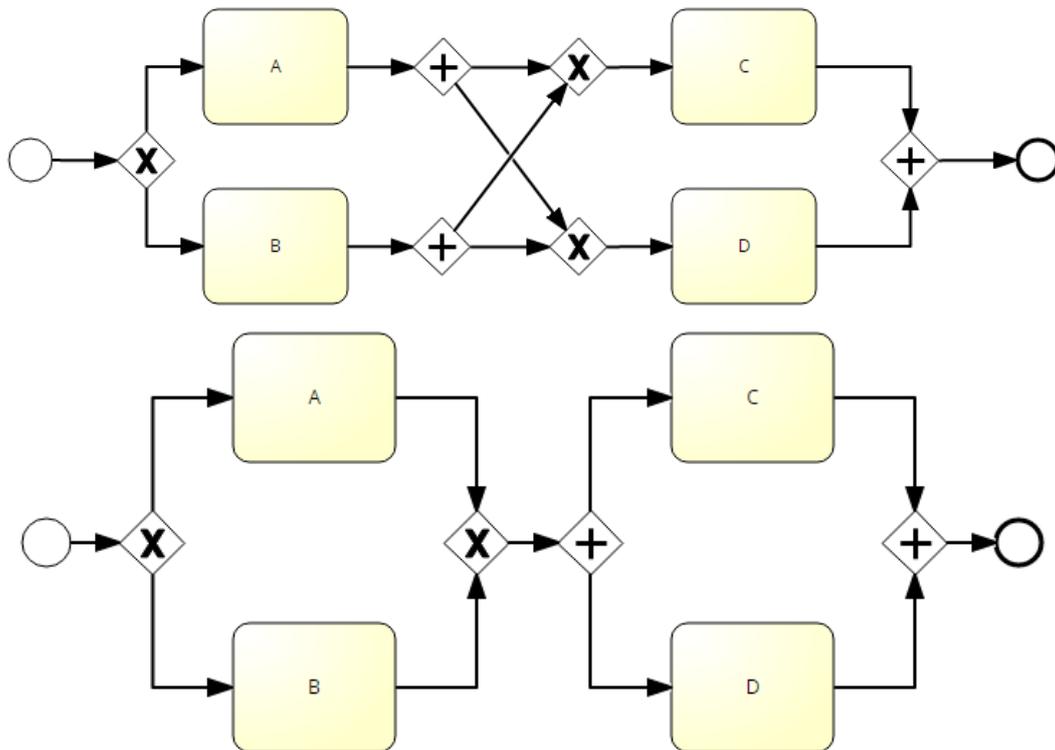


Fig. 6. Unstructured (on the top) and structured (on the bottom) process models.

As stated above, structuring plays important role in the chain of methods from process mining, that enable analyst and business experts to handle the requirements. On a figure 7 one can observe what role the automated discovery techniques play.

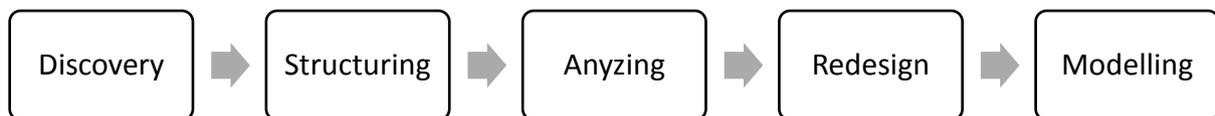


Fig. 7. Process mining value chain.

So as could be seen, structuring plays significant role to help the experts to work with the model, and to facilitate the redesign process, as it hardly depends on the quality of model provided, as well as the future improvement is easier to do with the model that could be clearly comprehended.

3 Conclusions

In the following research, the trends in the information technology sphere was considered, such as: a need of adding features to the existing systems and in some cases a need of a total redesign, that is, however, partially based on the existing requirements, models and flows.

Since the automated process discovery can help with creating the BPMN process models, that are helpful to the analyst as a starting point for creating an improved, redesigned version of the requirement to the new system, the crucial part here is to have the model the analyst starting work with as clear and not over engineered as possible.

For this goal, automated process model structuring, and the algorithm of structuring acyclic models, in a way that it tries to organize as much nodes as possible, is a perspective area to learn and improve. Proposed algorithm is already well-designed, however it cannot handle some complex cases and multioutput loops. This could be improved by providing additional information

to the input of the process, such as gateway conditions. This will make the algorithm less flexible, however will help to resolve the cases that cannot be re-solved now.

In the future, the more attention should be paid to the big diagrams with multiple outputs and intermediate event in the middle of the process model blocks, as that is how industry created models from the event logs are going to look like. Even artificially created know-out process type, which is commonly used in the application-to-approval, claim-to-settlement, etc. processes should be considered as a priority in tuning existing and developing new business process structuring algorithm.

References

- [1] Ankori, R., & Ankori, R. (2005, February). Automatic requirements elicitation in agile processes. In *Software-Science, Technology and Engineering, 2005. Proceedings. IEEE International Conference on* (pp. 101-109). IEEE.
- [2] BPStruct is a tool on Google Code, License: GNU GPL v3, from <https://code.google.com/archive/p/bpstruct/>
- [3] Claes, J., Vanderfeesten, I., Gailly, F., Grefen, P., & Poels, G. (2015). The Structured Process Modeling Theory (SPMT) a cognitive view on why and how modelers benefit from structuring the process of process modeling. *Information Systems Frontiers*, 17(6), 1401-1425.
- [4] Conforti, R., Dumas, M., García-Bañuelos, L., & La Rosa, M. (2015). BPMN Miner: Automated discovery of BPMN process models with hierarchical structure.
- [5] De Weerd, J., De Backer, M., Vanthienen, J., & Baesens B. (2012). A multidimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs. *Information Systems*, Vol. 37 (pp. 654-676).
- [6] Dumas M. (2015, June). Process Mining Reloaded: Event Structures as a Unified Representation of Process Models and Event Logs. Keynote talk at the 36th International Conference on Application and Theory of Petri Nets and Concurrency (Petri Nets 2015), Brussels, Belgium.
- [7] Dumas, M., García-Bañuelos, L., & Polyvyanyy, A. (2010). Unraveling unstructured process models. In *Business Process Modeling Notation* (pp. 1-7). Springer Berlin Heidelberg.
- [8] Dumas, M., La Rosa, M., Mendling, J., Mäesalu, R., Reijers, H. A., & Semenenko, N. (2012, June). Understanding business process models: the costs and benefits of structuredness. In *Advanced Information Systems Engineering* (pp. 31-46). Springer Berlin Heidelberg.
- [9] Eshuis, R., & Kumar, A. (2015). Converting unstructured into semi-structured process models. *Data & Knowledge Engineering*, 101, 43–61
- [10] Kalja, A., Reitsakas, A., & Saard, N. (2005). eGovernment in Estonia: Best practices. *Technology Management: A Unifying Discipline for Melting the Boundaries*, 500-506.
- [11] Karmarkar, U. (1989). Getting control of just-in-time. *Harvard business review*, 89(5), 122-130.
- [12] Kiepuszewski, B., ter Hofstede, A. H. M., & Bussler, C. J. (2013). On structured workflow modelling. In *Seminal Contributions to Information Systems Engineering* (pp. 241-255). Springer Berlin Heidelberg.
- [13] Object Management Group. *Business Process Model And Notation™ (BPMN™) Version 2.0*. October 13, 2001, from <http://www.omg.org/spec/BPMN/2.0/>
- [14] Polyvyanyy, A., García-Bañuelos, L., & Dumas, M. (2010). Structuring acyclic process models. In *Business Process Management* (pp. 276-293). Springer Berlin Heidelberg.
- [15] Vanhatalo, J., Völzer, H., & Leymann, F. (2007). Faster and more focused control-flow analysis for business process models through SESE decomposition (pp. 43-55). Springer Berlin Heidelberg.

MYKHAILO DOROKHOV
University of Tartu
Institute of Computer Science
J.Liivi 2, Tartu, Tartumaa
ESTONIA
E-mail: mykhailo.dorokhov@ut.ee

Mark It

Dragomir Andrei, Grozea Ioan
Teacher coordinator: Delilah Florea

Abstract

Nowadays most of the technology is automatized. However, we couldn't find an application with repeatable commands for the smartphone, other than the alarm clock or other simple apps. Therefore, we decided to help the people in their daily activities by creating an application that can track the users by GPS using the phone's location feature. These will feature the maps from Google Maps, and they can set some checkpoints on the map. A pointer will show the user's current location, with an error of a couple meters, if the signal of the phone is good enough.

The app will notice when the current location of the user is within a checkpoint and will change/update some phone settings based on this checkpoint. The app will work in background, so the users mustn't have the screen on.

Keywords: current location, GPS, marker, phone settings, map

1. Introduction

The following article describes the app named „Mark It”, developed for the operating system Android OS. The main purpose of the app is to simplify the users activities, whether they are students or employees, because it may happen, so that the user forgets for example to put his/her device on silent mode during an important meeting or during classes. These problems will disappear after the installation of the „Mark It” app. The users will be localized using the devices GPS and they can set markers on the map, with an action range of 100 meters and when they walk into the range, the app will modify some settings, like setting the mode on silent mode. This technology is called Geofencing and it was developed by the Google company with the reason of sending a notification to someone, when they are near a Google building. We have adapted this technology to create a zone with a marker in the middle.

All in all, if a student, for example, sets a marker on his school with the silent setting and he enters its range, the device will not disturb the class and when he leaves the zone, the device will be set on normal mode again.

The interface will be presented in section 2, the loading of the map and the localization. The third section contains details about the software implementation of the app. The fourth section sums up the conclusions of the article and describes further development ideas.

2. User interface

On startup, the app will track the users location and show it on the map, together with the whole map, shown in the image below:



Fig.1: The map

On pressing the  button, the app will zoom in the map, until the users location will be shown, together with the surrounding area, like the next image:



Fig. 2: The users location

As shown in the above image, the phone is set on silent mode, because it is in the range of the marker. If the user presses on the marker, two buttons will pop up, , which, if pressed, will show the distance between the user and the marker (button 1) or the coordinates of his location (button 2).

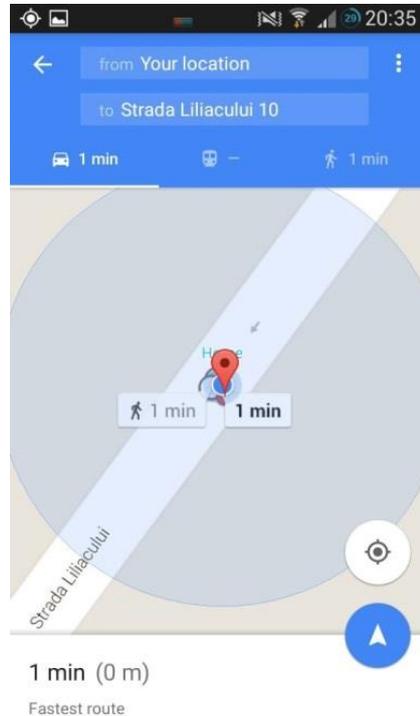


Fig. 3: Navigation to the marker

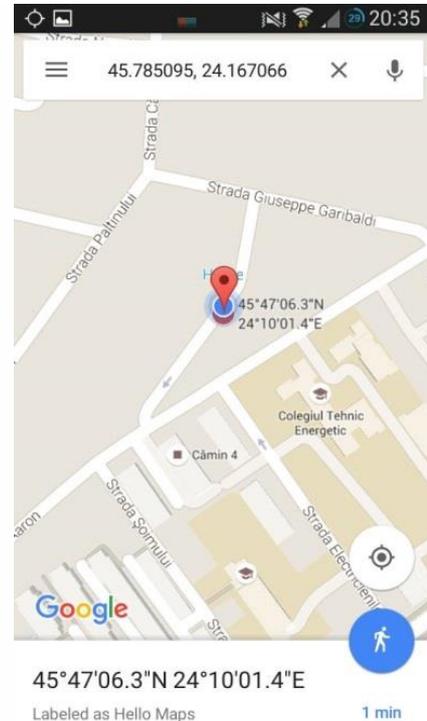


Fig. 4: Coordinates of the current location

3. Software development

3.1 Development environment

The app was created using the Android Studio v1.5.1 IDE, using the object oriented programming language JAVA and was tested directly on the smartphone, because its GPS sensor was needed. The development of the app needs an API Key and the registration in Google Developers Console in order to work. Because of compatibility reasons, the Android SDK was used for the development for the Android OS. We have chosen to develop the app for Android because the studies on the market[1] show that 49.2% of the devices sold in 2012 use the Android OS. Also, the Android platform is open source and because of its portability, more and more software developers choose this OS.

3.2 Code sections

3.2.1 Layout

One of the most important characteristics of the layout is the map itself [2]. Without it, the user could not benefit from the markers and the Geofencing. The next lines of code represent a part of the layout file, which contains a fragment with a map attached to the main activity:

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="com.google.android.gms.maps.MapFragment"/>
```

3.2.2 The map

If a map has not been created, the next function will create it [3] and will add buttons for the tracking of the user, the zoom in/out, the compass, which will rotate the map and the zoom in/out using the touch screen:

```
GoogleMap googleMap;
private void initializeMap() {
    if (googleMap == null) {
        googleMap = ((MapFragment) getFragmentManager().findFragmentById(
            R.id.map)).getMap();
        googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
        googleMap.setMyLocationEnabled(true);
        googleMap.getUiSettings().setZoomControlsEnabled(true);
        googleMap.getUiSettings().setMyLocationButtonEnabled(true);
        googleMap.getUiSettings().setCompassEnabled(true);
        googleMap.getUiSettings().setRotateGesturesEnabled(true);
        googleMap.getUiSettings().setZoomGesturesEnabled(true);
        if (googleMap == null) {
            Toast.makeText(getApplicationContext(),
                "Sorry the map could not be shown!", Toast.LENGTH_SHORT).show();
        }
    }
}
```

3.2.3 Tracking

To track the user it was needed to add a new JAVA class, named GPS.java. The minimum distance that the user needs to cover so that the location will be updated is of 1 meter and it is given by the variable:

```
private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 1;
```

To test if the GPS is on and to use it [4], it was needed to implement the `isGPSEnabled`, `isNetworkEnabled` and `canGetLocation` variables, which are of boolean type and are initialized with false, to test if the GPS and mobile data of the user are enabled:

```
protected LocationManager locationManager;
public GPS(Context context) {
    this.mContext = context;
    getLocation();
}
public Location getLocation() {
    try {
        locationManager = (LocationManager) mContext
            .getSystemService(LOCATION_SERVICE);
        isGPSEnabled = locationManager
            .isProviderEnabled(LocationManager.GPS_PROVIDER);
        isNetworkEnabled = locationManager
            .isProviderEnabled(LocationManager.NETWORK_PROVIDER);
        if (!isGPSEnabled && !isNetworkEnabled) {
        } else {
            this.canGetLocation = true;
            if (isNetworkEnabled) {
                locationManager.requestLocationUpdates(
                    LocationManager.NETWORK_PROVIDER,
                    MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                if (locationManager != null) {
                    location = locationManager
                        .getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                    if (location != null) {
                        latitude = location.getLatitude();
                        longitude = location.getLongitude();
                    }
                }
            }
            if (isGPSEnabled) {
                if (location == null) {
                    locationManager.requestLocationUpdates(
                        LocationManager.GPS_PROVIDER, MIN_TIME_BW_UPDATES,
                        MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                    if (locationManager != null) {
                        location = locationManager
                            .getLastKnownLocation(LocationManager.GPS_PROVIDER);
                        if (location != null) {
                            latitude = location.getLatitude();
                            longitude = location.getLongitude();
                        }
                    }
                }
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return location;
}
```

3.2.4 Geofence

Geofencing is a combination between the tracking of the users current location and the approach of the user to a point of interest. To mark a point of interest, it is needed to specify the latitude and longitude of the point and to set the approach to the chosen location, a range can be added. In other words, to create a Geofence, a range and a transition type must be specified:

```
mGeofenceList.add(new Geofence.Builder()
    .setRequestId(entry.getKey())
    .setCircularRegion(entry.getValue().latitude,
        entry.getValue().longitude,
        Constants.GEOFENCE_RADIUS_IN_METERS)
    .setTransitionTypes(Geofence.GEOFENCE_TRANSITION_ENTER |
        Geofence.GEOFENCE_TRANSITION_EXIT).build());
```

In the MainActivity, to add a geofence [5], you must add the following code:

```
LocationServices.GeofencingApi.addGeofences(
    mGoogleApiClient,
    getGeofencingRequest(),
    getGeofencePendingIntent())
.setResultCallback(this);
```

3.2.5 Marker

Markers are the ones that allow the displaying of the unique locations on the map. To add a marker you have to specify the latitude and longitude of the location and the information which they offer can be modified depending on the context:

```
MarkerOptions marker = new MarkerOptions().position(
    new LatLng(latitude, longitude))
marker.icon(defaultMarker(HUE_YELLOW));
googleMap.addMarker(marker);
```

3.2.6 Permissions

This app needs some permissions in order to work on the users device, like `ACCESS_FINE_LOCATION`, which allows the app to track and display a precise location of the user and `ACCESS_COARSE_LOCATION` which allows the app to display an approximate location. The next permissions can be found in the `AndroidManifest.xml` file of the app:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

4. Conclusion

All in all, our app „Mark It” could prove to be very usefull not just for the students, but also for the employees, because it simplifies their lives, through completing some tasks, which the users could forget to do. The battery consumption is not a problem, because the mobile data can be turned off, as soon as the user has been tracked, if the GPS is turned on and the screen can be turned off, because the app runs in background.

5. Future development

In the future, we would like to introduce a searching function, so that the user may search for an address and a marker will be placed in that place. We would also like to add a button to change the map type from normal to hybrid or satellite. We would also like to upload our app on the Google Play Store, to help other people.

Bibliography

- [1] Mednieks, Z., Dornin, L., Meike, G.B., Nakamura, M., 2012, *Programming Android: Java Programming for the New Generation of Mobile Devices*, Second Edition, O'Reilly Media.
- [2] Google Developers (layout map):
https://developers.google.com/maps/documentation/android-api/map#add_a_map_to_an_android_app
- [3] Stack Overflow (creating the map):
<http://stackoverflow.com/questions/30170271/android-google-map-how-to-check-if-the-gps-location-is-inside-the-circle>
- [4] Stack Overflow (GPS):
<http://stackoverflow.com/questions/24320989/locationmanager-locationmanager-this-getsystemservicecontext-location-servi>
- [5] Google Developers (Geofencing):
<http://developer.android.com/training/location/geofencing.html#BestPractices>

Dragomir Andrei
National College „Samuel von
Brukenthal”
Department of Mathematics-Informatics
Huet Place Nr. 5, Sibiu
Romania
Email: meandrei96@gmail.com

Grozea Ioan
National College „Samuel von
Brukenthal”
Department of Mathematics-Informatics
Huet Place Nr. 5, Sibiu
Romania
Email: ioan.grozea@gmail.com

Prof. coord. Florea Delilah
National College „Samuel von
Brukenthal”
Informatics Teacher
Department of Mathematics-Informatics
Huet Place Nr. 5, Sibiu
Romania
Email: delilah_florea@yahoo.com

Fireball

Bogdan Georgescu, Ciprian Petrean
Teacher Coordinator: Ramona Humeniuc

Abstract

The animation is made for a future “open world” game. As a first step we’ve done a medieval fantasy themed cinematic. We used Blender in making models and composing animations. Blender is the free and open source 3D creation suite. It supports the entirety of the 3D pipeline—modelling, rigging, animation, simulation, rendering, composing and motion tracking, even video editing and game creation. The project is made of dynamic scenes, in which every object is modelled from a basic mesh and textured. Video editing and a part of the composition capabilities were done in Sony Vegas Pro.

1 Introduction

3D modelling is a mathematical process by which a body or an object is represented three-dimensional using a specialized software. The 3D model is in fact a collection of dots positioned in three-dimensional space, connected by lines, triangles, curved surfaces and the image generated on a computer screen is composed of pixels: the higher the number of the pixels, the better the resolution of the screen and the quality of the image.

Since the theme is “medieval fantasy” we found inspiration in movies like “The Lord of the Rings” or “The Hobbit” or games like “World of Warcraft”, “The Witcher” or “Dragon Age”. Concerning the scenes, we used Blender to model objects and compose animations, Adobe Illustrator CC Tryout for vectors and Adobe Photoshop CS5 for textures and retouches.

2 Project composition

2.1 Scene I

The animation is composed of five scenes, the first scene(*fig.1*) consists of a dark room with a wooden floor and a table where there are placed some objects like a map, a ring, two books and a candle, every object being modelled from a basic mesh and then textured (*fig.2*). We added a “fireball” – a glowing sphere with a group of particles which floats around just to boost the scene with a more dynamic look.



Figure 1 - modeling

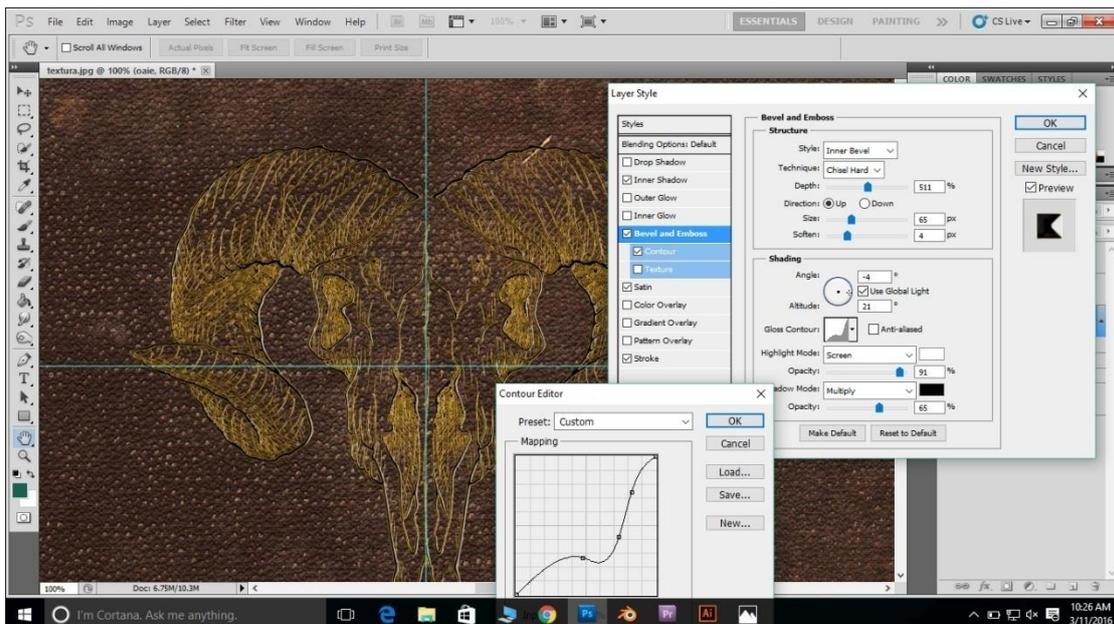


Figure 2 – book texture

2.1.1 Organising data

Title of Blender organizes data with different types of “data blocks” like Objects, Meshes, Lamps, Scenes, Materials, Images and more. Blender is the free and open source 3D creation suite. It supports the entirety of the 3D pipeline—modelling, rigging, animation, simulation, rendering, compositing and motion tracking, even video editing and game creation. In Blender an object is made of several “data blocks”, for example a cube has at least one datablock for Object and Mesh, and, usually, one material. This enables, for example, the duplication of objects keeping one single

datablock to which all the copies of the initial object are linked, but allowing the changing of the objects in the same time.

2.2 Scene II

The second scene consists of a landscape of snowy mountains which were created from a landscape-type mesh, modified so as to render a more realistic appearance of mountains. Concerning their texturing, we used two different textures, one for the rock and one for the snow to which we added some effects to create a product as realistic as possible. Textures are overlaid and connected by a “mix-shader” to which we added a color modifier for a better delimitation of the areas with and without snow. To render more “realism” we added the “mist” function”, which generates the mist in the landscape, on the distance camera- object (*fig.3*). The sun consists of a basic mesh, this time a sphere, to which we added light emission and a “blur effect” for the area around it (*fig.4*).

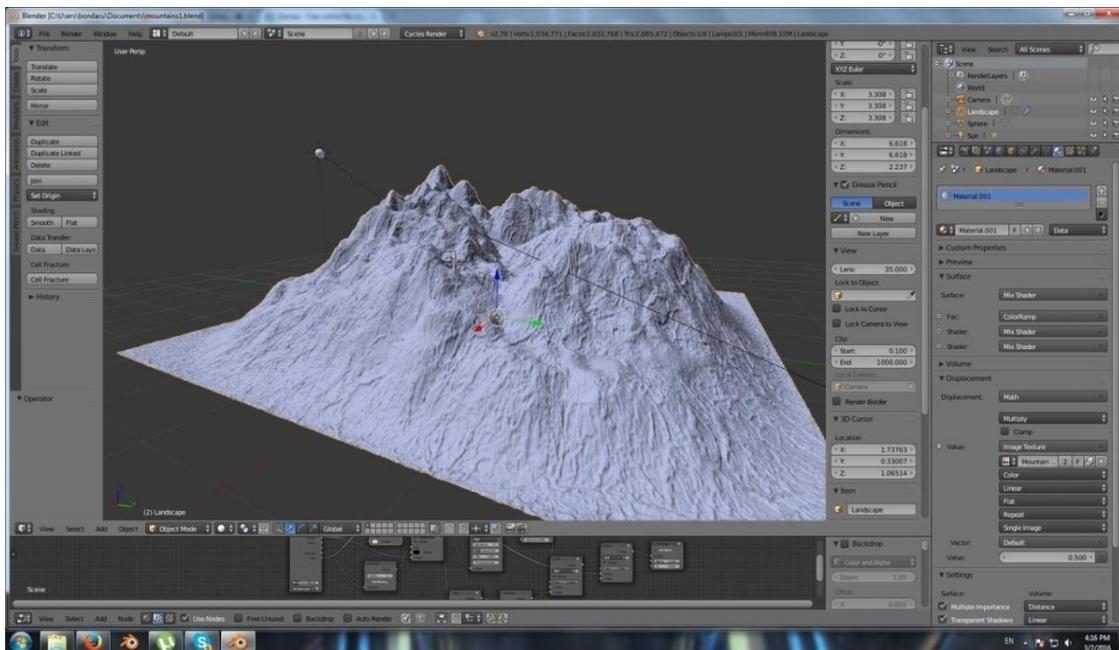


Figure 3 – modeling

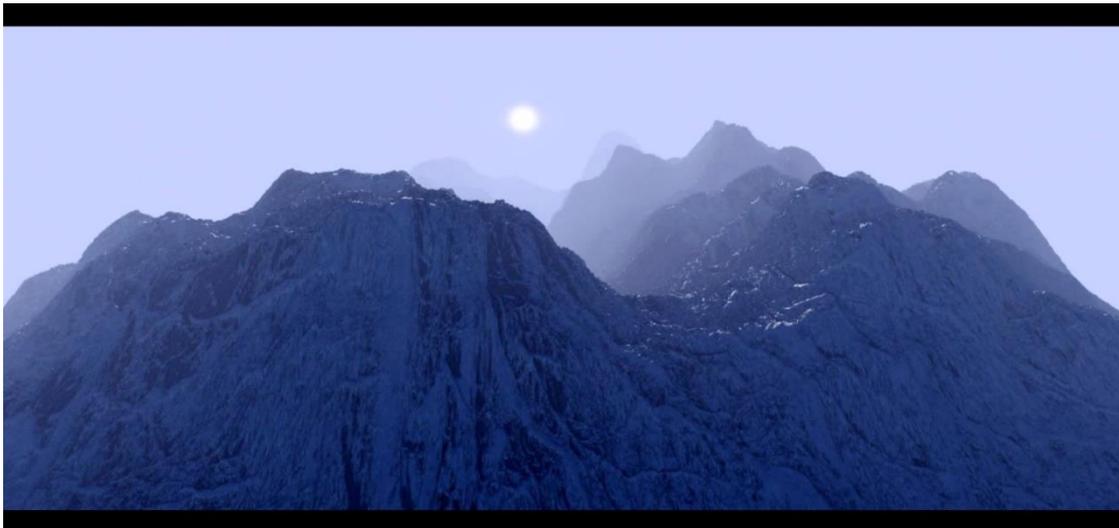


Figure 4 - result

2.3 Scene III

The third scene consists of a dark room with a stone textured floor, a few pillars, a coffin and a sword, each object being modelled from a basic mesh (fig.5-6). To obtain the objects surface, Blender allows the user to apply textures that can modify the reflection, specularity, roughness and other surface qualities; textures behave like additional layers that apply over the base material.

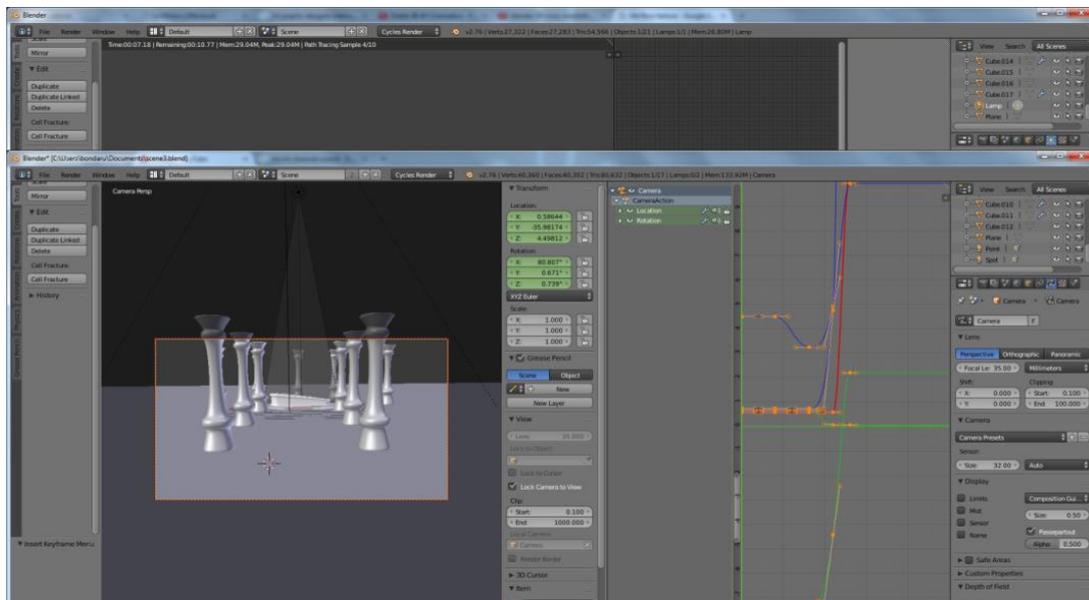


Figure 5 - modeling



Figure 6 - result

2.3.1 Tridimensional scenes

Creating a 3D scene requires at least three types of things: Models, Materials and Lights; objects can have many shapes and sizes and Blender provides tools that help modelling effectively, like:

- Objects- allows working with objects as a whole
- Meshes – allows working with the network that defines the shape of the object
- Curves – allows using curves in modelling and controlling objects
- Surfaces – allows modelling of a NURBS surface
- Meta Objects - Globs and Globules
- Duplications- allows duplication of objects
- Modelling Scripts – scripts for modelling(Python)

2.4 Scene IV

In the fourth scene we created an ancient warrior helmet modelled from a basic mesh, this time a sphere, with a scratched metal texture that has been passed through a series of nodes. The support is modelled from a basic mesh and we added a shiny steel texture over an ornamental texture (*fig. 7*).

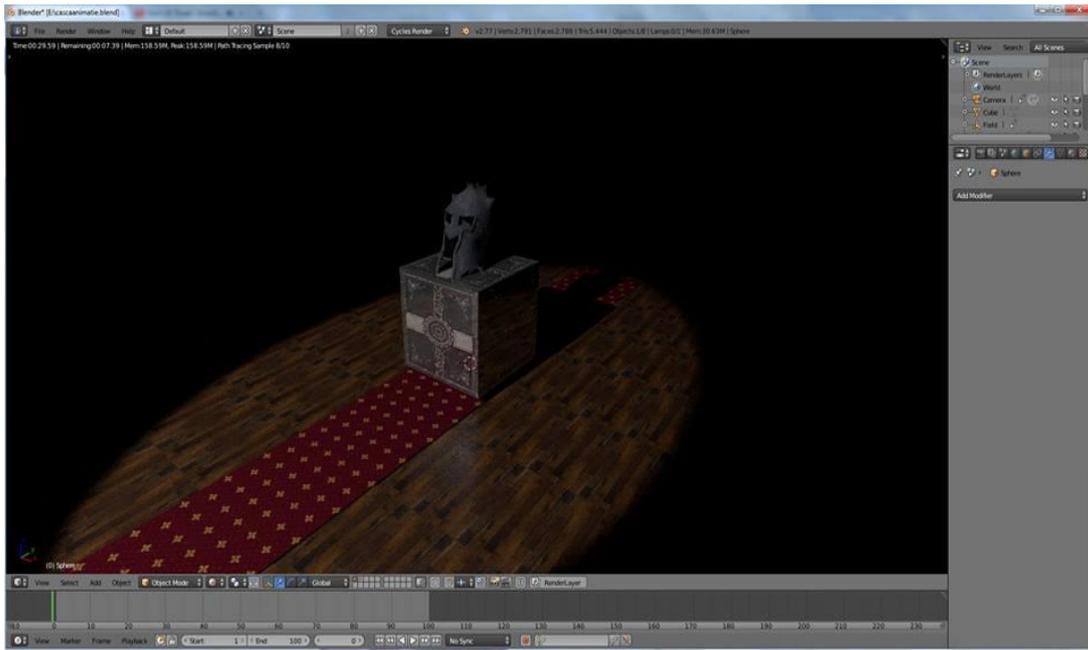


Figure 7 - result

References

- [1] AILENI, R.M., Contributii teoretice si experimentale la simularea dinamica 3D a produselor, Iasi, 2011.
- [2] Roman F. Hartmann. *Model in science*. Encyclopedia of Phylosophy, Springs 2006.
- [3] *Deviant Art official page* - <http://www.deviantart.com>
- [4] *Morguefile official page* - <http://www.morguefile.com>
- [5] *Free stock photos* - <http://www.stockvault.net>
- [6] *CGT official page* - <http://www.cgtextures.com>

Bogdan Georgescu
“Horea, Closca and Crisan” National
College
Computer science - mathematics
1 Decembrie 1918 Avenue
Romania
E-mail: bondaru@yahoo.com

Ciprian Petrean
“Horea, Closca and Crisan” National
College
Computer science - mathematics
1 Decembrie 1918 Avenue
Romania
E-mail: cipri.petrean@gmail.com

Prof. Ramona Humeniuc
“Horea, Closca and Crisan” National
College
Computer science, physics
1 Decembrie 1918 Avenue
Romania
E-mail: lichcalba@gmail.com

PhotoMaster

Handolescu Radu, Motoc Alexandru-Nicolae
Teacher Coordinator: Steavu Nicolae, Steavu Cristina-Elena

Abstract

The project we made, called "PhotoMaster", consists of a specialized program for editing photos or other images. The making of any type of program requires very strong knowledge of the programming environment especially when it is about an application with a well-defined practical role. While creating and designing this photo editor, we tried to closely study all the functionalities of a real editor, from the simplest ones to the most complex, and implement the new products as an original and easy to use program. We also wanted to highlight the usage of mathematical formulas for resizing, Zoom and other operations. The pre-existent editors on the market bring together a great number of features and functionalities which can make the user feel overwhelmed by the situation. We tried to remove this impediment through a very simple and clean interface, making the access to the diverse tools very easy and intuitive. The reason we chose to develop such an application is our desire to engage in a large scale project in order to test our C# programming knowledge, as well as our algorithmic thinking and ability to work in a team. We chose the photo editor because of the reasons we stated above and due to the absence of a free and quality program of this kind.

1 Introduction

Through PhotoMaster we plan to develop an image-editing application that is easy to use by anyone, especially those who worked in the past with such programs.

Being in the 12th grade we thought that we could crown the knowledge gained during high school with a project of higher proportions with which we would improve our programming skills and teamwork and mark the symbolic start of career our professional.

Other similar products are Photoscape, Pixlr, Paint, best known in the market as Photoshop, developed by Adobe and Paint, which comes as a feature included in Windows operating systems. Bearing in mind these models, we identified the problems a beginner faces at a first use of such a program. The biggest drawback is that a person can easily be overwhelmed by the plethora of tools which appear on the screen.

The desire of facilitating the work of those who want to make quick changes on some pictures determined us to start this project.

In this paper we present the application made by us and two of the basic operations that have been implemented so far, Zoom operation and selection operation, placing more emphasis on them. We should mention that the program was created in C # using Microsoft Visual Studio 2015 Community and that the project is based on Windows Forms and implementation possibilities offered by them ([2] - "Windows Forms").

2 Zoom

The Zoom operation means the enlargement or reduction of a portion of an image around a point with a predefined value. This operation is extremely useful because it allows the user to closely study an image, provides a more detailed image while allowing the user to perform minute operations that the original size of the image may not allow.

The Zoom operation was by far the most complex, not in terms of implementation because the idea and the code itself did not raise any problems, but in terms of debugging.

The work surface is composed of two elements: a PictureBox to load the image you are working with (called in the program workarea, yellow frame in Fig.1) and a Panel (the display, red frame in Fig.1) which represents the selected area of the image. In Figure 1 you can see how the image looks after several enlargement operations. In the program's interface is only visible the image inside the red frame, which with the help of scroll bars can be move across the entire image.

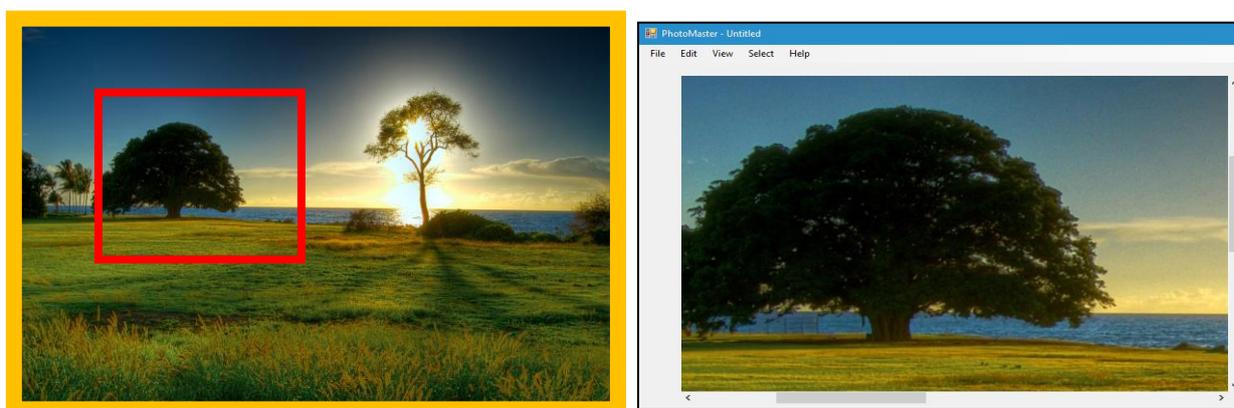


Fig.1 Image representation in the program

Zooming can be done in two ways: from the View menu (shown in Fig.2) or using the mouse wheel. When using the wheel, the Mousewheel event is called, and the Delta property of the MouseEventArgs argument is used to determine the direction in which the wheel was moved, figuring the type of zoom to be carried out ([2] - "Handling events generated controls"). The algorithm that resizes the work area is extremely simple and consists of only changing the size of workarea.

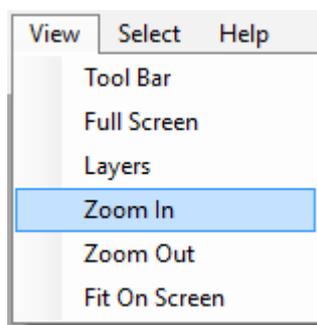


Fig.2 View Menu

What has raised problems and needed a longer thinking time was the algorithm that places the display (through the scroll bars) in the location where the Zoom has been executed, an operation for which we had to find the mathematical formula for the new location:

- centered on the place of the already positioned display if the View menu buttons were used;
- placed so that the mouse remains above the same pixel on which the mouse wheel was used to zoom.

I present here the method's code that performs this repositioning in case Zoom In was done using the mouse wheel. Code sequences have been added in this method as the developing progressed in order to solve bugs related to the Zoom operation.

