

**Editor: Dana Simian**

**Proceedings of  
The Fifth International Students  
Conference on Informatics**

**ICDD**

**IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT**

**Romania, Sibiu, May 21-23, 2015**

*Editor DANA SIMIAN*

---

**IMAGINATION, CREATIVITY,  
DESIGN, DEVELOPMENT**

Proceedings of International Students  
Conference on Informatics  
ICDD

**May 21<sup>th</sup> – 23<sup>th</sup>, 2015  
Sibiu, Romania**

---

**“Lucian Blaga” University of Sibiu**

“Lucian Blaga” University, Sibiu, 2015

**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 21<sup>th</sup> – 23<sup>th</sup>, 2015, Sibiu, Romania**

**Copyright @ 2015 All rights reserved to editors and authors**

# Preface

This volume contains refereed papers presented within the Fifth International Students Conference on Informatics - ICDD – 2015, which was held between May 21<sup>th</sup> – 23<sup>th</sup>, 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 4 countries (Bulgaria, Romania, Russia, 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 and to the quality of this proceedings volume.

We also thank to our sponsors COS SOFTWARE, EBS, GEMINI, GSD, IQUEST, OMERON, ROPARDO, PAN FOOD, TOP TECH whose support has greatly contributed to the progress of the conference.

May 2015

Dana Simian  
Conference chairman





**Motto:**

*“There are no limits, only your imagination”*



Proceedings of International Students Conference on Informatics, ICDD – 2015  
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 - Ovidius University of Constanța, Romania
- Prof. PhD Dan Cristea - "Alexandru Ioan Cuza" University of Iași, 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 - Duisburg-Essen University, Germany
- Prof. PhD Gheorghe Grigoras - "Alexandru Ioan Cuza" University of Iași, 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 Daniela Marinescu - Transilvania University of Brașov, Romania
- Prof. PhD Ioana Moisil - "Lucian Blaga University" of Sibiu, Romania
- 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 Dana Petcu - West University of Timișoara, Romania
- Prof. PhD Valer Rosca - "Lucian Blaga" University of Sibiu, Romania
- Prof. PhD Ernest Scheiber - Transilvania University of Brașov, Romania
- Prof. PhD Klaus Bruno Schebesch - "Vasile Goldis" University, Arad, Romania
- Prof. PhD Ing. Ioan Silea - Polytechnic University of Timișoara, Romania
- Prof. PhD Dana Simian - "Lucian Blaga" University of Sibiu, Romania
- Prof. PhD Luminita State - University of Pitești, Romania
- Prof. PhD Gheorghe Stefanescu - University of Bucharest, Romania
- Prof. PhD Mihai Talmaciu - University of Bacău, 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 Danciu - University of Craiova, Romania
- Assoc. Prof. PhD Stefka Fidanova - Bulgarian Academy of Sciences, Bulgaria
- Assoc. Prof. PhD Adrian Florea - "Lucian Blaga" University of Sibiu, Romania
- Assoc. Prof. PhD Victoria Jordan - West University of Timișoara, Romania
- Assoc. Prof. PhD Antoanela Naaji - "Vasile Goldis" Western University of Arad, Romania
- Assoc. Prof. PhD Sorin Noaghi - University of Petroșani, Romania
- Assoc. Prof. PhD Petrica Pop - North University of Baia Mare, Romania
- Assoc. Prof. PhD Sofia Visa - The College of Wooster, United States
- Lecturer PhD Alexandru Bobe - Ovidius University of Constanța, 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 Gabriela Moise - Petroleum-Gas University of Ploiești, Romania
- Lecturer PhD Alina Pitic - "Lucian Blaga" University of Sibiu, Romania
- Lecturer PhD Corina Rotar - "1 December 1918" University of Alba Iulia, 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 Brașov, Romania

## Contents:

<b>Didactic Visual Studio C# for Lee's algorithm</b> .....	13
Ladislau Andrasi Coordinator: Cornelia Ignat	
<b>Designing and implementing a programming language interpreter using Haskell</b> .....	24
Barbu Paul – Gheorghe	
<b>ProPrac</b> .....	33
Vlad Bădoiu, Vlad Teodorescu Coordinators: Mirela Mlisan, Antoanela Fărcășanu	
<b>Smart Home Open Framework</b> .....	39
Adrian Alexandru Bărbos, Kim Răzvan Velker, Andrei Călbăjos Coordinator: Camelia M. Pinte	
<b>Aviation HUD</b> .....	43
Cristian Beckert Coordinator: Delilah Florea	
<b>SymoEngine 2.0</b> .....	53
István Bialkó Coordinator: Grigore Albeanu	
<b>2D Skateboarding</b> .....	62
Paul Boldijar Coordinators: Liliana Comarnic, Ciprian Ghișe	
<b>Virtual reality game</b> .....	71
Petrică Bota, Bogdan Grecu, Darius Hațegan Coordinator: Dana Simian	
<b>GFastPay</b> .....	78
Georgi Chavdarov Coordinator: Katalina Grigorova	
<b>Groups Tracker - Don't get lost from your friends</b> .....	83
Bogdan – Mihai Fieraru, Cristian Oancea, Georgiana Secarea, Cristian – Ioneal Aldulea Coordinator: Anca Vasilescu	

<b>Public transport management in Sibiu</b> .....	90
Ionuț Hodiță	
Coordinator: Daniel Hunyadi	
<b>Modeling the population evolution in Romania in the period 1960 -2012 ....</b>	95
Mihaela Itu, Toporaș Andreea	
Coordinator: Alina Bărbulescu	
<b>Graphic Engine</b> .....	101
Arpad Kerestely	
Coordinator: Adrian Deaconu	
<b>Smart Hotel</b> .....	114
Mitrica Alexandru, Sperila Alecsandru	
Coordinator: Sorin Ilie	
<b>Playing Hex with Monte Carlo simulation</b> .....	125
Nikola Radovic, Ljubomir Raicevic	
Coordinator: Milan Tuba	
<b>Web design techniques with applications</b> .....	132
Petru Rogoianu	
Coordinators: Doina Solomon, Marius Amza	
<b>AIM - Kit for app development</b> .....	142
Robert Sandica, Marius Toma	
Coordinator: Dana Simian	
<b>Modeling of decision-making process by means of a method of a multicriteria alternative choice ELECTRA</b> .....	148
Maria Sokolova	
Coordinator: Vladimir Chernov	
<b>GoStudent</b> .....	156
Cosmin Stoian, Ioana Martin	
Coordinator: Elena-Liliana Martin	
<b>Interactive learning using LabVIEW – simple transformations of the ideal gas</b> .....	165
Marius-Nicolae Țoc	
Coordinator: Ramona Humeniuc	

<b>Handwritten digit recognition algorithm</b> .....	176
Eva Tuba Coordinator: Milan Tuba	
<b>JPEG algorithm compression adjustment</b> .....	182
Ira Tuba Coordinator: Milan Tuba	
<b>Blur detection in digital images</b> .....	189
Viktor Tuba Coordinator: Milan Tuba	
<b>S.N.A.P. - Student Network Administration Platform: The Core and Mobile Implementation</b> .....	195
Kim R. Velker, Adrian A. Bărbos, Cristian Bota, Camelia-L. Ianoş Coordinators: Camelia-M. Pinteă, Hajdu Măcelaru Mara	
<b>S.N.A.P. - Student Network Administration Platform: Web Design and Implementation</b> .....	203
Vlăduţ-F. Mureşan, Alin G. Bonaţ, Ramona Dragomir, Sergiu A. Molnar, Anamaria Palfi, Diana M. Codrea Coordinators: Camelia-M. Pinteă, Hajdu Măcelaru Mara	
<b>List of authors</b> .....	209
<b>Sponsors</b> .....	214