```
private void zoomIn(Point mouse)
{
    int scrollhorriz = display.AutoScrollPosition.X, scrollvert = display.AutoScrollPosition.Y;

    resizeImage(workArea.Image, (int)(multiplier * workArea.Width), (int)(multiplier * workArea.Height));

    centerPictureBox();

    resizePictureBox(workArea, workArea.Image);

    int x = mouse.X, y = mouse.Y;

    if (workArea.Width > display.ClientSize.Width || workArea.Height > display.ClientSize.Height)
    {
        display.AutoScrollPosition = new Point(0, 0);
    }

    if (workArea.Width > display.ClientSize.Width)
        workArea.Left = 0;

    if (workArea.Height > display.ClientSize.Height)
        workArea.Top = 0;

    display.AutoScrollPosition = new Point((int)(Math.Abs(x) * multiplier - x - scrollhorriz), (int)(Math.Abs(y) *
multiplier - y - scrollvert));
}
```

In the previous code sequence scroll bars are positioned so that after the Zoom the mouse cursor remains positioned on the same pixel as prior to the operation. *resizeImage* is the method that changes the *workarea* size after which the *resizePictureBox* method ensures that the PictureBox's lengths ratio remained unchanged after resizing. The variable *multiplier* is the factor of zoom in / out and it was arbitrarily set the value of 120%. The Point type parameter that was transmitted to the method represents the current position of the mouse on the PictureBox and is determined after triggering the event specific to the mouse's scroll wheel.

The formula of the scroll bars' new positions, as can be seen from the last line of code is:

$$x_{final} = [|x| * multiplier] - x - scrollhorriz \quad (2.1)$$

$$y_{final} = [|y| * multiplier] - y - scrollvert \quad (2.2)$$

In case a Zoom Out was desired, the multiplication inside the integer part in the equations (2.1) and (2.2) is changed in division. In these formulas *x* and *y* are the coordinates of the cursor within the borders of *workarea* and *x_{final}* and *y_{final}* are the new positions of the scroll bars.

The Zoom, as a basic function in any image editor, is not an operation that makes changes or settings in the program, but one that enhances and facilitates the user's experience, especially in times when he has to work at a level of detail relatively low.

3 Selection

The Selection is the operation that marks an area of an image for future processing. In the current version of the program, it can only have a rectangular shape, but we plan to implement irregular selections, giving the user a higher degree of manipulation of the selection.

When it comes to photo editing program, an essential and elementary tool that gives the user an extensive control over image processing is the selection. This provides the ability to mark a portion of the image through a rectangle. With this functionality help multiple applications can be made, like copy, paste or move the selected area or crop the photo.

The selection process consists of two steps:

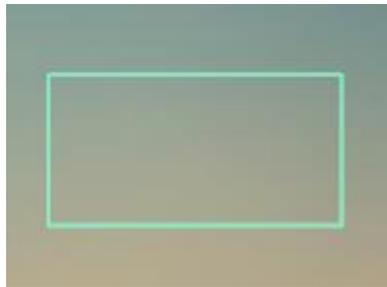


Fig. 3: Determining the selected area

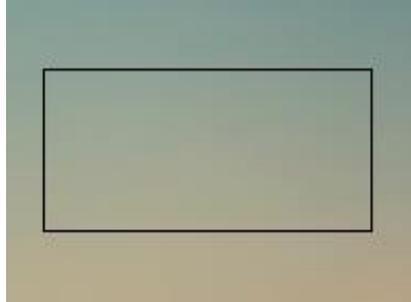


Fig. 4: Finalizing the selection

In order to determine the selection area the user has to click the initial point of the area he wishes to select and holding the mouse button clicked, drag the mouse towards the final point. In this moment, the selection is finished when the click is released.

We were able to implement this kind of functionality by using the predefined events of a PictureBox ([3] – PictureBox Events):

```
selectionBox.MouseDown += new MouseEventHandler(selectionBox_MouseDown);  
selectionBox.MouseUp += new MouseEventHandler(selectionBox_MouseUp);  
selectionBox.MouseMove += new MouseEventHandler(selectionBox_Move);
```

Using this events, we stored both the initial coordinates as well as the final ones of the selection and we checked if the mouse button was clicked.

We will present the method we used for drawing the selection box ([4] – Rectangle Selection):

```
private void drawSelectionThread()
{
    selectingActive = true;

    Rectangle r = createRectangle(selectStart, selectEnd);

    Graphics g = selectionBox.CreateGraphics();
    Pen p = new Pen(Brushes.Aquamarine);

    int x = 1;
    timerSelection.Interval = 500;
    do
    {
        if (x == 1)
        {
            x++;
            p.Brush = Brushes.Black;
        }
        else
        {
            x--;
            p.Brush = Brushes.White;
        }
        timerSelection.Start();

        p.DashStyle = DashStyle.Solid;
        g.DrawRectangle(p, r);

    } while (selectingActive);

    selectionBox.Invalidate();
}
```

We considered it is useful that the color of the box's margins alternates between black and white for better visibility in the program. The only problem of this process was the interface freeze, which made it impossible to use any other operations. Therefore, we avoided this undesirable effect by updating the program through a thread, which was the reason we had to learn more about multi-threading in order to have a thread-safe program ([1] – „Multithreaded programming”). The selectingActive variable has an important role because it specifies the actual state of the selection and it allows the usage of different operations in the Edit menu depending of its value. We accomplished this with the use of properties ([1] – „Properties, Arrays and Indexers”).

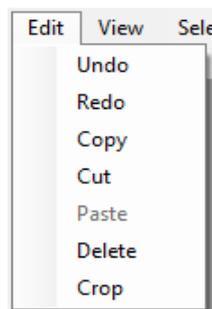


Fig. 5: Edit Menu

The operations that can be applied to a selection can be found in the Edit menu. Moreover, following the copy-paste operation of a selection, this can be freely moved depending of your needs. If the mouse cursor is located close to the selection's margins, the cursor icon will change and the user will be able to perform a resizing action.

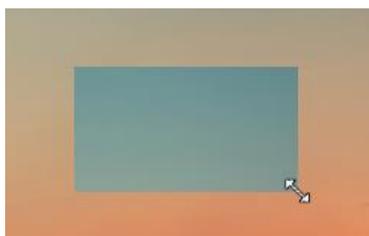


Fig. 6: Resizing

4 Conclusion

Regarding reaching the goals we set to this date, we succeeded in bringing the program to a stage where it is possible to perform basic editing such as copying, pasting, cropping or resizing an image or a selection.

We encountered some problems for a long period of time when performing the combination of making a selection and zooming. We succeeded to identify the cause as being the approximations to integers of the image size after it was resized with the zoom factor. Other difficulties we encountered are concerning multithreading and writing thread-safe code in a program that uses Windows forms.

After solving these problems and debugging the program we came to the conclusion that this experience has taught us in-depth many theoretical concepts and different methods to overcome situations in which a direct approach is not possible.

In the near future we wish to continue the development of this project, adding new useful and easy-to-use features. For example, the next challenge is working with layers which will function similarly to the one in any version of Photoshop. We will work to perfection the existent functionalities, testing the program in different manners and fixing every type of bugs that could lead to reducing user satisfaction.

We will use the next months to improve the design. Our objective is creating a modern and attractive environment, maintaining, at the same time, the simplicity of the program. More than that, the moment we reach a stage in which we will be pleased by our work, we will publish the application on Windows Store, hoping that other people will make use of it.

References

- [1] ARCHER, Tom, „*Inside C#*”, Microsoft Press, 2001
- [2] ROȘCA Valer, HUNYADI Daniel, „*Limbaje moderne de programare: Limbajul C#*”, Editura Universității „Lucian Blaga”, Sibiu, 2010
- [3] https://msdn.microsoft.com/en-us/library/system.windows.forms.picturebox_events (16.02.2016)
- [4] <http://stackoverflow.com/questions/716626/how-to-make-a-resizable-rectangle-selection-tool> (18.02.2016)

Handolescu Radu
Colegiul Național “Radu Negru”
Mathematics-Computer Science
Str. Școlii, Nr.1, Făgăraș
Romania
E-mail: raduhandolescu@yahoo.com

Motoc Alexandru-Nicolae
Colegiul Național “Radu Negru”
Mathematics-Computer Science
Str. Școlii, Nr.1, Făgăraș
Romania
E-mail: alexmotoc97@gmail.com

Prof. Steavu Nicolae
Colegiul Național “Doamna Stanca”
Computer Science
Str. Doamna Stanca, Nr.14, Făgăraș
Romania
E-mail: nicolae_steavu@yahoo.com

Prof. Steavu Cristina-Elena
Colegiul “Aurel Vijoli”
Computer Science
Str. Combinatului, Nr.3, Făgăraș
Romania
E-mail: cristeavu@hotmail.com

***What'S What* android application**

Darius Hațegan
Coordinator Professor: Dana Simian

Abstract

What'S What is an android project which want to increase the access level of information through an application. The application is based on camera which scans barcode from the product. The user can introduce the barcode manually, if lines are not visible. Afterwards, the app search in a lot of sources, call web services, through application web service and at the end, shows the information to the user in a well structured interface.

We made a research based on ten existed applications. The test proves that the use of one single application cannot provide enough information. The applications based on barcode work with small sources/databases containing hundreds of products or specification, introduced in databases from different users. On the same side, these apps are not focused on user experience. Our application aims to remove these drawbacks of existing apps.

For now, *What'S What* is only a proof of concept (POC) but for the future it will be a commercial product, designed to transform the need of using this application into a real experience.

1 Introduction

In actual context in which communication is very important and the need to be in contact with others it manifest in all current activities. Technology supports this need and the mobile devices represent the biggest exponent of this technology. New smartphone have operating systems increasingly performing, they want to be custom, to fit with every user: contacts, news, alarms, business meetings, diet, etc.

The usage rate of smartphones among the population of Romania has exploded in recent years, currently reaching about 40%, as well as consumption of mobile data, while 32% of Romanian use the internet on mobile as soon as they wake up. [2]

The aim of *What'S What* project have as target a variety of consumers. The first urban users, given that they have a wide range of packaged products. The present application focuses on the people having low allocated time for information about product, providing that kind of information in a short enough period of time. The app provides details about products and can be useful for people wanted healthy lifestyle who are interested on all the product ingredients.

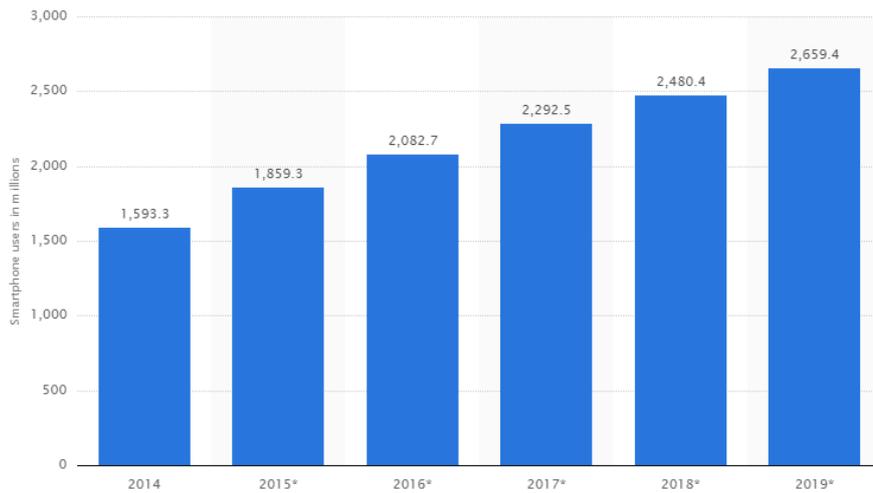


Fig.1 - Number of smartphone users [2]

As we can see the statistics from Fig.1, in 2016, two billions of people own smartphones and is growing [2]. The rate of smartphones usage is the reason for designing What'S What application on mobile platform.

The project idea started from difficulty encountered to find the holder of a brand from the market. Searching this kind of the information on the internet required us more time that we expected. The fact that this kind of information was not on the product label, raised a set of questions. One of them relate to people and their access to information. Consumers do not have access at all kind of details about product or provider due some reasons, including limited space which is given by the product or the details font size which is often very small. Another question refers to the possibility of increasing accessing information speed. These questions inspired us to create an application that provides information in a form as possible interactive and accessible. A solution is the traditional way, search on the browser which is not as fast as specific application which is designed to be faster.

What'S What uses barcode to identify unique the product. Barcodes are graphical representations of unique codes of products, each product type having assigned an unique code. In Fig.1 is presented the barcode. The data code looks like a row of black bars of various thicknesses on a white background. Each digit or letter is represented by one or more black lines at a distance and a specified thickness. Barcodes are read using scanners that interprets the lines as numbers.

On the market, exists different types of barcodes used on different types of products. EAN is used for retail products of type "European Article Numbering" [3].



Fig.2: EAN-13 Structure [3]

What'S What use barcode type EAN-13 to identify product because this is specific for products that the app aims. The structure of EAN-13 is simple. First part of barcode is used to

identify country, second part is to identify manufacturer, third part identify unique the product and the last part is check digit. Check digit is used to verify the correctness of the barcode using an algorithm based on the first three parts of the barcode.

2 Application Arhitecture

What'S What is build on workflow. As we can see in Fig.3, the application starts with a *Splash Screen* which is designed to introduce user into What'S What context. The second screen is authentication from where the user can go in several directions. The meaning of authenticate is that the user cannot make review or rate product without authentication. Third and four screens are for *User Registration* and password recovery. The main goal of the app is the scan and product details but for now we present the scanner. From *Barcode Scanner* we have to possibilities, one is to go at home screen and the second one, after the barcode is scanned, automatically send user to product details. If the product is not found then the user will receive the basic details about the barcode. *Home* has a menu from where the user can navigate through the application. In *Home* exists some functionalities, for instance search barcode manually or new functionality as barcode history.

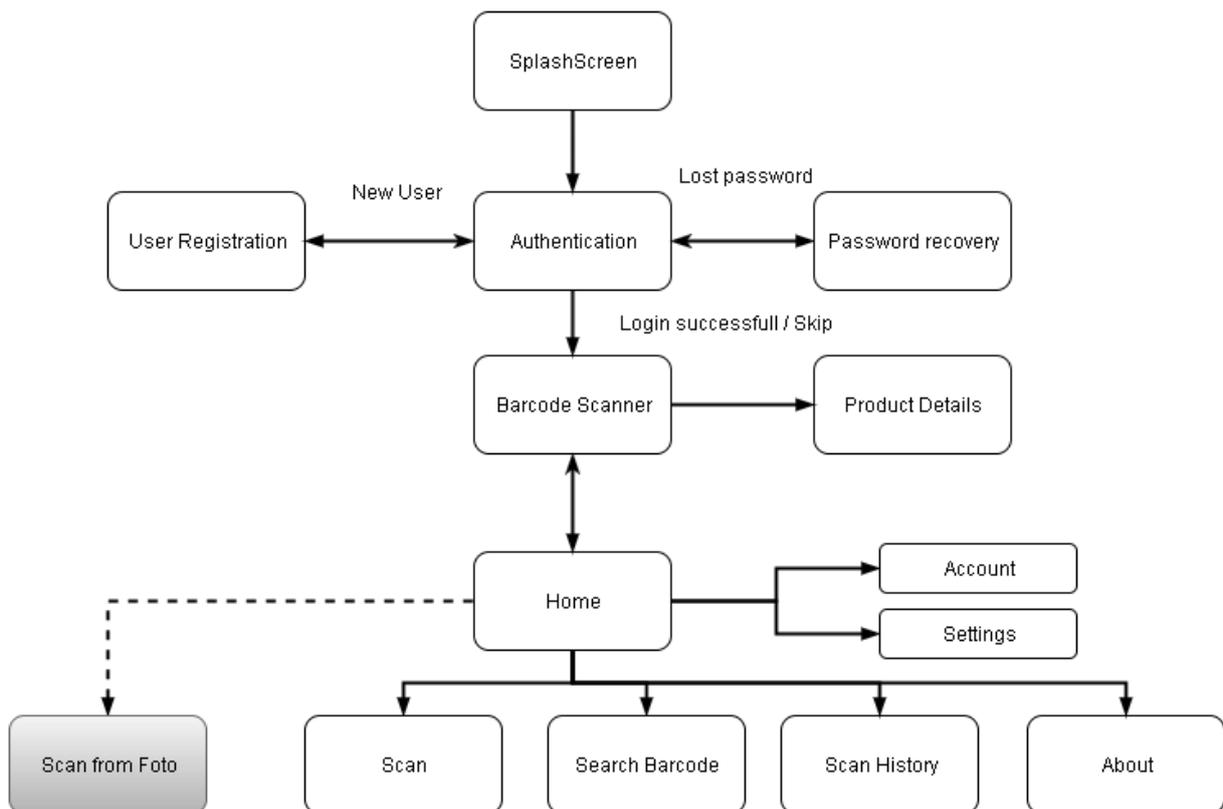


Fig.3: Workflow

3 Conclusions and other directions of study

The What's What android project is designed to offer information about product by scanning the barcode. Based on research, the project comes with new improvements in terms of design and functionality. One improvement is increasing the number of sources that provide information through web service. Another improvement is user experience, for this we created a better design.

The What'S What project is not in the final stage, the development is on-going and server need some additions. Beside these improvements, we want to add new features, for instance scan from photo. We want to create social module which allow the user to have friends, to view friend reviews and others functionalities.

References

- [1] *Smartphone usage statistics* - <http://www.zf.ro/business-hi-tech/aplicatiile-mobile-au-devenit-o-normalitate-peste-30-din-romani-folosesc-internetul-pe-mobil-imediat-ce-se-trezesc-14135443>
- [2] *Smartphone usage statistics* - <http://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>
- [3] *EAN-13 Structure*- <http://www.barcodeisland.com/ean13.phtml>

Darius Hațegan
Lucian Blaga University
Faculty of Science, Department of Informatics
Sibiu, str. Dr. Ioan Rațiu
Romania
Hategandarius71@gmail.com

3D Mobile cross platform game

Hodina Dragos
Coordinator teacher: Simian Dana

Abstract

Smartphones, since they appeared, were integrant part of our daily activities. We use them more to navigate on the internet, send messages, watch videos, listen to music, take pictures than for calling other people. Since human brain is able to accomplish complex tasks, daily activities tend to be boring and our brain simply “implore” us to take a challenge. Smartphones can provide this challenge through engaging mobile games from different genres, most notably: action, strategy, arcade, adventure and logic. These games made me to develop such a game, a strategy one, just because they put your brain to work, involving quick strategically decisions, attention, scheduling skills and objects management.

„Monster Defense” is a strategy tower based game offering a fast real-time path computation system based upon optimizations made on A* path finding algorithm, enemy motion tracking, saving progress capabilities, multiple upgrade options, multiple scenes, intuitive touch input, flexibility to screen resolutions, optimized graphics computations, low memory and CPU overhead, smooth enemy animations and maintainability using modular reusable scripting components.

1 Introduction

All tower based games work on the same principle: you have many waves and enemies grouped in waves that follow a path in order to destroy your basement and the player can stop them by placing towers/turrets in order to destroy each of those enemies. The player can place turrets only in predefined places without being able to alter in some way the path followed by enemies. Altering a path it's like in real world when a road is blocked due to an accident and the police man will force the player to drive on another available road in order to reach his destination. He is not allowed to change the path followed by enemy just because all paths are predefined. A predefined path is an array of way points through which enemy should pass in order to reach your basement.

To better understand the problem, we will take a real world example. Consider you want to drive from one city to another. You don't know how to get to the other city and you don't have a map, but your best friend told you how to reach your destination specifying some distances along with some steering indications. This is an example of what your friend could tell you: go forward for 500m, then steer left, then go another 2 km, then steer right, then go forward and at first road steer left. What happens, if along your way, a police man stops you because there is an accident ? Remember that you

don't have a map and you can't call your friend to ask for help. The same scenario would happen with an enemy in case a turret is blocking his path.

Predefined paths are made before a game is released and the player have no control over these paths. Generally, each different scene has different paths for enemies. It doesn't matter how many paths a scene have, because these are all static.

„Robo defense” is the only mobile game that has real-time path finding, which means you can alter enemies paths by placing turrets in front of them. This is the game that inspired me, because it is the only unforeseeable game from all tower based games. In our example, it's like you've got a map, or you are suddenly transformed in a geographic specialist and you now know to find another available road in order to reach your destination. That means our enemies have now artificial intelligence.

Our main goal was to create a tower based game named „Monster Defense” allowing the player to alter enemies paths.

Next chapters will describe technologies used, main systems that we've developed along with problems that we've encountered, performance, solutions provided and further implementations.

2 Technologies used

2.1 Unity 3D

2.1.1 Description

Unity 3D is a cross-platform engine developed by Unity Technologies and is used to develop video games for PC, consoles, mobile devices and websites.

The game engine targets the following API's: Direct3D on Windows platforms and XBox, OpenGL on Mac and Windows, OpenGL ES on Android and iOS and proprietary APIs on video games consoles. [1]

It provides real-time physics for 2D and also 3D environments powered by NVidia PhysX, complex animation controller called Mecanim, powerful dynamic lightning systems, easy to use editor, Navigation system for pathfinding, UI controls, Audio systems and high level scripting languages support like: C# and JavaScript.

2.1.2 GameObjects

GameObjects are the fundamental objects in Unity that represent characters, props and scenery. They do not accomplish much in themselves, but they act as containers for components, which implement the real functionality. [2]

2.1.3 MonoBehaviour class

This is the base class for every script that is attached to a game object. You can't attach a script to a game object without inheriting from MonoBehaviour class. It has base functions for manipulation and instantiation of game objects and special methods to interact with components.

It worth notice that a script that derives from MonoBehaviour can't be instantiated via classic C# constructor, that means you are not allowed to use „new” keyword for MonoBehaviour class or for derived classes. This is because every instance of such a script should be physically present in the scene. You use “Instantiate” method to create a new instance using a Prefab.

2.1.4 Serialization

Serialization is one of the biggest systems inside Unity engine core. This is because every single game object is serialized, all windows are saving their states using serialization, inspector also uses serialization.

This process uses objects known as Scriptable objects. These are capable to serialize big game objects with complex hierarchies and multiple references with advanced support for polymorphism or chains of inheritance.

2.2 Blender

2.2.1 Description

Blender is a professional free and open source 3D computer graphics software used for creating animated films, visual effects, art, 3D printed models, interactive 3d applications, simulations and video games.

Blender provide features like: 3d modeling, UV unwrapping, texturing, rigging and skinning, fluid and some simulation, particle simulations, sculpting, animating, camera tracking, rendering and video editing. It also integrates a game engine using Python scripting language.

Blender offers you the possibility to create any model you want. It provides a main scene where you can see in real-time your model, intuitive input controls, vast helper functions and a bunch of community add-on's. [3]

2.2.2 3D modeling

3D modeling is the main feature of blender and it is the mostly used. You can create meshes starting from a predefined mesh like: cube, lines, torus, humanoids or you can start from scratch. You will apply different operations on mesh vertices, edges or faces in order to model your 3D object. Blender provides many helper methods or iterative algorithms in order to help you to develop faster.

2.2.3 UV Unwrapping

It's the process of taking all your model "skin" from 3D and represent it on a 2D surface. Every more complex model will pass through this process which is a hard one. You need to think from beginning where to place your model edges, otherwise you will spend a lot of time trying to finish this process. This is done because it's easy to paint something in 2D than in 3D.

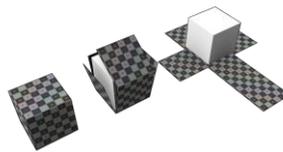


Fig. 1: 3D cube UV unwrapped [3]

2.2.4 Rigging, skinning, weight painting

When we talk about animating humans or any living creature, it's a bit more tricky than with other mechanical objects.

Rigging is the process of creating bones and putting them into place in order to create a skeleton for our model. A skeleton is a group of bones that controls any attached vertex. This process is done to ease the animation process. Generally, you will add bone constraints and root bones. It's like

creating a real body that has bones that can be moved or rotated, but within a certain range or angle. The same constraints must be applied in this process.

Skinning is the process of defining groups of vertices and bones that affects this groups. You must also provide a influence of the bones for each group of vertices, being able to use blended influences. Adding bone influence is known as weight painting.

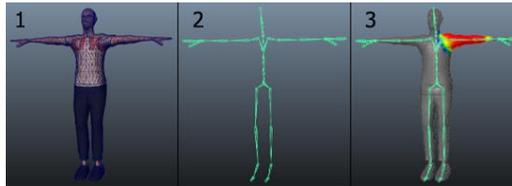


Fig. 2: *Skinning(1), rigging(2), weight painting(3) [3]*

2.2.5 Animating

Animating is usually the last process, but probably the hardest. Mainly, you will insert key frames for different properties at different moments of time. To achieve high quality animations you may use specialized software and hardware used in 3D animation films.

2.3 Texture packer

2.3.1 Description

It's a free tool if you just need basic capabilities. It's main feature is to create texture atlasing, which is the process of combining multiple UV textures into a big one. This is done to reduce draw calls in Unity, which improves a lot graphics computation.

2.4 Mixamo

2.4.1 Description

Mixamo is a website that provides 3D character models and animations, tools for online rigging, skinning and weight painting. It also provide a tool called "Decimator", which is used to lower polygons count for a 3D model. Mainly used for mobile platforms or games where you have lots of models.

3 Main systems developed

3.1 Real time path finding system

3.1.1 A* algorithm

A*, pronounced A star, is a computer algorithm that is widely used in path finding and graph traversal. It is widespread for it's high performance and accuracy and comes as an extension of Dijkstra algorithm. A* achieves better performance by using heuristics to guide its search.[4]

The algorithm is using some collections to keep track of visited nodes, neighbors of visited nodes and some sorted nodes based on a total cost(cost for move + estimation cost). It's crucial for performance to use the most appropriate type of collection to keep track of your data. We've tested

many data structures including: different types of trees, static lists, linked list, ordered list, hashtable and hashset.

We've picked the linked list approach. As we said, we needed a list with sorted nodes by total cost and that was done by maintaining the linked list sorted while adding new nodes. No sorting algorithm was applied, because even with the best sorting algorithm your path computation would be too slow.

Although on the internet you will find many implementations, we ended up creating a custom implementation. At first, we've used Unity's navigation system, which is a very powerful one, also based on A* algorithm. It was good, until I've created more enemies and all have to compute their own path and enemies started to freeze for few milliseconds, but was enough to observe that. Was a good reason for me to start searching for another, better, implementation.

We've found Aron Granberg implementation on the Unity Asset Store. There was a free version of the same A* algorithm. It was more efficient than what Unity provides. We've managed to integrate it with my project. The time was satisfying for about 7 - 8 enemies path computations at a given time. Average time: 21ms per path computation. You don't even notice that enemy stopped for few milliseconds. The free version has huge classes with very hard to understand methods. Documentation was provided, but it was poor. No unit test provided so it was impossible for me to track bugs when adding new feature and in the end we've dropped this implementation too. It wasn't what we needed.

Next, we've decided to create our own implementation with everything that we need. Taking in consideration that we were targeting mobile devices, creating such an implementation was a real challenge.

Our implementation average computation time is: 2 ms for each path. Worst case is when a path is blocked nearly destination, because in that case all nodes will be visited. In this case my algorithm computation time is: 3.5 ms. On the above implementations we get an average computation time of 25 ms for each path.

3.1.2 Heuristics

A heuristic is a technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution.

Notably heuristics are: Manhattan distance, Diagonal distance and Euclidian distance. We have used Manhattan Distance, because it is the ideal heuristic for grids that allows four way movement: up, down, left, right.



Fig. 3: Manhattan distance [4]

Green point represents starting position and red point represent destination position. Points coordinates are used to compute so called heuristic cost, also know as estimation cost. It's important to note that a better estimation of the cost means a faster and more precise algorithm. Our algorithm can be simply used on computing paths in any number of direction. All you need is to provide the appropriate heuristic function.

3.1.3 Editor precomputations

In Unity, you have the ability to add new features to the inspector, but also to the editor. This is done through special C# scripting. We've used this to generate the map and precompute estimation costs for all nodes. Since Unity serialization system knows how to save primitive types, all costs are already computed before game runs. This is very useful for performance reasons, but in case the target is moving at some point, I've also implemented the possibility to recalculate this cost during run-time with lowest CPU cost.

It uses a special process called „raycasting” which detects whether a node should be “walkable” or “unwalkable” based on the mesh under it. Your enemy is never gonna walk in air, it will always have something under his feet.

Map generation is also done from the Unity editor using another scripting component. You simply specify number of rows, columns, a prefab which contains scripting for nodes and starting point.

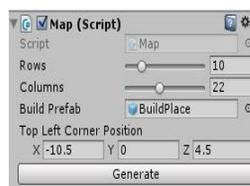


Fig. 4: Map generation script

One problem that we've encountered was that Unity's callbacks are called at different times and with different frequency while in editor mode. You must know how Unity serialization system works, because everything that you change in the editor must persist after closing Unity and this involves special callbacks implementations in order to accomplish data persistence.

3.1.4 Build places

In order to compute enemies path, you just need some nodes to walk on them. We've called them “build places”, because these are the game objects on each a turret can be placed. Basic scripting components are attached in order to know if a turret can be places or not and for changing place state.

3.1.5 Path checking

Since we wanted to make everything as fast as we can, it was simply very expensive to compute a path when is does not exist for all enemies. This situation appears because the player is free to place a turret, even if by placing the turret, there will be no more available paths. In this scenario there is one way to find a path, we try to compute one. Note that we compute only one. If there is one available, then all others enemies will recompute their paths, else, your turret is not placed and no other path computation is done.

3.2 Custom serialization system

Unity internal serialization system knows how to serialize complex game objects, but it does that only for scriptable object which can live only in editor and that means you can use those only for custom windows data serialization. You are not allowed to attach a scriptable object, which is a normal C# script inheriting from ScriptableObject base class, on in game instances. Since games tend

to have complex game object, it is impossible for one serialization system to take care of serializing any kind of game object, so you have to implement your own.

The biggest challenge was how to implement such a system in a generic way. It is very probably to add more types of enemies, more turrets and many other objects during time. The challenge was to create a system that is able to serialize all of these without changing any single line of code.

We took some informations from inside Unity serialization system. Unity programmers said something like that: we will not take care which is your serialization representation, we will only take care of saving that representation somewhere on the disk and we assure you will receive the right serialized file when you want to load your previous state. This is what we've done. We asked the object to serialize itself and then we've saved it on persistent memory storage. It's generic, easy to use and totally decoupled.

3.3 Touch input controllers

All engines are frame dependent, which means all your code is executed in a frame. A frame is a small time interval in which every script executes and every object on your screen is rendered and after that any touch input is stored. When you have too many computations, your frame rate is decreasing, which means your scripts are called fewer times and at low frame rates, your game is responding very slow or even freeze.

Since all engines are frame dependent, you have to create custom scripts to handle touch gestures, because a gesture is a time dependent action and you need to collect and process data from multiple frames in order to take a decision.

3.3.1 Turret instantiation

Since we've made a 3D game, we are using a camera with perspective view, which is placed at certain angle from where you can place your turrets. We've implemented turret dragging and we'll explain why it is a little bit tricky to create such an implementation.

All inputs are in 2D coordinates, which are pixels from screen. In 3D, these pixels have no meaning, because in 3D there is not unit to measure distances. Everything is relative to something. So we need a way to convert those pixels base on what camera can see, from what angle and height in order to get some world coordinates. Note that additional conversions are made if mobile screen is rotated and some constraints apply also.

3.3.2 Zooming

The same scenario from turret instantiation is applying to zooming scripts. We need to collect input data and process it over frames. We are using linear interpolation to know the exact value of zooming. Zooming is done modifying the "field of view" property of the camera. The user may adjust settings like zoom speed, sensitivity, or even to disable zooming.

3.3.3 Dragging camera

After you zoom in, you are able to move around to explore your map. It's like you are zooming a photo and than dragging it around. The same principle is applying here, but remember that we are on a frame dependent system, not time dependent.

We need to provide logic for letting the player to drag around, but not too far from map edges. When the player wants to zoom out, camera needs to modify its position accordingly, because if it stays where it is, the player will see only a part of the map, which is not the intended behaviour.

3.3 Turret artificial intelligence

On a turret are attached many scripts which take care of enemy tracking, attacking, turret upgrading and turret placing.

Enemy tracking is achieved using co-routines and some Unity callback for collision detection. We've tried to make this script as simple as we can, because collision detection has a CPU footprint and using this redundantly will slow your game. This script is takes in consideration turret range.

Attacking script is based on turret fire rate. It handles types of missile used, what movements should missile have, where to spawn the missile and if turret is able to fire.

Turret placing is used at turret instantiation.

3.4 Turret upgrading system

It's responsible for initiating a turret upgrade, handling turret selection, performing physical turret upgrade. More than one script is involved, just because some of them handle UI input and others logical actions.

When user wants to upgrade a turret, these steps are executed: turret upgrading is initiated which means that the turret upgrade script is queried for all available upgrades, then all updates are loaded and passed forward in order to be displayed, then a player input is waited. After input comes a script process it and there are two cases: canceling or selecting a turret upgrade. If player selects a turret, a script handle physical turret upgrade, otherwise UI that is showing available upgrades is closed.

Note that upgrading require some approval from „Cash system”, because you can place a turret only if you have the amount of cash required.

3.5 Waves system

It's responsible for triggering enemies waves. We've spent time thinking how to create it in order to be generic, extensible and easy to maintain. It uses special functions called “co-routine”. A co-routine is a function that is using the yield return instruction to avoid every frame update which has impacts on performance side. This task is solved by executing something from a method, but not the entire body and when it is called second time, it resumes from where it let off.

We can change number of waves, spawning positions, numbers of enemies for each wave and also types of enemies spawned at each wave.

3.6 Object pooling

In many applications and games, you will encounter situations where you need many instances of the same class that have short life time. In Unity, as in any other framework, creating and destroying objects is not an easy task. Since performance is important, we ensured that we are reusing short living game objects.

Script from this system receives queries from others scripts to give them instances of different types. If no such instance is available, it will return a brand new one, else it will return an available one in a “fresh” state. A fresh state means that object “looks” like a new one.

This system is not very complex, but it is a must in such a game where bullets and enemies are created all over the time.

Main advantage of this system is that it is not framework dependent. You can easily use it on other app or game that is not made with Unity. We achieve that using Humble Object Pattern, which says to create a non-monobehaviour class with all the functionality and everything that is framework dependent to be putting in other class. Then we’ve used composition to use the first class instance.

Note also that this must be a Singleton script, meaning you are allowed to have only one such instance in the entire game.

3.7 UI

In the Main menu of the Monster Defense game are located three buttons as described:

New game - this button will open a menu from where the player select a scene and it will trigger deletion of all saved data

Load - this button will load last active scene preserving all saved data and all saved settings



Fig. 5: Monster Defense main menu

Settings menu contains different sliders for customizing the in game player experience and it also support saving all those settings which spares player from resetting them when he plays again.



Fig. 6: Monster Defense settings menu

In game panel offers information about you current score, available cash, number of lives remained and wave number.

The player has access to „Fast forward” button which will make all time dependent scripts to run twice faster.

In the middle bottom of screen is a button that supports dragging and is used by the player to place it’s turret.



Fig. 7: Monster Defense in game scene

4 Conclusions and further directions of study

Due to its highly optimized systems, „Monster Defense” is a strong competitor among tower based games. Our game has the advantage in front of its major competitor, which is „Robo Defense”, that it is able to handle multiple input touch controls over different mobile platforms like: Android, Windows Phone, iOS, it provides platform independent serialization system, highly optimized path finding system based upon A* search algorithm and decoupled scripting components which allow us to reuse them in other projects even outside Unity framework, because path finding in framework independent.

On top of priorities we will create dynamic scene loading from server which provides in game challenges to players, without being forced to download again the entire game from store and online high scoring using Azure service, which is already done, but it needs to be integrated with Unity framework because it is using something that Unity does not support currently and further changes are required.

References

- [1] Unity 3D, [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
- [2] Unity components, <http://docs.unity3d.com/Manual/index.html>
- [3] Blender, [https://en.wikipedia.org/wiki/Blender_\(software\)](https://en.wikipedia.org/wiki/Blender_(software))
- [4] A* search, https://en.wikipedia.org/wiki/A*_search_algorithm

Hodină Dragoș
Faculty of Science “Lucian Blaga”
Department of Informatics
Str. Dr. I. Ratiu, No. 5-7, Sibiu, 550012, România
E-mail: hodina.dragos@gmail.com

”Dodgy Ninja”

Flavius Holerga, Andrei Milik, Inocențiu Moga
Teacher Coordinator : Gabriela Florea

Abstract

The application “Dodgy Ninja” is a game designed for all. It is created for both gamers and people who want to spend their spare time in a pleasant way. The creators aim to catch the users interest by provoking him to a challenging game in terms of difficulty. The theme of the game is survival in ‘ninja style’, as long as possible, on a narrow piece of wood, under the pressure of obstacles thrown towards the character. Its difficulty will increase during the round. The obstacles come from a wide assortment of comic objects, such as balls, rockets, etc.

1 Introduction

This game is designed for people of any age aiming to captivate and entertain the audience in a challenging way. The character must be controlled in order to dodge the projectiles that are coming in his way therefore gaining points. With the points earned, in the shape of rubies, the player can buy and unlock new characters and backgrounds to personalize the game.

The application was made in GIDEROS Studio, using Lua as programming language. The graphics were made in Inkscape and the animations in Spriter. As for collision, a version of Box2d was used.

In order to control the character, the player can choose from a couple of methods:

- Using the first mode, the player can control the character by pressing the left or the right side of the screen making the ninja move accordingly.
- The second mode provides a more interesting way to control the ninja by tilting the device left or right, process that makes the character move respectively.
- The third mode, even if it doesnt aim for practicality, provides an engaging way to move the ninja using the microphone incorporated in the device, therefor making the character move depending on the intensity of the external noise, which allows the player to control the game using his voice.



2 Description

Gideros Studio is a software designed for Android, iOS, Windows Phone and HTML5 application development, with the advantage of it being free, easy to use with a simple but basic interface. Furthermore, Gideros Studio presents infinite possibilities offering an easy way to integrate plugins (written in either C++ or Java) in the development process. Even more, Gideros Studio allows the developer to test his application in middle development, offering the option to test the app through Wi-Fi or through a built-in emulator, which makes testing very easy and extremely fast compared to other softwares.

In order to access the accelerometer and retrieve the rotation of the device for the second mode we used the following code which basically stores three numbers in three different variables, each corresponding to a rotation axis, which then are used to implement the character motion.

```
local function compensateAccelerometer(x, y, z)
    local orientation = application:getOrientation()
    local deviceOrientation = application:getDeviceOrientation()
    local p1 = orientation == "portrait" or orientation == "portraitUpsideDown"
    local p2 = deviceOrientation == "portrait" or deviceOrientation == "portraitUpsideDown"
    local rotation = (p1 == p2) and deviceOrientation or orientation
    if rotation == "portrait" then
        return x, y, z
    elseif rotation == "landscapeLeft" then
        return -y, x, z
    elseif rotation == "portraitUpsideDown" then
        return -x, -y, z
    elseif rotation == "landscapeRight" then
        return y, -x, z
    end
end

local function onEnterFrame()
    local x = accelerometer:getAcceleration()
    x = compensateAccelerometer(x, y, z)
    fx = x * 0.1 + fx * 0.9
    ninja:setRotation(80 * fx)
end

stage:addEventListener(Event.ENTER_FRAME, onEnterFrame)
```

In the function *compensateAccelerometer(x,y,z)* the orientation of the device is identified, retrieving the orientation values accordingly.

The function *onEnterFrame()* is basically the main function of the game in which the accelerometer is called, then the values received are processed through the first function, to which there is applied a filter making the rotation constrained by a rotation maximum of 80 degrees, as well as a speed maximum.

In order to make the third mode possible, we used the following code to retrieve the volume detected from the built-in microphone from the user's device:

```
local microphone = Microphone.new(nil, 22050, 1, 16)
microphone:addEventListener(Event.DATA_AVAILABLE, function(event)
    local ev = event.peakAmplitude
    ev=(ev-0.5)* 2
    ninja:setRotation(82 * ev) --rotate the image of the character
    ninja.box:setRotation(82 * ev) -- rotate the collision box as well
    ninja.box:setPosition(xh+ev * 82,yh)
end)
```

Game characteristics:

- A big variety of characters and backgrounds
- The possibility of sharing the score on social media (Twitter, Facebook)
- Sound effects
- Simple and professional graphic style



Fig 1: Some of the backgrounds used in the project

3 Functionality

The theme of the game is survival in "ninja style", the character being challenged to resist and dodge as many projectiles as possible (as you can tell by the name of the application). The player is informed by the direction from which the projectiles are coming through a sign which can be either yellow, which means that the projectile is coming from one of the corners with a diagonal trajectory, or brown, which means that the projectile is coming from the sides with an horizontal trajectory.



Fig.2: Character tilted to the left in order to avoid the projectile

If the warning sign is brown, then the projectile will take the shape of a box which the player can avoid only if he tilts the character to the maximum in one of the ways.

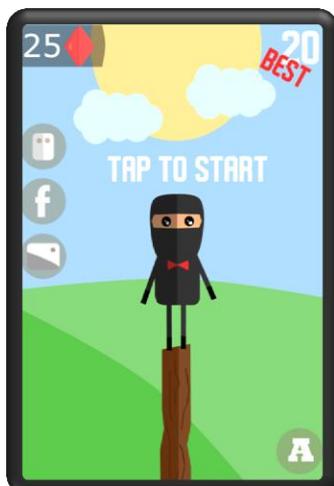


Fig.3: Game menu

Furthermore, at random intervals of time, falling rubies will show up, serving as currency for the game, the player has the option to rotate the ninja intercepting with their trajectory in order to collect them. Using the rubies collected in game, the player can afford new characters and new backgrounds which he can equip from the shop menu.

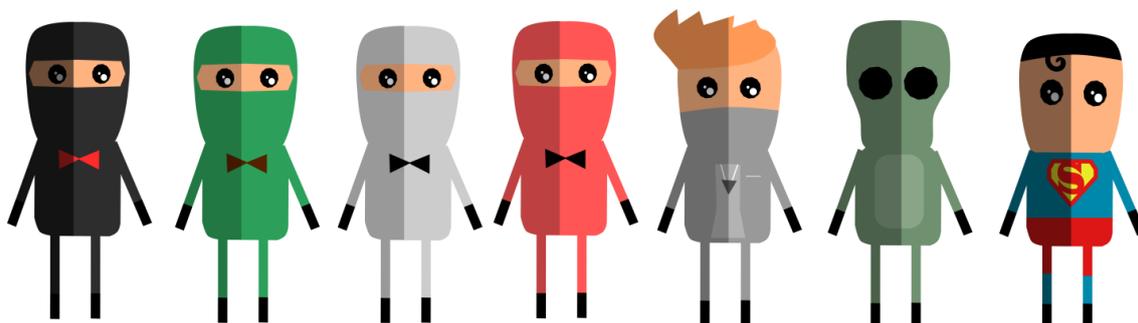


Fig.4: The first seven characters found in the shop

4 Conclusion

In conclusion, the application does not have an ending, being a repetitive game which aims to be entertaining for a long period of time, depending on the player. It will be found on Google Play for free. Furthermore, we intend, based on future reviews, to improve the art style and to add more elements to the shop aiming to make the game as addicting as possible.

Flavius Holerga
L. T. "Axente Sever"
Matematică informatică
Mediaș, România
E-mail: flavius.holerga@gmail.com

Andrei Milik
L. T. "Axente Sever"
Matematică informatică
Mediaș, România
E-mail: andreimilik30@gmail.com

Inocențiu Moga
L. T. "Axente Sever"
Matematică informatică
Mediaș, România
E-mail: inocentiumoga@gmail.com

OpenGL Game Engine

Svetlozar Iliev
Coordinator Professor: Katalina Grigorova

Abstract

This paper describes the creation of a 3D game engine, for use in an introductory video game programming. Having the right tools available can make illustrating the concepts of game development and design substantially easier. In creating the engine, a number of factors such as ease of use and accessibility, were considered. Successful implementation of tools allow clients to produce games quickly and the concepts of game development to be explored more effectively.

1 Introduction

A typical game engine is a software framework designed for the creation and development of video games. Developers use them to create games for consoles, mobile devices and personal computers. The core functionality typically provided by a game engine includes a rendering engine ("renderer") for 2D or 3D graphics, a physics engine or collision detection (and collision response), sound, scripting, animation, artificial intelligence, networking, streaming, memory management, threading, localization support, scene graph, and may include video support for cinematics. The process of game development is often economized, in large part, by reusing/adapting the same game engine to create different games, or to make it easier to "port" games to multiple platforms.

In many cases game engines provide a suite of visual development tools in addition to reusable software components. These tools are generally provided in an integrated development environment to enable simplified, rapid development of games in a data-driven manner. Game engine developers attempt to "pre-invent the wheel" by developing robust software suites which include many elements a game developer may need to build a game. Most game engine suites provide facilities that ease development, such as graphics, sound, physics and AI functions. These game engines are sometimes called "middleware" because, as with the business sense of the term, they provide a flexible and reusable software platform which provides all the core functionality needed, right out of the box, to develop a game application while reducing costs, complexities, and time-to-market — all critical factors in the highly competitive video game industry. Gamebryo, JMonkey Engine and RenderWare are such widely used middleware programs.

2 The Game Engine

2.1 Main functionality

The basis of this Game Engine is to propose a dynamic and flexible way of creating an actual virtual world. It offers various different development tools and reusable software components to generate such a world. The Engine is divided in two main parts. The first one is the actual

Rendering pipeline and the second is the Physics pipeline. Combined those two components give the client everything he needs to build up a world from the ground. The Physics engine is tightly incorporated with the Rendering pipeline which gives a realistic and dynamic world manipulation. The Engine includes various other types of Graphics techniques such as:

- Lighting – Diffuse, Specular and Ambient
- PCF and Variance Shadow Mapping
- Real Water Simulation
- Day/Night Cycle, OBJ Models
- Height Map Model Generation
- Blend Mapping
- Collada Model Animation
- Normal Mapping
- Parallax Mapping
- Particles Engine
- Ray Picking
- Dynamic Space partitioning
- Saving/Reading Scenes from file
- Sprites and Textures
- Dynamic Fog Generation

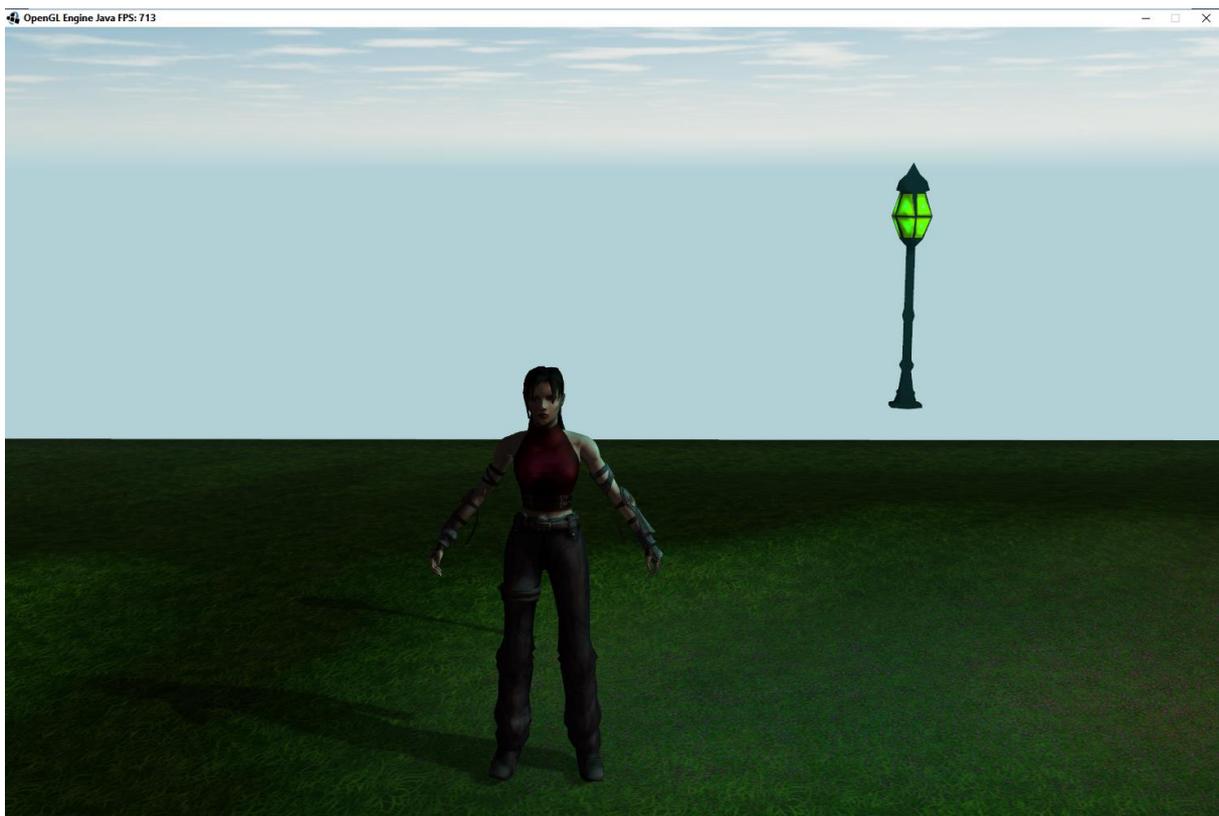


Fig.1 Static Models Overview

2.1.1 The Engine and client interaction

When the application loads, an interactive window opens which shows the last or currently loaded scene by the scene loader. The Interface with the client is currently in beginner stage and there are not very many interaction tools to help the client build up the world as fast as one would

imagine. At the moment most of the Dynamic world loading is done using the command line. Some basic commands are implemented to help the client interact with the engine. The Graphical user interface is still in early stages, but the next step is to implement a fully functional GUI to completely ease the communication between the application and the user.



Fig.2 Bump and Specular Mapping

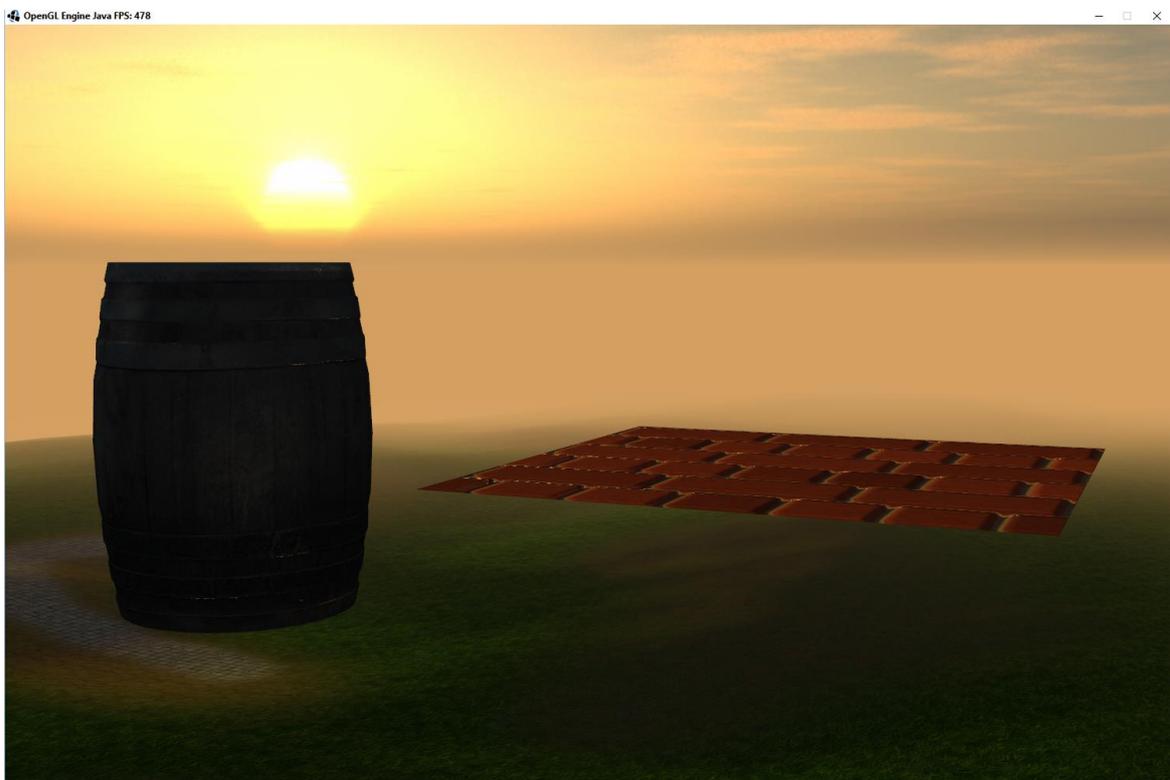


Fig.3 Day/Night Cycle and Parallax Mapping

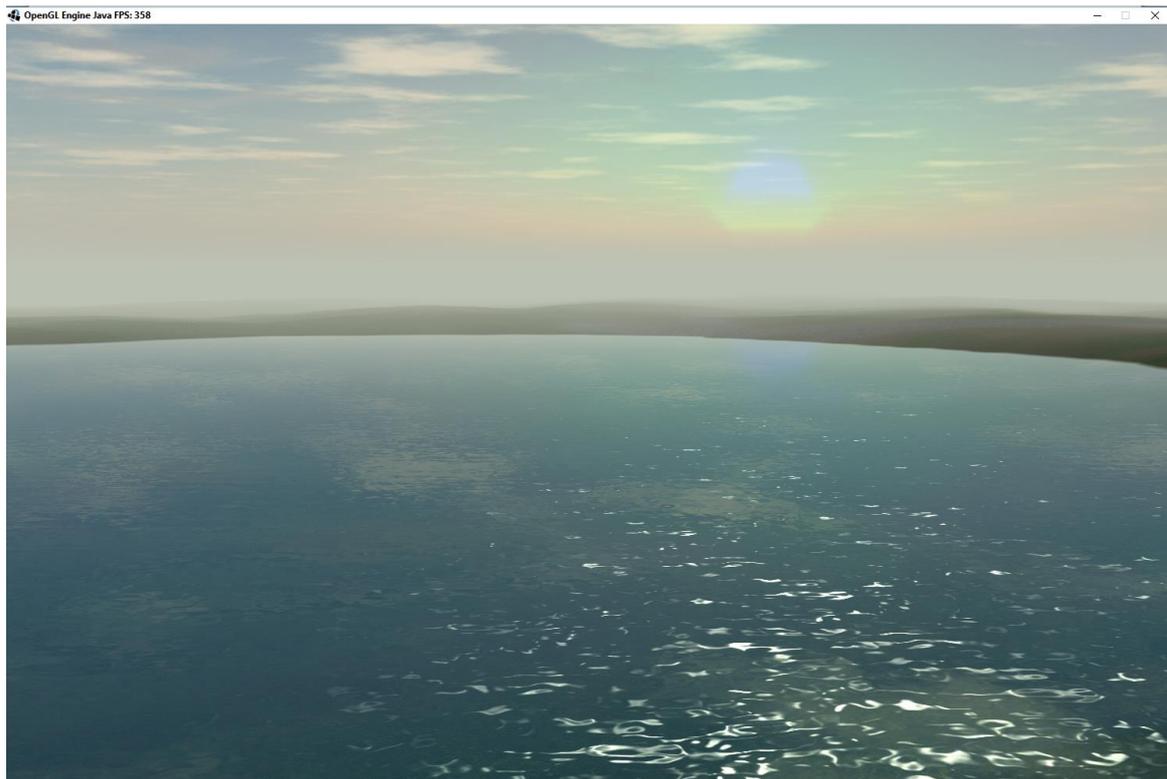


Fig.4 Water Simulation



Fig.5 Particle Engine

2.1.2 The Rendering pipeline

The first major part of the engine itself is the graphics pipeline. This module is concerned with the visual display onto the screen. Most of the core functionality is implemented from this module

such as: Lighting, Shadows, Water, Blend, Diffuse, Normal Parallax mapping and so on. This module is responsible for the view the client receives while the application is running.

2.1.3 The Physics pipeline

The second major part of the engine is the physics pipeline. This module is concerned with the dynamic and real time interaction between the objects and models in the virtual world environment. Physics pipeline includes a Physics Engine which is the main driving force of the pipeline, it also includes various modules which take care when the physics pipeline interacts with the graphics pipeline.

2.2 Engine Implementation

The Engine is implemented using the programming language Java and the OpenGL API. The main library used for this project is called LWJGL which is a wrapper library for the native OpenGL libraries. LWJGL includes very many library API's within itself such as: OpenGL, Math, System Utilities and more. The application is in a very early stage of its development, there are many more functionalities that will be added in the future.

Once finished the Engine will offer users to create huge dynamic worlds in a blink of an eye which will operate in real time.

3 Conclusion

The designed application serves as an interactive tool for users to create a virtual world of their liking without any prior knowledge in graphics or physics. The Engine offers its users easy and fast to use tools for dynamic and real time world creation. Furthermore the author plans at expanding the capabilities in the following areas:

- Friendly GUI for much better interaction between client and application
- Multiplayer support
- Post processing effects
- Vehicle physics
- Enhanced animation interaction
- Saving or Loading a dynamic virtual world from external source
- Support for script languages - LUA

References

- [1] OpenGL Superbible: Comprehensive Tutorial and Reference (7th Edition) by Graham Sellers (Author), Richard S Wright Jr. (Author), Nicholas Haemel (Author)
- [2] OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3 (8th Edition) 8th Edition
- [3] OpenGL 4 Shading Language Cookbook - Second Edition Kindle Edition by David Wolff (Author)
- [4] OpenGL ES 2.0 Programming Guide 1st Edition by Aaftab Munshi (Author), Dan Ginsburg (Author), Dave Shreiner (Author)

Svetlozar Iliev
University of Ruse
Department of Informatics and
Information Technologies
8, Studentska str., Ruse 7017
BULGARIA
svetlozar321@gmail.com

Smart Backpack

Vlad Ionescu, Emanuel Giurgiu, Vlad Murzea, Sebastian Vidrea
Coordinator Professor: Livia Sangeorzean

Abstract

In our days children are very attached with technology and their thirst for knowledge is growing up day by day. Their continuous curiosity to discover new thing together with their “spongy minds” inspire us to increase our possibilities and to oversteps our limits.

As we all know school and technology are hand in hand and the “old school methods of learning” are overcome for our days. Looking back, the traditional ways of writing with pencil and paper or the dust of the white Crete bring us a nostalgic feeling that futures generations wont get to know it. Even for a child from the first grade all this sound hilarious and definitely are not “cool” anymore. Not only that the 21-century school technology is more attractive but also the studies proved that is more efficient. All this progress of the educational system have made the child more open minded, eager and excited about learning. In the past years the government spend millions of dollars to update and modernize the equipment in order to satisfy the needs of the pupils. All in all their effort have just encouraged the importance of having them in the educational process.

This era of evolution affect also another category and we refer here at the parents. They need to have a job, they build a carrier, they work at their own education, maintain a family and so they need to perform multiple roles in our society. One of the biggest and heaviest would be the parent role. The technology requirements may be often over their heads and all this busy life makes them sometimes to forget things like the schedule of their children.

Last but not least the teachers are also affected. Besides they are working at school, they also need to work at home, for preparing the lessons, and the test papers. In this way the professor works for little and unimportant things instead of according sufficient time for the students.

Therefore we have tried to create a concept that combines all the solutions to the problems from the three perspectives discussed above.

1 The beginning of the project

Like most of the greatest projects the beginning was a brainstorming.

Without a starting point we have discussed subjects from different areas. After a while we realized that we don't have a strong idea for the project, so we have started thinking about the main problems that we have in everyday life.

We wanted our project to be a part of the Internet Of Things family. That meaning that our project won't need to many interactions to fulfill its purpose. We wanted to keep it simple so that in the future it could be easily integrated in many fields. So that is how we came up with the Smart Backpack

The thing that a child never forgets when going to school is his backpack, no matter if he actually

did not change the books from one day to another. As a child the responsibility and attention levels are low and in order to actually skip a motherly fight or a talk with the teacher they need to increase their level of maturity.

The Smart Backpack will ultimately remind the child of his daily schedule with the help of the display and remind him of the books he needs to take with him.

The parents will also be relieved to not actually need to look in the backpack and just check the display. It is also a good way for them to check the child's homework for the day.

The real issue also is in regard of the teacher. If a child forgets his/her books the class is compromised for him thus making the teacher's work much harder. The backpack is the perfect solution for the children's lack of attention and also a stress reliever for the parents and teachers.

2 Scenarios and Personas

2.1 Personas

The client suited for the backpack would be children between 6-12 years old, due to its cubic form and uses. Between the ages mentioned, children often forget to pack more than one of the things needed for school time. More than anything else, the most important things often forgotten are manuals and notebooks useful for the classes. Also, in the list of always forgotten for school are the clothes you need for gym class and sandwich for lunch.

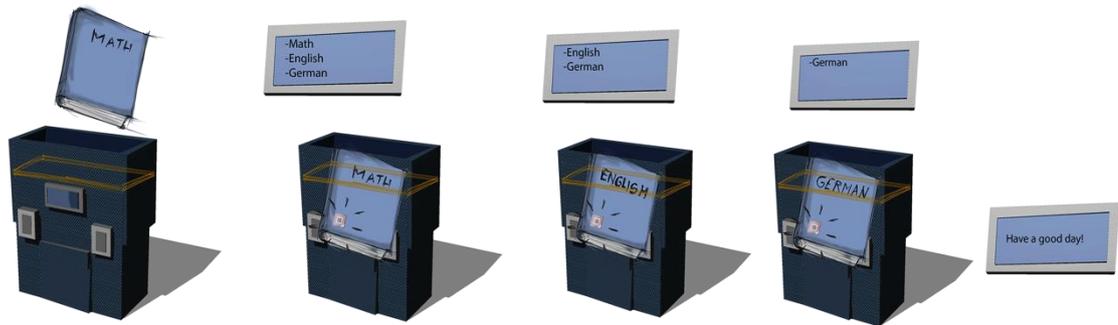
The Smart Backpack can easily scan the objects needed as they are put in the backpack. This easy technology does not need technological or informatics related knowledge thus making it perfect for children. Once the object is scanned it will appear on the screen while a led will light up confirming the action.

2.2 Scenarios

The Smart Backpack was designed to meet every possible problem and primarily solve it by itself. The only problems that could occur and could not be handled by the backpack are the technical ones which need to be taken to a professional. But as I mentioned, the backpack was designed to meet every problem and have none to minimum technical ones. As you start the backpack the only problem, which could occur, is the low battery, the problem that could be resolved by just connecting the battery to a power source. If the problem does not appear, the backpack should automatically go in 'Stand By' mode.

The next step is for it to search for an Internet connection through wireless. If the device recognizes the Wi-Fi signal, it will automatically update its schedule for the day. If the device does not find the connection or is limited, it will take the schedule stored in the device's memory. If there is no schedule in the memory device for the specified day, then on the screen will appear the message 'Error'. If this message occurs, we advise the user to find a wireless connection. Once the schedule is created, through wireless connection or the backpack's memory, the LCD will be automatically be updated to show the remaining objects needed for that day. The backpack will go on idle and will be waiting for the input, or scanned objects. In this situation the only thing that could go wrong is if the scanning is not properly done and in such case the LCD will show the message 'Error' in which case you should try again. If the input item is read and accepted by the device, the schedule will be update and it will Loop back to a previous step, which is updating the LCD with the remaining items needed for the day.

3 How dose it work



As we say it by now our product is quite simple. We didn't want something that is difficult to use or has a complicated interface.

This are the hardware components that we used:

- Arduino UNO Board
- RFID reader
- Arduino WiFi
- RFID Tags
- Battery

In order to make the connection between teachers and the backpack we made a cloud. Information will be uploaded to the cloud, only parents and teachers have access to the cloud. This is the cloud design: We will expose more details about this part of the project in the next chapters. First we created the code for the Arduino WiFi because the backpack needs to take the schedule from the cloud. The Arduino will connect with an SSID and the Password to the cloud and will collect all the information that the backpack will need.

The next step was to make the Arduino UNO Board with the RFID reader to interpret the data provide from the Arduino Wifi. And then we verify if the date are inside the backpack.

With the help of an LCD, integrated in front of the backpack, we will show to the student if one of the books is missing or if everything is all right.

Basically this is how it works:

1. You put the book (that contains an RFID tag) inside the backpack.
2. The backpack will check if the book is in the schedule for today, if it is he will deleted from the LCD.
3. When you finish putting all the books inside the LCD will print a message on the LCD.

4 Software and Materials

The software and materials that we used for our project will be presented shortly in this chapter. We tried to merge the greatest performance with our limited budget, so we chose the best software and materials that could be found in our budget

4.1 Arduino Uno Micro-controller .

For programming the micro-controller we used Arduino IDE. This contains all the software which will run a computer in order to program and communicate with an Arduino board. The Arduino IDE contains an editor which we can use to write sketches (that's the name of Arduino programs) in a

simple programming language modelled after the processing language. Using the IDE the program we have wrote is converted to C language and then compiled using AVR-GCC. This process produces binary code which the micro-controller on the Arduino board will be able to understand and execute. When the Arduino board is connected to a computer using the USB cable, by using the IDE we are able to compile and upload to the board the program.

Arduino is a kind of a “mini-system” based on a AVR micro-controller, together with an other chip that provides USB connectivity and a voltage regulator.

The Arduino micro-controller is also cheaper than other micro-controllers. That was also a thing that made us choose Arduino instead of other marks. This way we could save a little money and our prototype can have a better price and can be also accessible to a larger variety of people.

Arduino includes a “boot-loader” to take advantage of the USB chip, which simplifies programming, plus a dedicated language ("C" based) that includes a lot of rewritten routines. So the programming would be easier on an Arduino board than on any other micro-controller.

We consider the Arduino to be a "development tool" rather than a micro-controller. It's kind of a mini-motherboard (though very, very simple.). Also the Arduino programming language is more accessible and we find it the most perfect way to program and develop our project.

4.2 RFID Shield

RFID/NFC Shield is a Near Field Communication interface for Arduino built around the integrated circuit and SPI communication between Arduino and NFC Shield RFID/NFC is a short-distance radio technology that enables communication between devices that are held close together. RFID/NFC traces its roots in RFID technology and is an open platform technology.

RFID/NFC technology is regulated by Near Field Communication Forum which standardises NFC communication between devices -- how they pair, share data and allow a secure transaction to happen. NFC operate on ISM band of 13.56 MHz Frequency. NFC communication range is up to 10 cm. But, this is limited by the antenna and power radiation design. Most devices work within a range of 10 mm. NFC Shield antenna is designed to work within a range of 1 cm.

4.2.1 Features

- ♣ Arduino Shield compatible. No soldering required.
- ♣ SPI interface. Hence, most Arduino pins are available for other applications.
- ♣ Built in PCB Antenna.
- ♣ Supports both 3.3 V and 5 V operation.
- ♣ Socket to connect other shields. WIFI module ESP8266

4.3 ESP8266

The ESP8266 WIFI Module is a self contained SOC with integrated TCP/IP protocol stack that can give any micro-controller access to your WIFI network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WIFI-ability as a WIFI Shield offers. The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front- end module, is designed to occupy minimal

PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth interfaces at the same time, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support.

4.3.1 Features:

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2 ms
- Standby power consumption of < 1.0mW (DTIM3) Radio-frequency identification (RFID) RFID is the wireless use of electromagnetic fields to transfer data, for the purposes of automatically identifying and tracking tags attached to objects. The tags contain electronically stored information. Unlike a barcode, the tag does not necessarily need to be within line of sight of the reader and may be embedded in the tracked object. RFID is one method for Automatic Identification and Data Capture.

Since RFID tags can be attached to cash, clothing, and possessions, or implanted in animals and people, the possibility of reading the linked information is very simple and we found this to be the best solution for our prototype.

4.4 LCD Screen

A liquid-crystal display (LCD) is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals. Liquid crystals do not emit light directly.

The LCD screen is more energy efficient and can be disposed of more safely than a CRT. Its low electrical power consumption enables it to be used in battery-powered electronic equipment like our smart backpack. It is an electronically modulated optical device made up of any number of segments filled with liquid crystals and arrayed in front of a light source (backlight) or reflector to produce images in monochrome.

The LCD screen on our backpack is used to show which of the books/materials that are supposed to be in the backpack are left outside. The children are automatically announced when something is not in their backpack so they can add it right away.

It is a simple method that we chose to use to simplify the child's job. He can see very easily what is missing and he wouldn't lose any time to think about what is left out.

5 Conclusions

We consider that we achieved our initial goals to make a IoT object that is autonomous and easy for the end user to use. Our design could be implemented in other fields, such as enterprises, shipping business etc. We also want to continue and improve our project until be it could be implemented in the daily life.

References

- [1] Wi-Fi module ESP8266, data sheet, <https://cdn.sparkfun.com/datasheets/Wireless/WiFi/ESP8266ModuleV1.pdf>.
- [2] RFID Shield , data sheet, https://cdn-shop.adafruit.com/datasheets/PN532_AntennaDesign_v1.0.pdf
- [3] Brian W. Kernighan, Dennis M. Ritchie. *The C Programming Language, 2nd Edition*, 1988.
- [4] Simion Monk, *Programing Arduino*, 2011.

Ionescu Vlad
Universitatea Transilvania
Facultatea de Matematica Informatica
BLVD-ul Eroilor nr.3
Romania
E-mail: ionescuvlad12@gmail.com

Murzea Vlad
Universitatea Transilvania
Facultatea de Matematica Informatica
BLVD-ul Eroilor nr.3
Romania
E-mail: vlad.murzea@gmail.com

Giurgiu Emanuel
Universitatea Transilvania
Facultatea de Matematica Informatica
BLVD-ul Eroilor nr.3
Romania
E-mail: emanu.giurgiu@gmail.com

Vidrea Sebastian
Universitatea Transilvania
Facultatea de Matematica Informatica
BLVD-ul Eroilor nr.3
Romania
E-mail: vidreasebastian@gmail.com

Angry Driver

Marinescu Marius Daniel, Preda Mihail Irinel
Teachers Coordinators: Martin Elena Liliana, Dinu Alina, Raicu Viorica

Abstract

The application represent an entertainment software portable and interactive, implemented with UNITY, designed for ANDROID platform. It can be used by anybody within recreational scope, improving attention and reaction time.

The main objectives of the game are: avoiding obstacles by the player, which drives a vehicle and collecting coins with the aim of purchasing other cars. As far as the player goes, the cars' velocity increase. It allows purchasing other new cars, weapons which facilitates going forward and protective shield which allows to destroy the other cars. It will store a high score, collected coins and purchased vehicles. Also the application has two game modes: "Normal Mode", where the player has to drive the car during the day and "Night Mode", where the rewards are doubled and the player has to drive the vehicle during the night.

Unity uses Object Oriented Programming (OOP) and allows working with windows, buttons and other objects. The application becomes interactive, dynamic with personalized GUI (Graphic User Interface). It is integrated on an online edited web page (blog type). Any user can rapidly and surely download the application.

Key words: recreational, portable, dynamic, reaction time, Unity, Object Oriented Programming, Android

1 Introduction

The application is an example of utilization of UNITY's features and Object Oriented Programming.

By developing this application we identify working methods which allowed us to use PHOTOSHOP to enhance graphical aspect of the used objects within the game (vehicles). Also we use C# to manage the movement of vehicles and monitor the consumption of used coins, game rounds etc. So, we used the programming knowledge for implementing an interactive and easy-to-use game. Another challenge for making this product consisted on creating an attractive graphic design.

2 Functionality and applicability

In order to insure attractively and easy using, we integrated the program in a WEB page, online edited, making possible for any user to download the application quickly and surely. The application's frame is designed in the way that user can optimally use it.

[Download application](#)

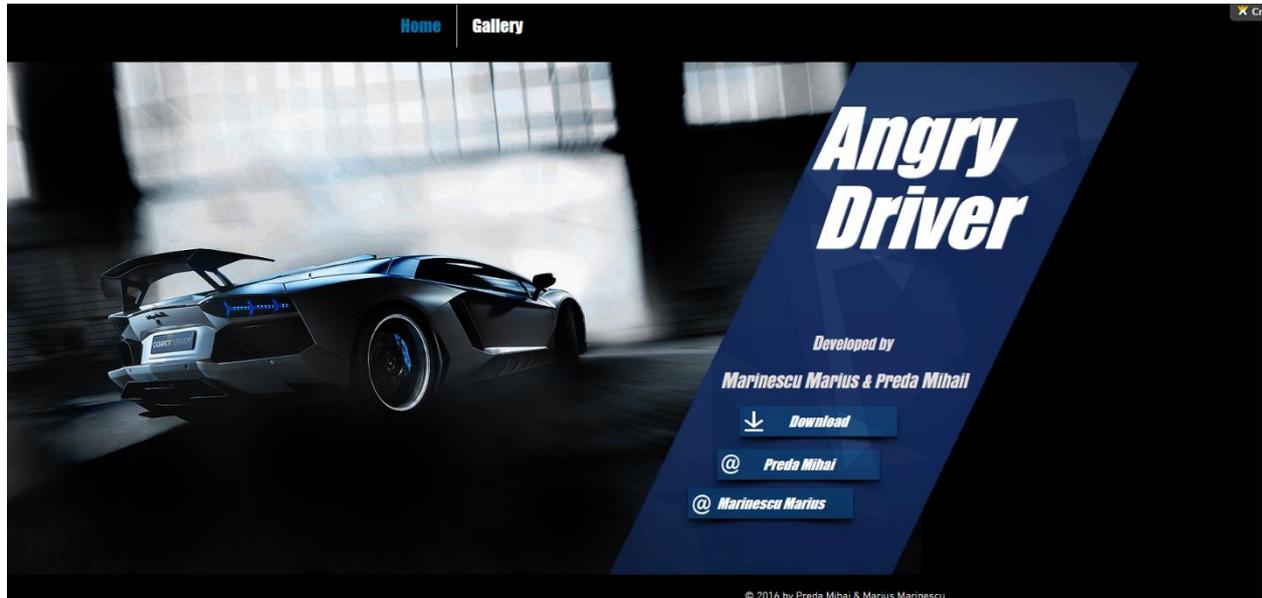


Fig. 1: The main page for downloading the application

2.1 Real-time testing

Accessing the below icon, the application opens. The image was edited in PHOTOSHOP so be as much as suggestive considering the game type.



Fig. 2: Icon

From main page, through accessing the “Information” button, a web frame which contains information about the game in both Romanian and English opens.

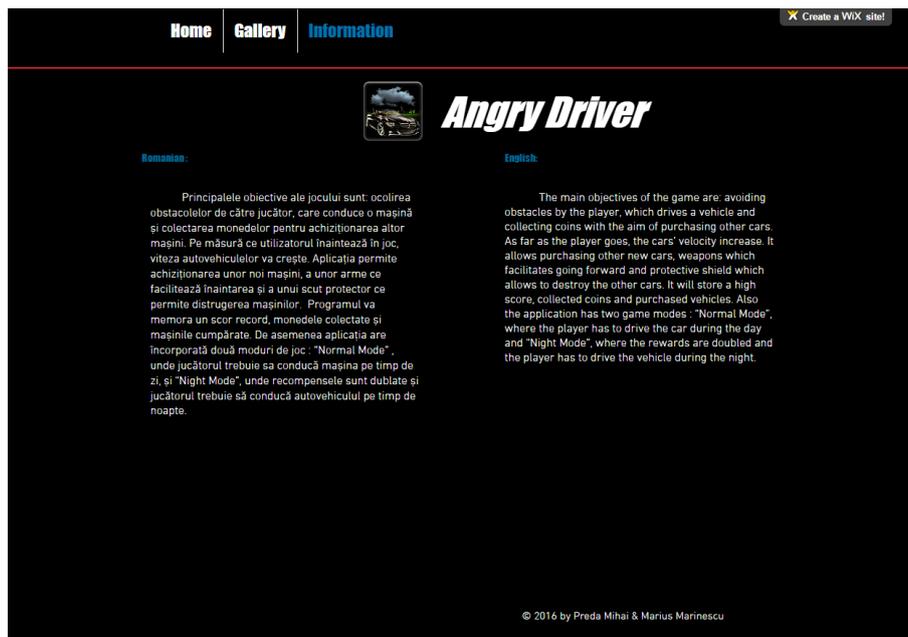


Fig.3: Information button

In order to launch the program, the user must click on the game icon.

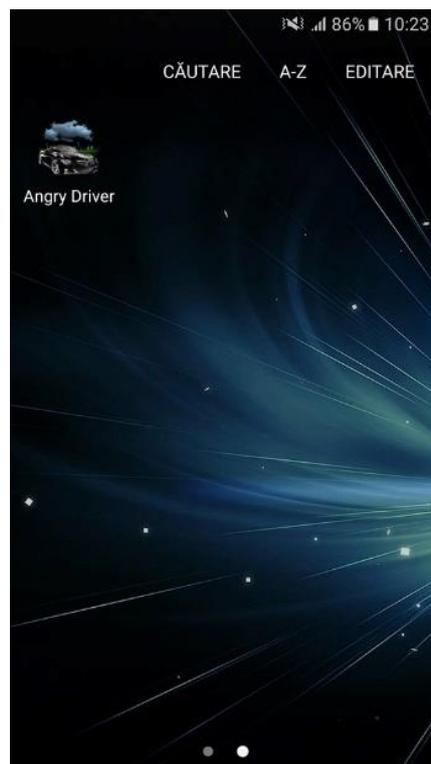


Fig.4: The icon of the installed game

This frame contains a sequence of buttons suggestive named for opening next frames.



Fig.5: Main menu

By using the “Play” button a submenu will open and will contain other buttons that serve for fulfilling the scope of the game.

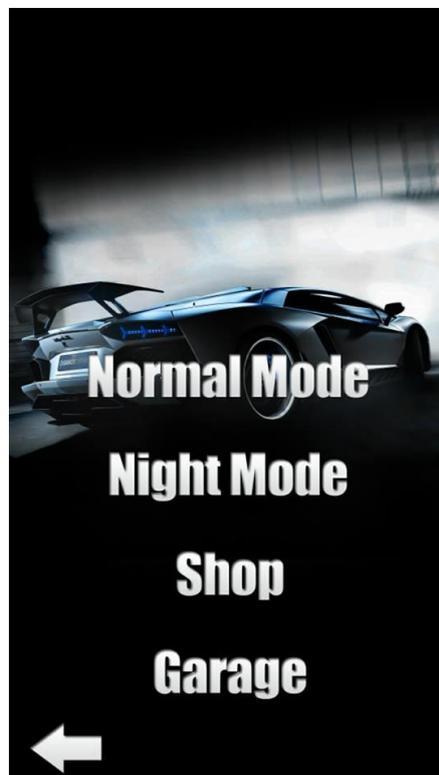


Fig.6: Submenu

By using the “Garage” button, a frame which contain owned cars and cars which can be purchased, will open. Each car will have listed the price below.



Fig.7: Garage

By using the “Shop” button, a frame which contain the upgrades, having attached the acquisition price.



Fig.8: Shop

Exemplification for “Normal Mode” button:

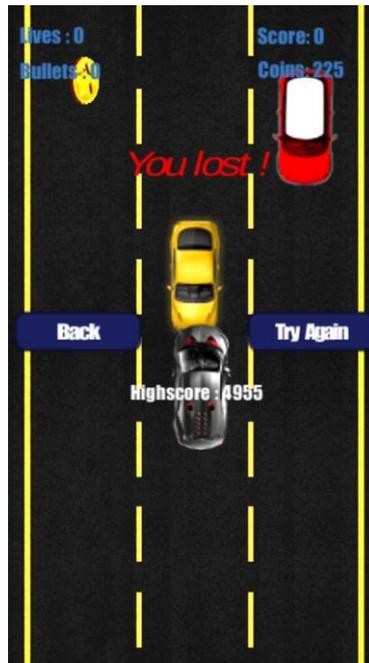


Fig.9: Gameplay “Normal Mode”

Exemplification for “Night Mode” button :

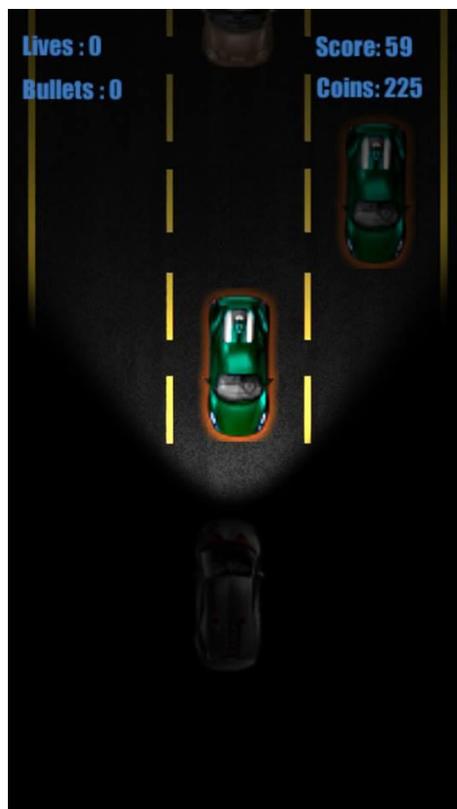


Fig.10: Gameplay “Night Mode”

2.2 Requirements and specifications

The application is reliable and portable and for a good usage of the software it requires that the phone have installed minimal version of Android: Android 2.3.1 “Gingerbread”. The application has approximately 50 MB and can be played in Portrait mode.

2.3 Source code fragments:

C# source code for CarSpawner function:

```
if (carshowed > 100) {
    EnemyCarMove.speedcm -=carshowed/300;
    TrackMove.speedtr +=carshowed/300;
    densitatemasini = 450;
    delaytimer -= 0.6f;
    if(delaytimer<0.30f)
        delaytimer = 0.30f;
}

if (carshowed > 101) {
    EnemyCarMove.speedcm = -20.5f;
    TrackMove.speedtr = 0.9f;
    densitatemasini = 470;
    delaytimer -= 0.6f;
    if(delaytimer<0.25f)
        delaytimer = 0.25f;
}

if (timer < 0) {
    Vector3 car1 = new Vector3(41 , Random.Range(1500,1800),0);
    Vector3 car2 = new Vector3(41 , Random.Range(1500,1800),0);
    Vector3 car3 = new Vector3(220 , Random.Range(1500,1800),0);
    Vector3 car4 = new Vector3(220 , Random.Range(1500,1800),0);
    Vector3 car5 = new Vector3(401 , Random.Range(1500,1800),0);
    Vector3 car6 = new Vector3(401 , Random.Range(1500,1800),0);
    Vector3 coin1=new Vector3(41 , Random.Range(1500,1800),0);
    Vector3 coin2=new Vector3(220 , Random.Range(1500,1800),0);
    Vector3 coin3=new Vector3(401 , Random.Range(1500,1800),0);

    if (asd == 1) {
        if (ok < 2) {
            if (Mathf.Abs (car1.y -car2.y) < 190)
            { //verificare banda 1 ciocniri
                if (car1.y > car2.y) {

                    Instantiate (cars [carno], car1, transform.rotation);
                    y1 = car1.y;

                    //car1.x = car1.x + 210;
                } else {
                    Instantiate (cars [carno], car2, transform.rotation);
                    //car2.x = car2.x + 210;
                    y1 = car2.y;
                }
                ok++;
                carshowed++;
            }
        }
    }
}
```

```
    } else {  
        if(Mathf.Abs (y2 -  
y1) < densitatemasini || Mathf.Abs (y1 - y3) < densitatemasini)  
            car1.y = car1.y + 250;  
        if(Mathf.Abs (y1 - coin1.y)<210){  
            coin1.y = coin1.y+230;  
        }  
    }
```

3 Conclusion

This application is relatively simple game for all ages, having a relaxing mean. In future the software will benefit of updates on the car models, a database for keeping the scores and a user manual. The game can be now downloaded via “Google Play”

References

- [1] John P. Doran, *Unity Game Development Blueprints*, Editura Packt Publishing ,ISBN 978-1-78355-365-5
- [2] Venita Pereira , *Learning Unity 2D Game Development by Example* , Editura Packt Publishing, ISBN: 978-1-78355-904-6
- [3] UNITY Documentation - <https://unity3d.com/learn/tutorials>

Marinescu Marius Daniel
B.P.Hasdeu National College
Multimedia Systems
Buzău , România
E-mail: yow_dany@yahoo.com

Preda Mihail Irinel
B.P.Hasdeu National College
Multimedia Systems
Buzău , România
E-mail: mihaipreda1997@gmail.com

Test Generator Tool for Embedded Testing

Fineas - Iacob Mureşan
Coordinator Professor: Dana Simian

Abstract

This paper was put together in the Continental Automotive Systems Company, ADAS department (Advanced Drive Assistance Systems). The work object of ADAS department is represented by Short Range Lidar, a system which helps to avoid or minimizes the accidents at a low speed. At speeds below 30 km/h, the sensor which is positioned above the windscreen, scans the area in front of the vehicle, about 7.6m looking for possible obstacles. If the sensor detects an obstacle, a speed reduction or if a vehicle in front of it is in a stationary position, it will indicate that a crash is imminent. In this case, the brakes will be recharged without the intentionality of the driver. If the driver remains passive (no reaction to avoid or to brake), the machine uses the brake automatically and reduces the torque engine. At a relatively lower speed, namely 15 km/h, the automatic breaks help the driver to entirely avoid the crash if an obstacle is ahead. If the speed between two vehicles is between 30 and 50 km/h the impact is expected to decrease.

The purpose of the application is to eliminate CANoe¹ from the creation process of the integration tests and to have a special tool. This tool should offer the possibility to build tests, run them individually, and select multiple sequential tests to run. It should also generate a test report, text-format, containing real-time information about the performance of the tests.

Currently, there are two ways to create integration tests. One way is a tool by Vector, called CANoe, where the tests are designed for the validation. This tool is used also in the integration process. Testing is performed from actual editing (writing code) up to their running. Another way to create integration tests is simply writing some command lines in a text editor (Notepad, Notepad ++, etc.), then saving with the extension *.cmm* as a file and finally sending it as a parameter to Trace32, from a batch file. Trace32 is an executable that debugs the code on the sensor.

The main disadvantage for CANoe is the high price of the license for its use. Building this tool leads to saving a considerable amount of money, a sum that can be used for other purposes of the company. Furthermore, if writing tests would be addressed as in the second case, namely, writing commands in a text editor, this would lead to hard work favoring also the appearance of many errors, because the syntax is very strict. Another disadvantage of the existing testing represents the lack of possibility to select more tests for successive running without human intervention. In both cases, there are some drawbacks which have led to the need to build a new tool that can be affordable and build tests with ease.

¹ CANoe is a tool offered by Vector dedicated to the validation software tests, not integration tests

1 The context description

1.1 Advantages/ disadvantages of the project

In the beginning it took a fairly extensive research work in order to familiarize with Lauterbach syntax, with the commands from *Trace32* and with their application mode. *Trace32* offers the possibility of writing orders in command line, manually, and in text files with the extension *.cmm* that can be sent as a parameter to *Trace32* [2]. The biggest difficulty is given by the very strict syntax that designed these commands. A simple space - blank - which in other programming languages would not raise any problem, here can lead to errors and therefore unabling to complete the proposed tests.

For a closer understanding of the testing process, environment and context in which it occurs we present the following elements involved in it.

For a closer understanding of the testing process, its environment and the context in which it occurs, we present in Tabke 1 the main involved elements:

Power supply	The testing environment is assured by a power source that supplies 12 V to the sensor or to other voltages depending on the requirements of the tests.
Lauterbach	Together with <i>Trace32</i> , the interface that enables the communication with it, is the main tool in the testing process. It is connected to the PC via an USB port having its own power supply.
Sensor	In its memory, there is the software program that needs to be tested for various manufacturers, in this case, <i>Mitsubishi</i> . It is connected to the power source and the Lauterbach.
PC	It has the <i>Trace32</i> application installed, together with the afferent drivers. Moreover, the PC runs the <i>TestGenerator.exe</i> application.

Table 1. The devices for the working environment

1.2 Definition of an integration test

Integration tests are defined by a Project Manager (PM). They are composed by certain requirements in chronological order that must follow certain steps.

Every step is numbered, starting from 1 and increasing during the test. Moreover, there are also mentioned the specific results expected for each step after running the test. Thus, the test will be considered as *passed* or *failed* based upon the matching results obtained from the test run and the results specified by the PM.

The test aims to verify the functionality of the software situated on the braking sensor and its behaviour under certain conditions. Therefore, the test consists of specific debugging commands, specific to embedded testing. The most common commands are: setting a breakpoint, deleting a breakpoint, specific for orders of *go / run* or the *setup* and *read* commands of the value of a variable. For example, the *WRAP_v_ApplUpdatedBuffer* test consists of 9 steps placed in chronological order and it tests if *Trace32* stops in the 2 breakpoints in alternative mode while the value of the variable *ui8_Status_Private_Can* changes from the value of 0x00 to 0x04 and later returning to the value 0x00.

1.3 Identification of project

The project consists of two major parts, both referring to implementation. The first part consists of files with the extension *.cmm*, imitating the role of the functions that can be called upon in a specific

moment and the processing of data handling such files and other data. The second part of the project consists of the application built in C #.

2 General view about the project functionality

For a better understand how to conduct a test, further correlations between different parts of the project will be presented. In order to create a test in a specific language for *Lauterbach* and writing the test actually, all it takes is a simple text editor (e.g. Notepad, Notepad ++) in which specific commands are written for Lauterbach's, strung one after another, as the client specifies. When finishing the editing, the file is saved with the extension *.cmm* (e.g. WRAP_v_ApplUpdateDevBuffer.cmm) and it is sent as a parameter to Trace32. Sending the test as a parameter can be done as a *.bat* file. Trace32.exe is the application that enables communication between the computer and Lauterbach. Once the test file is send as a parameter, the commands will read the file one by one and will be sent by Trace32 to Lauterbach to be executed.

2.1 Automation part

Because of strict syntax and because usually there are used specific set of commands for testing, we automated the process of writing tests, precisely for not needing to write them in a trivial text editor. A command may extend over multiple lines in a file, so I realized automation by writing each command in separate files with the extension *.cmm*. Each file bears a suggestive name, according to the command contained therein. The test is another file with *.cmm* extension , calling on separate lines. Such command files are executed through the syntax *Do nameFile.cmm*. Calling the files is not made by hand, as mentioned earlier, but by using the TestGenerator. Some of these command files need to transmit and receive parameters. For the values of these parameters, it is sometimes necessary to make certain data processing, which is now done automatically by using the implemented classes. After selecting these files with the application's help, we can run both test and save it.

2.2 CMM part

The *.cmm* files as functions that can be called upon, have been developed in order to avoid writing them each time. These files have a specific functionality suggested by their names. In those files, most used and common commands were synthesized for creating integration tests. Once implemented, they can be called directly from the GUI. In turn, in the form of call files, thus composing a table with two columns that will ultimately result in a test file.

Each file *.cmm* simulates a specific functionality and its implementation is made in Lauterbach language. The file name is also the call syntax of the GUI. Moreover, the filename is also suggestive, indicating the content of the file. Each file contains instructions to write in the report resulting data and commands to verify if the step is Passed or Failed. The files have to be in the same directory, with the application built in C # - TestGenerator.exe. In the GUI the title of the desired command will be typed by the user / tester and in the test file a call will be written to that file that corresponds to the respective order. At the same time, the desired parameters are also sent. When Trace32 receives as parameter a *.cmm* file, it takes the contents of this file and executes it line by line. If while browsing, it meets another *.cmm* file, it proceeds with his execution (line by line) and finally, it resumes the execution of the previous file. For debugging mode, it is possible the execution of files in step-by-step, step-over or go-to-end mode. All cmm files are presented in *Fig. 1*.

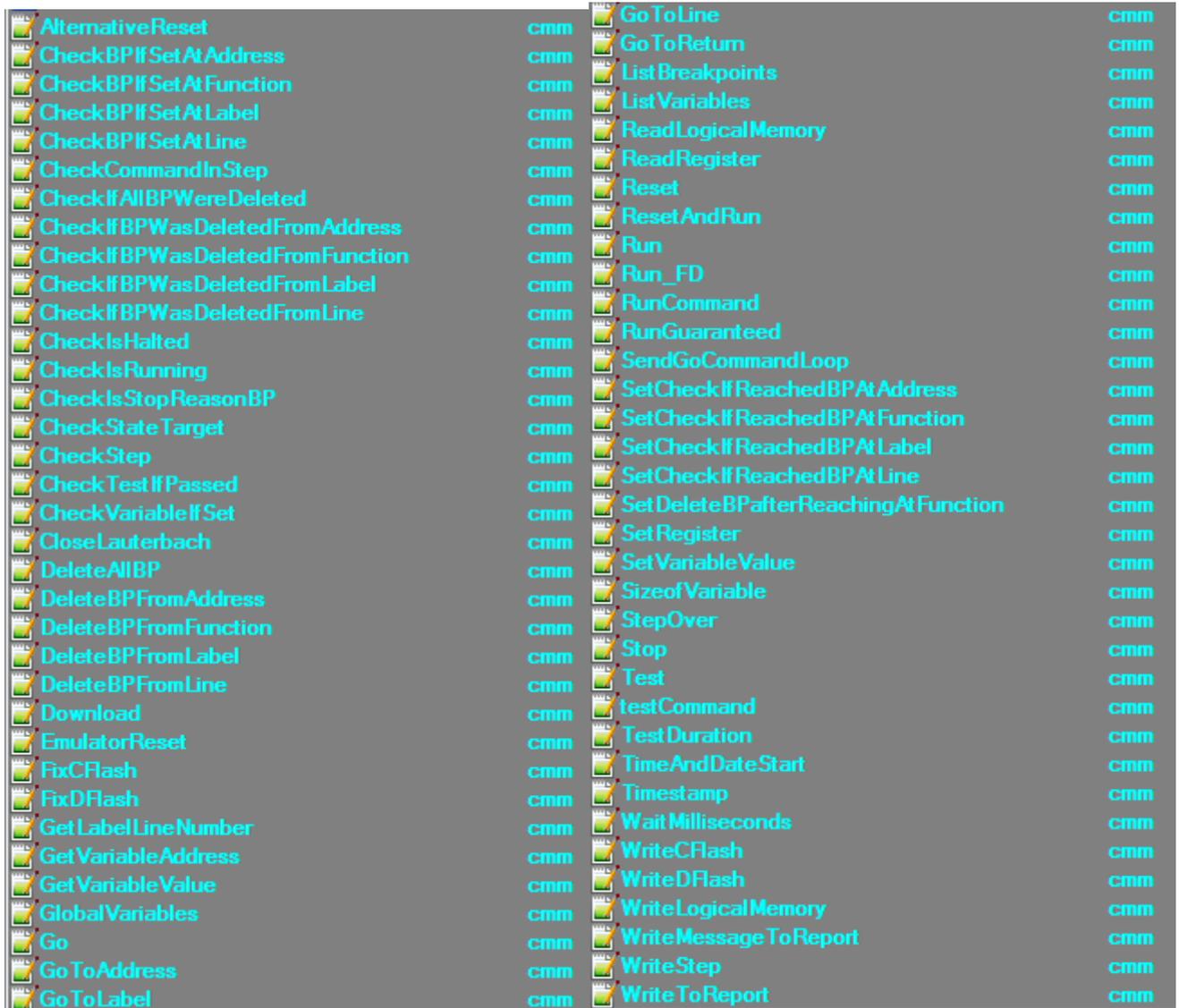


Fig. 1: CMM Files

3 C# Project

The second part of the project consists of the application built in C #, the Microsoft Visual Studio. This application is of Windows Forms application type. In the next section, it will be presented the application data processing, the graphical interface of the application and its functionality.

3.1 Functionality

For handling all .cmm files presented in the first part and even for creating some of them that are to be presented, it was necessary to build an application having a graphical interface. Graphical interface makes testing much easier being built by an inexperienced person in the language of Lauterbach. As follows, using this application, the tester has only to select the files with which the test is built in a window and in the end to save the test or even run it. A saved test can be opened anytime to run or to be edited with the help of graphic interface.

3.2 Windows

3.2.1 TestGenerator – Main

Basically, in order to create a test, the application TestGenerator.exe is launched and the main window – TestGenerator – Main is opened (Fig. 2). From here the navigator goes to the desired folder and presses the button Add New.

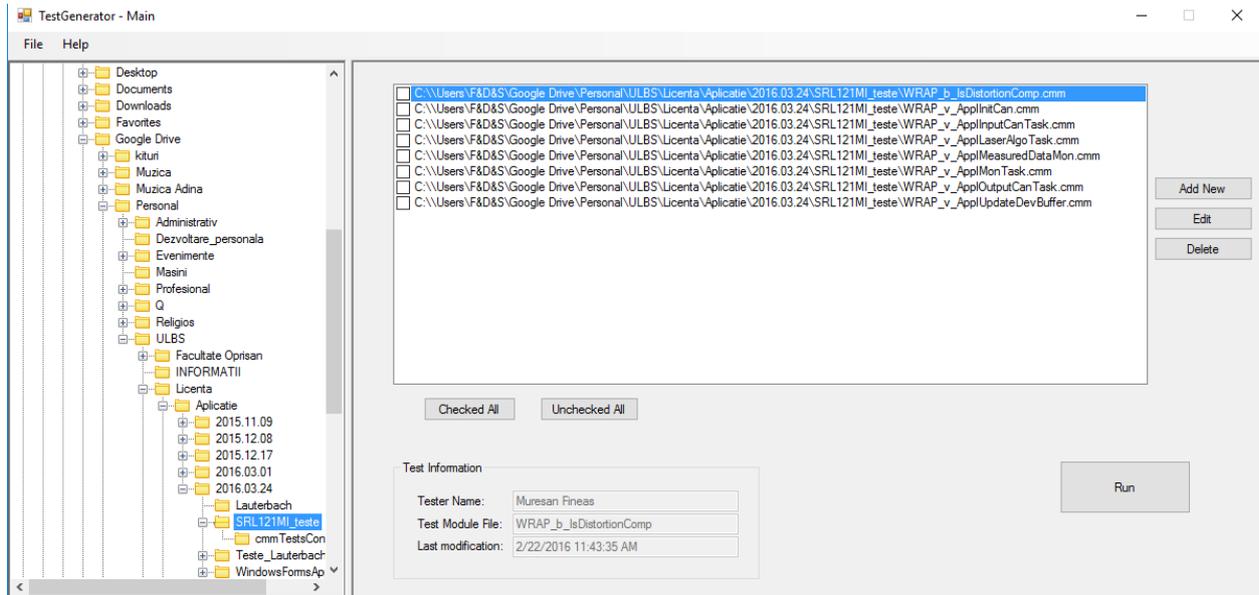


Fig.2: Test Generator – Main Window

The button Add New refers to the creation of a new test. By pressing it, a new window called Make/Edit Test will open. When the button Edit it is pressed, it will open the window Make/Edit Test but this time in an edit mode test. Basically, the selected test will open in CheckedListBox for editing. When the window Make/Edit Test is opened by one of the buttons Add New or Edit, it will also have consequences on the window Test Generator – Main; this window will change to the mode *hide*, until the Make/Edit Test will be closed. When one or more tests are checked, with the button Delete the tests will be erased. While erasing it, a dialog box pops up with a warning for the user to confirm the deletion. The functionality of the button Run is to run Trace32, giving it as parameter the start.cmm file which is formatted each time you press the Run button. Start.cmm file will contain as many lines as files were selected. On each line there is a command *Do*, followed by one of the tests. Sending the Start.cmm file as a parameter to Trace32, the tests given on each line will be sequentially executed.

3.2.2 Make/Edit Test

Make/Edit Test window (Fig. 3) can be opened by pressing Add New or Edit button. In the first case, the window will be open for creating a new test; in the second case, for editing a previously created test. The window consists of several labels, text fields, buttons radio, and a ListBox [1]. However, the main role is played by a DataGridView with two columns: Commands and Parameters. The first column the command that build the test are introduced, and in the second one are introduced specific parameters for each command. Each command in the first column suggests the number and type of parameters which must be written in the second column. These must be separated by a comma.

Each cell from the first column is the ListBox type with AutoComplete, which means that when launching an editing event of a cell in the first column, the file contents of the Commands.txt file will be loaded in a variable of AutoCompleteStringCollection [3] type. This variable stores a string data type, with an auto-completion feature. That is, with every new character typed into that specific cell, it

will make a filtering of the data in this collection and display only the data corresponding to the filtering. If there are no suitable data, it will not show any. In the case that there are multiple valid results, you can navigate with the keyboard arrows to the desired result, or directly select with the mouse the desired option.

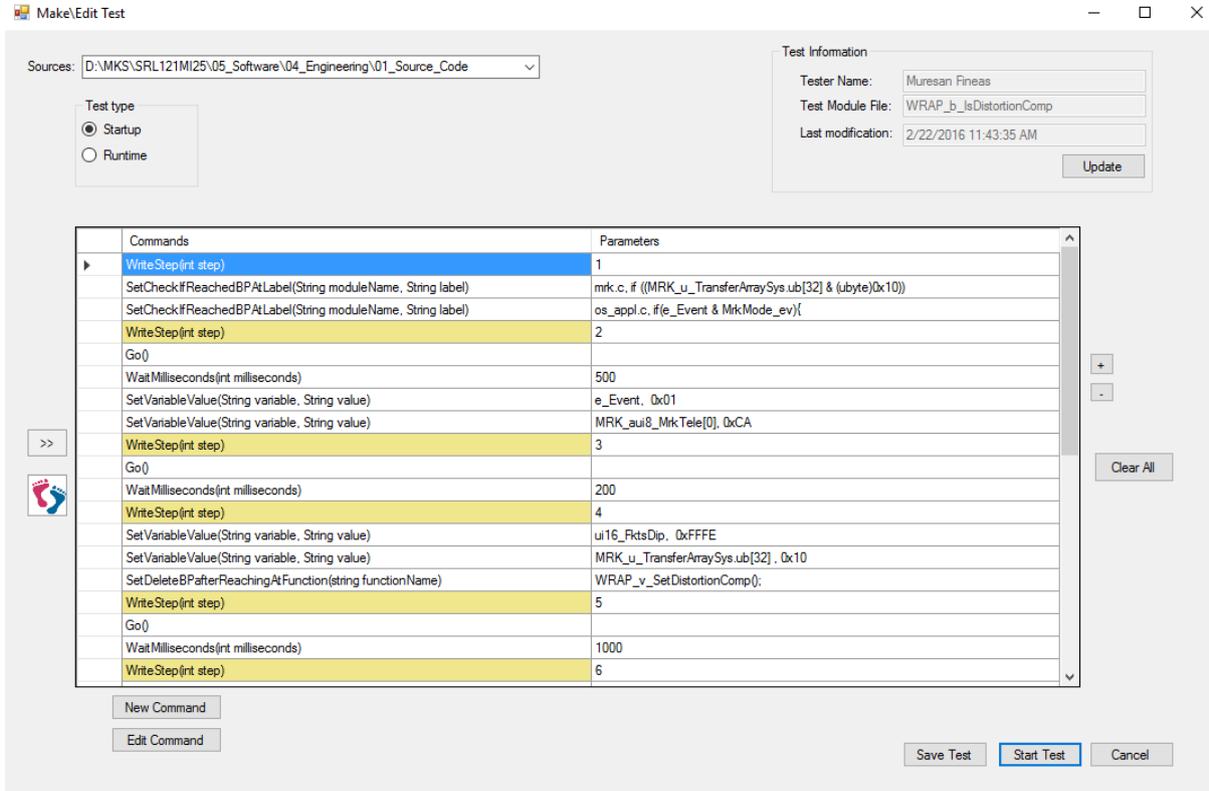


Fig.3: Make/Edit Test Window

3.3 Classes

Classes used for the creation of the application are divided into two categories. In the first category are included classes designed to treat incoming events from the graphical interface and create some special features for the graphic interface. In the second category are classes of data processing. In the following will be presented some of the most representative examples of classes (Fig. 4).

Classes used to treat graphical interface events are: MakeEditCommand, MakeEditTest, TestGeneratorMain, TreeView, SaveTestInfo and TestInfo.

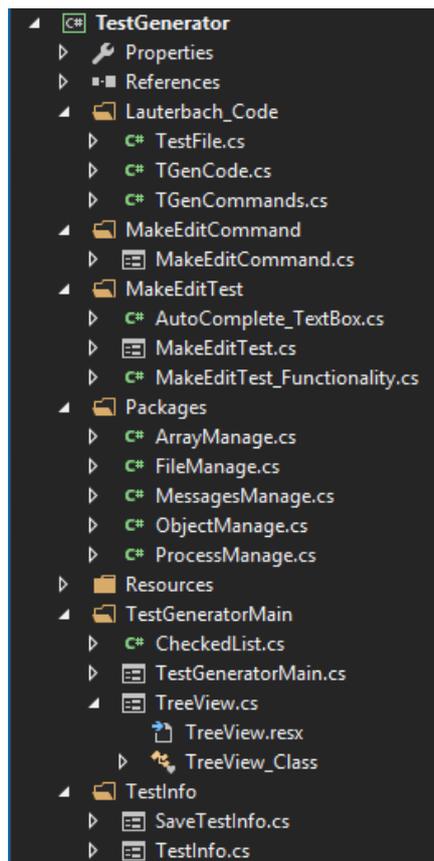


Fig.4: C# Classes

3.3.1 TestGeneratorMain

The class which stands behind Test Generator - Main window and provides treatment of events that happen in this window is TestGeneratorMain.

However, before the window is displayed at the time of its creation certain items will be initialized, precisely because while being displayed, it contains certain necessary information. This information relates to the structure of files located on the hard disk and navigating through the expanding tree of files, towards the node stored in file TreeView_XML_TestGenerator.xml. Data will be displayed in the panel on the left side of the window Test Generator – Main, with the help of TreeView. Another class that will be instantiated is CheckedList. As previously mentioned, the panel on the left will scroll automatically when creating the window, towards the node previously stored. After it will be found and expanded, it will also be selected. When selected, its content will be displayed in the second panel on the right side of the window. This requires an object of class CheckedList. Another instantiated object here is the class TGenCode. This class contains various functions that will be called upon when saving a created test in the window Make/Edit Test. These are all functions of translation of the data entered by the keyboard; the result is formatted in the specific language of Lauterbach.

3.3.2 MakeEditTest

The class that is designed to manage events creating or editing a test uses a lot of important functions. Two example presented here are SaveTest() și StartTest().

SaveTest() will be called upon by pressing SaveTest button in the window of Make/Edit Test. Firstly, it is checked whether the information was introduced in the Test Info window and also if all fields are completed. Otherwise, for both checks a warning message it will be display. Then, the steps of the test will be automatically reviewed to prevent logical errors, using the function UpdateSteps(). The proper

function of saving the test is called, and if in the window TestGenerator – Main no test is selected, the first test will be automatically selected from the list.

Classes included in the second category, namely, those processing data are: TestFile, TgenCode, TGenCommands, MakeTest_Functionality and CheckedList. In the following, the most representative ones will be explored (Fig. 5).

3.3.3 TestFile

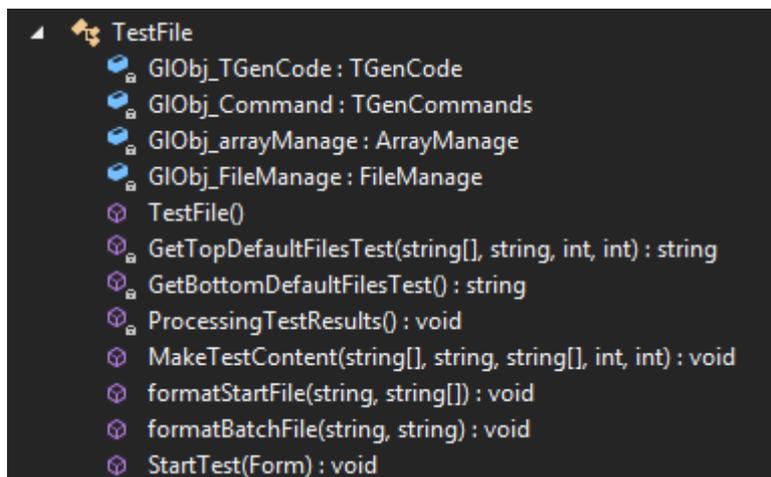


Fig.5: TestFile Class

The TestFile class deals with formatting the text file and other files such as start_debugger.bat or start.cmm. The files are formatted by using functions MakeTestContent, formatStartFile or formatBatchFile. The class constructor initializes some items that will be needed in class functionality. The TestFile class contains certain functions and if called upon with corresponding values, just as parameters, will 'arrange' the contents of these files. For example, to create a test, there are some files that are included in each test automatically, at the beginning and at the end of the test.

The commands contained in these files are actually some general preconditions. If called upon, they will create the necessary environment for testing. In fact, the GetTopDefaultFilesTest function will store in a string all titles of files that will be inserted in the beginning of the test. In the test file, for the continuation of these files, the data introduced by the tester will be insert in the window Make / Edit Test, and in the end, a new set of files will be automatically inserted by using the tool GetBottomDefaultFilesTest.

4 Conclusions and Improvements

This project presents an original solution regarding a new way to generate integration tests.

The project was realized by building a common set of commands specific to Lauterbach's, structured in files and through a GUI application built in C #.

The presented solution is used successfully within our team, fulfilling all the initial requirements: excudes the CANoe's tool from testing process, gives the possibility to select several tests for execution and to generate a report for the test. The benefits of the presented solution are: saving of time, costs and human resources.

The ultimate focus of the paper represents the project in Microsoft Visual Studio, a Windows Forms project type.

Other possible improvements for the future could be:

- DataGridView lines (window Make / EditText) able to allow the multiple selections and able to be arranged in a different order by dragging with the mouse.
- Automatic change of date when editing a text.
- Development of the application so that it will more easily configurable for different computing machines.
- Total independence from the testing project.

References

1. Valer Roşca, Daniel Hunyadi, "Modern programming languages: C # - edition. revised and enlarged " (in Romanian language), Editura Universităţii „Lucian Blaga” Sibiu, 2013.
2. Training Manuals - Lauterbach, 2016 - <http://www.lauterbach.com/frames.html?training.html>.
3. DataGridView Autocomplete TextBox, 2016 - <http://csharp.net-informations.com/datagridview/autogridview.htm>

MUREŞAN FINEAS-IACOB
Lucian Blaga University from Sibiu
The Faculty of Science
Dr.I.Ratiu Street, 5-7 No, 55001
Romania
E-mail: fineasmuresan.ro@gmail.com

StarDust

Oltean Alexandru, Prăvariu Alin Gabriel
Teacher Coordinator: Popescu Carmen, Preda Georgeta

Abstract

Since this year we celebrate Tomohiro Nishikado's^[1] 72 anniversary at the end of the month, 31st of March to be more precise, our project will be based on one of his games, Space Invaders, but with a tint of new. Our game will keep the original 8bit theme but with brighter colours, small added details and attractive effects suitable for such a renowned game from 1978, remade after nearly 40 years from its release. It's recreated with the game engine Unity and scripts from C#.

1 Introduction

The game titled StarDust was created using the Unity platform [2], a game engine, which also allows the use of scripts in C# [4]. Items belonging to this game were taken from the standard assets for Unity, both spaceships and asteroids. The special effects were created by us using the particle system and the lighting options that Unity offers. Each element has one or more scripts attached to enable it to perform certain movements or functions. For example, the spaceship has attached a script that allows it to move and fire bullets. The technology used is up-to-date, being used by larger companies to create many popular games such as Cities: Skylines, Rust or Firewatch. Some of the original elements of our game are the user-friendly interface, in-game music and special effects. Compared to other applications of its kind, our app keeps the nostalgic character of games from the 80s, but at the same time it brings modern and innovative elements. We will give a quick explanation of the game's universe, the steps in its creation and difficulties encountered during programming and its configuration. We will offer a short conclusion in Section 6. *Conclusion*, where we talk about possible changes in the future in terms of what platforms you will be able to play StarDust on, requirements and complexity of the game you will play.

2 Brief initiation to the game

StarDust is an arcade shooter that creates a dangerous environment for the player, the captain of the spaceship, who is on a dangerous mission through the vast and unexplored space. It is at risk of being hit by asteroids and shot by enemy spacecrafts. The player moves using the keyboard arrows or WASD keys, CTRL to fire the guns. Ship's weapons are used to destroy asteroids coming towards the player's ship or just try avoiding them while they are in a constant battle with

the enemy. The game is based on a scoring system that is updated after destroying an asteroid or enemy ship.

3 Creation stages

The project began with the creation of player controlled ship (see Fig. 5). Then the camera that follows the player ship and light effects were added. A background is attached (see Fig. 6) and player movement scripts. It followed with the creation of lasers and their script. The next step was to create enemies and asteroids, special effects owned by all the enemy ships and asteroids, followed by the script that controls the entire game. The project was completed with the addition of the audio and scoring system. Then came the game compilation.

4 Explanatory figures

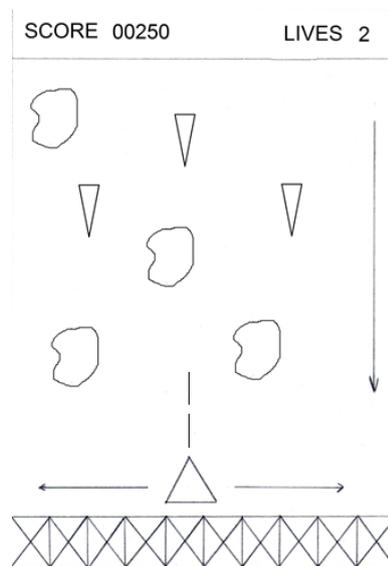


Fig. 1: In-game interface (simplified)

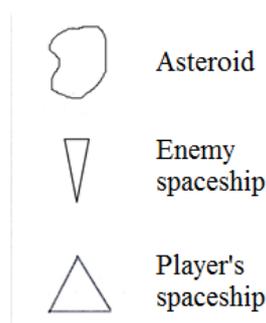


Fig. 2: Moving Objects



Fig. 3: The area protected from the enemy ships



Fig. 4: Status bar



Fig. 5: Player's spaceship model

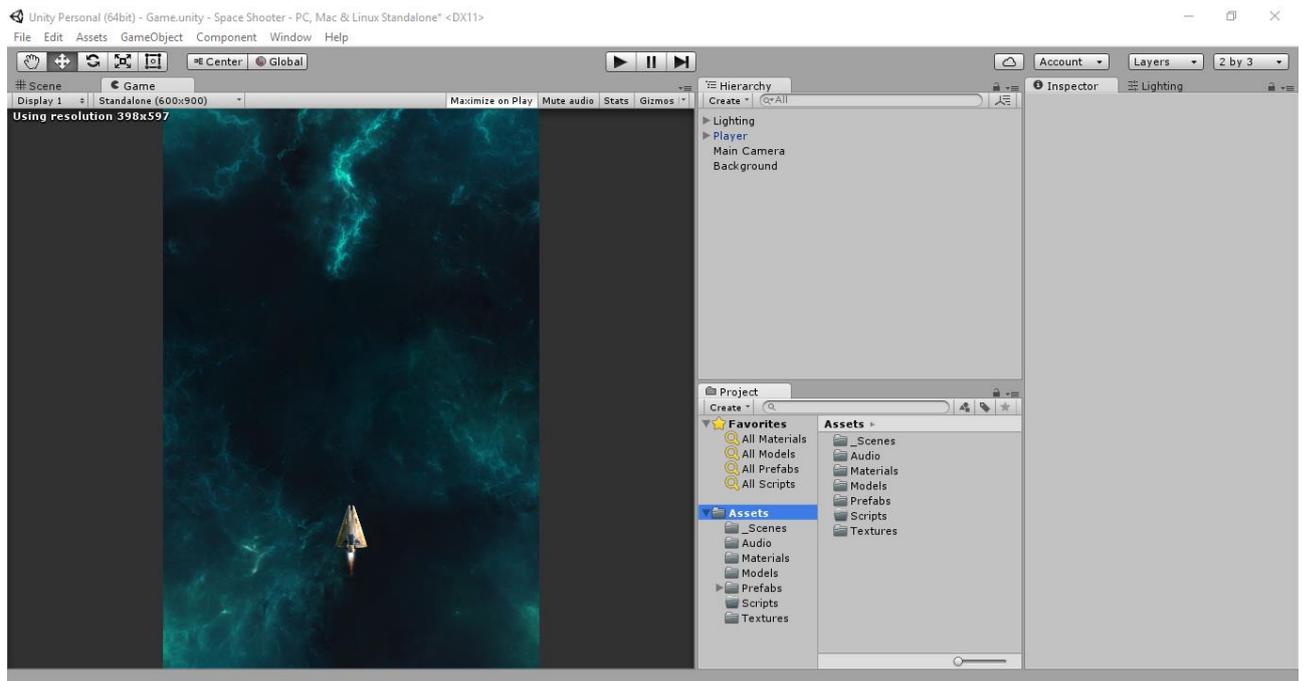


Fig. 6: In-game image of the player's spaceship on the chosen background

In Fig. 6 we can observe the later stage of the game development, having ideas from Fig. 1 and the spaceship model seen in Fig. 5.

5 Difficulties encountered

Difficulties occurred while adding special effects, start menu creation, making the enemies appear in waves and the script that controls the game.

6 Conclusion

In the future, our game could be played on other platforms such as Android, iOS, PlayStation, web browser or even the platform that Space Invaders[3] was originally released on, the Atari 2600. Another platform could be the Oculus Rift, platform that would radically change the gameplay and even in-game user experience. Of course, this game can be turned into a business by adding advertisements, but we do not want that at the moment.

Although the idea of the game itself does not simply want to test player's reflexes, we plan to add other requirements of the game, such as testing player's leadership abilities. In the near future we are seeking to add a new mode that allows the player to control other spacecrafts at his disposal.

References

- [1] About Tomohiro Nishikado https://en.wikipedia.org/wiki/Tomohiro_Nishikado
- [2] Unity official site - <https://unity3d.com/>
- [3] Information about Space Invaders - https://en.wikipedia.org/wiki/Space_Invaders
- [4] Information about C# - <https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>

Oltean Alexandru
Colegiul Național "Gheorghe Lazăr" Sibiu
Matematică-informatică intensiv informatică
Sibiu, Sibiu, România
E-mail: alexandru.oltean98@yahoo.com

Prăvariu Alin Gabriel
Colegiul Național "Gheorghe Lazăr" Sibiu
Matematică-informatică intensiv informatică
Sibiu, Sibiu, România
E-mail: alin_gabriel_pravariu@yahoo.ro

Roev - Event Planner

Theodor-Ovidiu PALAMARU
Coordinator Professor: Dana Simian

Abstract

The purpose of this article is to present a web application used for events management. This application is developed with Liferay and it's designed mainly for companies and enterprises, but not exclusively. With an intuitive and friendly interface and with the help of reliable and modern technologies, it is a perfect tool for day to day meetings and events planning. Further in this article there are different sections which describe the application's components and how they work. There are also different screenshots from the application and a couple of code snippets included.

1. Introduction

There are many types of organizations; be them related to work, school, volunteering and so on. In these groups, as with any other, there will come a time when a decision must be made regarding the different aspects of the way the members will meet: when, how, for how long, who should come, how many people can join, where are they going to meet? All these questions, even though they may sound simple and maybe at first not too important, are in fact a decisive factor in the group's success.

Sure, there are already many ways to contact friends, family, associates or whatever the members are called; such as e-mail addresses, social websites, different chat applications and almost unlimited amount of other identical pieces of software that keep you in touch with the world. It's a golden age of technology after all and the possibilities are endless. And yes, it is true that some of these technologies offer a decent form of organization for a group of people. But most of them have some limitations, since the concept of strictly managing a meeting (for example) is not the project's main focus. These limitations also include and are not limited to: lack of a user friendly UI, too few options, content loading too slow, advertisement spam, profit oriented, unfixed bugs and so on.

One of the main ideas behind *Roev - Event Planner* is to break those limits and offer a better and easier way towards creating, organizing and managing a specific event or series of events for a group, community or other form of social environment that brings people together. The project being developed with Liferay, makes all this become possible. The Liferay Portal isn't just another platform from which you can run a random number of applications. It also offers a structure of groups and roles that help define in detail different aspects of this application, such as: which users can view an event, who can join or create events, how and where the events are displayed and so on.

Next, in this article you can read more about different elements of the project, look at few snippets of code and learn about future developments.

2. Application description

Roev - Event Planner is a web application used for the single purpose of creating and managing events. It is built on a platform called Liferay, which puts the concept of this project in an ideal position from which it can reach its maximum potential.

Liferay is an open source enterprise portal, written in Java. It includes a build-in web content management system and a wide range of features used for website and portal development. In 2007 it was declared "Best Open Source Portal" by InfoWorld. Although it offers a very advanced and sophisticated programming interface for developers, basic website administration does not require any programming skills at all. The Portal is Java-based and runs on any computing platform capable of running JRE (Java Runtime Environment) and an application server. The Liferay bundle is available with Apache Tomcat servlet.[4][5]

Liferay, among other things, has a very complex and well structured system of Groups, Communities and other User Management elements. It also comes with useful tools such as Roles, Permissions, Media Library and different other smaller frameworks and options which are quite a blessing when it comes to development. Basically, for lack of a better description, Liferay is in my opinion one of the best platforms for web application development, simply because it can do pretty much anything. Java, PHP, MySQL, PostgreSQL, HTML, CSS, JavaScript, JQuery; you name it. You can use almost anything, link it to anything and make it work. For your idea to succeed you just need imagination, and maybe a lot patience.

In this section I will mention the project structure, the UI and other elements, telling you a bit about how they work and what they are for.

2.1. Application Structure

Roev - Event Planner is made up of 3 Liferay Plugins:

- **Service Builder** Plugin that is the main link to the database and also contains the 2 Portlets of the application: the main project Portlet that displays the events and other UI elements seen by everyone, and a Configuration Portlet that can be accessed just by the Administrator in the Control Panel of the Liferay Portal. (A Portlet is a user interface component managed and displayed in a web or enterprise portal) [6]
- **Hook** Plugin that is used for the creation of the custom Roles and Permissions [1]. These elements will be created when the project is being deployed on the Liferay Portal. The hook will be deployed first.
- **Liferay Theme** that improves the visual aspect of the project's interface. For this Theme, I used Bootstrap 3, instead of the usual Bootstrap 2 that Liferay uses. [2]

2.2. Roles and Permissions

For it to be used in an efficient way, the application has a set of custom Roles and Permissions. Each of these elements has a specific functionality and it is necessary for a good management of the application and its contents.

2.2.1. Roles

- **Roev Manager** - this role is meant for the managers of a company (or whatever the person in charge of the group is called). All users with *Roev Manager* set as one of their roles can View the main Portlet and all its content, can Join events, can Add new events and also approve or decline any events that are marked as *Pending*.

- **Roev Employee** - this role is used by all the other members that are not in any leadership roles (managers). The only difference between them and the managers is that they cannot approve or decline the *Pending* events.
- **Roev Special Guest** - this role is the most limited of all 3 custom roles. The main idea behind this role is quite simple: maybe the group has different collaborators, people that are not members of the group, but they have different ties to it. Users with this role can only View the event lists and Join an event. They cannot create new events. For all intents and purposes, as their role name clearly states, they are guests.

2.2.2. Permissions

- **View Portlet** - this permission allows the users to see the contents of the main Portlet. They can see the lists of events, their own draft list and create a new event. All this, of course, according to the role set to them.
- **Join Event** - users who have this permission can Join an event marked as *Approved*.
- **Add Event** - this permission gives the users access to the *Drafts* and *New Event* tabs, and their content.
- **Approve Event** - the only ones who get this permission are the users with the role *Roev Manager* and the Portal Administrator(s) (the Administrator role is pretty much *God Mode*, so to speak, since the user(s) that have it can see and do anything related to the Liferay Portal. Thus it is only logical that they get this permission and all the other by default). This permission is necessary to approve or decline events marked as *Pending*.

2.3. Event Status

In total there are 5 possible statuses an event can have, but no event will ever reach all of them. Each event will follow a certain path, according to the way it will be managed by the users. These types are:

2.3.1. Draft Event

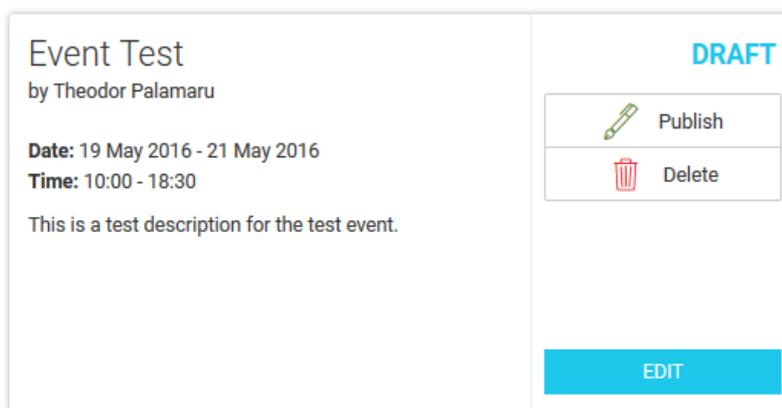


Figure 1 - Draft Event (the first status of any event)

In *Figure 1* you can see the first stage of any Event. After successfully passing the initial creation form, the owner will be redirected to the **Drafts Tab**, where he can see his Event in the list.

Every user can see only their own drafts. While in this state, the new event can be edited, in case the owner wants to change some information about it. Once it's decided that the event is ready, it can be *Published*. If the user no longer wishes to pursue the idea, then he'll have the option to *Delete*.

But once the event is published, it can no longer be deleted. The user can also simply keep it there in the drafts list, since he is the only one who can actually see it anyway.

2.3.2. Pending Event

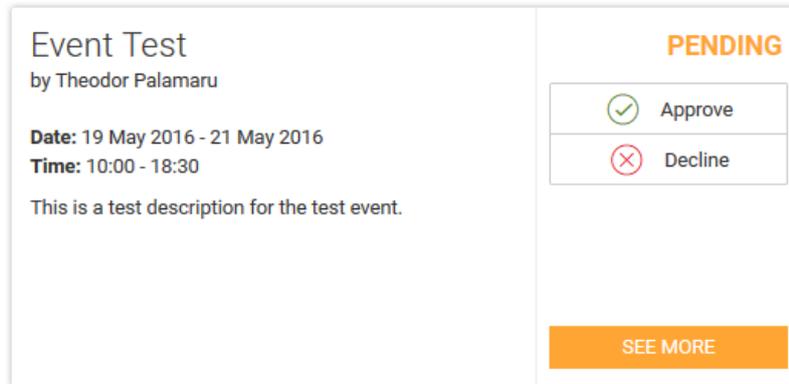


Figure 2 - Pending Event

Figure 2 shows what happens after the draft event is published. In this state, the event can be seen only by the owner and those who have the *Roev Manager* role, since they will be the ones who can approve or decline the event. This event can be seen in the **Events Tab**.

In this state, for the first time, the *See More* option will be displayed. If you click on it, the Detail View section of that event will be displayed. Here, apart from the information that can already be seen on the original event card, you can read the long description and view other details regarding the event.

2.3.3. Approved Event

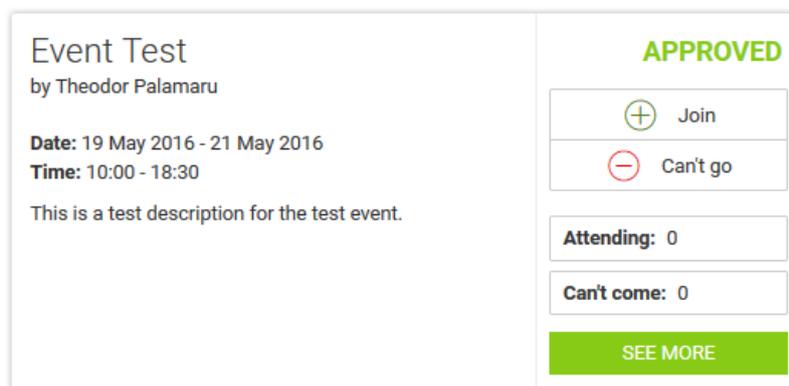


Figure 3 - Approved Event

If the event is approved, then its status will change accordingly. The cards in the list will also get a few extra elements, such as the join/can't go buttons, and the number of people that are attending and those that are not. This event can be seen in the **Events Tab** and will look like in *Figure 3*.

2.3.4. Declined Event

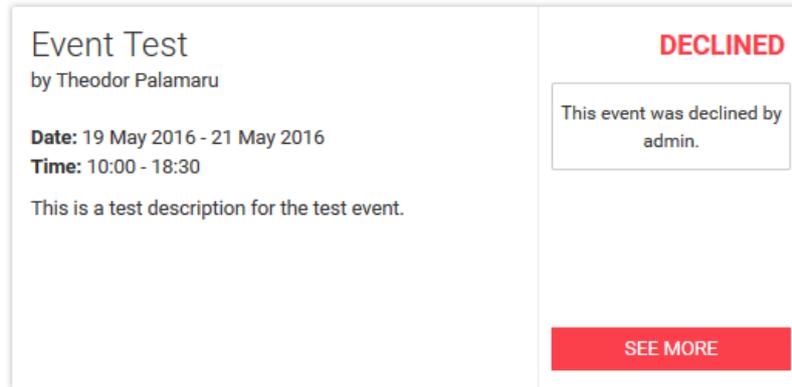


Figure 4 - Declined Event

As shown in *Figure 4*, the event can get this status if the manager has declined it. You can still check the Detail View of the event, but other than that you cannot interact with it. This type of event can be found in the **History Tab**.

2.3.5. Passed Event

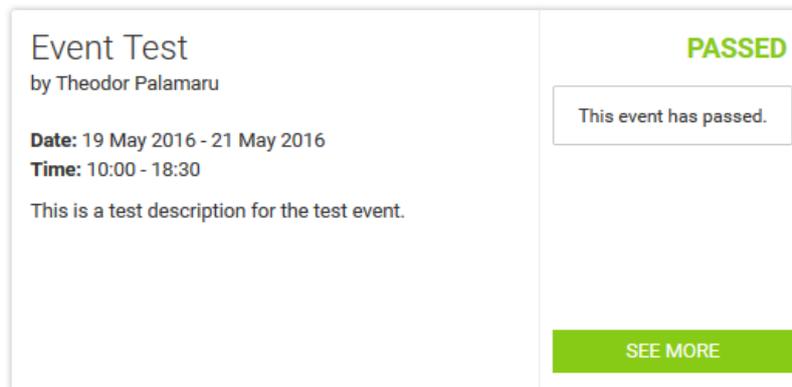


Figure 5 - Passed Event

Figure 5 illustrates the last possible status. This particular status can only be seen on an approved event that has already taken place. Basically, if the set end hour in the last day has passed, then this event will be marked as *Passed* and will no longer be seen in the Events Tab. It will be displayed in the **History Tab** instead.

2.4. Menu

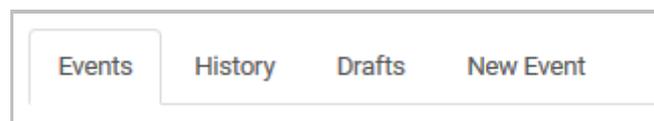


Figure 6 - Main menu of the application

The application is made up of 4 main parts (tabs) as shown in *Figure 6*. Each tab contains a specific type of content as follows:

- **Events Tab** - in this tab you can view and interact with all the *Pending* and *Approved* events, which can be seen by the appropriate users according to their specific roles and permissions.

- **History Tab** - the events displayed in this tab are those marked as *Passed* or *Declined*. This tab is nothing more than a record of events that are no longer available.
- **Drafts Tab** - every user that has either one of the roles *Roev Manager* or *Roev Employee* can see this tab. Here the user can View and Edit only his own draft events.
- **New Event Tab** - from this tab, the users that have the necessary permissions, can create a new event. At first there will be a list with different types of events (such as: Meeting, Poll, etc.) from which the user will choose the one that better suits his needs, thus giving him a better way of expressing his idea.

2.5. Navigation Tools

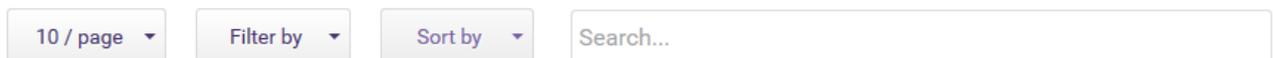


Figure 7 - Navigation Tools

The Events, History and Drafts tabs also have a few tools that can help the users navigate easier through the list of elements and find what they are looking for. As can be seen in *Figure 7*, these are:

- **Pagination** - by default this is set to 10 items per page, but it can be changed to either 6 or 16. At the bottom of the list you have the option to swap between pages.
- **Filters** - depending on what type of events are displayed in the list (this will be according to the tab you are currently on), you can filter the list to display only one type. Thus making it easier to find what you are looking for.
- **Sorting** - there are 4 sorting options: by status and by name (both ascending and descending)
- **Search** - the search bar is probably the best way to track the event(s) you are interested in. It searches by title and description.

All these tools work in real time, without any tedious page loading or extra button clicking. It's all dynamic. This way you find what you are looking for in the best and fastest way possible.

2.6. Event Form

After the user has chosen the desired type of event they wish to create (Meeting, Poll, etc), the appropriate form will appear. In here the user will fill in the necessary data regarding their new event. After all validations and requirements have been met, the data will be saved and the user will be redirected to the **Drafts Tab** where he can see his idea come to life.

3. Implementation

This application was developed mainly in Java for the backend part and JavaScript for frontend. Other technologies were also used, such as HTML, CSS, SASS, a different range of JavaScript libraries and of course specific Liferay elements for the UI. Also for the database management, I used PostgreSQL.

The Liferay Plugins were created using Maven and Maven Artifacts (for the Liferay Theme). As for the server, the initial Liferay bundle comes with Tomcat 7, therefore this was an obvious choice. For the development environment (IDE) I chose Eclipse [3], simply because it offers a wide range of tools that make the development part somewhat smoother.

Since quite a number of languages and technologies were used for the development of this application, in this part of the document I will present small code snippets of some of the more interesting parts, along with short descriptions.

3.1. Java