## **Didactic Visual Studio C# for Lee's algorithm**

**Ladislau Andrasi**  
**Teacher Coordinator: Cornelia Ignat**

### **Abstract**

The idea of this application was to create a friendly interface which can help you understand one of the basic algorithms in secondary school. It can be used both for teaching and self-learning. Main features include: friendly interface and a lot of customization. Both the two dimensional array and the means of execution can be customized. Four different ways to execute a certain configuration can give you both a close and an overview look. Also the cells inside the matrix can be customized, the starting, destination and blocked points can be changed at any time. There are very few applications found on the internet that can help you understand this algorithm, but even those don't offer this level of customization and means of execution. Different chromatics and markings in cells provide a very easy way of following the execution, also the queue, which is one of the hardest parts to understand in this algorithm, can be viewed in real-time as it evolves.

## **1 Introduction**

Lee's algorithm [2] is one of the first more difficult algorithm that you learn in secondary school so understanding it is something very important. Therefore this application was made to help students learn it and understand how it works. This is done by using a friendly and easy to use interface with different colouring and markings. A more simplified version of this application was presented at the PCID Conference [3] since then the program got updated with more functions, stability and bug fixes. Some improvements are: the ability to save a configuration, change between orthogonal and diagonal movement, switch languages between Romanian and English.

The idea was born to satisfy two needs: firstly I wanted to create something to help you understand this algorithm easily and quickly and secondly to satisfy my desire to learn C# and visual programming.

The application was made in Microsoft Visual Studio, language C# [1].

## **2 The application**

Opening the application the first thing we notice is that it is structured on two main parts: the left hand side with the features and settings and the right hand side which is a panel containing the matrix [Fig. 1].

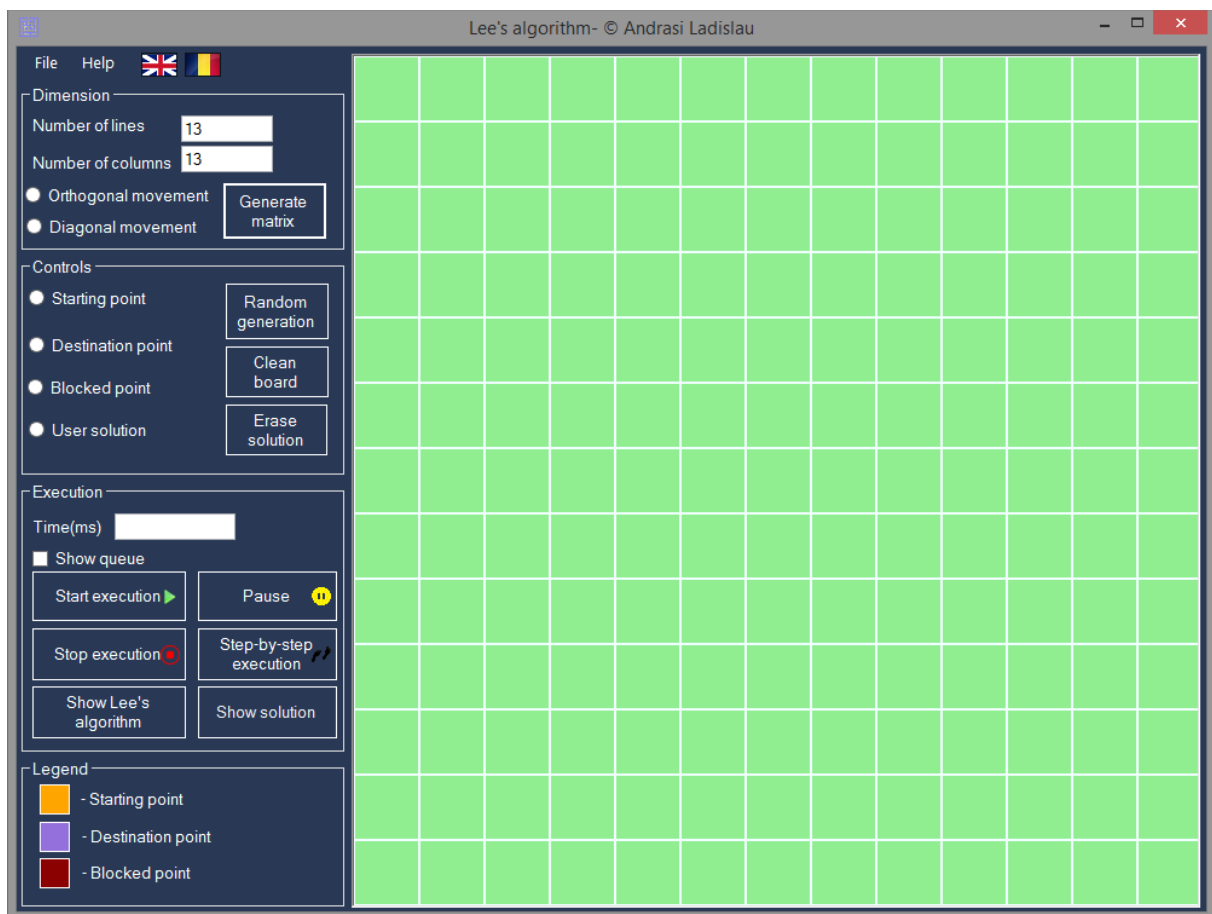


Fig. 1: Main window

## 2.1 Features and settings

### 2.1.1 Open and save

This feature allows you to save a certain configuration and later accessing it again. This is shown in Fig. 2. The saved file can be seen in Fig. 3. The g and r letter symbolizes green and red which means open and blocked points. The s and e symbolizes the starting and destination point. If a wrong file is opened you'll get a warning. Also there is the exit option which closes the application.

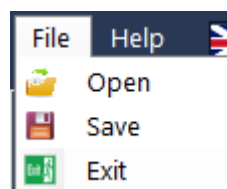


Fig. 2: Open & save

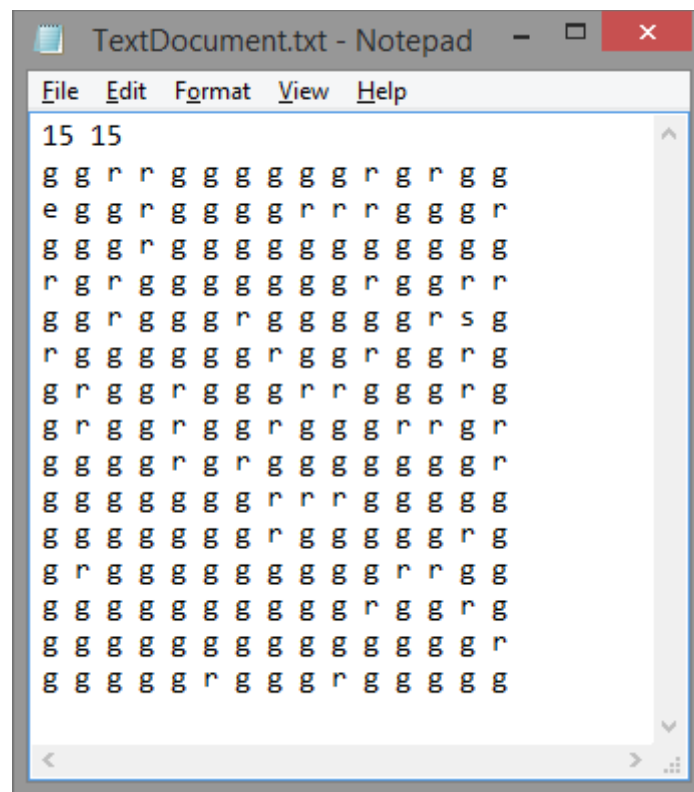


Fig. 3: Saved file

### 2.1.2 Dimensions

The first groupbox [Fig. 4] on the left hand side is called dimension. This groupbox consists of two textboxes in which the user must enter the number of lines and columns of the matrix, two radiobuttons which control the movement inside the matrix which can be orthogonal or diagonal and one button which creates the two dimensional array and unlocks the other groupboxes.