```
public Event updateEvent(long eventId, long mainAuthor, String name, Date startDate,
    Date endDate, String shortDescription, String longDescription, boolean isForSpecialGuest)
    throws SystemException, PortalException {

    Event event = eventPersistence.fetchByPrimaryKey(eventId);

    event.setName(name);
    event.setMainAuthor(mainAuthor);
    event.setStartDate(startDate);
    event.setEndDate(endDate);
    event.setShortDescription(shortDescription);
    event.setLongDescription(longDescription);
    event.setIsForSpecialGuest(isForSpecialGuest);

    eventPersistence.update(event);

    return event;
}
```

Figure 8 - Java method for event update

Figure 8 represents a piece of Java code used by the Service Builder to update an event. As you can see, it receives the new data in parameters. After finding the event by its id, the next step is setting the new information and then updating it.

3.2. JavaScript

```
function setTextareaLength(maxLength, containerId, numberContainerId) {
    var fieldVal = $('#' + containerId).val();
    var valueLength = fieldVal.length;
    $('#' + numberContainerId).text(maxLength - valueLength);

    $('#' + containerId).on('keyup', function() {
        $.when ( $(this).blur() ).done(function(x) {
            $(this).focus();
        });

        var fieldValue = $(this).val();
        var fieldValueLength = fieldValue.length;

        if (fieldValueLength >= maxLength) {

            var limitedValue = fieldValue.substr(0, maxLength);

            $(this).val(limitedValue);

            $('#' + numberContainerId).text(maxLength - fieldValueLength);
        } else {

            $('#' + numberContainerId).text(maxLength - fieldValueLength);
        }
    });
}
```

Figure 9 - JavaScript function used for field length count

The JavaScript function from *Figure 9* is used for calculating the text length inside the description fields found in the event form. At first it starts off with the default maximum number (200 for short description and 6000 for long description) which is already set from global variables. Afterwards, on each key press inside the *textarea* field, the script will run. It gets the current text, counts how many characters it has and subtracts from total. A few validations were added for when the current total will be bigger than the average number. When the limit has been reached, no matter how much you try to type, the script will simply take the current text and return exactly the maximum number of characters permitted.

Of course, this is not the only validation in place for the description fields. There are also 3 more validations. One of them is in the Service Builder and it allows the description field in the database to receive a certain length for its value. This particular validation is the most important, because without it the whole thing would crash. PostgreSQL sets the default field length to 75. And that is not enough. The other two validations are both in frontend. The first one is the Liferay validation and the other is a simple check that iterates through all fields in the form, the descriptions included, and checks if they meet the necessary requirements. Basically this check is more like a reinforcement for the Liferay validation and makes sure the proper error messages are displayed (if needed).

3.3. CSS/SASS

```
/* Meeting Type Event Card */
.new-meeting-event {

    .mev-details,
    .mev-add-sign {
        padding: 8px 12px;
        height: 100%;
    }

    .mev-add-sign {

        .fa-plus {
            font-size: 140px;
            width: 100%;
        }
    }
}
```

Figure 10 - CSS code used for responsive design

In *Figure 10* you can see a SASS syntax. This technology allows the nesting of CSS, which can be quite useful when you are working with many containers, especially when a grid is involved as well. This particular piece of code is being applied in a media query and it is used to improve the responsive design of the *Add Meeting Event* button (**New Event Tab**).

3.4 JSTL

```
<c:choose>
  <c:when test="${ev.eventStatus == }
    <button class="btn-custom-orc
  </c:when>
  <c:otherwise>
    <c:if test="${ev.eventStatus
    <button class="btn-custon
    </c:if>
  </c:otherwise>
</c:choose>
```

Figure 11 - JSTL tags used to display a button

For the initial display of the event lists, I used different JSTL tags (JavaServer Pages Standard Tag Library). The actual project pages are **.jsp** and since these tags are made for such files, using them was an obvious choice. In the example shown in *Figure 11*, I am using the **c:choose** structure to display the *See More* button on the event card.

4. Conclusion & future developments

This article is meant to present an application for Events Management, a concept which is not yet explored as it should be. Therefore I believe that this application will be a way to improve the idea and offer a unique experience and a sure way of communication and better development, both socially and business related, between the members of a community or a number of communities or groups, no matter their location, creed or status. The design is simple and elegant, thus anyone can use this application and accomplish their goals.

In this part of the article I will also specify a couple of features that will be implemented in the future.

- New types of events, such as Workshop (basically an event that is a collection of events), Scrum (for Agile Project Development), Candidate Vote (similar to the Poll event, but with a specific structure for elections) and many other.
- Images upload
- Message board (an area where different groups can communicate with each other; this establishing a link between communities)
- Different tools for managers such as: a list of employees, different statistics and charts about employees activities and what kind of events they prefer to join.
- Contact Us tab

Acknowledgment: I'd like to offer special thanks to the management team and to my colleagues at Ropardo, for the training and their guidance. Also I'd like to thank Prof. dr. Dana Simian for encouraging me to participate at this year's ICDD Conference.

References

- [1] Theodor Palamaru. *Liferay 6.2 - Create Liferay Role from Hook*
<http://blog.ropardo.ro/2016/02/24/liferay-6-2-create-liferay-role-from-hook/>
- [2] Theodor Palamaru. *Liferay 6.2 Theme with Bootstrap 3*
<http://blog.ropardo.ro/2015/10/16/liferay-6-2-theme-with-bootstrap-3/>
- [3] Theodor Palamaru. *Liferay 6.2 - Installation Guide*
<http://blog.ropardo.ro/2015/10/16/liferay-6-2-installation-guide/>
- [4] Liferay website - <https://www.liferay.com/>
- [5] Liferay Wikipedia - <https://en.wikipedia.org/wiki/Liferay>
- [6] Portlet Wikipedia - <https://en.wikipedia.org/wiki/Portlet>

Theodor-Ovidiu PALAMARU
"Lucian Blaga" University of Sibiu
Department of Informatics
No. 5-7, Dr. Ioan Ratiu st., 550012 Sibiu,
Romania
E-mail: palamaru.theodor@gmail.com

Pseudocode Learning Companion

Popa Cătălin
Teacher Coordinator: Ignat Cornelia

Abstract

Be it the introduction into elementary algorithmic notions in the 9th grade, the preparation for the national exams in the 12th grade, or reviewing information during university, good knowledge of the pseudocode language is mandatory for any pupil studying Informatics. The aim of the program is not only to provide a modern and effective means of teaching (or updating) this language, but also to support self-learning. Made using Microsoft Visual C#, the program offers the possibility to explain the language using multiple modules: learning environment, code checker and translator. The learning environment allows pupils to master the syntax and characteristics of an individual statement using on-screen instructions, and then to try and write it themselves and see whether it's correct. Later, the pupil may check if a whole algorithm that uses multiple statements has been written correctly. Last but not least, the translator provides an easy means of transition towards C++ or Pascal, by effectively translating pseudocode into real code.

1 Introduction

This program is meant to provide a friendly way of learning the pseudocode language, but also a good helper for those who wish to learn themselves. It has a clear, easy-to-use user interface, which makes it ideal for usage in pre-tertiary education, while studying Informatics. It encourages students to start learning independently and to discover things themselves, without being assisted.

The project came to life out of the desire to facilitate pseudocode learning as a true foundation for object-oriented programming, but also out of being curious about learning visual programming. As such, the project is a Windows Forms Application made by using Microsoft Visual C# 2015 [1].

A program similar to the one proposed is „Pseudocode” made by ValiNet Romania (Radu Valentin and Dudescu Alexandru), which is a professional pseudocode translator only [2], as opposed to a larger suite.

This paper is going to give an insight into the program's usage (supported by screenshots from the program itself) and the way the program works (supported by a small code sequence from the program's core function library along with explanations on what each function does and how it is done).

The originality of the program is provided by the fact that it brings together, in a single place, a pseudocode learning environment, a code checker and a translator into C++ or Pascal, out of which the code checker is truly unique.

2 Using the program

2.1 The Main Window

The program's main window, shown in Fig. 1, allows launching the three available modules: the learning environment, the code checker, or the translator. Also available from the main menu are the help section and the language selector, placed in the button-left corner of the window.

The program includes support for English and Romanian by default, together with the possibility of translation into any other language that can be written using ASCII characters, by editing the OTHER.TXT located in the program's resources directory. The formatting of the file has to be followed exactly, in order to maintain the accuracy of menus, messages, on-screen text, and, of course, of the language itself. For now, it includes a rough French translation.



Fig.1: Main Window

2.2 The Learning Environment

The first module available is the **learning environment**. Its main window, shown in Fig. 2, allows the selection of one of the eight pseudocode instructions to be learned.

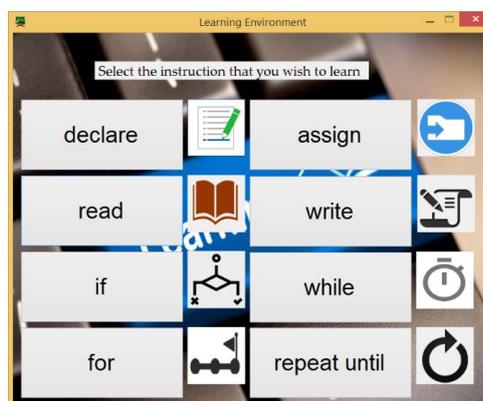


Fig.2: The learning environment's selector

After selecting one of the eight instructions, another window will appear. It contains several RichTextBoxes: one of them shows the syntax of the selected instruction, another one displays a short explanation of how that instruction works (the explanation is hidden by default), and the last one allows the student to write the instruction himself. By pressing the check button, he'll be informed if what he has written is correct or not. The program will check the syntax, the format and the general accuracy of the code (aspects like the correct usage of parentheses, commas, etc. [3]) and will display a message based on the check's result.

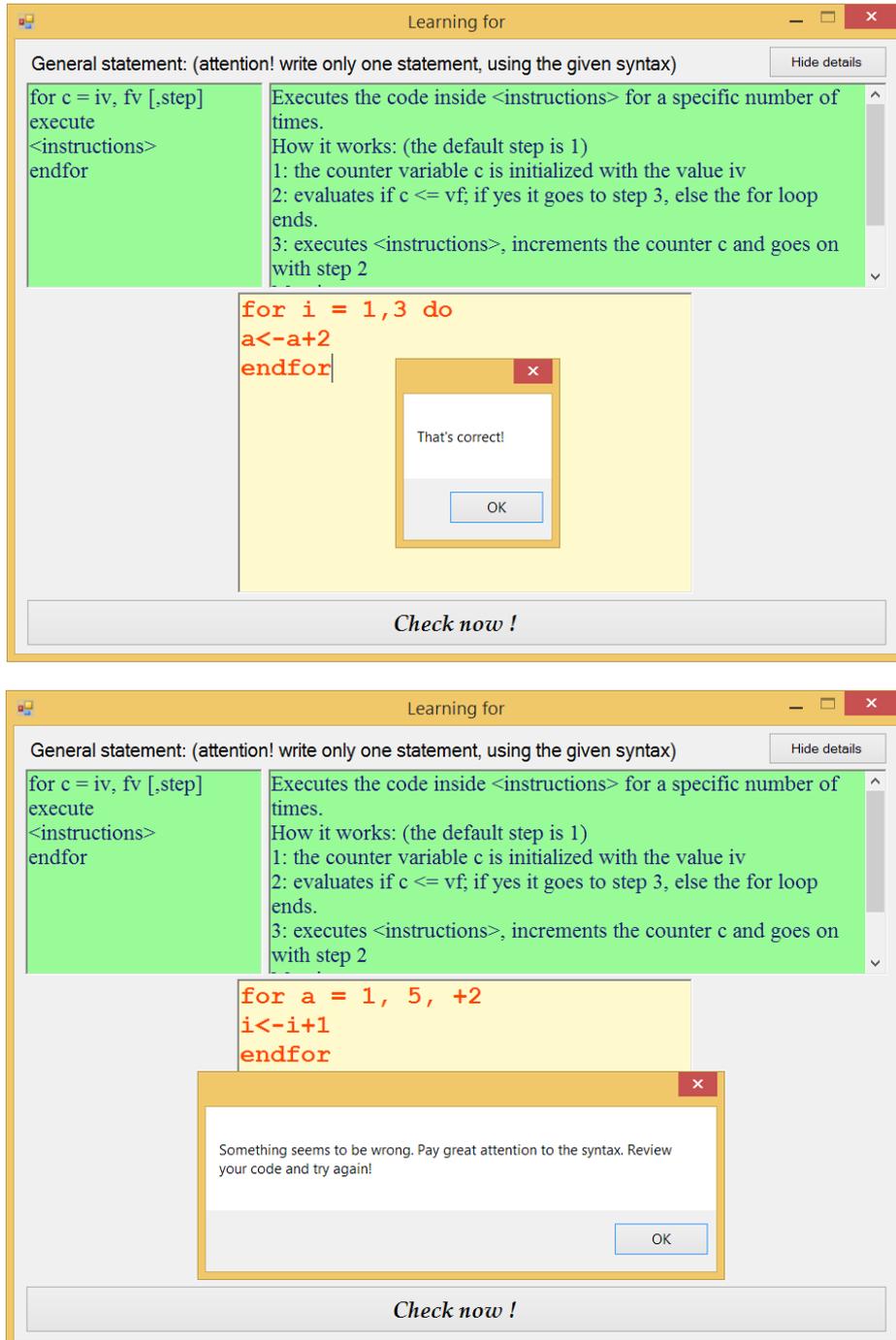


Fig.3: Learning the FOR loop

2.3 The Code Checker

The second module available is the **code checker**, which displays a RichTextBox and a check button. This module is an extension of the learning environment. Assuming a full passage and a complete understanding of the learning environment, it allows the analysis of a broader algorithm, which can be made out of multiple instructions, unlike the learning environment, which only allows the inquiry of a single instruction at a time.

As such, the student will see if he's ready to write a whole algorithm and not just isolated instructions or not. All he has to do is to type his attempt (or to open it from a text file) and to press the check button. The program will then check the given code line by line, stopping at the first line where an error is detected. Thus, it can pinpoint the line where there is an error within the algorithm.

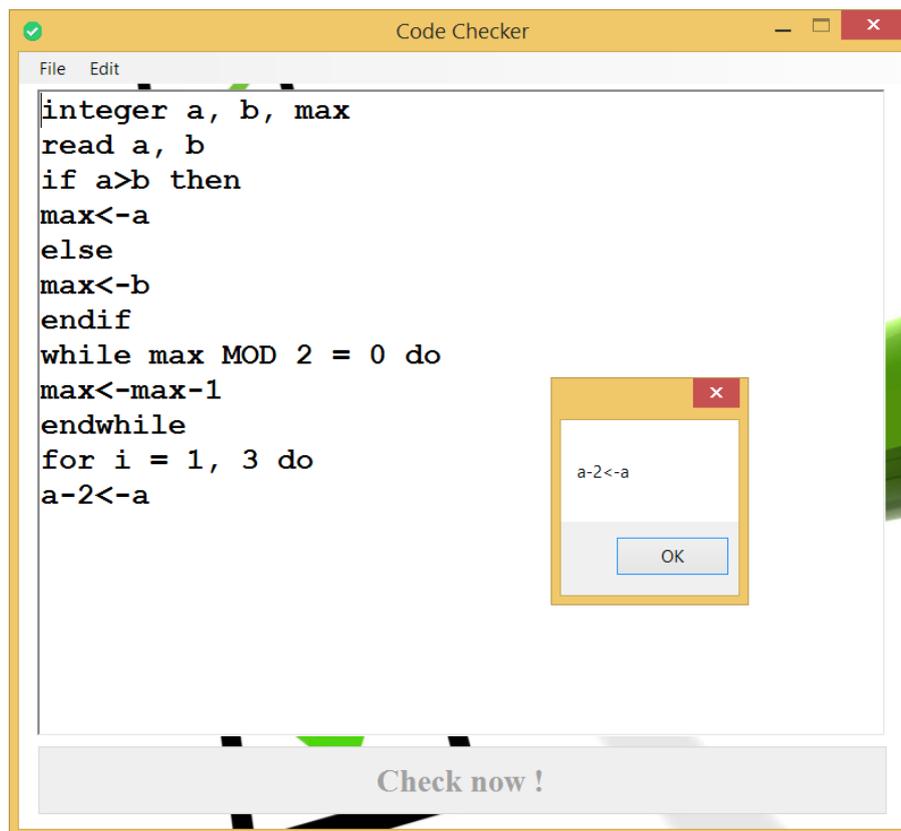


Fig.4: The Code Checker. It stops after the first line that contains an error.

2.4 The Translator

The third module available is the **translator**, which is meant for usage when the students have finished studying pseudocode, and the transition to a real programming language, one that can be compiled directly, such as C++ or Pascal is imperative. It allows the pupil to translate not only individual instructions, but also whole algorithms written in pseudocode into C++ or Pascal, greatly easing the transition from code written on paper to code written on-screen.

The student only has to type his code in the RichTextBox on the left (or to load it from a text file) and press the translation button. The program will check the accuracy of the algorithm and will signal if the code contains an error (and will refuse to translate it), and will translate it if it is detected as being correct. The result can also be saved later saved as a text file.

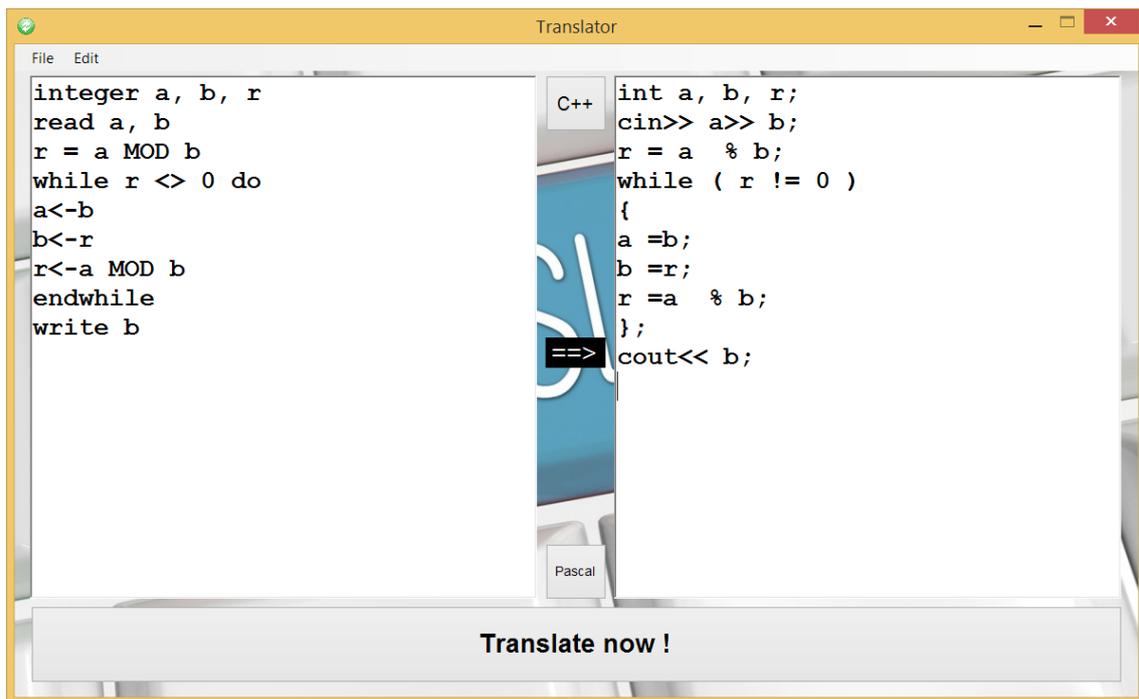


Fig.5: The translator displaying Euler's algorithm for the greatest common divisor of two numbers

3 Program Code

All the three modules of the program are based, more or less, on a **central class** defined in the program as "**class.cs**", which incorporates multiple audit functions that are shared by all other modules, but are of special use to the code checker. Anyway, all the modules also have their own parts that are not dependent on the class: the translator has its translation and equivalence engine, the learning environment can effectively check if the selected instruction is present or not in the testing box, and the code checker has a line-by-line module that is capable of pinpointing the line where there is a mistake. Also, the main window is the also the multilingual engine behind the app.

Part of the class's code, which includes certain audit functions, can be seen below:

```
public static bool verifica_prezenta_sir(string sir, string prezent) // the presence of a string is checked,
but also the presence of the assigning operator and of commas (this function is used when a certain portion
of code is not allowed to contain a certain string or the specified two operators
```

```
{
    if (sir.Contains(prezent) == false || sir.Contains("<-") == true || sir.Contains(",") == true)
        return false;
    sir = sir.Replace(prezent, "");
    if (sir.Length > 0)
        return false;
    return true;
}
```

```
public static bool verifica_sfarsituri(string sir) // the accuracy of using closing statements is checked for
any instructions that require it (such as if and endif)
```

```
{
    int i;
    for (i = 0; i <= 2; i++)
```

```
        if (sir.Contains(chestii[i]) == true && Verificare_Sintaxa.verifica_prezenta_sir(sir, chestii[i]) == false)
            return false;
        return true;
    }

    public static bool verifica_pana_cand(string sir) // the syntax from the beginning of the repeat...until loop
    is checked
    {
        if (sir.Contains(operatii[6]) == false || sir.Contains("<-") == true || sir.Contains(",") == true)
            return false;
        return true;
    }

    public static bool verifica_paranteze(string sir) // the correct usage of parantheses is checked by
    counting how many have been opened and closed
    {
        int c1 = 0, c2 = 0;
        int i;
        for (i = 0; i <= sir.Length - 1; i++)
        {
            if (sir[i] == '(')
                c1++;
            if (sir[i] == ')')
                c2++;
            if (c2 > c1)
                return false;
        }
        if (c1 != c2)
            return false;
        return true;
    }

    public static bool verifica_operatori(string sir) // the correct usage of operators is verufued by turning all
    of them into + and searching for two consecutive +es (this allows to see if there are two consecutive
    parameters, which is not allowed)
    {
        while (sir.Contains(" "))
            sir = sir.Replace(" ", "+");

        sir = sir.Replace("<-", "+"); // we cannot write an assignment that begins with an operator

        sir = sir.Replace("-", "+");
        sir = sir.Replace("*", "+");
        sir = sir.Replace("/", "+");
        sir = sir.Replace("DIV", "+");
        sir = sir.Replace("MOD", "+");

        sir = sir.Replace(">", "+");
        sir = sir.Replace("<=", "+");
        sir = sir.Replace(">=", "+");
        sir = sir.Replace("=", "+");
        sir = sir.Replace("<>", "+");

        sir = sir.Replace("AND", "+");
        sir = sir.Replace("OR", "+");
        sir = sir.Replace("NOT", "+");
        sir = sir.Replace("XOR", "+");

        if (sir.Contains("++") == true || sir.Contains("+") == true)
            return false;
        return true;
    }
}
```

4 Conclusion

Compared to „Pseudocode” by ValiNET, which is an efficient tool made for translating pseudocode into different programming languages, this program aims to be a real help to those starting to uncover the secrets of structured programming in the first year of high school (or, why not, even sooner), being made especially for self-learning. This is done by means of the code checker module that provides functionality absent from the aforementioned program.

There is a series of potential future directions of study, such as the addition of a fourth module, a builder that would allow creating a pseudocode algorithm straight away, by selecting and filling in the desired instructions. Another improvement would be the addition of syntax highlighting.

The educational potential of the program (especially the one for self-learning) is greatly enriched by the possibility of learning the individual instructions of pseudocode and then checking them individually at first and then as part of a broader algorithm later. This potential, along with the code checker itself (which is the unique module), are the main contributions.

The application is easy-to-use, thanks to its friendly interface and large buttons and textboxes, that contain clear, legible text, in such a manner that not only the code written by the student, but also the instructions meant to help him are clearly shown.

The multiple modules offered allow the user to have an overview and a detailed view of the specific way in which the pseudocode language operates. The possibility of checking and translating individual instructions and blocks of them, even if they are not part of a complete algorithm, is the main advantage of the program, especially from a self-learning perspective.

References

- [1] DUMITRIU-LUPAN, Nuşa, PINTEA, Rodica, NIŢĂ, Adrian, NIŢĂ, Mioara, *Introducere în programarea .Net Framework*, Edited by BYBLOS SRL under coordination from Agora Media SA, for Microsoft Romania, 2008
- [2] RADU, Valentin, DUDESCU, Alexandru, *Official ValiNet page*, <http://www.valinet.ro/ro-ro/about-pseudocode.html>
- [3] KELLY, Michael, *Pseudocode: An Introduction*, <http://faculty.ccri.edu/mkelly/>
- [4] IGNAT, Cornelia, *Informatics Portfolio for first year of high school*

Popa Cătălin
Liceul Tehnologic „Școala Națională de Gaz” Mediaș
Matematică-Informatică
Strada Metanului, Nr 1, Mediaș, județul Sibiu
ROMÂNIA
castle2145@gmail.com

Multicriteria alternative choice in the conditions of uncertainty on the basis of the fuzzy preference relations

Maria Sokolova
Coordinator Professor: Vladimir Chernov

Abstract

In this work a variant of a method of a multicriteria alternative choice ELECTRE, which allows to account for uncertainties of expert estimates, was developed and studied. Modified method ELECTRE is tested for solving a particular problem and describes different possible solutions of this problem in the program FuzzyCulc.

1 Introduction

The multicriteria methods of decision science considering uncertainty of estimates of alternatives criteria compliance are considered in this work.

It is possible to allocate three types of methods for solution this task according to the characteristics of information on preferences of the decision-maker:

- 1 - decision-making in the conditions of definiteness;
- 2 - decision-making in the conditions of information absence about a system of preference of the decision-maker;
- 3 - decision-making in the conditions of gradual obtaining information about a system of preference of the decision-maker.

This classification allows to allocate quite accurately three groups of methods, the first group means decision-making in the conditions of definiteness, the second and the third groups mean decision-making in the conditions of risk or uncertainty.

From a set of known methods of decision-making the most interesting are the methods giving the chance to consider uncertainty and allowing to carry out a choice of decisions from sets of alternatives of different types in the presence of the criteria having different types of measurement scales (these methods belong to the second and the third groups).

If there is no information on system of preferences of the decision-maker (the second group of criteria), standard procedures of detection of effective decisions are applied. Then the decision-maker chooses the best alternative on the basis of estimation.

The methods of the tasks solution relating to the first and second classification groups are well-known. So it is expedient to concentrate attention on those methods which give the chance to consider uncertainty and are the most applicable nowadays (the third group of criteria).

2 The choice of the method

The majority of methods of a multicriteria alternative choice is based on the estimates folding by criteria for the purpose of receiving a uniform estimation. However criteria can have various physical and logical sense so their folding isn't always reasonable. The ELECTRE 1 and ELECTRE 2 methods are the most interesting among the methods of a multicriteria alternative choice, because they assume the creation of the preference relations of alternatives instead of estimates folding [1].

The key advantage of the ELECTRE methods is the phasing of preferences identification of the decision-maker in the process of setting agreement and disagreement levels and studying of kernels. The ELECTRE methods provide wide tools for the problem solution of decision-making in the different fields of activity.

The working principle of this method: for each couple of alternatives is made the statement that the first alternative is better, than the second. There are two indexes for each couple of alternatives: an agreement index (the number confirming the statement of superiority of the first alternative over the second) and a disagreement index (the number disproving this statement). One or several best alternatives ("kernel" of alternatives) are chosen on the basis of the analysis of these indexes.

Opportunities:

1 - the method allows to carry out the analysis of the problem. Thus the problem of the decision-making is presented in this form:

- criterion weight - the numbers characterizing importance of the corresponding criterion;
- scale length of criteria;
- criteria estimates - information of qualitative character about the importance of criteria;

2 - the method allows to carry out data collection on a problem, for this purpose it is necessary to define:

- set of criteria of the alternative estimates;
- set of alternatives;

3 - the method allows to estimate importance of the accounting of each decision by setting of criterion weights.

The classical ELECTRE methods assume that experts set point estimates to alternatives. However expert estimates always contain uncertainty so it is offered to set them in the form of fuzzy numbers. The theory of fuzzy sets is very flexible mechanism allowing to receive more adequate results of the analyzed situation. In particular, uncertainty in the weight coefficients and scales values estimation used in the ELECTRE method can be considered.

Thus, using together of a method of a multicriteria alternative choice ELECTRE and the fuzzy sets is expedient. It will allow to use fuzzy linguistic variables in the process of an estimation of problem options of decision-making. So it is necessary to modify the ELECTRE method.

3 The definition of the problem

Let's suppose that there is a set of alternatives, for example, A, B, C, D, E, a set of criteria $C = \{c_i, i=1, \dots, I\}$, scale lengths of criteria $L = \{l_i, i=1, \dots, I\}$ and the criteria weights $W = \{w_i, i=1, \dots, I\}$ (Fig. 1). Thus criteria have various physical nature, for example distance, cost, time, etc. Alternative estimates by criteria are set not by dimensionless units (points), but by the corresponding physical units. In the classical ELECTRE method these estimates have an appearance of point numbers that shows the real situation inadequately. So, for example the cost of any project which realization takes some time can't be absolutely determined.

A more appropriate presentation of these statements will be in the form of fuzzy numbers (Fig. 2). In the transition to the fuzzy numbers choosing the best alternative will be based on fuzzy preference relations, which would require changes in the algorithm of the ELECTRE method.

Criteria weight		Scale length	
W1	8	L1	500
W2	7	L2	1
W3	5	L3	1
W4	6	L4	1
W5	8	L5	500
W6	6	L6	100
W7	8	L7	1
W8	8	L8	1

Figure 1 – Criteria weights and scale lengths

Criteria	A	B	C	D	E
1)	35	15	15	17	500
2)	0,7	0,8	0,5	0,5	0,9
3)	0,6	0,3	0,6	0,6	0,5
4)	1	0,6	0,5	0,5	0,5
5)	15	50	150	200	500
6)	70	30	10	10	10
7)	0,3	0,5	0,4	0,4	0,9
8)	0,4	0,5	0,4	0,4	0,5

Figure 2 – Alternative estimates

4 The decision procedure

The calculation of the agreement and disagreement indexes is performed by using fuzzy table FuzzyCalc [2] (Fig. 3 and 4).

		A	B	C	D	E
2						
3	A	-	0,505917	0,867286	0,867286	0,361369
4	B	0,505917	-	1,01183	1,01183	0,596259
5	C	0,379438	0,144548	-	0,867286	0,361369
6	D	0,379438	0,144548	0,867286	-	0,361369
7	E	0,650465	0,560122	0,867286	0,867286	-

Figure 3 – The matrix of agreement indexes

		A	B	C	D	E
17						
18	A	-	0,3	0,1	0,1	0,93
19	B	0,4	-	0	0,004	0,97
20	C	0,6	0,3	-	0,004	0,97
21	D	0,6	0,3	0,1	-	0,966
22	E	0,6	0,9	0,7	0,6	-

Figure 4 – The matrix of disagreement indexes

In the classical ELECTRE method the decision-maker (DM), estimating the level of agreement and disagreement, builds the kernel of the dominant alternatives. Direct participation of a decision-maker in this process is a drawback of the classical ELECTRE method. The choice of fuzzy levels of agreement and disagreement becomes more difficult, because it requires comparison of fuzzy

numbers. Another way for solving this problem, which is used in this work, is free of these shortcomings.

A fuzzy number is the fuzzy set which is defined on a real axis that is why it is possible to apply sum operation to fuzzy numbers for structuring alternatives:

$$\tilde{\alpha}_i = \bigcup_k \tilde{q}_{ik} , \quad (1)$$

$$\tilde{\gamma}_i = \bigcup_k \tilde{r}_{ik} . \quad (2)$$

The estimates received on formulas (1) and (2) are integrated values of agreement and disagreement coefficients by all set of criteria. Thus the formula (1) characterizes "soft" approach, and the formula (2) - "rigid" approach. It is obvious that the most preferable alternative should have the maximum value of $\tilde{\alpha}_i$ (Fig. 5) and minimum value of $\tilde{\gamma}_i$ (Fig. 6). The received values of these parameters confirm the result received earlier.



Figure 5 – The maximum value of $\tilde{\alpha}_i$

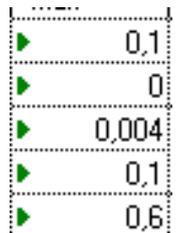


Figure 6 – The minimum value of $\tilde{\gamma}_i$

Thus the alternative B unambiguously has the maximum value of $\tilde{\alpha}_i$ (Fig. 5) and has the minimum of $\tilde{\gamma}_i$ (Fig. 6). Thus, results don't contradict each other, but it is necessary to make additional verification of the received decision.

One more solution of the considered task is possible, it is caused by existence of uncertainty in basic data and results (Fig. 7). Matrixes of agreement and disagreement coefficients can be considered as matrixes of the fuzzy values characterizing uncertainty of a situation, so fuzzy entropy can be used for estimating of alternative decisions, [3] it is determined by a classical formula of Shannon [4]:

$$H_i = \sum_k \mu_{i,k} \log_2 \mu_{i,k} , \quad (3)$$



Figure 7 – Entropy calculation

An entropy is an estimation of an uncertainty level, so the best decision should have the minimum entropy value. The calculations which are carried out by formula (3) confirm already received results – the best is the alternative B. Thus, three various ways of the the same task solution led to the same results that is in full accordance with methodology of the stability theory according to which the result of data processing is invariant concerning a method of processing data.

5 Conclusions

The received results give the grounds to speak about opportunity of practical application of the modified ELECTRE method in the conditions of uncertainty of expert estimates. Receiving the same results by various independent methods confirms their truth.

References

- [1] Roy B. *Multicriteria Methodology for Decision Aiding*, Dordrecht : Kluwer Academic Pulisher, 1996.
- [2] Chernov V. G., *Solution of business tasks by means of fuzzy algebra*, Moscow, Diagramm, 1998.
- [3] Deshmukh, K.C., Khot, P.G., *Generalized Measures of Fuzzy Entropy and their Properties*, World Academy of Science, Engineering & Technology, Issue 56, 2011.
- [4] Chernov V. G., *Entropy criterion of decision-making in the conditions of full uncertainty*, Management information systems, Issue 6, 2014.

Sokolova Maria
Vladimir State University
Faculty of Information Technology
Department of Management and Informatics in
technical and economic systems
Str. Gorky, 87, Vladimir
RUSSIA
E-mail: sok.masha2011@yandex.ru

Bluetooth Rover

Radu Stoican

Teacher Coordinator: Georgescu Oana

Abstract

The project consists in a Bluetooth controlled rover. Any Bluetooth device can be used to communicate with the robot, for example computers, smartphones or tablets that have Android as an operating system. The main part of the project is the Arduino Uno microcontroller and the robot is programmed in the Arduino software with C++ language. The robot also has other parts as motors, sensors, LEDs and the ability to use SD and microSD cards.

1. Introduction

The project is a robot built as a rover, robotic vehicle that can cross different obstacles and irregular planes through tracks.

2. Description of the project

2.1. Functions

The rover uses the Arduino Uno microcontroller as the main part, which is connected to two motors powered by four AA batteries. The robot is equipped with a Bluetooth module that provides communication and remote control of the microprocessor. It also has an ultrasonic sensor that can be used for determining the position and distance of an object to the rover. Thus, the robot can be used to perform quick and accurate measurements in spaces of different sizes or areas inaccessible to humans. The rover can also store measurement values on an SD or microSD card that can be attached to the robot. The project can be seen in Fig. 1 and Fig. 2.

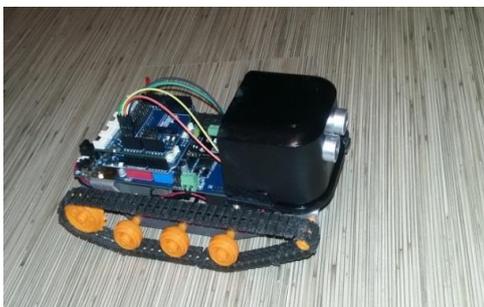


Fig. 1: Bluetooth Rover

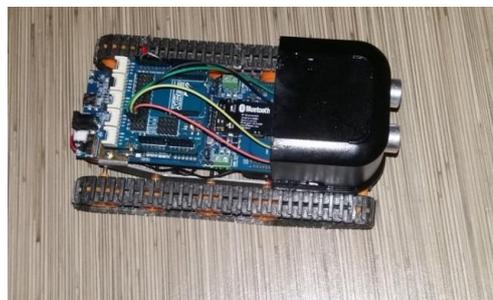


Fig. 2: Bluetooth Rover

The project is used to perform accurate measurements and save the results on the SD or microSD card that is attached to the robot. The movement of the robot can be easily controlled remotely by the user.

The project can also receive commands through different characters sent via Bluetooth, the robot having different functions.

A first function that can be used is to measure, through an ultrasonic sensor, the distance to the object in front of the robot by simply pressing the key 'i'. The rover confirms the measurement by lighting a LED for a short period of time, saving the value measured in centimeters on its memory card.

Another function of receiving rover is automatically and continuously measuring the distance every half a second. This function can be activated by pressing 'k'. The function is activated regardless of the state in which the robot is - moving or stationary. The function will remain active regardless of changing the state of motion in which the rover is. When the function is active, the rover signals by lighting an LED at each measurement. To deactivate the function, the 'k' key must be pressed again.

The temperature sensor the rover has equipped allows the robot to perform accurate measurements of temperature in Celsius and Fahrenheit degrees. The sensor is initially off, but it can be activated by pressing 't', measurements being made like the distance ones by pressing the keys 'i' and 'k'. Temperature values are also saved on the SD card.

Also, when the robot is connected to a computer via USB cable, the key 'r' can be used for displaying the data saved on the memory card of the robot, provided that on the computer exists an application that can communicate with the robot, such as the Arduino software.

2.2 The Hardware

The project consists of a robot built from several parts and a program that determines the robot's behavior, which is written in the Arduino software, C ++ language. The project consists of the following parts:

- Arduino Uno microcontroller (Fig. 3) - is the most popular microcontroller from the Arduino company. This is the most important piece of the project, being the "brain" of the robot. Through uploading the program with a USB cable to the robot, the microprocessor uses the instructions to communicate with the user. Communication is possible in both directions. Arduino Uno is also used for the "thinking" function of the robot, which is able to make decisions by itself depending on the situations it is in. The microprocessor is also used for controlling other electronic components of the project.



Fig. 3: Arduino Uno

- Motor controller - is the element that can act motors that are used to move the robot, which require a higher voltage than 5V, the maximum voltage that can be provided by the Arduino Uno microprocessor. The motor controller does not analyze instructions, but it receives them from the microprocessor, acting according to the command transmitted.

The two components are usually separate parts, however, for making the use of them and for building project easier they are attached, forming a single component.

- Engines - the robot uses two motors that produce circular motion, working on direct current voltage. They operate the gears that make the tracks move and, therefore, moving the entire robot. The rover can move in any direction, steering by slowing or stopping the opposite track of the desired direction.
- Bluetooth module (Fig. 4) - provides the robot's ability to communicate and receive instructions from the user. It can be paired with any device that can perform a Bluetooth communication. Thus, any function of the robot can be accessed via Remote Control by using portable devices, such as smartphones that have the Android operating system, provided that they have an application that enables communication with the Arduino Uno microprocessor.



Fig. 4: Bluetooth module

- Ultrasonic sensor (Fig. 5) - is used by the robot to determine whether or not there is an object in its proximity and to calculate the distance to it. The sensor system represents the "vision system" of the robot and works on the basis of ultrasonic with a frequency of 40kHz. Thus, the sensor sends a sound signal and measures the time from transmission to reception of the echo that occurs when meeting an obstacle. After this operation, the sensor returns the value obtained in microseconds, which can be converted into distance by using the mathematical formula 1.

$$\text{distance} = (\text{duration} / 2) / 29.1 \quad (1)$$

The formula uses the speed of sound, which is 340 m / s or 29.1 microseconds per centimeter. The result thus obtained is divided by 2 because the sound is sent to the object and its echo back to the sensor, resulting in two roads. The sensor can calculate the distance of minimum 2cm and maximum 4m.



Fig. 5: Ultrasonic sensor

-Shields - these are extensions that can be added to a microprocessor to take advantage of new features. There are numerous types of such shields such as: Ethernet Shield (which provides the ability to connect the project to the Internet), LCD screen Shield (which can be used to display different values and messages), digital camera Shield (which makes it possible to add and use a camera) etc. The biggest advantage of the shields is their ability to be stacked, so multiple Shields can be used for a microprocessor, provided that there are no pins used by both shields. The project presented uses two shield: one of them offers the ability to connect multiple sensors in different ways, which would not be possible only by using the microprocessor alone (Fig. 6). The second shield used offers the possibility of adding a SD or microSD card that can store data (Fig. 7).

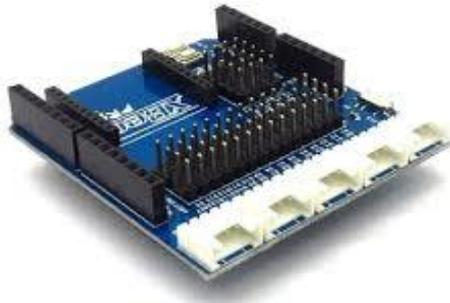


Fig. 6: *Sensor Shield*



Fig. 7: *SD Shield*

-Temperature sensor (Fig. 7) - is used for temperature measurement of the environment in which the robot is. The robot receives the results of measurement in volts (their number is between 0 and 5), these being transformed in Celsius degrees using the formula (2), and the transformation from Celsius to Fahrenheit is realised with formula (3).

$$\text{degreesC} = (\text{voltage} - 0.5) * 10.0 \quad (2)$$

$$\text{degreesF} = \text{degreesC} * (9.0 / 5.0) + 32.0 \quad (3)$$

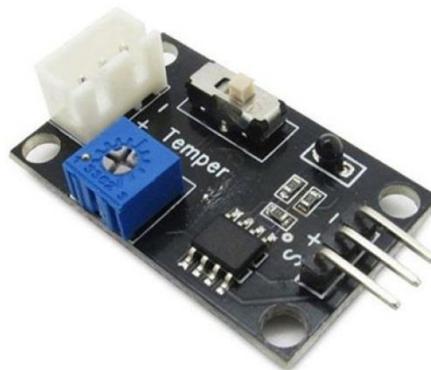


Fig. 7: *Temperature sensor*

2.3. The program

The robot is programmed in the Arduino software, C ++ language. The current program uses the following pins from the microprocessor: 2,3 are used for ultrasonic sensor; 9 for LED; 4,10,11,12 and 13 are used for the SD shield; and from 5 to 8 for the two engines. The pin analog 1 is used for the temperature sensor.

Further the program used and the most important sequences in it will be described.

In the subprogram void setup (void) is assigned to each pin a value of INPUT or OUTPUT, depending on their role. Thus OUTPUT means that through the chosen pin will be sent instructions from the microprocessor to the connected component, for example the engines that work according to the instructions received, and the INPUT pins transmit data to the microprocessor, using it according to the instructions. An example is reading the distance from the ultrasonic sensor. Also, the speed of data transmission from the user to the robot and vice versa can be chosen here, the current speed being of 9600 bits per second. The subprogram void setup (void) can not be absent from any program because it ensures the necessary settings before starting the loop of instructions.

The main part of the program is called void loop (void) and represents instructions that will be repeated as long as the robot is running. Variable "val2" is used to allow receiving commands for performing measurements during movement, when variable "val" is already used.

In the following sequence is assigned to the variable "val" the value sent by the user if "val2" has already received a value of 'i' or 'k'.

```
if (val2 == 'i' || val2 == 'k' || val2 == '\0')
{
  while (Serial.available() < 1) {}
  val = Serial.read();
}
else
  val = val2;
```

By using the instruction switch (val) is called a subprogram, depending on the value received. These subprograms can be one of the five that ensure the movement of the robot, including the function void stop (), which is used to stop the rover from moving, or one of the two measurement functions.

In the next sequence is presented the case of receiving the value 'w':

```
case 'w':
  forward (leftspeed, rightspeed);
  if (!Serial.available())
  {
    if (k > 0)
      pingtime();
  }
  else
  {
    val2 = Serial.read();
    if (val2 == 'i')
      SDwrite();
    if (val2 == 'k')
      k = (-1) * k;
  }
  break;
```

The subprogram "forward", that receives the parameters a and b, has the role of transmitting the speed and direction necessary to move forward to the engines. The subprograms "reverse", "left" and "right" function in the same way, according to the chosen direction. Function "stop" fixes both engines speed to 0 (zero).

In the following sequence is shown the subprogram used for moving forward:

```
void forward(char a, char b)
{
    analogWrite (E1, a);
    digitalWrite(M1, LOW);
    analogWrite (E2, b);
    digitalWrite(M2, LOW);
}
```

After calling the subprogram "forward" it is checked if measurements are necessary, the instructions needed to perform them can be received during movement without affecting the robot's motion through the use of variable "val2". The next sequence of the program is accessed when receiving value 'k' and performs measurements every half-second.

```
case 'k':
    k = (-1) * k;
    if (k > 0)
        pingtime();
    break;
```

The variable 'k' is used to identify the state in which the robot is, meaning whether or not it should start or end the measurement cycle, so its value is initially -1, and by pressing the key 'k' it becomes 1, thereby starting the measurement.

In the following sequence is shown the subprogram "ping" that is designed to measure and return the distance.

```
int ping ()
{
    long duration, distance;
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = (duration / 2) / 29.1;
    digitalWrite(LED, HIGH);
    delay(pingdistance);
    digitalWrite(LED, LOW);
    return distance;
}
```

After the signal is sent and returned to the sensor, it is transformed in distance in the variable "distance", which is intended to measure the time it takes the signal to travel from the sensor, to the object and, respectively, back. Using the sequences `digitalWrite (trigPin, LOW);` and `digitalWrite (trigPin, HIGH);` are sent some initial signals that are designed to ensure the normal operation of the ultrasonic sensor. The measurement is done by instruction `duration = pulseIn (echoPin, HIGH);` where the constant "echoPin" is used to receive the ultrasound echo.

The function "ping time" is used to call the subprogram "ping" every half second. While the microprocessor receives no value from the user, or receives values that are different from 'k', the function will continue to enter the subprogram every half second.

```
int pingtime ()
{
    SDwrite();
    if (!Serial.available())
```

```
{  
  delay(pingdistance);  
  pingtime();  
}
```

The function "temperature" is used both for the performing measurements through the temperature sensor and for converting the received values from volts to Celsius and Fahrenheit degrees, using formulas mentioned above.

```
void temperature()  
{  
  voltage = analogRead(temperaturePin) * 0.004882814;  
  degreesC = (voltage - 0.5) * 10.0;  
  
  degreesF = degreesC * (9.0/5.0) + 32.0;  
}
```

Writing and reading to the SD card are performed by calling the subprograms "SDwrite" and "SDread".

3. Conclusions and Perspectives

In conclusion, the project has various practical applications which were submitted in the paper.

There are numerous possible future developments such as using a camera that transmits live images or replacing the tracks with wheels, thereby gaining speed in exchange for the ability to traverse rough areas and obstacles. Adding other sensors such as light sensors, humidity, sound, etc. helps measuring various environmental conditions in which the robot is.

There are other projects like this one that are used for similar purposes. An example would be presented at website [4]. It has no ability to move, but it displays the measurements results in real time via the LSD.

A more complex future project, which can evolve from this one to have more practical applications, is a robot that can record and transmit images in real time via a wireless cameras, which can be used for monitoring, responding to various disasters and emergencies, for example scanning ruins after a natural disaster etc.

4. Bibliography

- [1] <https://www.arduino.cc/>
- [2] https://en.wikipedia.org/wiki/Main_Page
- [3] <http://www.robotshop.com/blog/en/robots/gorobotics/tutorials/how-to-make-a-robot>
- [4] <http://www.instructables.com/id/Arduino-LCD-Project-for-Measuring-Distance/>.
- [5] <https://www.robofun.ro/>
- [6] Kimmo Karvinen, Tero Karvinen, "Make: Arduino Bots and Gadgets", Ed. O'Reilly Media, 2011
- [7] Massimo Banzi, "Getting Started with Arduino", Make:Books, 2011

Stoican Radu
Colegiul National de Informatica "Grigore Moisil"
Profil real / specializarea Matematică Informatică, intensiv Informatică
Matematica-Informatica
Brasov, Romania
E-mail: radasca16@gmail.com

Embedded application controlled by a mobile device

Marius TOMA

Coordinating Professor: Dana SIMIAN

Abstract

The aim of this paper is to present an idea of a simple robot which is moving on four axes and is controlled by a mobile device through an android app using a Bluetooth module. This robot could have applications in various fields that we come across. With his computational and agile skills the robot can perform assignments that are difficult or hazardous to humans. One area where robots are beginning to come through, and which causes less controversial debate on their efficiency is as explorers.

The proposed work is to implement a robot that practically simulates a remote control vehicle. Due to the large wheels and the fact that the chassis is flexible, the robot well approaches any kind of hard surface ground. Also, the most basic component of the project is a microcontroller Arduino Uno. This is the "master" or "the brain" of the entire robot vehicle, the Arduino board uses a piece of written code similar to C++ language loaded into its microcontroller memory to initialize or control the other components.

1 Introduction

We live in a century where science and technology have a major impact on our daily life; robotics is one of the today science booms. Recently, there has been increasing interest in the emerging field of robotics. Today's robots are mechanical arms controlled by computers that are programmed to perform a range of handling activities. They are establishing themselves in manufacturing automation systems to produce a range of goods with great precision. The emerging era of robots calls for different types of skills. On non-industrial areas, the first trainee robots for domestic use are coming off the production lines. Also robots are being used in hazardous places, such as army missions; outer space or under the sea [4].

Project idea started some years ago when I was watching the show "Robot Wars" on Discovery channel. Back in those days I was very excited about the idea of building a robot by myself, time passed away... And a year ago after I consulted some specialized sites and forums, I bought the parts needed to implement and develop a basic type of explorer robot. Also behind of following a personal research I was started to own some skills about the embedded development. Must be mentioned that specialized online sites of today are providing a large range of embedded components such as: development boards; sensors; RF communication modules (IR, Bluetooth, Wi-Fi) etc. at very low cost.

With recent advances in consumer electronics, mobile phones have displayed an incredible range of possible applications. Given that unmanned robot industry systems are a rapidly growing, this project was interested in developing a system based around Android mobile system that could meet the basic requirements for controlling a remote vehicle, while minimizing cost and user training.

Through the development of this robotic platform, the users will be able to concurrently growth experience with application development and robotics.

The purpose of the present research is to provide a simple robot hardware architecture based on a powerful computational platform so that can focus on the research and tests, instead of Bluetooth connection infrastructure. This simple robot vehicle architecture is also useful for educational robotics courses, because students can build their own robots with low cost and use them as platform for experiments in several courses.

2 Robot vehicle architecture

The whole robot architecture is built around a microcontroller Arduino UNO V3 board which is the vehicle brain, and around the microcontroller we have a few basic components such as: a Bluetooth module Roving RN-42 which is used to sustain the communication process between mobile phone and Arduino; the driver shield L298; and the four servo-motors (MG-6-120) attached on the flexible chassis. The entire robot is powered by six AA batteries generating 9V.

The robot vehicle is designed to optimally provide mobility to the platform as a whole. In order to meet those requirements, four servo DC motors, are attached to L298 motor driver. By using an H-bridge, the microcontroller can easily change the direction of each motor independently, while running off a common power supply. The H-bridge is implemented in the Arduino driver shield.

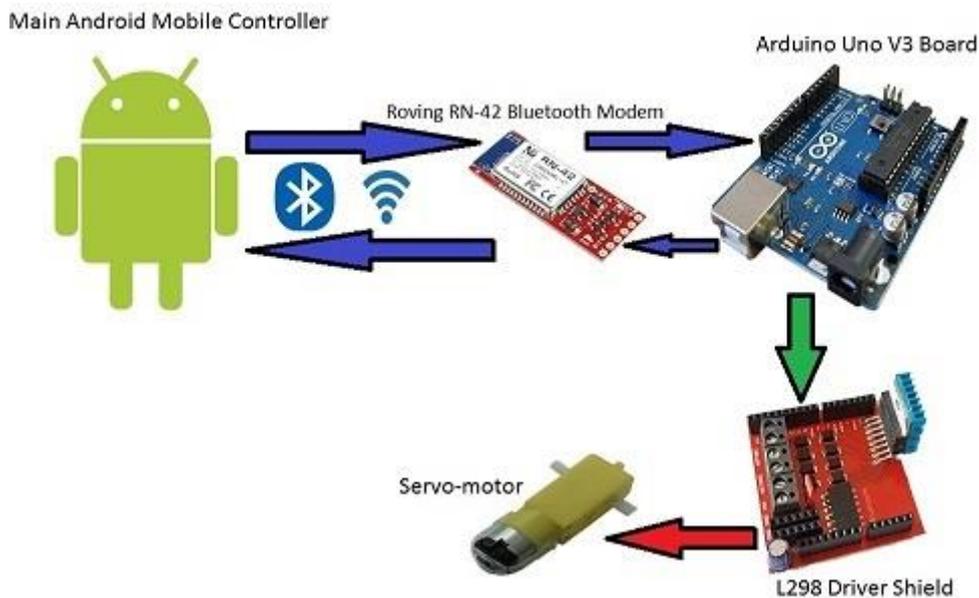


Fig.1: Communication Diagram

Our various assets communicate with each other as presented in Fig. 1. The mobile communicates to a Bluetooth module which delivers the code to the Arduino Uno. The Uno decodes it, passes the information to the driver shield and controls the motors.

2.1 Arduino UNO V3 Board

Arduino UNO V3 Board is an open source physical computing platform based on a I/O board and a development environment that implements the Processing language (similar to C++).It can be used to develop stand-alone interactive objects, the board consisted in a small platform (6.80 / 5.30 cm) As we reported in (Fig. 2).

It is built around a signal processor so is being able to retrieve data from the environment by a series of sensors, and to perform actions on the environment through lights, motors, actuators, and other mechanical devices.



Fig. 2: Arduino Board

The Arduino Uno is the most “standard” Arduino board currently in the market, and is probably the best choice for beginners just getting started with the platform. The board is well-suited with more shields (add-on boards) than other models.

Arduino simplifies the process of working with microcontrollers, and offers some advantages for students, teachers and interested amateurs over other systems Arduino is:

- Simple, clear programming environment.
- Open source and extensible software.
- Open source and extensible hardware.

The Uno's main limitation is ATmega328 chip, which doesn't have lot of SRAM or flash memory, which limits the kinds of programs we can load on the chip. In my project I used about 35% of entire 32kb of flash memory and 41% of dynamic memory and must say that for my kind of project it suits very well. But If your project involves a display or otherwise needs to store and use any form of images or audio data, 2KB of memory isn't going to be enough [1].

The main advantage of the Arduino Uno board is that microcontroller can be removed and replaced from the socket in case of breakdown.

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It contains 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, USB connection, power jack, 16 MHz ceramic resonator, an ICSP (in-circuit serial programming) header, and a reset button. It contains everything which is needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 programmed as a USB-to-serial converter. Revision 2 of the Arduino Uno board has a resistor pulling 8U2 HWB line to ground, making it easier to put into DFU(Device Firmware Update) mode [1].

Revision 3 of the Arduino Uno is the board which I used to my project, and has the following new features: 1.0 pinout: added SDA(serial data out) and SCL(serial clock line) pins that are nearer to the AREF pin and two other new pins placed near to the RESET pin, IOREF that allows the shields to adapt to the voltage provided from the board. In future, shields would be compatible with both the board that uses AVR, which operates with 5V and Arduino Due which operates by 3.3V. The second one is not a connected pin and is reserved for future purposes.

The Arduino programs are written in a programming language similar to C++. An Arduino IDE (Integrated Development Environment) comes with a software library called "Wiring" from the original Wiring projects, which makes many common input/output operations much simple. Users only need to define two functions to make a runnable cyclic executive program:

- `setup()`: a function run one time at the start of a program that can initialize settings.
- `loop()`: a function called repeatedly till the board powers off.

The Arduino IDE utilizes the GNU tool chain and AVR Libc to compile programs and uses avrdude to upload programs to the board. As the Arduino platform utilizes Atmel microcontrollers, Atmel's development environment, AVR Studio or the newer Atmel Studio, may also be used to develop software for the Arduino [3].

2.2 Driver shield L298

L298 motor driver as shown in (Fig. 3), is used for controlling DC motors using Arduino [5].

Technical reason why we used the driver is that Arduino is able to draw on his power ports a very small amount of power, totally insufficient to actuate a motor. If we connect an electric motor directly to a port on Arduino, most likely we get the burning of the signal processor on the board.

In order to avoid such a bad technical malfunction, we need a power amplifier which takes strength from the power supply (battery, in our project), and transmit that power to the motors as Arduino command.

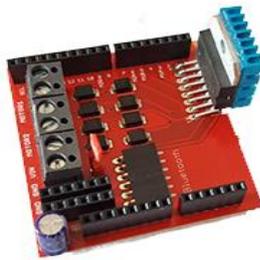


Fig. 3: Driver Shield

The driver shield connects to the Arduino platform using four digital pins (3, 5, 6 and 9) knock in to the pins directly into the Arduino board. And it also have a dedicated Bluetooth socket.

As a feature various sensors can be directly fed from the shield, using the 6 connectors that provide direct connection to the 5V and the 6 connectors that provide direct.

2.3 Bluetooth module Roving RN-42:

Description:

The Bluetooth module is connected to the Arduino board which communicate with each other through Analog signals. The module receives its instructions from the android phone app. L298 Driver takes these signals and instructs to the motors.

The RN42 is a small form factor, low power, Class 2 Bluetooth radio for designers who want to add wireless capability to their products. The RN42 supports multiple interface protocols, is simple to design in, and is fully certified, making it a complete embedded Bluetooth solution.

In our project, we are controlling the robot wirelessly from an android device via a RN-42 Bluetooth modem (Fig. 4). This requires a UART (Universal Asynchronous Receiver Transmitter) the receiver end of the Bluetooth modem at which is connected serially with the Arduino board.

The Bluetooth modem is powered by Arduino board and is encrypted with a MAC address which when entered on the android phone, achieves communication with the Arduino board.

The RN-42 Bluetooth module works great with Arduino, and It is very simple to use (the data written on pins Rx / Tx are available on receiver) from this point of view we can consider as pins Rx / Tx are connected by common threads between transmitter and receiver.

Physical connection of the Bluetooth adapter to the Arduino board is made extremely simple, on the shield we have a dedicated 6-pin connector designed specifically for Bluetooth. Arduino Bluetooth module communicates with the board via a digital Software-Serial initialized on pins 2 and 4.



Fig. 4: Bluetooth Modem Module

Features:

- Bluetooth v2.0+EDR support
- Secure communication, (128 bit encryption)
- Low power (26uA sleep, 3mA connected, 30mA transmit)
- UART (SPP or HCI) and USB (HCI only) data connection interfaces.
- Sustained SPP data rates - 240Kbps (slave), 300Kbps (master)
- HCI data rates - 1.5Mbps sustained, 3.0Mbps burst in HCI mode.
- Class 2 radio, 60 feet (20meters) distance, 4dBm output transmitter [6].

2.4 Servo motors and the flexible chassis

The robot uses the four MG-6-120(Fig.6) DC servo motors that take electrical energy and produce mechanical energy in order to operate the four wheels. Using the kinetic energy from the wheels the robot easily moves on four axes, and the principle movement on the left or right is like the "panzer" method: two side wheels rotating forward and other two side wheels rotating backward and this will cause a circular movement on the entire robot(Fig.5) [7] .

Robot chassis is made of two panels of plexiglass attached with a bolt. Due to the large wheels and the fact that chassis is flexible, the robot vehicle well approaches any kind of hard surface.

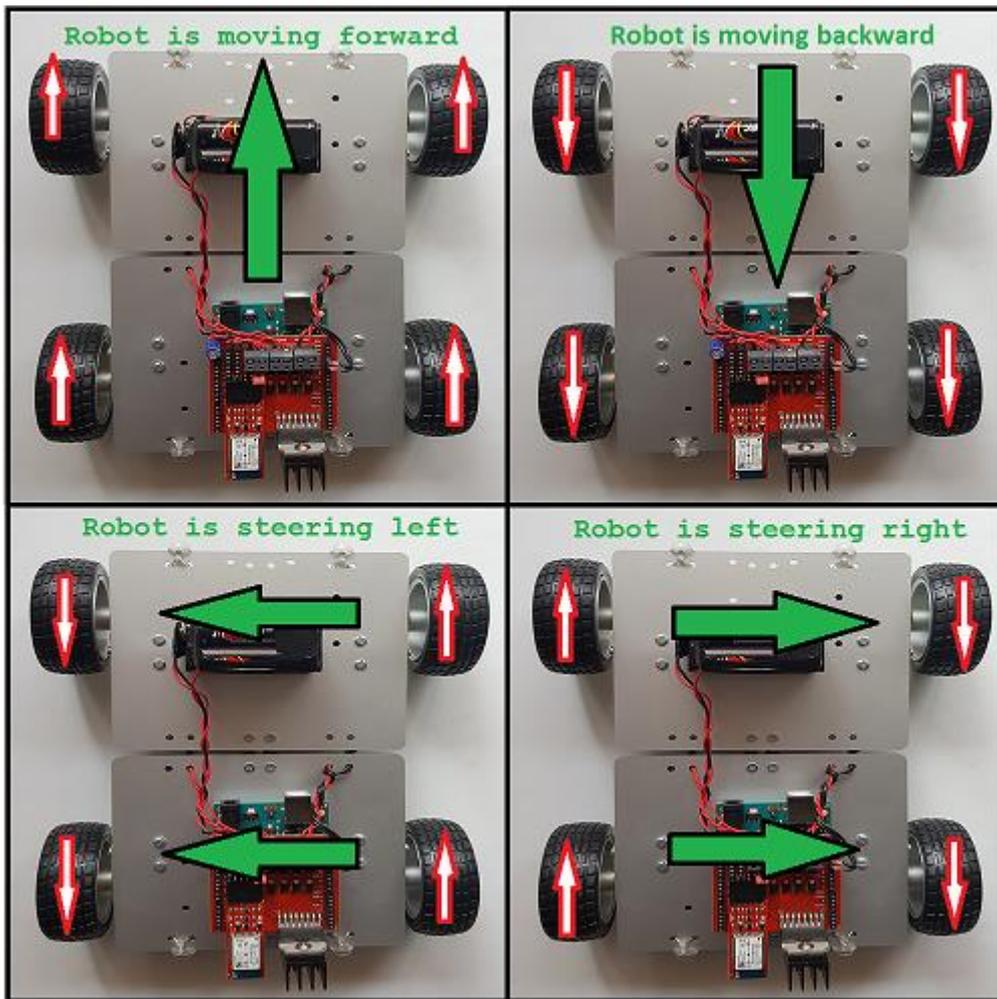


Fig. 5: Robot Movement Principle



Fig. 6: Servo-Motor

Features of the servo-motor:

- Double Shaft Gear Motor
- Rated Voltage: 6VDC
- Rated Current: <300mA
- Rotation speed: 100/min
- Torque: 1.0 kgf.cm [7].

3 The embedded code

The robot vehicle components consist of one android mobile device, an Arduino Uno, L298 Driver Shield and a Bluetooth module. The user android device is used as a Bluetooth controller, with a simple free to download android app available on google play store [2] for free, that anyone can pick up and use without prior training. The Bluetooth signals are interpreted by the Arduino Uno, which process the commands into mechanical motion via signals sent to the L298 Driver Shield which drives the motors.

The Arduino code was designed to interpret signals received through Bluetooth from the Android mobile controller. For each state that the robot can be in, a different code was assigned. A switch case is used, and for each different state, sends the proper signals to the servo-motors. In addition, our code implements a Bluetooth disconnect fail safe, such that if it does not receive a command after a set period of time, it will trigger a flag which tells the robots motors to stop.

Programming of the robot is performed in the Arduino software, and Arduino programming language (similarly to C++). The actual programming code initialize the digital microcontroller pins:3,5,6,9 as output for motors controlling. And the digital pins 2 for input / pin 4 for output to initialize the software serial receiver/transmitter needed for Bluetooth communication. And also we initialize the digital pin 13 as a output for the chassis led. The whole steps described above is a part of function: void setup() which is a function that run one time at the start of our program and its main purpose is to initialize settings. Also in this function It takes place the initialization of serial communication following this section of code:

```
Serial.begin(9600);
mySerial.begin(115200);
mySerial.print("$");
mySerial.print("$");
mySerial.print("$");
delay(100);
mySerial.println("U,9600,N");
mySerial.begin(9600); // Start bluetooth serial at 9600
pinMode(ledPin, OUTPUT);
Serial.println(VERSION);
while (mySerial.available()) mySerial.read();// empty RX buffer
```

The main part of the program is void loop () and this function repeats the instructions as long as the Arduino board is power on. Further I will present a short code sequence part of the function void loop ():

```
void go (int speedLeft, int speedRight) {
  if (speedLeft > 0) {
    analogWrite(MOTOR1_PIN1, speedLeft);
    analogWrite(MOTOR1_PIN2, 0);
  }
  else {
    analogWrite(MOTOR1_PIN1, 0);
    analogWrite(MOTOR1_PIN2, -speedLeft);
  }
}
```

```
if (speedRight > 0) {  
  analogWrite(MOTOR2_PIN1, speedRight);  
  analogWrite(MOTOR2_PIN2, 0);  
} else {  
  analogWrite(MOTOR2_PIN1, 0);  
  analogWrite(MOTOR2_PIN2, -speedRight);  
}  
}
```

This function receives two parameters meaning the speed of the two directions (left and right) according to their values the DC motors start to spin which cause the robot to move forward or steer.

4 Conclusions and future directions of work

This project proved to be a little bit challenging for me, but the rewarding experience allowing us to gain a breadth of knowledge in the area of designing and implementing. Though the prototype would need to be redesigned for mass use, it would certainly be a achievable and economical solution to anyone's safety and admission needs.

The future work of robot vehicle can only be limited by your imagination. This highly customizable chassis was designed so that the user can easily add on parts. By adding a GPS module, infrared or ultrasonic sensor an autonomous mode can be implemented and the robot will drive around collecting data. Also, we have the possibility to add a video camera and implement live video images feed to the Android Mobile so the robot can be transformed into a surveillance vehicle.

Embedded applications controlled by a mobile device could find applicability in other fields like: Military Services; Underwater Exploration; Investigating Hazardous Environments, or even Space Exploration.

References

- [1] [Arduino Uno Board](#)
- [2] [Android app author site](#)
- [3] ["Using Atmel Studio for Arduino development"](#)
- [4] [Wikipedia main page](#)
- [5] [Sparkfun.com datasheets Of L298 H Bridge pdf](#)
- [6] [Sparkfun.com datasheets Of Wireless Bluetooth RN42 pdf](#)
- [7] [Olimex.com Gear Motors MG-6-120](#)
- [8] [Allon Robots Types of robots](#)
- [9] McGraw-Hill. "Programming Arduino Getting Started with Sketches". Nov 8, 2011.

TOMA MARIUS
University "Lucian Blaga" of Sibiu
Faculty of Sciences
ROMANIA
E-mail: kid.morvius@gmail.com

ChessTime – a simple chess clock for Android

Dimitar Trifonov
Coordinator Professor: Katalina Grigorova

Abstract

Chess is a classic 2-player game popular all around the world and played by people of all ages. Nowadays you can easily play chess online, but playing in person, on a real chess board, is just a lot more fun. Unfortunately, while you may have a chess board that you can play on, very few people own a chess clock as well, which is mandatory for determining how long a game should be. ChessTime solves that problem in a convenient way for both players.

1 Introduction

Everybody likes playing chess. It's simply a fun game, which engages your mind, improves your critical thinking [1] and makes time fly by pretty fast. Nowadays it's extremely easy to just go on the Internet and find an opponent to play against online. But before the Internet, believe it or not, people actually sat down in front of a real chess board and they played for hours. All professional games are played on real chess boards too.

Although playing in person is generally considered a far superior experience, it has its pros and cons. For example, most official chess tournaments have a time limit for each player, which adds additional challenge to the game. Because of this, the games are played with a chess clock, which you tap whenever you're done with your move and then the other player's timer starts running down. When one of the players' time is up, the game is over. The problem is that very few people own a chess clock of their own and as a result, their games' length is random and chaotic. Furthermore, the lack of a chess clock often causes the awkward question "Uh, did you move?", as your opponent stares at you in utter disbelief after waiting for you for 5 minutes straight to make a move.

When you're out and playing against a friend, you may not wish to have a timer, because you just want to savor the experience. Unfortunately, this is not always the case and sometimes you just don't have all day to play or wait for your overly analytical friend to make their first move. You could use your phone's alarm features, but it's not very convenient and requires additional mental focus to keep track of what's going on.

In order to avoid these problems, you can simply use ChessTime, which is pretty much a chess clock in the form of a smartphone app. What makes ChessTime unique is its utter simplicity. While most other apps of this nature try to "capture" the feeling of a real chess clock by making the app's layout resemble a chess clock or by adding unnecessary options such as time penalties,

they're forgetting that sometimes the user wants just one thing - a portable chess clock that works. Furthermore, ChessTime is an open-source app, which means that anyone can modify and adjust it to meet their own personal needs. As an added bonus, in case you want to track how many moves were played in a particular game, the app allows you to do that as well (to make the game a bit more “official” and help you out if you wish to submit it to a chess game database).

The application is written in Java (functionality) and XML (layout/design) and makes use of the Android SDK. You need a smartphone running Android 4.0.3 (IceCreamSandwich) or later to

2 How the app works

2.1 Main functionality

ChessTime’s charm is in its simplicity. There’s only one screen, which includes only the bare essentials, presented in a convenient and user-friendly manner.

In a nutshell, ChessTime allows you to pick a standard length of the game you’re about to play (that is, the amount of time each player gets for the duration of the game). Then you can either start the game directly or also enable the move counter (if you wish to keep track of how many moves were played in a game) and “Flip mode”, which basically rotates the timer, so that it’s more accessible to the current player (this is explained in more detail below).

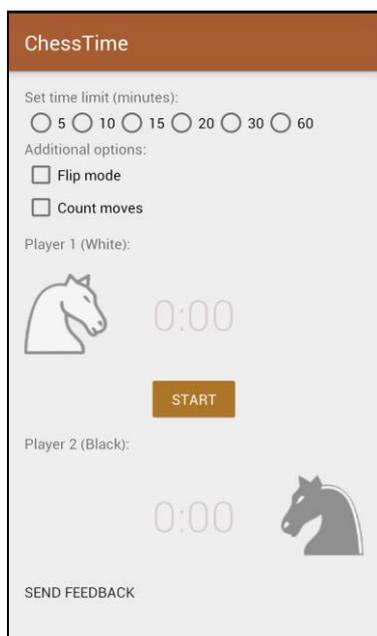


Fig. 1. The main (and only) screen of the app

2.1.1 Setting the time limit

When the app loads, you can set the time limit for each player. You can either go for a “blitz” game (also known as speed chess or “bullet chess” [2]) with a really short duration (5 minutes per player) or you can go for a normal game, which gives each person 60 minutes. The app does not support custom time limits, as it is primarily intended for relatively short games between friends. If you try to start a game without setting the time limit, you will get a notification prompting you to first choose a time limit. Once the time limit has been set and the Start button has been clicked, you cannot alter it until the game is over or until you restart the app.

2.1.2 Flip mode

The so-called “flip mode” allows you to literally flip the timer, so that it is conveniently placed for the current player. To better explain this, here is an example of how Black would see the timer with “flip mode” enabled and with “flip mode” disabled, provided that the phone is placed in a vertical position near the chess board and facing White. This feature eliminates the need to place the phone in a horizontal position or to constantly rotate the phone in order to make it convenient for the other player.

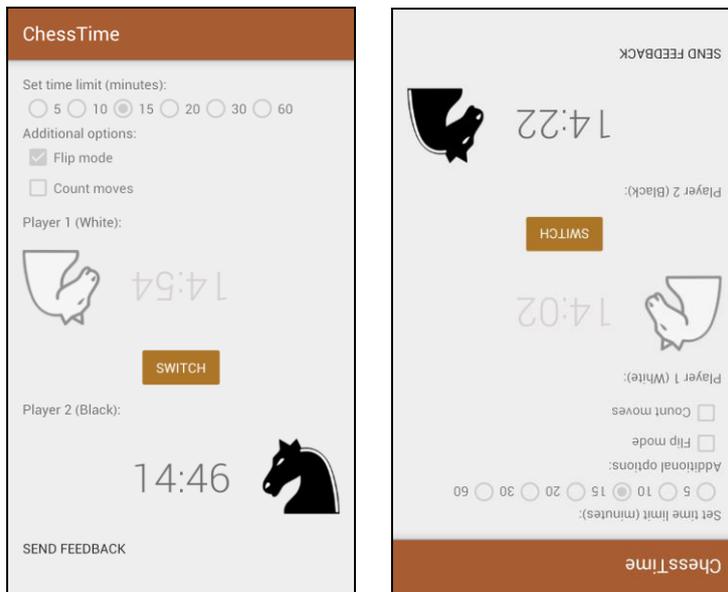


Fig. 2. How Black sees the screen with “flip mode” enabled/ disabled when the phone is in a vertical position and placed in front of White.

2.1.3 Counting moves

Counting moves is not a necessary part of playing chess, but it’s a useful feature for people that wish to keep track of how many moves their games take to complete on average or for people that submit their games at online chess databases. All official chess games are usually recorded by writing down the actual moves that were played and naturally the number of moves as a whole [3]. Plus, it gives the winner a sweeter taste of victory (“Hey, I beat that guy in only X moves!”). Just a small additional option that is supposed to be the cherry on top of the cake and not the cake itself.

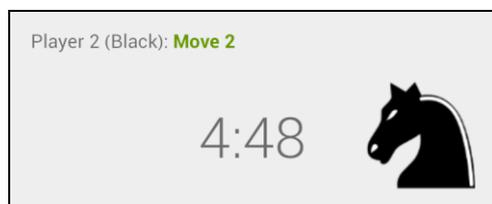


Fig. 3. How the app changes with the “Count moves” option enabled

2.1.4 Starting the actual game

Once the countdown timer has started, you can’t pause it or make adjustments to the time limit or the additional options, unless you close the app and re-launch it. The timer always starts

on White's side, as according to the official chess rules [4], White goes first every time. Besides the countdown timer, the transparency of each knight is adjusted to make it more noticeable by the human eye whose move it currently is.

2.1.5 Game over

When one of the players' time is over, the timer gets replaced by the text "Time's up!", which indicates that the game is over and as a result, you can once again set the time limit, as well as enable/disable any of the additional options after clicking the NEW GAME button.

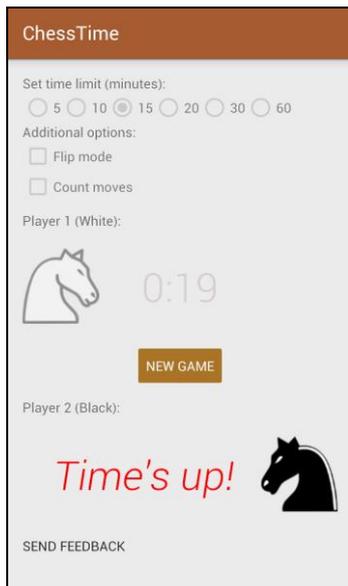


Fig. 4. How the app looks like when a certain player's time is up

2.1.6 Quitting the app

Most Android apps can be closed with one click of the Back button, but ChessTime is programmed to close after two clicks. The reason for this is that usually whenever you're playing a chess game, you don't wish to be disturbed and you may accidentally close the app by clicking the wrong button (especially if you're starting the app for the first time). As a result, whenever you click the Back button, you'll get a notification prompting you to click the button once more to confirm that you really wish to exit and the click wasn't an innocent mistake.

2.1.7 Sending feedback to the author

ChessTime offers a feedback option, in case the players want to suggest a feature for a future version of the app. Whenever the player clicks on the SEND FEEDBACK text, he'll be prompted to choose his preferred e-mail client and when that is done, the e-mail address of the author is automatically added to the "Send to" field and the subject of the e-mail is set to "ChessTime Feedback" (to save the player some time and effort).

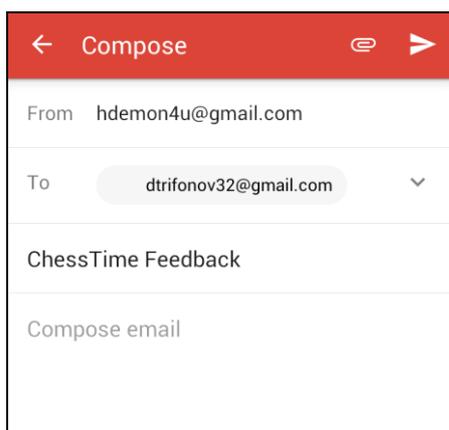


Fig. 5. Sending an e-mail through the Gmail app

2.2 Usage of the Android SDK, Java and XML

While ChessTime is a rather primitive mobile app, it makes use of some of the Android SDK's fundamental classes and widgets, such as TextView, ImageView, RadioButton, CheckBox, Button, Toast, Intent and so on.

The layout and design are done entirely via XML. Both LinearLayout and RelativeLayout are used, as well as ScrollView (to add support for devices with smaller screens).

The app is open-source and its code can be accessed by anyone at its repository on GitHub [here](#).

3 Conclusion

The designed application holds convenient and easy to use features. Chess lovers have at their disposal a simple, yet effective chess clock in the form of an app, which they can take literally anywhere with minimal effort.

The author plans expanding the capabilities in the following areas:

- ability to set a custom time limit;
- ability to log information about previously played games;
- ability to set names for the players and count number of won/lost games for each player.

References

- [1] EducationWorld, *Strategies that Work: Chess and Critical Thinking*, http://www.educationworld.com/a_curr/strategy/strategy021.shtml
- [2] Scimia, E. *Rapid chess*, 2016, <http://chess.about.com/od/chessvariants/a/Rapid-Chess.htm>
- [3] The Chess House, *How to Read and Write Algebraic Chess Notation*, http://www.chesshouse.com/how_to_read_and_write_chess_notation_a/166.htm
- [4] FIDE, *Laws of Chess*, <http://www.fide.com/fide/handbook.html?id=171&view=article>

Dimitar Trifonov
University of Ruse
Department of Informatics and
Information Technologies
8, Studentska str., Ruse 7017
BULGARIA
dtrifonov32@gmail.com

Parameters tuning for support vector machines

Eva Tuba

Coordinator Professor: Milan Tuba

Abstract

Machine learning is one of the important field in computer science. It represents a type of artificial intelligence and it is used in many different applications. One of the tasks of machine learning is classification. Classification is widely used for applications in many scientific fields and represents very active research field. As result, different algorithms were proposed for solving the classification problem. One of the latest and widely used classifier is support vector machine (SVM). In order to get satisfying accuracy of classification by SVM a few parameters need to be tuned. In this paper we experiment with values of parameters for different data.

1 Introduction

One of very active research field in computer science is machine learning. It can be defined as ability of computers to learn without being explicitly programmed. Machine learning is usually part of applications such as data mining [1], natural language processing [2], image recognition [3], etc. Machine learning can be categorized in different ways. One categorization refers on learning algorithms that can be used, unsupervised learning, supervised learning and reinforcement learning. Another categorization of machine learning algorithms depends on output of machine learning system. Based on output, machine learning can be categorized as one of the following category: classification, regression, clustering, density estimation or dimensionality reduction.

Classification is a task where inputs are divided into two or more classes and learning model assigns unknown input to one of the classes. It is use in medicine for tumor classification [4], electromyography signal classification [5], text recognition [6], face recognition [7], etc. Many different algorithms and methods were developed for classification. Some of the well know classifiers are decision trees, learning set of rules, Naive Bayes classifier, Bayesian Network, etc. Neural network also can be used for solving the classification problem. One of the latest and widely used classifier is support vector machine.

Support vector machines originally were binary linear classifiers. The main idea is to find optimal hyperplane that separates data from different classes. In order to find that hyperplane, soft margin's parameter C must be determined. Even though SVM was made as binary classifier, by introducing the kernel trick it can perform non-linear classification. For this, different kernel function can be used and each of them has some parameter. In this paper, we will experiment with Gaussian radial basis function (RBF). Parameter of RBF is γ . Parameter γ has an important influence in learning performance since it controls the linearity of the induced SVM, while the parameter C trades off misclassification of training examples against simplicity of the decision surface [8]. Determination of these two parameters is very important step in building a good SVM model. In this paper, we will experiment with different pair of values for (C, γ) .

In the next section a brief description of machine learning, especially classification task is given. In Section 3, linear SVM is described and parameter C is examine. In Section 4 kernel trick is

introduced along with Gaussian radial basis function and its parameter γ is described. Finally, at the end experimental results and conclusion are given.

2 Machine learning

As mentioned before, machine learning represents systems that automatically learn programs from data. In the last decades the use of machine learning was widely spread in computer science and more. Some examples of use of machine learning are: web search [9], spam filters [10], credit scoring [11], fraud detection [12] and many more.

Wide need for machine learning algorithms inspired researchers to develop many different approaches. For different purposes different learning algorithms are used (supervised, unsupervised and reinforcement learning). In the case of supervised learning, data in training set are given with the desire output. The task is to learn general rule that maps input data with the output. This learning algorithm is used when the outputs are clearly defined and they are known for some inputs. On the other hand, input data for unsupervised learning do not have corresponding labels. The goal of learning algorithm is to find some structure in input data or better to say, to find some hidden pattern in them. Reinforcement learning is used in the cases when a computer program interact with a dynamic environment in which must perform a certain goal, but without telling whether it has come close to goal. This is used for applications such as driving a vehicle, learning to play a game by playing, etc.

Beside learning algorithm, in machine learning is very important to know what is supposed to be learned. Many machine learning techniques require information about the probabilities of various events involving the data. This information usually is not available and must be estimated from available data. For this, several density estimate technique are proposed such as histogram density estimate, Bayesian approach, kernel density estimate, k -nearest neighbor and others. Dimensional reduction refers to mapping of data to a lower dimensional space, but the uninformative variance in the data is discarded. This technique is often used for data visualization, to represent image that is in high dimensional image in two dimensional space. Regression is supervised problem where output data are continuous. Regression is used for weather prediction [13], stock market prediction [14] and others. On the other hand, classification and clustering have discrete outputs. Clustering is unsupervised learning algorithm that input data organizes into classes and classification is supervised algorithm where data can be labeled with one of earlier determined classes.

2.1 Classification

In machine learning, classification is the problem of determination to which class a new instance belongs. Based on instances from training set whose labels are known, learning model is built and used for predicting the labels of new, unknown instances.

Each instance is represented with set of quantifiable properties named features. All instances must be represented with the same set of features. Features can be binary, continuous or categorical. Classifier refers to the algorithm that implements classification. Also, it can refers to the mathematical function that maps input to the classes.

In [15], classification techniques were categorized as logical (symbolic), perception based techniques and statistics. Decision tree is one of the logical techniques where classification of instances is done by sorting them based on feature values. Another logical technique is learning set of rules, where each class is represented by logical expression. Single layer perception is used for linearly separable data. In this techniques, output is defined by sum of weighted input's features. If data are not linearly separable, that multilayer perception is used. The most popular and well known multilayer perception is artificial neural network. Last category, statistical learning algorithms are characterized by having an explicit underlying probability model. Instead of the simple classification, these models determine probability that the instance belongs in every class. Linear discriminant analysis and Fisher's linear discriminant are the simplest methods used in machine learning. These techniques find the linear combination of features that best separates classes. Another used technique is maximum entropy. The idea is that if nothing is known, distribution should be as uniform as possible, or better to say, to have maximum

entropy. According to [15], Bayesian network is the most well known statistical learning technique. It is a graphical model for probability relationships among a set of features. Support vector machine (SVM) is one of the newest classifier and in the next section the brief description of it is given.

3 Support vector machine

Support vector machine (SVM) is one of the latest and widely used non-probabilistic binary classifiers. It was proposed by Vapnik [16]. The main idea of the SVM is to find hyperplane that separates instances from two different classes. All instances are represented as points in n -dimensional space and each one has corresponding class label where n is the number of features that represent instances. SVM model is built based on instances from the training set. That model is used for classification of new examples.

Formally, SVM is defined as follow. Assume that training set consists linearly separable instances $x_i \in R^d$, where d is number of features, $i = 0, 1, \dots, n$ and n is number of instances. Each instance has corresponding label $y_i \in -1, 1$. Hyperplane that separates instances from different classes is defined with the following equation:

$$w \cdot x + b = 0, \quad w \in R^d, \quad b \in R \quad (1)$$

where w is the normal vector to hyperplane and $\frac{b}{\|w\|}$ represents the offset of the hyperplane from the origin along the normal vector w . In Fig. 1 it can be seen that more hyperplanes satisfy condition to separate instances from different classes.

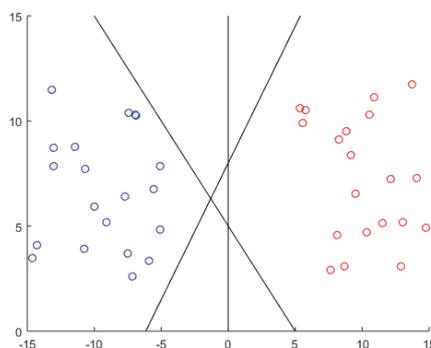


Figure 1: Examples of hyperplanes that separates instances from different classes

Optimal hyperplane lies in the middle of the margin. Margin is the space between two hyperplanes that separates instances and the distance between them is as large as possible. Margin borders are defined with the next equation:

$$\begin{aligned} (w \cdot x_i + b) &= 1, \\ (w \cdot x_i + b) &= -1 \end{aligned} \quad (2)$$

In Fig. 2 two interrupted lines represents borders of the margin and in the middle optimal hyperplane is shown with full line. All instances must be on the right side of margin, which is formally written with the following expression:

$$y_i(w \cdot x_i + b) \geq 1 \quad \text{for } 1 \leq i \leq n. \quad (3)$$

Eq. 3 defined so-called hard margin. Distance between these two hyperplanes that define margin should be maximized. The distance between these two hyperplane is $\frac{2}{\|w\|}$, where $\|w\| = \sqrt{\sum_i w_i^2}$. With some

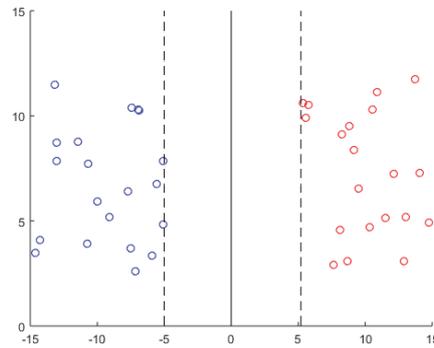


Figure 2: Optimal hyperplane for SVM

basic mathematic it can be concluded that optimal hyperplane is found by solving the next quadric programming problem:

$$\begin{aligned} & \min \frac{1}{2} \|w\|^2 \\ \text{s.t. } & y_i(w \cdot x_i + b) \geq 1, \quad i = 1, 2, \dots, n \end{aligned} \quad (4)$$

3.1 Soft margin

Previously described model work for linearly separable data. Problem with this model is that real data usually are not linearly separable because of a few outliers. Outliers are data that are significantly different from other instances from the same class. Also, it is not rare situation that huge amount of data is entered in computer by human. Human mistakes can also lead to set of data that can not be classified with described model. In order to solve that problem, instead of hard margin, soft margin is used. Soft margin allows that some instances be on the wrong side of hyperplane or to lie somewhere between the margin. To allow this, instead of Eq. 3 next equation is used:

$$y_i(w \cdot x_i + b) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0, \quad 1 \leq i \leq n \quad (5)$$

where ϵ_i is slack variable. With this parameter, some slackness is allowed and it is represented with ϵ . It can be noticed that if $0 < \epsilon \leq 1$, instances lie can be somewhere between the margin, but on the right side of hyperplane, while for $\epsilon > 1$ some data will be misclassified.

In this case, instead of Eq. 5, the next quadric programming problem should be solved:

$$\begin{aligned} & \min \frac{1}{2} \|w\|^2 + C \sum \epsilon_i \\ \text{s.t. } & y_i(w \cdot x_i + b) \geq 1 - \epsilon_i, \quad i = 1, 2, \dots, n \\ & \epsilon_i \geq 0 \quad i = 1, 2, \dots, n \end{aligned} \quad (6)$$

In this expression, C represents a free parameter that controls the relative importance of minimizing the norm of w or better to say maximizing the margin and controls that each data point satisfies margin constraint. For relatively small values of free parameter C , the cost for instances inside the margin is not large. This means that very wide margin will be made and many instances will fall into it. On the other hand, if C is close to infinity the cost of data points that violating margin constraint is large, so the model will be close to the model with the hard margin.

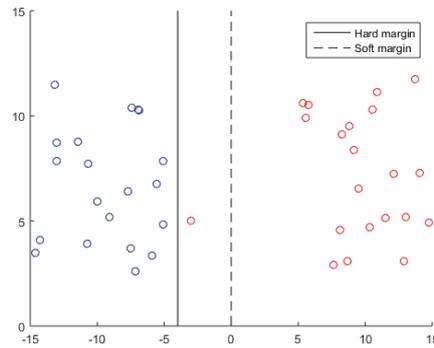


Figure 3: Soft margin and hard margin for data with outliers

4 Kernel function

Previous section explains SVM for linearly separable data. Not many real world data has that characteristic. In order to SVM can be useful for dealing with classification problem it must be able to classify non-linearly separable data. This can be done by introducing the kernel trick. Kernel trick transform input data so that dataset become linearly separable. By kernel function input dataset is transformed into higher dimensional space where they are linearly separable. To train data, actual data points are not needed but only dot product of $\langle x_i^T, x_i \rangle$. Kernel trick replace $\langle x, y \rangle$ with $k(x, y)$ where k is kernel function. Kernel function correspond inner product in some higher dimensional space. Any function that satisfy Mercer's theorem. Mercer's theorem is a representation of a symmetric positive-definite function on a square as a sum of a convergent sequence of product functions.

For SVM, several kernel function are usually used such as polynomial, sigmoid and Gaussian radial basis function (RBF). In this paper we will consider only RBF. Radial basis function is defined with the following equation:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \quad \gamma > 0 \quad (7)$$

Gaussian kernel function map input data into infinite dimensional space. Because of that it has infinite choice of dimensions to split data with, thus every input dataset with appropriate chosen γ parameter can become linearly separable. Fig. 4 is an example of classification of non-linearly separable data. Data points with the black star are support vectors.

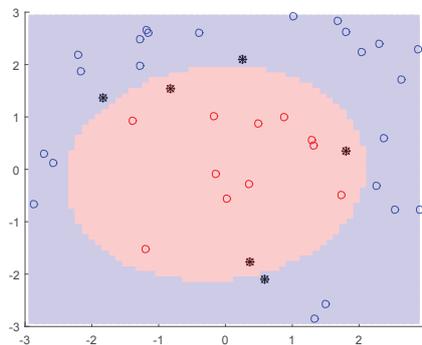


Figure 4: Decision boundary and support vectors

Finding an optimal value for parameter γ is crucial for building a good SVM model that will provide

satisfying classification accuracy. If too large value for parameter γ is chosen, influence of the support vectors only includes the support vector itself and it can lead to overfitting. Too small values for γ will make the model too constrained so the complexity of the data will not be seen. Influence of any selected support vector would include the whole training set.

For building a good SVM model that will provide satisfying accuracy, determination of the pair of values for parameters (C, γ) is crucial. In the next section, we will examine a few examples with different values of parameters.

5 Experimental results

All experiments were done in Matlab R2015a. For building the SVM models, LIBSVM was used [17]. We experiment with three different data sets. These data sets are shown in Fig. 5. First data set represents instances that are organized in the circle. One class is inside of circle, and the other class is outside that circle. Second data set can be seen as almost linearly separable with outliers. Third data set will be consider more detailed later in this section.

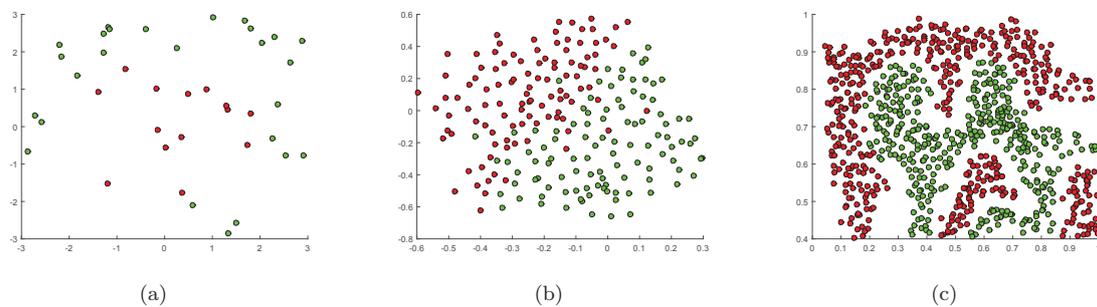


Figure 5: Data sets used for experiments

As mentioned before, SVM model for non-linearly separable input datasets that uses radial basis function as kernel function has two parameters that need to be tuned. Both C and γ are scale parameters, so optimal values should be searched in a logarithmic scale. Based on previous definition of parameter C , it can be assumed that too small and too large value of it will not give a satisfying results. Also, too large value for parameter γ will lead to every data point is support vector for it self. In Fig. 6, support vectors of models for our three examples with too small C or too large γ are shown. Support vectors are marked with black stars. As it can be seen, almost every input data is support vector and no decision boundaries were found.

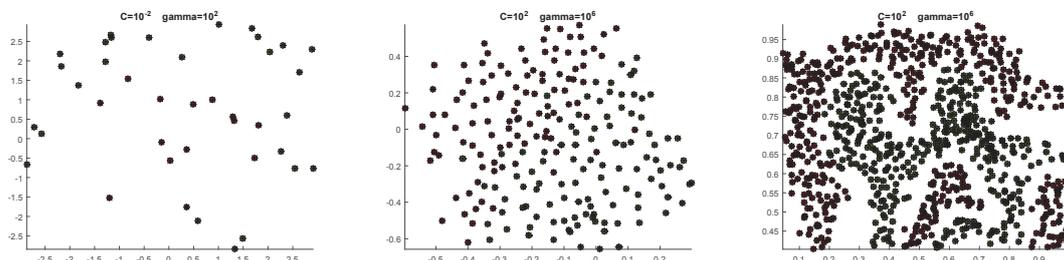


Figure 6: Classification with bad parameters for three different examples

On the other hand, it should be consider pair of values for (C, γ) instead of looking parameters individually. For the first dataset, the best separation of classes were with $(C, \gamma) = (10^6, 10^{-2})$. Individually it can be consider that C is too large and γ is too small, but for this example, this is

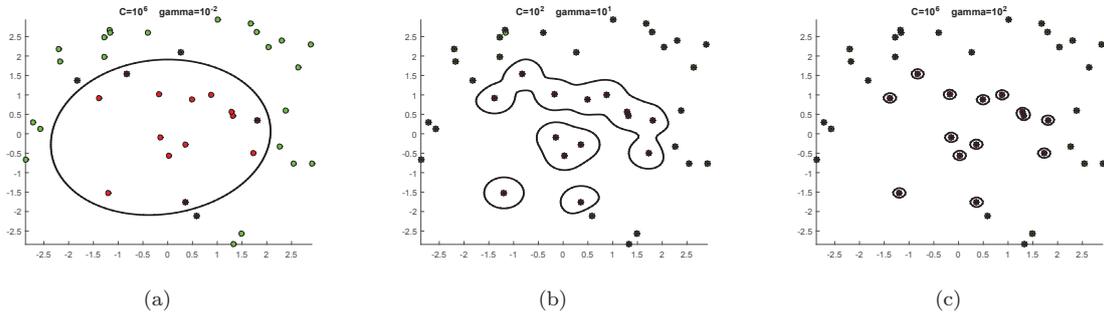


Figure 7: Classification of the first example with different (C, γ) (a) $(10^6, 10^{-2})$, (b) $(10^2, 10^1)$ and (c) $(10^6, 10^2)$

the best combination of values for these parameters. With parameters $(C, \gamma) = (10^6, 10^2)$ model was over fitted. Also, $(10^2, 10^1)$ is not a good combination for parameters for this dataset. With these values, only near instances were grouped. Decision boundaries for these parameter combinations are shown in Fig. 7.

For the second dataset, the best classification were obtained with $(C, \gamma) = (10^2, 10^1)$, while for $(C, \gamma) = (10^6, 10^2)$ model were correctly classified all training data points, which usually mean poor classification accuracy for test instances. In Fig. 8 are shown decision boundaries for the second dataset with mentioned parameters.

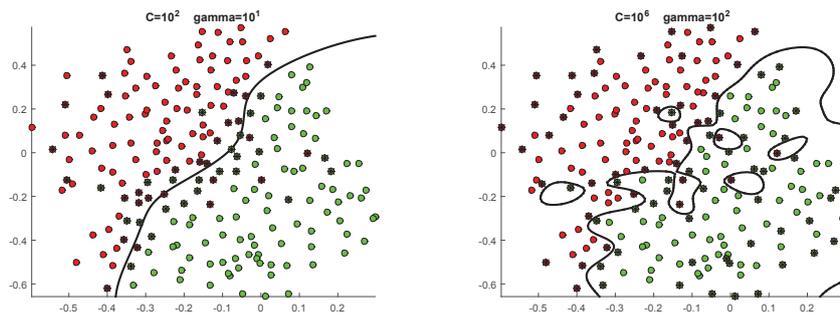


Figure 8: Classification of the second example with different (C, γ) (a) $(10^2, 10^1)$ and (b) $(10^6, 10^2)$

On the third example it can be best seen difference between models with different parameters. Fig. 9 shows decision boundaries with different pair of values for C and γ . If the γ is too small (in this example $\gamma = 10^{-2}$), parameter C can improve the models only a little bit. No complexity of data is considered. If $C = 10^2$, number of instances that can fall into the margin is large. With this pair of values, SVM model separate instances very poorly, almost linearly (Fig. 9(a)). If we increase parameter C to 10^6 , number of instances that can fall into the margin is reduced, so a little bit better classification is done (Fig. 9(b)). Model shown in Fig. 9(c), where $(C, \gamma) = (10^2, 10^1)$ has a good classification. By increasing the parameter γ , complexity of input data is taken in consider and relatively large value for C limit the number of instances inside of the margin. If we increase C up to 10^6 , we will reduce the number of instances in the margin space. Fig. 9(d) shows decision bounders when $(C, \gamma) = (6, 2)$. Compare to the previous example γ is also increased. This model perfectly separates instances from different classes. It can be noticed that on the parts with larger distances between instances from different classes, decision boulder goes right in the middle of that empty space. Also number of support vectors is smaller than in previous case. So with this parameters we have less support vectors and more accurate classification. In the first and the second datasets with this parameters we had overfitting, but in the third case, this parameter are the best choice for classification. It can be concluded that parameters have to be tuned for each dataset individually and for different datasets, different parameter values are needed.