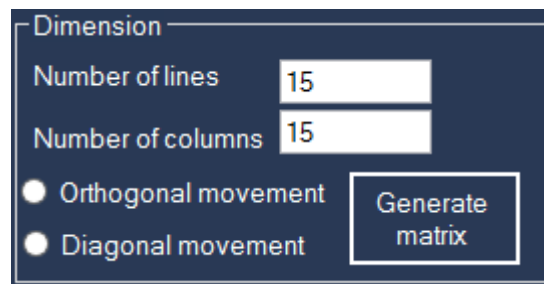


Fig. 4: Dimension groupbox

### 2.1.3 Controls

The *Controls* groupbox [Fig. 5] offers a variety of means to construct the configuration of the board. The first thing you can do is generate a random configuration by clicking the *Random generation* button. The starting and destination point is selected randomly as well as the number of blocked points and their position. If this doesn't suits your needs you can create your own configuration or modify a randomly generated one by using the radiobuttons on the left. The *Starting point*, *Destination point* and *Blocked point* radiobuttons adds or removes the starting, destination and blocked points. The *User solution* radiobutton allows you to input your own solution of a configuration and getting a feedback about the correctness of your solution. Clicking

the *Clean board* button you can clean the entire board or if you just want to wipe a current solution and keep the configuration you can do this by clicking the *Erase solution* button.

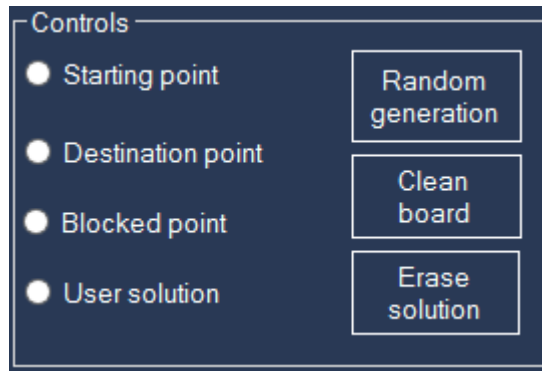


Fig. 5: Controls groupbox

#### 2.1.4 Execution

Here, in this groupbox [Fig. 6], we have the different ways to start the solving of a certain configuration of the board. The *Start execution* button starts a full execution, and the speed can be controlled in the *Time* textbox which operates in milliseconds. Also using the *Pause* button the execution can be either paused or resumed at any time.

Then there is the *Step-by-step execution* button which starts a step by step execution. A step represents a move on the board. Each cell is marked by a colour and by typing the line and the column of the cell inside of it. At the end of the execution the path with minimum steps is marked by a certain colour.

The *Stop execution* button allows you to stop the execution.

The *Show solution* button solves the configuration and shows the shortest path, if there is one. This is useful if you want to check if there exists a solution for your configuration.

Checking the *Show queue* checkbox works in both types of execution and it opens a new form in where you can see the evolution of the queue which perhaps is the most difficult part of the algorithm.

Also the *Show Lee's algorithm* button lets you see the algorithm used.

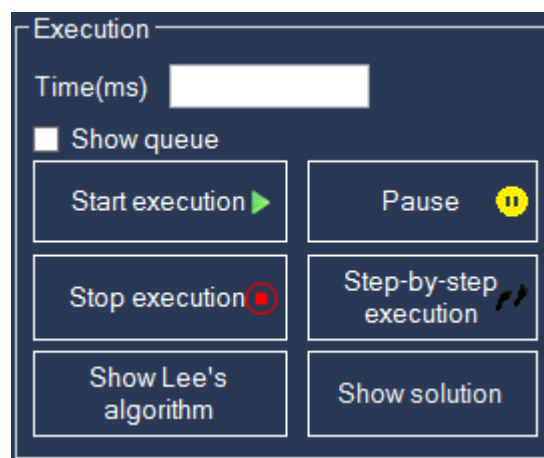


Fig. 6: Execution groupbox

### 2.1.5 Queue

In Fig. 7 the first thing we must notice is that the Show queue checkbox is checked so when Start execution or Step-by-step execution is clicked a new form pops up containing the line and column of each cell that helps form the solution. Start execution is influenced by the value entered in Time textbox, the bigger the value the slower the execution so it makes it easier to follow the queue. Also cells that are part of the solution are coloured white in the Queue form.



Fig. 7: Queue

## 2.2 The two dimensional array

The example in Fig. 8 is a 13 by 13 two dimensional array generated randomly. The meaning of the colours are explained in Fig. 9.

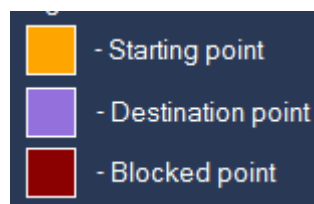
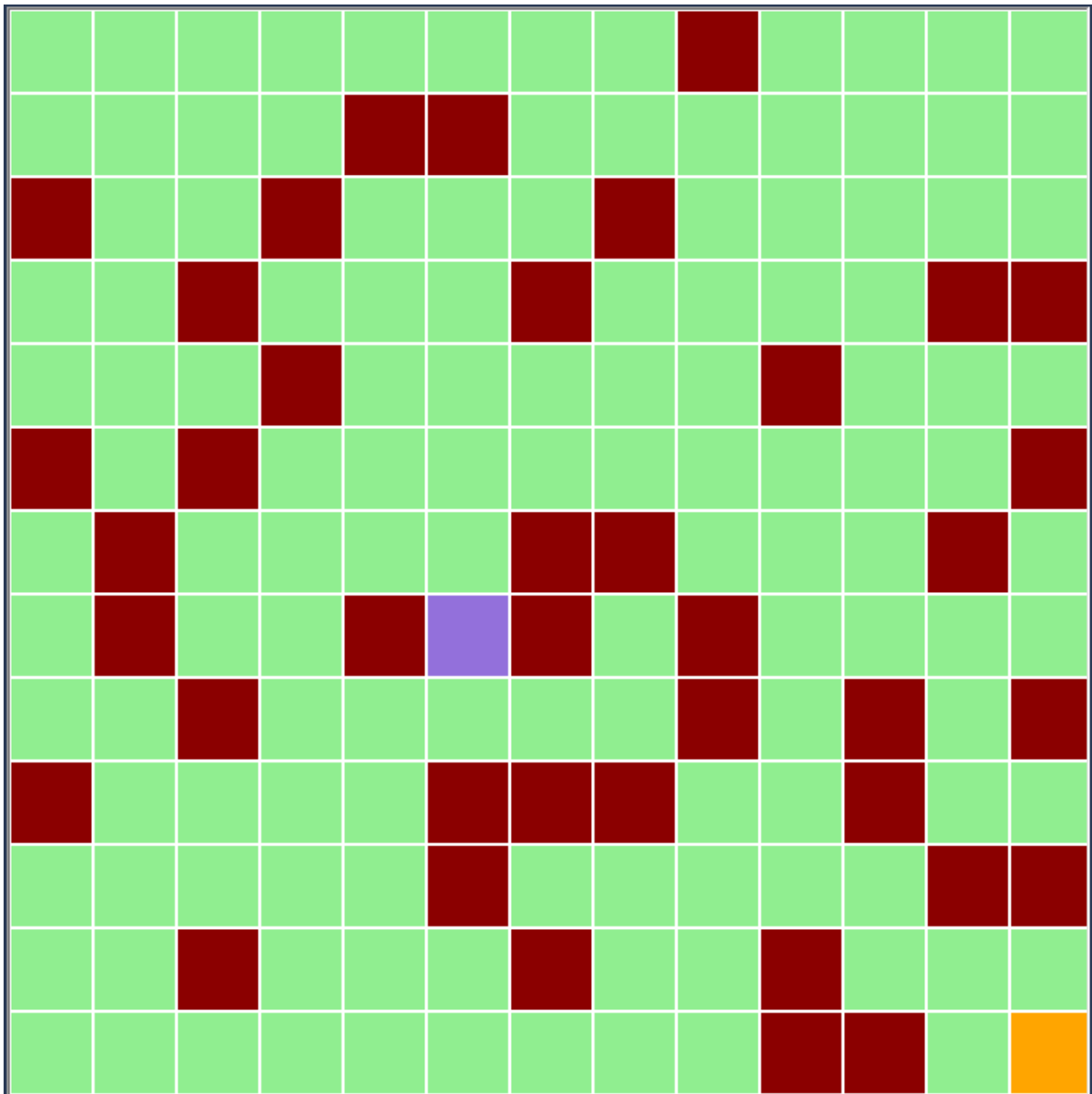


Fig. 9: Points



*Fig. 8: Matrix*

Also another two colours used are light blue and white which symbolizes the correct path and the cells visited to find that path. This is shown in Fig. 10. In the figure the Start execution function was used and only orthogonal movement was allowed. For diagonal movement there is another solution.

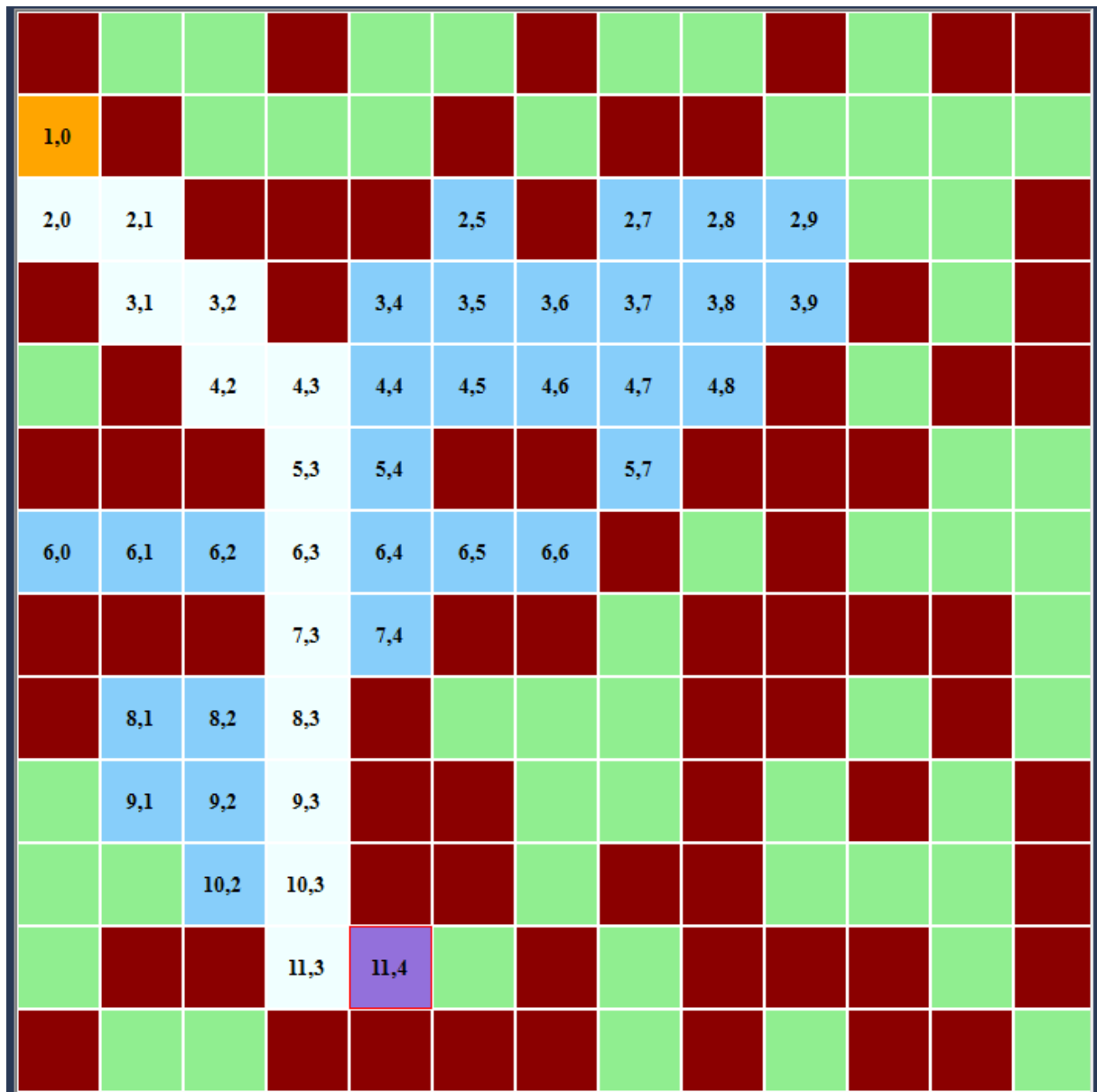


Fig. 10: Solution

### 3 Important parts of the code

The following code is Lee's algorithm I used in this application:

```
void insereaza(int inou, int jnou, int valnoua)
{
    int k;
    k = u;
    while (k > p && valnoua < coada[3, k])
    {
        coada[1, k + 1] = coada[1, k];
        coada[2, k + 1] = coada[2, k];
        coada[3, k + 1] = coada[3, k];
        k--;
    }
}
```



```

        coada[1, k + 1] = inou;
        coada[2, k + 1] = jnou;
        coada[3, k + 1] = valnoua;
        u++;
    }
    void LEE()
    {
        int k, gata=0, inou, jnou, valnoua, SMIN=0;
        p = u = 1;
        coada[1, 1] = locstartx;
        coada[2, 1] = locstarty;
        coada[3, 1] = 1;
        selectat[locstartx, locstarty] = 1;
        while (gata == 0)
        {
            for (k = 0; k < 4; k++)
            {
                inou = coada[1, p] + di[k];
                jnou = coada[2, p] + dj[k];
                valnoua = coada[3, p] + 1;
                if (inou >= 0 && inou < m && jnou < n && jnou >= 0)
                {
                    if (selectat[inou, jnou] == 0 && matr[inou, jnou] != -1)
                    {
                        insereaza(inou, jnou, valnoua);
                        matr[inou, jnou].Text = (p).ToString();
                        selectat[inou, jnou] = 1;
                        if (inou == locfinalx && jnou == locfinaly)
                        {
                            SMIN = valnoua;
                            gata = 1;
                        }
                    }
                }
            }
            p++;
        }
        SMIN = SMIN - 1;
        drum(locfinalx, locfinaly, 1);

        for (int i = 1; i < SMIN; i++)
        {
            matr[traseu[1, i], traseu[2, i]].Text = (SMIN - i).ToString();
            matr[traseu[1, i], traseu[2, i]].BackColor = Color.Azure;
        }
    }

```

Using this algorithm I encountered the following problem: I knew the minimum steps between the two points but I didn't know the path, which I solved with the following algorithm:

```

void drum(int i, int j, int ct)
{
    int k = 0;
    int inou, jnou;
    int x, y;
    int i2 = 0, j2 = 0;
    int minimul = 9999999;
    for (k = 0; k < 4 && movement == true; k++) // orthogonal movement
    {
        inou = i + di[k];
        jnou = j + dj[k];
        if (inou >= 0 && jnou >= 0 && inou < m && jnou < n)
        {
            if (matr[inou, jnou].Text != "" && matr[i, j].Text != "")

```

```

    {
        x = Convert.ToInt32(matr[inou, jnou].Text);
        y = Convert.ToInt32(matr[i, j].Text);
        if (x < y)
        {
            if (minimul > x)
            {
                minimul = x;
                i2 = inou;
                traseu[1, ct] = inou;
                j2 = jnou;
                traseu[2, ct] = jnou;
            }
        }
    }
}
for (k = 0; k < 8 && movement==false; k++) // diagonal movement
{
    inou = i + did[k];
    jnou = j + djd[k];
    if (inou >= 0 && jnou >= 0 && inou < m && jnou < n)
    {
        if (matr[inou, jnou].Text != "" && matr[i, j].Text != "")
        {
            x = Convert.ToInt32(matr[inou, jnou].Text);
            y = Convert.ToInt32(matr[i, j].Text);
            if (x < y)
            {
                if (minimul > x)
                {
                    minimul = x;
                    i2 = inou;
                    traseu[1, ct] = inou;
                    j2 = jnou;
                    traseu[2, ct] = jnou;
                }
            }
        }
    }
}
ct++;

if (ct < SMIN)
    drum(i2, j2, ct);
else
{
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            matr[i, j].Text = "";
}
}
}

```

What this does it goes from the destination point to the starting point following a certain rule. In Lee's algorithm I am marking each cell in the matrix with the current step which is symbolized by  $p$ . So the rule is that starting from the destination point the algorithm looks for neighbours with the smallest numbering inside the cell and moves to that one and continues on until the starting point. This can be seen in Fig. 11. After the path is found the cells are emptied and marked with their line and column. This happens fast so it is not noticeable by the user.

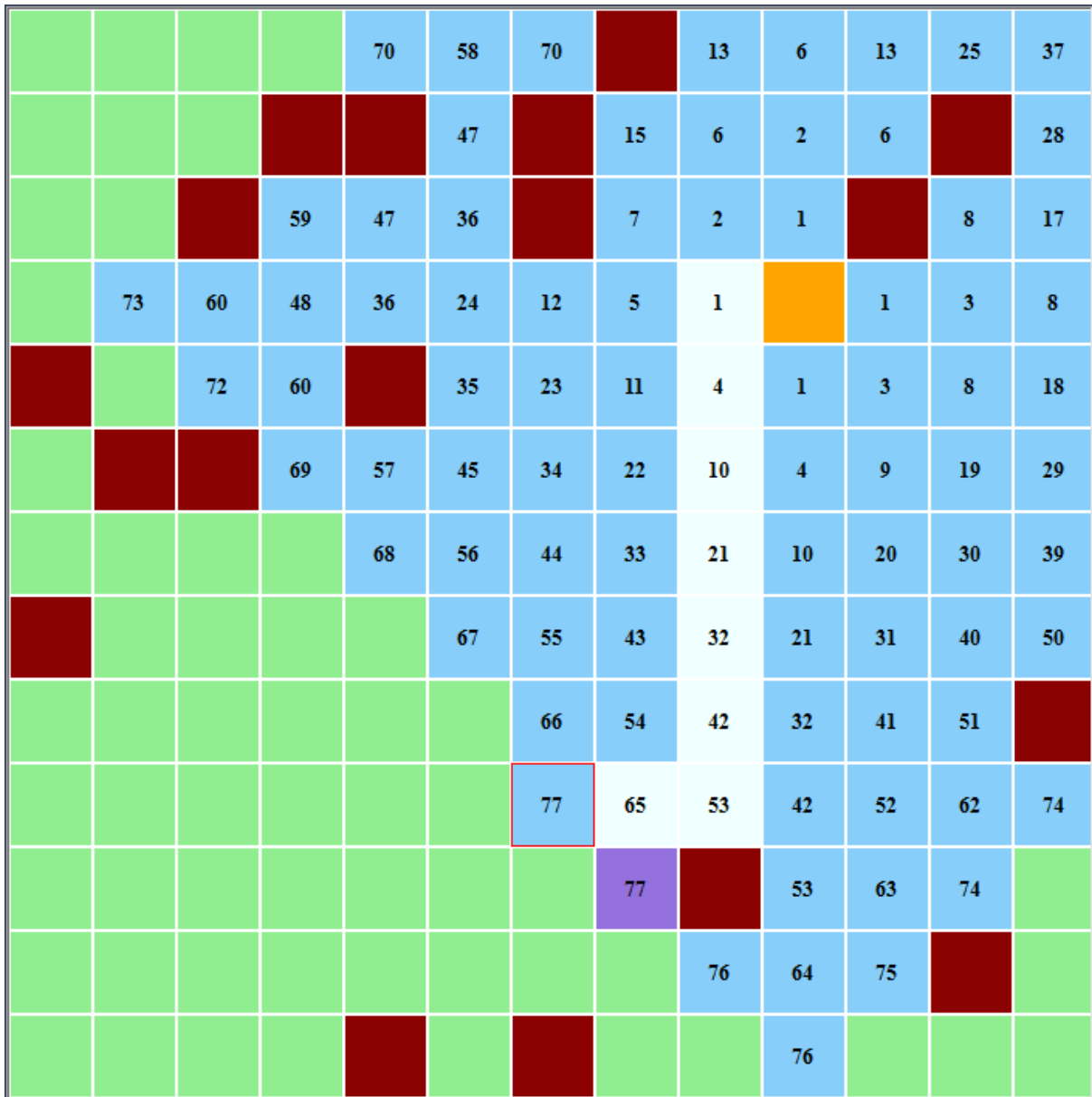


Fig. 11: Path finding algorithm

## 4 Conclusion

The main difference between this application and all the others I could find is the fact that you can create your own matrix, you're not stuck with a predefined one.

It is also easy to follow because of the different colouring, which are explained in the in the last group box of the application.

The variety of ways find the solution of a certain configuration gives the user the ability to have both an overview and a detailed vision of the execution. Also the possibility to have a step by step execution is the strong point of this application.

Future improvements would include some sort of animations and a possibility to create big sized matrixes without the application crashing or having any kind of delay.

## References

- [1] DUMITRIU-LUPAN, Nuşa, PINTEA, Rodica, NIŢĂ, Adrian, NIŢĂ Mioara, Introducere în programarea .Net Framework, Editat de BYBLOS SRL sub coordonarea Agora Media SA, pentru Microsoft România, 2008.
- [2] VIŞINESCU, Radu, VIŞINESCU Violeta, Algoritmul lui Lee. Teorie și aplicații, Gazeta informatică nr.15/2, Cluj-Napoca, 2005.
- [3] SIMIAN, Dana, STOICA, Laura, Programare, Comunicare, Imaginație, Design. Lucrările Conferinței Naționale de Informatică pentru Elevi, Ediția I, Sibiu, 2015.

Ladislau Andrași  
Liceul Tehnologic „Școala Națională de Gaz”  
Mediaș  
Matematică – informatică  
Mediaș, România  
ladiladi85@gmail.com

## Designing and implementing a programming language interpreter using Haskell

Barbu Paul - Gheorghe  
Teacher Coordinator: -

### Abstract

Nowadays programming languages are ubiquitous, it is important to understand how they are created and how can we improve them, thus this paper aims to present the steps taken to design a programming language and implement an interpreter for it. This paper is also motivated by the fact that we have to understand the tools that we use and, as programmers, our main tool is the programming language and its corresponding interpreter or compiler.

## 1 Introduction

Programming languages are the basis of any programmer's work and not only. Everything that runs on a computer or a smart device has to be programmed by some means. Those means are the programming languages that allow us to express ideas in code, instructions that machines can understand and execute. Taking into account the rapid growth [1] of internet-connected devices it is safe to assume that the demand for applications (both online and offline) is rising. Due to the constant need for applications, it is also safe to assume that programming languages will also continue to evolve, new ones will appear and old ones will get deprecated by others. Regardless of what will happen to particular programming languages, the ideas and the concepts surrounding them will continue to persist (one such example is the closure [2]), thus it's necessary for us to have a basic understanding for how they work and how can we improve them, in other words we need to get a grip for the basics. This paper aims to do this by looking at the design and implementation process of an interpreted programming language, called ePseudocode [3], by providing a fully functional programming language. This language is aimed both at beginner programmers, that are still learning, by providing an easy to use syntax and pseudo-code appearance (hence the name) and at more experienced programmers by providing advanced features such as user defined data types, anonymous functions and closures. It is important to note from the start that all the keywords of the language are in Romanian, since the language is aimed primarily at Romanian students willing to get into programming. It's also interesting to see running programs written on the computer as if they were on paper.

## 2 Related work

Programming languages [4] are quite diverse, in paradigm, syntax or just purpose.

There are a lot of research projects [5] that aim to improve programming languages and this leads to more and more languages being developed, keeping this in mind, I will list here only languages that influenced me while designing ePseudocode.

### 2.1 Haskell [6]

Haskell is a programming language developed in academia, used as research ground for functional programming, but also used in industry [7] due to its characteristics, which include first-class functions, laziness, purity and pattern matching, among others.

From Haskell, ePseudocode takes the first-class functions, which means that functions can be passed around freely as arguments to or returned from other functions. ePseudocode, unlike Haskell, does not completely embrace the functional paradigm and has a more permissive type system, which results in faster development times, but also more bugs that may show up at runtime, in contrast with catching them at compile time.

Also Haskell is the host language for ePseudocode, chosen for the ability to work with algebraic data types, which simplifies the representation and traversal of the abstract syntax tree of the guest language. The Parsec [8] library was a great plus, too, making the parser look very natural, almost like reading English, suppose the sequence for parsing if-else blocks:

```
reserved tIf <?> tIf
cond <- expr
reserved tThen
thenStmts <- many mainParser
reserved tElse
elseStmts <- many mainParser
reserved tEndIf
return $ CompleteIf cond thenStmts elseStmts
```

A snippet [9] of ePseudocode showing the ability to use functions as first-class citizens of the language:

```
func applyToRange(a, b, step, f)
  pt i=a; i<=b; i=step(i) executa
    f(i)
  sfpt
sffunc

func main()
  applyToRange(1, 5, func(x) ret x+1 sffunc, func(x) scribe(x*2) sffunc)
sffunc
```

Here, applyToRange is being passed not only the lower and upper bounds (a and b), but also the function that should provide the next value for the counter, step, and the function that uses the counter in the loop's body, f. The output will be: 246810 (the scribe built-in function just writes whatever it is passed to it as argument to the screen without appending spaces or newlines).

## 2.2 C [10]

The main influence on ePseudocode was the C programming language, because it is the main language used in Romanian high schools it's the first one encountered by pupils, thus it's important to keep some familiarity between ePseudocode and C in order for the transition to be easy.

The syntax, like C, is block oriented, with all the blocks surrounded by instructions that mark the beginning and the end of the block.

Also the file handling functions in the standard library are inspired by their counterparts in the C standard library.

Unlike C, ePseudocode is a higher level language, it doesn't allow the programmer direct access to memory, so there is no pointer data type and no functions like malloc or free. The reason behind this decision is that beginners do not have to worry about manual memory management. It is important to note that C is a compiled language, whereas ePseudocode is interpreted, so the speed of programs written in my programming language is not nearly as good as programs written in C, this is a known problem and there are various solutions to it [11], but ePseudocode implements none, in order to keep the interpreter simple to understand.

## 2.3 Python [12]

From Python ePseudocode inherits the type system, or more precisely the lack of one thereof. Python is a duck typed language and so is the language I developed. The lack of explicit types allows the student to concentrate on the actual algorithms and ideas behind programming. One feature of Python is also the fact that some operators have English names like "and" and "or" this is also continued in ePseudocode, but, as said earlier, the operator names are in Romanian (e.g.: si and sau). The Python language is an interpreted one, as is ePseudocode, this allowed for a short development time of the interpreter and a quick minimum viable product.

## 3 The language

ePseudocode is an imperative, dynamic, high level, general purpose, Turing complete programming language, with a very permissive type system.

The entry point in a ePseudocode program is the main function, if it's missing, that constitutes an error. As usual the main function may take no arguments or one argument (the argument list).

```
func main(argv)
    firstArg = argv[0]
sffunc
```

Functions may be used as closures and anonymous functions, a feature inherited from Haskell:

```
func plusN(n)
    // this is a closure, defined as an anonymous function
    // it will capture its environment,
    // and thus n will be defined inside its body
    ret func(x)
        ret n + x
    sffunc
sffunc
```

At the language level it is important to note that ePseudocode, unlike Haskell and Python, only provides a small number of basic data types. Apart from integers and floats, the list is the main data type, they are different from Haskell lists because one can store elements of different types, e.g.:

```
l = {"foobar", 42, 3.14}
```

Strings are also lists of characters, but for which some syntactic sugar exists allowing them to be delimited by double quotes, one can also use escaping sequences inside strings as shown in the example code that implements Conway's Game of Life [13].

A programmer may also define his own data types using structures:

```
struct KeyVal
    key = ""
    val = ""
sfstruct
```

In the snippet above the KeyVal structure holds two fields, a key and a value, with the default values being the empty string. In the example below Q is a list that holds elements of type KeyVal, the fields in the structure are accessed similarly to C, by using the dot operator:

```
func minNode(Q)
    min = Q[0]
    pt i=1; i<lung(Q); i=i+1 executa
        daca Q[i].key < min.key atunci
            min = Q[i]
        sfdaca
    sfpt

    ret min
sfunc
```

As can be seen from the examples, variables need not be declared before they are used, a variable declaration and definition is done in the same place.

## 4 Standard library and tools

The standard library and the tools are what influence a programming language's adoption. Having this in mind I also developed a small standard library [14] containing some utility functions along with the built-in ones like: scrie, citeste, deschide, inchide, int, float, ceiling [15]

Apart from the standard library there also exists syntax highlighting for Notepad++ [16], proving that ePseudocode could be used exactly as are other well known programming languages as well as showing that tools may be developed in order to aid the programmer's job of writing programs in this new language.

The interpreter itself may be considered a tool and it has two modes of operation. The first mode allows the programmer to execute code interactively in a REPL [17] if invoked simply as "epseudocode" without arguments. This mode allows for quick experimentation, the user can use the full power of the language similarly to the REPL provided by Python or the one provided by GHC



[18]. The GHC REPL doesn't allow the programmer to define functions, but the ePseudocode REPL allows one to define functions provided that they are written on a single line.

The second mode of operation can be used to run a full ePseudocode program. By running "epseudocode huffman.epc" in a command line the code in the "huffman.epc" file will be executed by the interpreter and its results will be written on the stdout or in some files, depending on what the program does.

## 5 The interpreter

The interpreter for ePseudocode is written in Haskell, a high level, pure functional programming language with one of the strongest type systems out there. One of the reasons Haskell was chosen is for its ability to define algebraic data types, which makes the representation of the abstract syntax tree trivial, the AST for a ePseudocode statement looks like this:

```
data Stmt = TypeDef String [Stmt]
          | Assign Expr Expr
          | Completelf Expr [Stmt] [Stmt]
          | Simplelf Expr [Stmt]
          | While Expr [Stmt]
          | For (Maybe Stmt) (Maybe Expr) (Maybe Stmt) [Stmt]
          | Ret Expr
          | Break
          | Continue
          | E Expr
```

The interpreter is implemented in a modular way by separating the concerns as is done in any interpreted language.

### 5.1 The lexer [19]

The lexer defines the available keywords in the language as well as the possible identifier names, an identifier has to start with a letter or an underscore and may contain letters, underscores and numbers, it cannot match any built-in keywords. The comments are also defined by the lexer and are identical to C, with the added feature that they can be nested, some C compilers do not allow this.

### 5.2 The parser [20]

In an usual setup, the parser is in charge of translating the token stream produced by the lexer into a parse tree. ePseudocode, because of the parser combinator library it uses, takes another approach: the parser is fed directly with the text source code and, using parts of the lexer, translates the source code directly into the abstract syntax tree. This is possible because the syntax of the language is not complex and at first it allowed great flexibility in designing the language.

Both the lexing and the parsing parts are built on top of the Parsec Haskell library.

### 5.3 The evaluator [21]

The heart of the interpreter for ePseudocode is the evaluator, it handles the translation of the abstract syntax tree to Haskell code that is ultimately executed to get the desired results.

It is written in a pure functional way, relying heavily on recursion, monads and monad transformers as is the case with the majority of big Haskell applications. Being implemented in a pure functional way has both advantages and disadvantages: the bright side would be that the code is pretty straight

forward to read and to modify, functions are composed together to form the interpreter, but on the down side, the speed is not so great since modifying nested lists implies recreating the whole nesting structure and applying the modifications at every level, which brings in not only speed limitations, but also space leaks [22], defects for which Haskell, due to its lazy nature, is famous. When this is done in a loop the speed issues start to appear. This effect can be seen in the example code that implements Conway's Game of Life. Running a single iteration (two passes that check the rules of the game in a 40x20 matrix) both in Python [23] and in ePseudocode yields the results illustrated by the Fig plot.

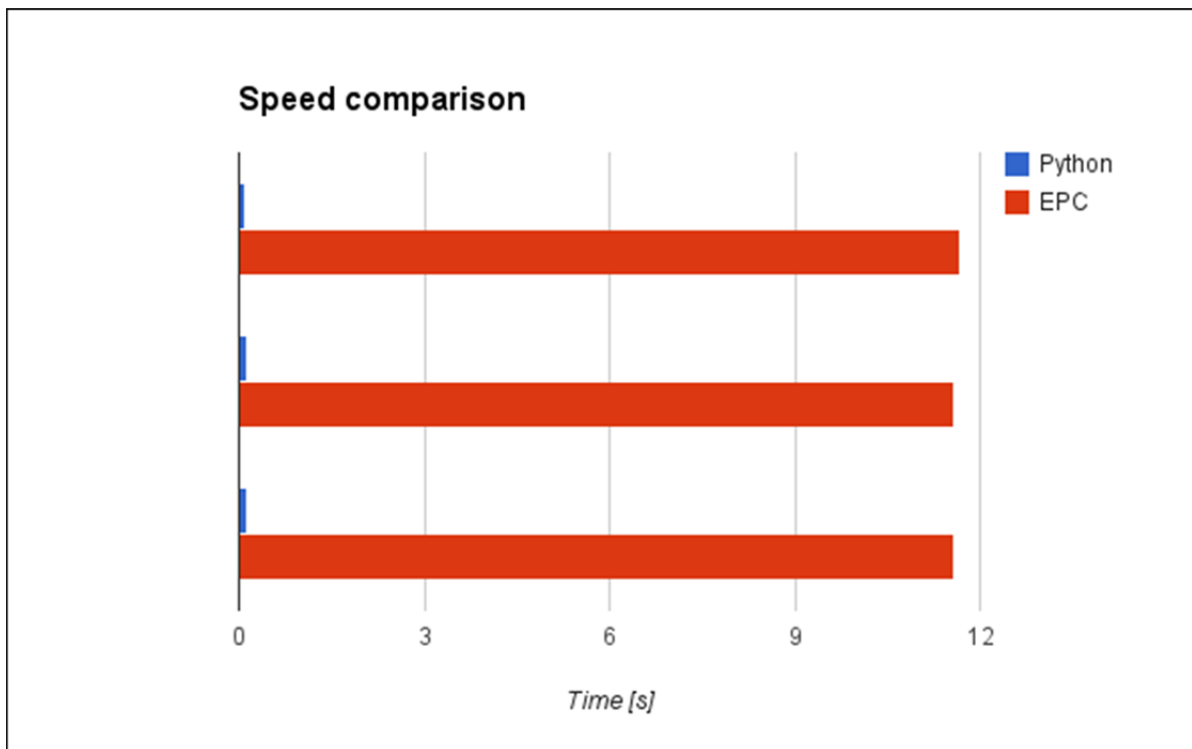
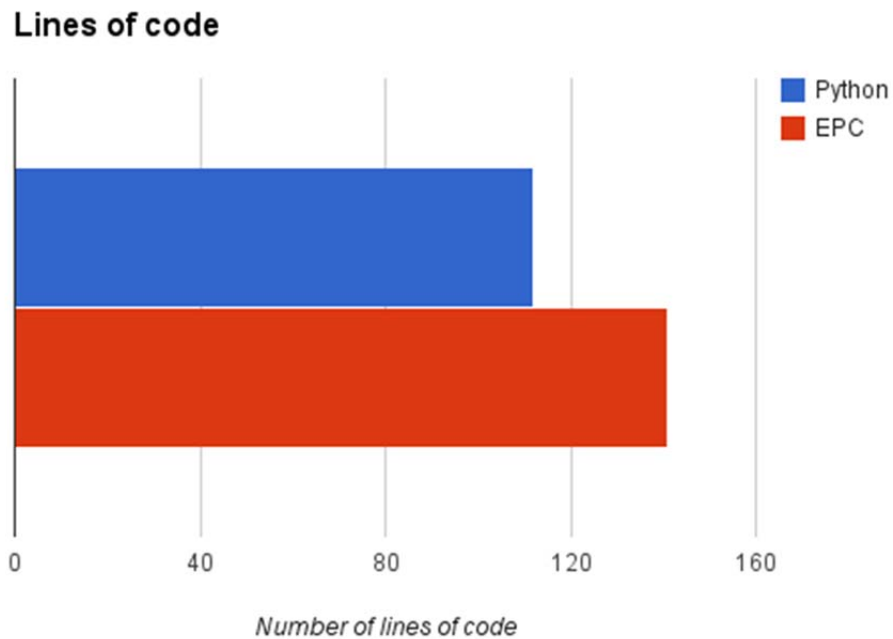


Fig. 1: Speed comparison

Of course this defect can be optimized away by using the Data.IORef Haskell module and modifying the lists in place instead of taking the functional approach.

Although the performance is not great, the programmer productivity is enhanced by the high level nature and the clean syntax of the language. The line counts of the two example implementations are illustrated below:



*Fig. 2: Lines of code*

As can be seen from the Fig plot the ePseudocode line count is very close to the the Python one, the additional lines explained by the fact that in ePseudocode the programmer has to mark the block endings.

## 6 Future work

Considering that the language is still in its early days there is a lot of work to be done. For example at the moment there can only be single file programs, no module system or namespaces are implemented, every top level function or variable is defined in the global namespace.

Of course the speed can be improved as can be seen in the above plot. At first Data.IORef may be used to get a significant speedup by modifying lists in place, then the evaluator may implement optimizations on the abstract syntax tree similarly to other languages. Finally the evaluation process may use just in time compilation, a well known technique for implementing optimizing interpreters.

The language also lacks some tooling support, like a debugger, right now the programmer has to debug his programs by printing values to the screen, also known as "printf debugging". The REPL can be enhanced, too, by allowing the user to input multiple lines at once, now the REPL accepts a single input line at a time.

## 7 Conclusion

In spite of its lack of speed, ePseudocode can do anything that a mature programming language can. It also offers a simple and clean syntax, not only designed to help beginners get started with programming, it also allows professional programmers to rapidly create prototypes due to its high level nature. Although the keywords are written in Romanian, the language could be easily translated because it is open source and the lexer is clearly separated from the rest of the interpreter. The 600 unit tests [24] that ensure that more than 95% of the code behaves as it should, stand as proof of the language's quality.

## References

- [1] Mobile World Live, *Number of devices to hit 4.3 per person by 2020 – report*, <http://www.mobileworldlive.com/connected-devices-to-hit-4-3-per-person-by-2020-report>, 2014
- [2] Wikipedia, *Closure - History and etymology*, [https://en.wikipedia.org/wiki/Closure\\_%28computer\\_programming%29#History\\_and\\_etymology](https://en.wikipedia.org/wiki/Closure_%28computer_programming%29#History_and_etymology), 2015
- [3] ePseudocode, <https://github.com/paulbarbu/epseudocode/>, 2015
- [4] Wikipedia, *List of programming languages*, [https://en.wikipedia.org/wiki/List\\_of\\_programming\\_languages](https://en.wikipedia.org/wiki/List_of_programming_languages), 2015
- [5] DrDobbs, *Microsoft M#, An Extension Of C#*, <http://www.drdobbs.com/windows/microsoft-m-an-extension-of-c/240165199>, 2014
- [6] Haskell, <https://www.haskell.org/>, 2015
- [7] Wikipedia, *Haskell (Programming language) – Industry*, [https://en.wikipedia.org/wiki/Haskell\\_%28programming\\_language%29#Industry](https://en.wikipedia.org/wiki/Haskell_%28programming_language%29#Industry), 2015
- [8] Haskell Wiki, *Parsec*, <https://wiki.haskell.org/Parsec>, 2015
- [9] ePseudocode, *callback.epc*, <https://github.com/paulbarbu/ePseudocode/blob/master/examples/callback.epc>, 2015
- [10] Wikipedia, *C (programming language)*, [https://en.wikipedia.org/wiki/C\\_%28programming\\_language%29](https://en.wikipedia.org/wiki/C_%28programming_language%29), 2015
- [11] Wikipedia, *Just-in-time compilation*, [https://en.wikipedia.org/wiki/Just-in-time\\_compilation](https://en.wikipedia.org/wiki/Just-in-time_compilation), 2015
- [12] Wikipedia, *Python (programming language)*, [https://en.wikipedia.org/wiki/Python\\_%28programming\\_language%29](https://en.wikipedia.org/wiki/Python_%28programming_language%29), 2015
- [13] ePseudocode, *conway.epc*, <https://github.com/paulbarbu/ePseudocode/blob/master/examples/conway/conway.epc>, 2015
- [14] ePseudocode, *.stdlib.epc*, <https://github.com/paulbarbu/ePseudocode/blob/master/.stdlib.epc>, 2015
- [15] ePseudocode, *Builtins.hs*, <https://github.com/paulbarbu/ePseudocode/blob/master/src/Epseudocode/Builtins.hs>, 2015
- [16] ePseudocode, *epcSyntax.xml*, <https://github.com/paulbarbu/ePseudocode/blob/master/tools/epcSyntax.xml>, 2015
- [17] Wikipedia, *Read–eval–print loop*, [http://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print\\_loop](http://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop), 2014
- [18] Wikipedia, *Glasgow Haskell Compiler*, [https://en.wikipedia.org/wiki/Glasgow\\_Haskell\\_Compiler](https://en.wikipedia.org/wiki/Glasgow_Haskell_Compiler), 2015

- [19] ePseudocode, *Lexer.hs*, <https://github.com/paulbarbu/ePseudocode/blob/master/src/Epseudocode/Lexer.hs>, 2015
- [20] ePseudocode, *Parser.hs*, <https://github.com/paulbarbu/ePseudocode/blob/master/src/Epseudocode/Parser.hs>, 2015
- [21] ePseudocode, *Evaluator.hs*, <https://github.com/paulbarbu/ePseudocode/blob/master/src/Epseudocode/Evaluator.hs>, 2015
- [22] Haskell Wiki, *Memory Leak*, [https://wiki.haskell.org/Memory\\_leak](https://wiki.haskell.org/Memory_leak), 2015
- [23] ePseudocode, *conway.py*, <https://github.com/paulbarbu/ePseudocode/blob/master/examples/conway/conway.py>
- [24] ePseudocode, *Test suites*, <https://github.com/paulbarbu/ePseudocode/tree/master/test>, 2015
- [25] Leijen Daan, *Parsec, a fast combinator parser*, <https://research.microsoft.com/en-us/um/people/daan/download/parsec/parsec.pdf>, 2001
- [26] HaskellWiki, *Parsing Expressions And Statements*, [https://wiki.haskell.org/Parsing\\_expressions\\_and\\_statements](https://wiki.haskell.org/Parsing_expressions_and_statements), 2012
- [27] Coursera, *Compilers*, <https://www.coursera.org/course/compilers>, 2014
- [28] Wikipedia, *Conway's Game of Life*, [https://