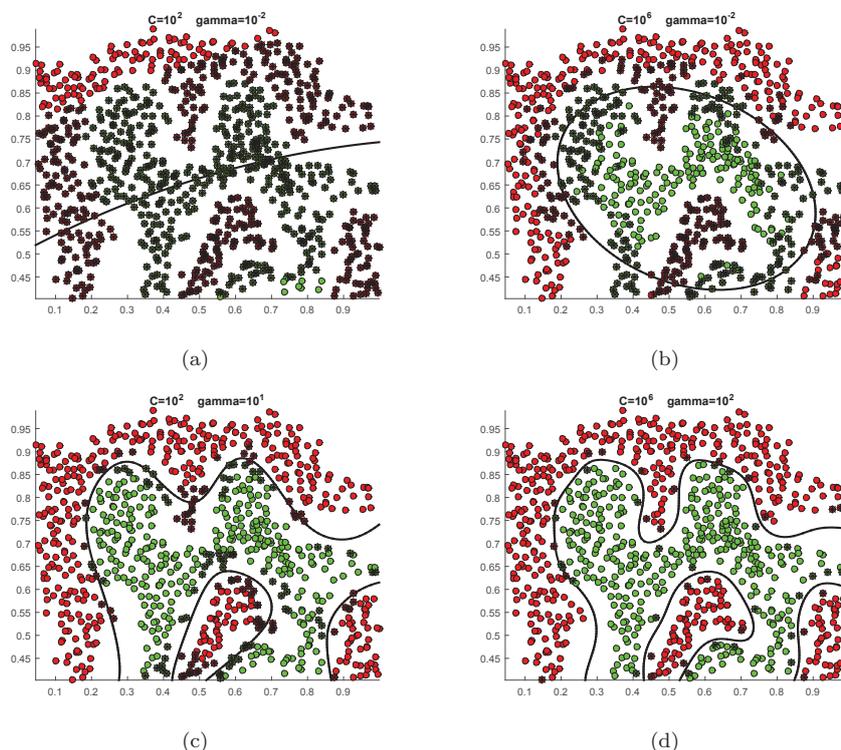


Figure 9: Classification of the third example with different (C, γ)

6 Conclusion

Classification is very important part of numerous applications. One of the widely used classifier is support vector machine. Success of classification done by SVM strongly depend on chosen parameters. In this paper accuracy of classification with different data and parameters were explored. It was proved that rather different parameters give the best model for different problems. The importance of parameters tuning is also shown. In the further work, automatic selection of parameters can be added. For parameter tuning some optimization algorithms can be used, such as swarm intelligence. Also, in this paper we focused on only one kernel function, so in the further work some other kernel function can be used.

References

- [1] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*. New York, NY, USA: Cambridge University Press, 2011.
- [2] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernandez, "Syntactic N-grams as machine learning features for natural language processing," *Expert Systems with Applications*, vol. 41, no. 3, pp. 853–860, 2014. *Methods and Applications of Artificial and Computational Intelligence*.
- [3] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 105–119, Jan 2010.
- [4] Y.-H. Liu, M. Muftah, T. Das, L. Bai, K. Robson, and D. Auer, "Classification of mr tumor images

- based on Gabor wavelet analysis,” *Journal of Medical and Biological Engineering*, vol. 32, no. 1, pp. 22–28, 2012.
- [5] A. Subasi, “Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders,” *Computers in biology and medicine*, vol. 43, no. 5, pp. 576–586, 2013.
- [6] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, “Improving offline handwritten text recognition with hybrid HMM/ANN models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 767–779, April 2011.
- [7] J.-F. Connolly, E. Granger, and R. Sabourin, “An adaptive classification system for video-based face recognition,” *Information Sciences*, vol. 192, pp. 50–70, June 2012. Swarm Intelligence and Its Applications.
- [8] T. A. Gomes, R. B. Prudêncio, C. Soares, A. L. Rossi, and A. Carvalho, “Combining meta-learning and search techniques to select parameters for support vector machines,” *Neurocomputing*, vol. 75, no. 1, pp. 3–13, 2012.
- [9] R. L. Santos, C. Macdonald, and I. Ounis, “Selectively diversifying web search results,” in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10*, (New York, NY, USA), pp. 1179–1188, ACM, 2010.
- [10] A. H. Wang, “Detecting spam bots in online social networking sites: A machine learning approach,” in *Data and Applications Security and Privacy XXIV: 24th Annual IFIP WG 11.3 Working Conference, Rome, Italy, June 21-23, 2010. Proceedings* (S. Foresti and S. Jajodia, eds.), (Berlin, Heidelberg), pp. 335–342, Springer Berlin Heidelberg, 2010.
- [11] C.-F. Tsai and M.-L. Chen, “Credit rating by hybrid machine learning techniques,” *Applied Soft Computing*, vol. 10, no. 2, pp. 374–380, 2010.
- [12] J. Perols, “Financial statement fraud detection: An analysis of statistical and machine learning algorithms,” *Auditing: A Journal of Practice and Theory*, vol. 30, no. 2, pp. 19–50, 2011.
- [13] T. L. Thorarinsdottir and T. Gneiting, “Probabilistic forecasts of wind speed: ensemble model output statistics by using heteroscedastic censored regression,” *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 173, no. 2, pp. 371–388, 2010.
- [14] C.-Y. Yeh, C.-W. Huang, and S.-J. Lee, “A multiple-kernel support vector regression approach for stock market price forecasting,” *Expert Systems with Applications*, vol. 38, no. 3, pp. 2177–2186, 2011.
- [15] S. B. Kotsiantis, “Supervised machine learning: A review of classification techniques,” in *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real World AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, (Amsterdam, The Netherlands), pp. 3–24, IOS Press, 2007.
- [16] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [17] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, May 2011.

Eva Tuba
John Naisbitt University
Faculty of Computer Science
Bulevar umetnosti 29
SERBIA
E-mail: etuba@acm.org

Optimization of Otsu's method for multilevel image thresholding by grid search

Una Tuba

Coordinator Professor: Milan Tuba

Abstract

Digital images are widely used in every day life by many people. With increasing the use of digital images, different needs for their adjustment and processing was developed. Image thresholding is one of the very important algorithm used in computer science. Different algorithms for image thresholding was developed. One of the widely used is Otsu's method. Because of computational complexity , in order to perform multilevel image thresholding, some optimization is needed. In this paper we experiment with grid search for optimization of Otsu's method. Execution time can be significantly reduced with grid search.

1 Introduction

Nowadays, digital images are part of many different applications. Widespread use of digital images has contributed to the development of different research topics for image processing and analysis [1]. One of the very important task in image processing is thresholding. Thresholding refers to the problem of determination threshold values that gray scale image transforms to image with less gray levels. Many applications use thresholding to prepare image for further use and analysis. Thresholding is used for image segmentation [2], [3], [4] and image denoising [5], [6]. These algorithms, especially image segmentation are part of many application such as character recognition [7], map processing [8], medical image analysis [9], [10], etc.

Because thresholding is such important task many different algorithms were developed. Generally speaking, thresholding algorithms can be classified as parametric and nonparametric methods. Parametric approaches consider probability density function for gray-level distribution of each class. These methods attempt to determine the parameters of distribution that will best fit the given histogram data. Determination of the parameters usually needs to solve nonlinear optimization problems which is computationally expensive and time consuming [11]. Because of that nonparametric were introduced. Nonparametric methods for thresholding optimize some discriminating criteria. One of the most popular optimization criteria is entropy. Firstly, entropy of intensity histogram was introduced as a new method for thresholding [12]. This algorithm was improved later by Kapur and nowadays it is one of the most popular method for thresholding [13]. Beside Kapur's method, Shanon's concept of entropy, Renyi's entropy and others are widely used.

Important group of parametric methods for thresholding is clustering-based methods. The main idea is to cluster gray values into several classes and that exists similarity of gray values in same class, but also that there is dissimilarity between classes. One of the main method from this group is Otsu's method [14]. Exhaustive search is used for fonding the optimal threshold value by maximizing dissimilarity between classes. Otsu's method is shown to be very good and effective for bilevel image thresholding, but for

multilevel thresholding is very computationally expensive. In order to reduce this cost some optimization is needed. Instead of exhaustive search through all search space, grid search can be used. Grid search reduce search space for optimal threshold value by dividing it into equidistant grid.

In this paper we will experiment with different size of grids for Otsu's method optimization with grid search algorithm.

After Introduction, thresholding problem as well as multilevel thresholding are described. In Section 3 Otsu's method with grid search for optimization is presented. Section 4 shows experimental results, while final section concludes this paper.

2 Thresholding

Thresholding is usually used for image segmentation. In gray scale digital images intensities of pixels that belongs to some object are often different than intensities of pixels of background. It can be also used for differentiation of objects with different pixels' intensity. Simplest thresholding transform gray scale image into binary image. It is called bi-level thresholding. Bilevel thresholding is also used for printing gray images on black and white devices. Beside bi-level thresholding, multilevel image thresholding is used.

In bi-level thresholding one threshold value is needed. All pixels that have intensity higher then threshold value become white and the rest of pixels become black. Determination of threshold value is very important task and it can be done by looking at an intensity histogram. Intensity histogram is a graph that represents number of pixels with same value of intensity. For 8-bit gray scale images there are 256 different intensities and the histogram will show distribution of pixels amongst those gray scale values. Intensity histogram is a one-dimensional vector h of length K where K is number of pixels on image. Number of black pixels (intensity value is 0) is represented with $h(0)$ and number of white pixels is $h(L - 1)$, where L is number of gray levels. On Fig. 1 is shown image *Lena* that is standard image used in image processing and its histogram.

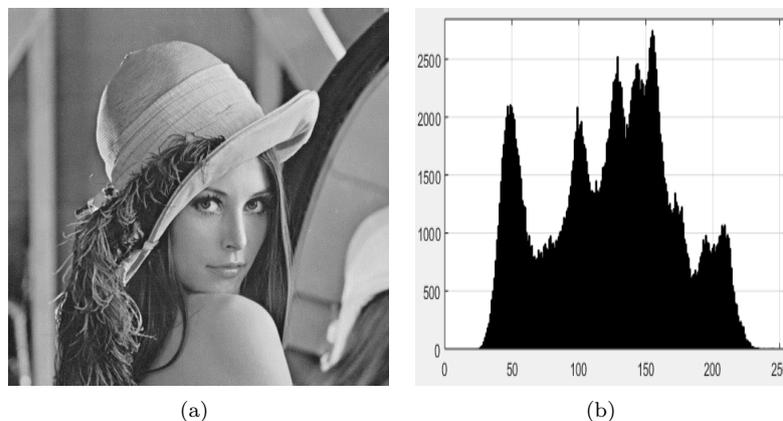


Figure 1: Example of intensity histogram (a) gray scale image and (b) corresponding intensity histogram

The main function of intensity histogram is to provide statistical informations about the image. Intensity histogram contains no information about the position of pixels, so as it can be seen on Fig. 2 different images can have the same histogram. If the images contain same number of pixels with specified intensity, histogram will be the same.

Optimal value for threshold can be visually determined by looking into intensity histogram. In some images different intensities of gray levels are grouped (for example, background is light, while objects are dark or vice versa). In cases like this, on histogram will appear two peaks.

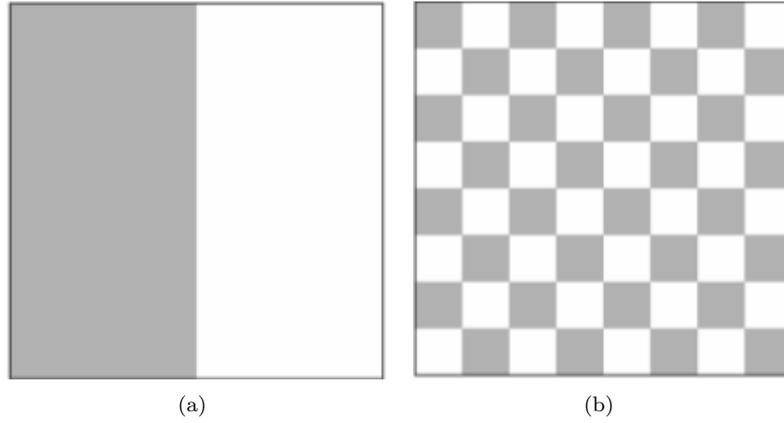


Figure 2: Example of different images that have same intensity histogram

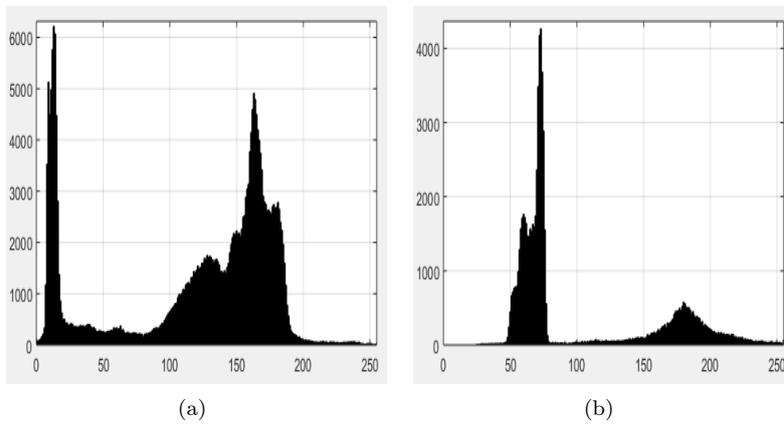


Figure 3: Examples of intensity histograms with clearly separated dark and light parts

2.1 Multilevel thresholding

While bilevel thresholding divide gray level into two classes, multilevel thresholding is a process that divide a gray levels into several disjunct classes. In this case, pixels that belong to the same class are getting the same shade of gray from interval (t_i, t_{i+1}) , where t_i is i^{th} threshold value. Selection or computing of multilevel thresholds is very important for image segmentation, because good segmentation is mainly result of adequate determination of thresholds.

For multilevel thresholding several threshold values are needed. This process is useful for image segmentation of images that have complex background [15]. If I is gray scale image with set of pixels A , multilevel image thresholding is a method that with n thresholds values divides set A into $n + 1$ disjunct subsets A_0, A_1, \dots, A_n so it satisfy following conditions:

$$\begin{aligned}
 A_0 &= \{x : 0 \leq f(x) < \alpha_0\} \\
 A_1 &= \{x : \alpha_0 \leq f(x) < \alpha_1\} \\
 &\vdots \\
 A_n &= \{x : \alpha_{n-1} \leq f(x) < L - 1\}
 \end{aligned} \tag{1}$$

where $x = (x_1, y_1)$ is pixel represented with coordinates (x_1, y_1) in Cartesian coordinate system and $f(x), 0 \leq f(x) \leq 255$ represents intensity level of pixel x .

Multilevel thresholding methods compute optimal threshold values $\alpha_0, \alpha_1, \dots, \alpha_{n-1}$, where disjunct subsets $A_j, j = 0, 1, \dots, n$ represent disjunct regions of the image and union of them is actually original image I . Determination of optimal thresholds can be done with optimization of some functional criteria defined on intensity histogram. As mentioned before, one of the wide used method for thresholding is Otsu's criteria of maximal between class variance.

3 Otsu's method with grid search

Otsu's method is one of the most popular method for thresholding. It was named after its author and it was proposed in 1979 [14]. It is method based on discriminant analysis and it belongs into group of clustering-based techniques. The main idea of clustering-based methods is to gray values divides into several classes with condition that gray values in the same class have similarity and to have dissimilarity of gray values between classes.

Otsu's method determines optimal threshold by dividing gray values into classes so that the spread or inter-class variance is minimal. Equivalent to this is maximization of between-class variance. Formally, Otsu's method can be defined as follow. Assume that I is $n \times m$ image that is represented with L gray levels. The probabilities of pixels at level i can be calculated from intensity histogram. If N_i is the number of pixels with gray level i , probability p_i is defined as follow:

$$p_i = \frac{N_i}{n * m} \quad i = 0, 1, \dots, L - 1 \quad (2)$$

where $p_i > 0$ and $p_0 + p_1 + \dots + p_{L-1} = 1$. Cumulative probabilities for classes A_i can be calculated by next equation:

$$w_0 = \sum_{i=0}^{t_0-1} p_i, \quad w_1 = \sum_{i=t_0}^{t_1-1} p_i, \quad \dots, \quad w_k = \sum_{i=t_{k-1}}^{L-1} p_i \quad (3)$$

where t_j represent thresholds that separate classes. It can be noticed that in case of bilevel thresholding k is 1 and for multilevel thresholding, $k > 1$. For $k + 1$ classes $A_i, i = 0, 1, \dots, k$ the goal is to maximize between-class variance that is calculated by following expression:

$$f(t_0, t_1, \dots, t_{k-1}) = \sum_{i=0}^k \sigma_i \quad (4)$$

where the sigma function is variance of the class defined by next expression:

$$\begin{aligned} \sigma_0 &= w_0 \left(\sum_{i=0}^{t_0-1} \frac{ip_i}{w_0} - \sum_{i=0}^{L-1} ip_i \right)^2 \\ \sigma_1 &= w_1 \left(\sum_{i=t_0}^{t_1-1} \frac{ip_i}{w_1} - \sum_{i=0}^{L-1} ip_i \right)^2 \\ &\vdots \\ \sigma_k &= w_k \left(\sum_{i=t_{k-1}}^{L-1} \frac{ip_i}{w_k} - \sum_{i=0}^{L-1} ip_i \right)^2 \end{aligned} \quad (5)$$

Finding the optimal threshold with Otsu's method involves iterating through all possible threshold values, calculating between-class variance as it is defined in Eq. 4 and finding its maximal value.

In the case of bilevel thresholding and image with 256 gray levels it is necessary 256 times to compute between-class variance so it is not computationally expensive. On the other hand, multilevel thresholding,

where more than one optimal threshold values is needed, which is time consuming and expensive operation (complexity is $O((m + 1)(L - m)^m)$).

In order to reduce time of search for optimal threshold value grid search can be used. It is one of the most simplest method for optimization. The main idea is to bound the parameter space by upper and lower limit and search through grid (manually discretized search space) instead of search through all search space. Size of grid is determined with search range and step in it. By grid search number of calculation can be drastically reduced. If the image is 8-bit gray scale (256 gray levels) for finding threshold value it is necessary to calculate 256 variances with Otsu's method, one for each possible intensity level that can be threshold, but if grid with for example step size 8 is used, only 32 variance is needed to compute. If smaller step is used less benefits from this method is gained, but on the other hand, for larger step it is larger possibility to miss optimal threshold. In the next section will be experimented with different step size and different number of level thresholding.

4 Experimental results

In this paper we used 6 different test images for experiments. These are images that are used, namely, Barbara, Living room, Boats, Goldhill, Lena and Cameraman are of size (512 x 512). Barbara, Boats and Goldhill images are available at <https://homepages.cae.wisc.edu/~ece533/images/>. The Living room and Lena images were chosen from http://www.imageprocessingplace.com/root_files_V3/image_databases.htm. The Goldhill image can be found at https://ece.uwaterloo.ca/~z70wang/research/quality_index/demo.html. Original of all six test images are shown in Fig. 4. Application for testing the grid search on Otsu's method was written in Matlab R2015a.



Figure 4: Test images used in this paper

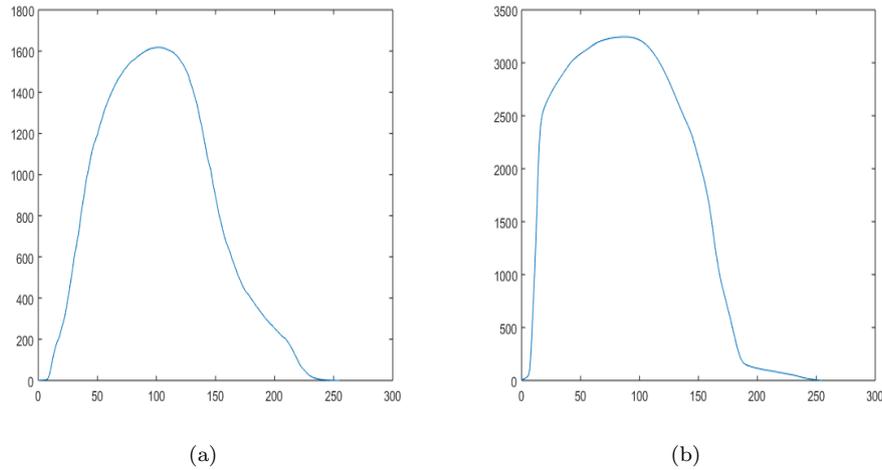


Figure 5: Between-class variance for all possible thresholds for (a) Boat and (b) Cameraman

In Section 3 is described Otsu’s method and its optimization with grid search. Basically, Otsu’s method use exhausting search and by checking all possible variants finds an optimal thresholds. By increasing number of levels computation complexity raise drastically. In order to reduce execution time we divided search space with different step for grid search. Table 2 shows execution time in seconds for Otsu’s method with different grid step size or better to say when less computation for finding optimal threshold is done, while Table 1 shows optimal thresholds calculated with exhausting search and corresponding computational time. In Table 1 was not shown results for 5-level thresholding, because computation time was too long.

If the grid size is 5, and all image has 256 gray levels for finding optimal threshold instead of 256 variance calculation only 51 calculation is needed. Optimal threshold can be different from value obtained with this method maximally for 4. Fig. 5 shows between-class variances for all possible threshold values. As it can be seen, peak of the function is pretty wide, so if optimal threshold is missed for only 4 will not make big difference, but it will significantly reduce computation time and complexity.

Table 1: EXECUTION TIME FOR CALCULATING MULTILEVEL THRESHOLDING WITHOUT DIFFERENT GRID SEARCH

	3-level threshold		4-level threshold	
	Opt.th.	time (s)	Opt.th.	time (s)
Barbara	81, 147	1.2368	74, 127, 176	133.7963
Living room	86, 145	1.2626	75, 123, 163	134.2582
Boats	92, 155	1.2808	72, 126, 179	134.3905
Gold hill	93, 161	1.2574	82, 126, 179	134.3919
Camera man	69, 144	1.2789	82, 126, 179	135.1578
Lena	92, 151	1.2665	80, 127, 171	133.7691

From Table 1 and Table 2 it can be seen that for image *Barbara* for 3-level threshold and grid step 5 computation time was reduced from 1.2369s to 0.0535s. Similar improvements was obtained for all benchmark images. Moreover, if the grid step is set to 20, time reduction is very high. In Fig. 6 are shown example of three images where 3-level image thresholding is done. Values for thresholding are obtained with no grid search and with grid search step 20. As it can be seen, there is no major difference between them but the execution time is reduced drastically. The best example is reduction of time for 5-level thresholding. If the grid step increase 4 times (from 5 to 20), execution time is reduces

approximatively 10 times in case of 3-level thresholding, while for 5 level thresholding time reduction is more then 400 times. As it was mentioned complexity of Otsu's method is $O((m + 1)(L - m)^m)$. Execution time drastically grows for more level threshold. As it can be seen in Table 1 for computation 3-level thresholding 1.3 seconds is needed in average for images with 256 gray intensities, while for 4-level thresholding execution time raise up to 133 seconds. If we add just one more level, computation time is more then 1000 seconds. This shows importance of optimization.

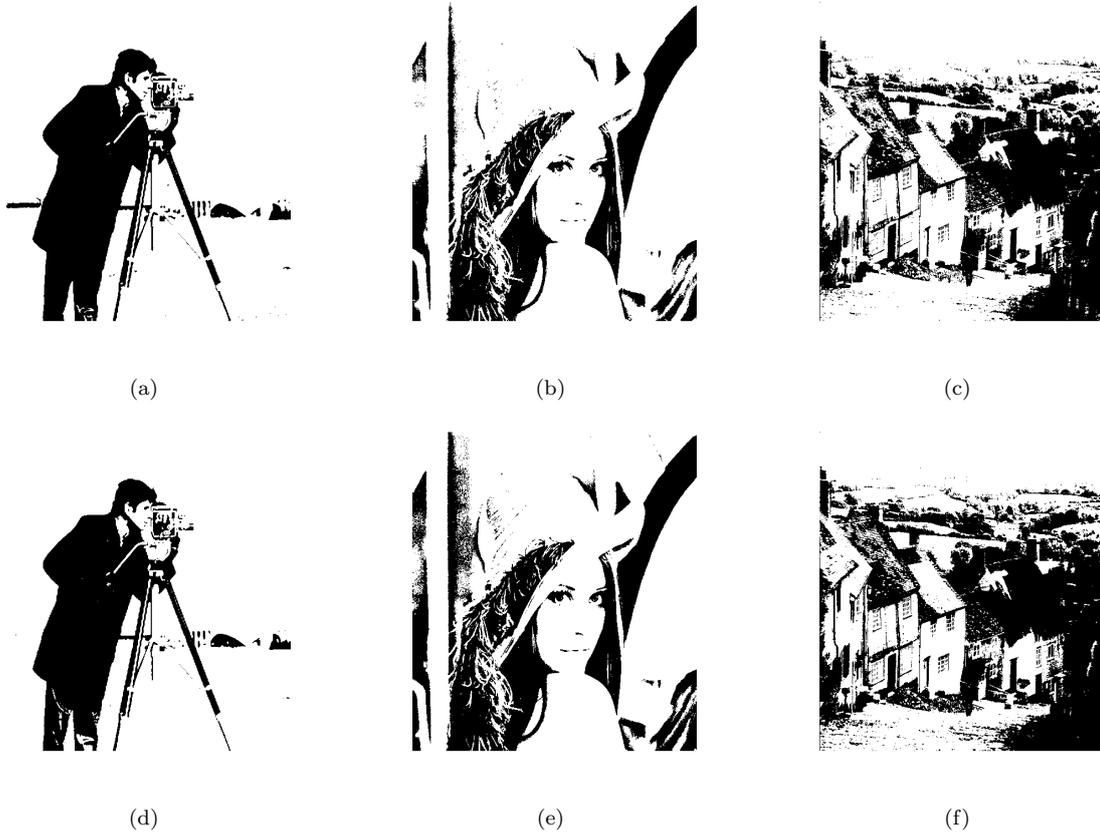


Figure 6: Example of 3-level threshold images with no grid search and grid step 20

Table 2: EXECUTION TIME FOR CALCULATING MULTILEVEL THRESHOLDING WITH DIFFERENT GRID STEP

grid step	3-level threshold			4-level threshold			5-level threshold		
	5	10	20	5	10	20	5	10	20
Barbara	0.0535	0.0147	0.0048	1.1451	0.1381	0.0166	16.1652	0.9090	0.0450
Living room	0.0531	0.0144	0.0047	1.1320	0.1372	0.0165	16.1010	0.9025	0.0451
Boats	0.0536	0.0147	0.0048	1.1377	0.1363	0.0166	16.1649	0.9119	0.0466
Gold hill	0.0555	0.0144	0.0047	1.1339	0.1332	0.0166	16.1771	0.9267	0.0469
Camera man	0.0557	0.0151	0.0048	1.1294	0.1340	0.0165	16.1299	0.9182	0.0456
Lena	0.0558	0.0147	0.0047	1.1367	0.1335	0.0169	16.0174	0.9286	0.0465

In table 2 can be noticed that execution time is significantly reduced by setting the grid step. Success of grid search method for optimization of Otsu's method the best can be seen on example of 5-level thresholding. With relative small grid step, where obtained thresholds are usually good enough, compu-

tation time was reduced from over a 1000 seconds to 16 seconds. If less accuracy is enough and grid step 20 is used, execution time is reduced to less than 0.05 seconds.

For some purpose thresholds obtained with Otsu's method optimized with grid search with large step can give satisfying results. If more precise threshold is needed it can be done with multilevel grid search. First will be found range for optimal threshold, and then grid search can be done on that range with smaller step. It will find optimal threshold if the step is set to 1 at the final grid search and computation time can be significantly reduced.

5 Conclusion

In this paper we experiment with grid search for Otsu's method optimization. Grid search is relatively simple method for optimization, but it is shown that it can reduce computational time significantly. Cost of time reducing is relatively small, threshold values will be in range of grid step size from optimal thresholds. In further work multilevel grid search can be used to determine optimal thresholds. Also it can be tried grid search for some other thresholding methods, Kapur's method for example.

References

- [1] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*. United Kingdom: Cengage Learning, 4 ed., 2014.
- [2] K. S. Tan and N. A. M. Isa, "Color image segmentation using histogram thresholding fuzzy C-means hybrid approach," *Pattern Recognition*, vol. 44, no. 1, pp. 1–15, 2011.
- [3] S. Sarkar, G. R. Patra, and S. Das, "A differential evolution based approach for multilevel image segmentation using minimum cross entropy thresholding," in *Swarm, Evolutionary, and Memetic Computing*, pp. 51–58, Springer, 2011.
- [4] H. Cai, Z. Yang, X. Cao, W. Xia, and X. Xu, "A new iterative triclass thresholding technique in image segmentation," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1038–1046, 2014.
- [5] C. Chesneau, J. Fadili, and J.-L. Starck, "Stein block thresholding for image denoising," *Applied and Computational Harmonic Analysis*, vol. 28, no. 1, pp. 67–88, 2010.
- [6] A. D. Ali, P. D. Swami, and J. Singhai, "Modified curvelet thresholding algorithm for image denoising," *Journal of Computer Science*, vol. 6, no. 1, pp. 18–23, 2010.
- [7] R. Mithe, S. Indalkar, and N. Divekar, "Optical character recognition," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 2, pp. 72–75, 2013.
- [8] W. Hu, X. Li, G. Cheung, and O. Au, "Depth map denoising using graph-based transform and group sparsity," in *15th IEEE International Workshop on Multimedia Signal Processing (MMSp)*, pp. 001–006, IEEE, 2013.
- [9] A. Nyma, M. Kang, Y.-K. Kwon, C.-H. Kim, and J.-M. Kim, "A hybrid technique for medical image segmentation," *BioMed Research International*, vol. 2012, 2012.
- [10] S. Aja-Fernandez, G. Vegas-Sanchez-Ferrero, and M. A. M. Fernández, "Soft thresholding for medical image segmentation," in *IEEE Annual International Conference of the Engineering in Medicine and Biology Society (EMBC)*, pp. 4752–4755, IEEE, 2010.
- [11] M.-H. Horng, "Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13785–13791, 2011.
- [12] T. Pun, "A new method for gray-level picture thresholding using the entropy of the histogram," *Signal Processing*, vol. 2, pp. 223–237, July 1980.

- [13] J. N. Kapur, “A new method for gray-level picture thresholding using the entropy of the histogram,” *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 273–285, March 1985.
- [14] “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, Jan 1979.
- [15] S. Arora, J. Acharya, A. Verma, and P. K. Panigrahi, “Multilevel thresholding for image segmentation through a fast statistical recursive algorithm,” *Pattern Recognition Letters*, vol. 29, no. 2, pp. 119–125, 2008.

Una Tuba
John Naisbitt University
Faculty of Computer Science
Bulevar umetnosti 29, Belgrade
SERBIA
E-mail: tuba.una@gmail.com

List of authors:

- Petru AMORTOAE
„Lucian Blaga” University of Sibiu
Faculty of Sciences
Str. Dr. I. Rațiu, Nr.5-7, Sibiu
ROMANIA
E-mail: petru.amortoae@gmail.com
- Adrian BĂRBOS
Technical University Cluj-Napoca
Faculty of Science
76 Victoriei, Cluj-Napoca
ROMANIA
E-mail: ady.barbos@gmail.com
- Paul – Gheorghe BARBU
"Lucian Blaga" University of Sibiu
Department of Computer Science and
Electrical Engineering
10, Victoriei Bd., 550024, Sibiu
ROMANIA
E-mail: barbu.paul.gheorghe@gmail.com
- István BIALKÓ
Spiru Haret University, Bucharest
Faculty of Mathematics and Informatics
str. Ion Ghica, nr. 13, 030045
Sector 3, Bucharest
ROMANIA
Email: bialko.istvan@gmail.com
- Alexandru BÎRSAN
Bucharest Academy of Economic Studies
Management
Piata Romana 6, 010374, Bucharest
ROMANIA
E-mail: birsan.alexandru@gmail.com
- Cristian BOTA
Technical University Cluj-Napoca
Faculty of Science
76 Victoriei, Cluj-Napoca
ROMANIA
E-mail: cristianbota94@gmail.com
- Petrică BOTA
„Lucian Blaga” University of Sibiu
Faculty of Sciences
Str. Dr. I. Rațiu, Nr.5-7, Sibiu
ROMANIA
E-mail: bota.petrica2@yahoo.com
- Momcilo BRAJIC
Faculty of Computer Science
Boulevard Umetnosti 29, Belgrade
SERBIA
E-mail: momcilob@live.com

Maria – Alexandra CRAIU	<p>„Lucian Blaga” University of Sibiu Faculty of Sciences Str. Dr. I. Rațiu, Nr.5-7, Sibiu ROMANIA E-mail: alexandra_mariaa27@yahoo.com</p>
Crina CRISTEA	<p>Bucharest Academy of Economic Studies Management Piata Romana 6, 010374, Bucharest ROMANIA E-mail: crina.cristea@gmail.com</p>
Alexandru DANIEL	<p>Bucharest Academy of Economic Studies Management Piata Romana 6, 010374, Bucharest ROMANIA E-mail: dan74a@yahoo.com</p>
Mykhailo DOROKHOV	<p>University of Tartu Institute of Computer Science J.Liivi 2, Tartu, Tartumaa ESTONIA E-mail: mykhailo.dorokhov@ut.ee</p>
Andrei DRAGOMIR	<p>National College „Samuel von Brukenthal” Department of Mathematics-Informatics Huet Place Nr. 5, Sibiu ROMANIA Email: meandrei96@gmail.com</p>
Bogdan GEORGESCU	<p>“Horea, Cloșca and Crișan” National College Computer science - Mathematics 1 Decembrie 1918 Avenue, Alba-Iulia ROMANIA E-mail: bondaru@yahoo.com</p>
Emanuel GIURGIU	<p>Transilvania University Faculty of Mathematics and Informatics Eroilor Bd., no. 3, Brașov ROMANIA E-mail: emanu.giurgiu@gmail.com</p>
Ioan GROZEA	<p>National College „Samuel von Brukenthal” Department of Mathematics-Informatics Huet Place Nr. 5, Sibiu ROMANIA Email: ioan.grozea@gmail.com</p>

Radu HANDOLESCU	<p>“Radu Negru” National College Mathematics - Computer Science Școlii St., No.1, Făgăraș ROMANIA E-mail: raduhandolescu@yahoo.com</p>
Darius HAȚEGAN	<p>„Lucian Blaga” University of Sibiu Faculty of Sciences Str. Dr. I. Rațiu, Nr.5-7, Sibiu ROMANIA E-mail: hategandarius71@gmail.com</p>
Dragoș HODINĂ	<p>“Lucian Blaga” University of Sibiu Faculty of Science Department of Mathematics and Informatics Dr. I. Ratiu St., No. 5-7, 550012, Sibiu ROMANIA E-mail: hodina.dragos@gmail.com</p>
Flavius HOLERGA	<p>Axente Sever Theoretical High School Mathematics-Informatics, Avram Iancu St., No. 18, Mediaș ROMANIA E-mail: flavius.holerga@gmail.com</p>
Svetlozar ILIEV	<p>University of Ruse Department of Informatics and Information Technologies 8, Studentska St., Ruse 7017 BULGARIA E-mail: svetlozar321@gmail.com</p>
Vlad IONESCU	<p>Transilvania University Faculty of Mathematics and Informatics Eroilor Bd., no. 3, Brașov ROMANIA E-mail: ionescuvlad12@gmail.com</p>
Marius Daniel MARINESCU	<p>“B. P. Hasdeu” National College Multimedia Systems No.1, Bd. Gării, Buzău ROMANIA E-mail: yow_dany@yahoo.com</p>
Andrei MILIK	<p>Axente Sever Theoretical High School Mathematics-Informatics, Avram Iancu St., No. 18, Mediaș ROMANIA E-mail: andreimilik30@gmail.com</p>

Inocențiu MOGA
Axente Sever Theoretical High School
Mathematics-Informatics,
Avram Iancu St., No. 18, Mediaș
ROMANIA
E-mail: inocentiumoga@gmail.com

Alexandru-Nicolae MOTOC
“Radu Negru” National College
Mathematics - Computer Science
Școlii St., No.1, Făgăraș
ROMANIA
E-mail: alexmotoc97@gmail.com

Fineas-Iacob MUREȘAN
„Lucian Blaga” University of Sibiu
Faculty of Sciences
Str. Dr. I. Rațiu, Nr.5-7, Sibiu
ROMANIA
E-mail: fineasmuresan.ro@gmail.com

Vlad MURZEA
Transilvania University
Faculty of Mathematics and Informatics
Eroilor Bd., no. 3, Brașov
ROMANIA
E-mail: vlad.murzea@gmail.com

Alexandru OLTEAN
“Gheorghe Lazăr” National College of Sibiu
Mathematic-Informatics
Gh. Lazăr St., No 1-3, 550165, Sibiu
ROMANIA
E-mail: alexandru.oltean98@yahoo.com

Theodor-Ovidiu PALAMARU
"Lucian Blaga" University of Sibiu
Faculty of Sciences
No. 5-7, Dr. Ioan Ratiu st., 550012, Sibiu
ROMANIA
E-mail: palamaru.theodor@gmail.com

Ciprian PETREAN
“Horea, Cloșca and Crișan” National College
Computer science - Mathematics
1 Decembrie 1918 Avenue, Alba-Iulia
ROMANIA
E-mail: cipri.petrean@gmail.com

Cătălin POPA
"National School of Gas" Technology High
School, Mediaș
Mathematics - Informatics
Metanului St., No 1, Mediaș
ROMANIA
E-mail: castle2145@gmail.com

Alin Gabriel PRĂVARIU
“Gheorghe Lazăr” National College of Sibiu
Mathematic-Informatics
Gh. Lazăr St., No 1-3, 550165, Sibiu
ROMANIA
E-mail: alin_gabriel_pravariu@yahoo.ro

Mihail Irinel PREDA
“B. P. Hasdeu” National College
Multimedia Systems
No.1, Bd. Gării, Buzău
ROMANIA
E-mail: mihaipreda1997@gmail.com

Robert SĂNDICĂ
„Lucian Blaga” University of Sibiu
Faculty of Sciences
Str. Dr. I. Rațiu, Nr.5-7, Sibiu
ROMANIA
E-mail: sandica.robert@gmail.com

Darko SHULESKI
Bucharest Academy of Economic Studies
Management
Piata Romana 6, 010374, Bucharest
ROMANIA
E-mail: darko.shuleski@man.ase.ro

Maria SOKOLOVA
Vladimir State University
Faculty of Information Technology
Department of Management and Informatics
in Technical and Economic Systems
Str. Gorky, 87, Vladimir
RUSSIA
E-mail: sok.masha2011@yandex.ru

Radu STOICAN
The National College of Computer Science
“Grigore Moisil”
Computer Science, Brașov
ROMANIA
E-mail: radasca16@gmail.com

Marius TOMA
„Lucian Blaga” University of Sibiu
Faculty of Sciences
Str. Dr. I. Rațiu, Nr.5-7, Sibiu
ROMANIA
E-mail: kid.morvius@gmail.com

Dimitar TRIFONOV
University of Ruse
Department of Informatics and Information
Technologies
8, Studentska St., Ruse 7017
BULGARIA
E-mail: dtrifonov32@gmail.com

Una TUBA

John Naisbitt University
Faculty of Computer Science
Bulevar umetnosti 29, Belgrade
SERBIA
E-mail: tuba.una@gmail.com

Eva TUBA

John Naisbitt University
Faculty of Computer Science
Bulevar umetnosti 29, Belgrade
SERBIA
E-mail: etuba@acm.org

Sebastian VIDREA

Transilvania University
Faculty of Mathematics and Informatics
Eroilor Bd., no. 3, Braşov
ROMANIA
E-mail: vidreasebastian@gmail.com

List of sponsors in alphabetical order:



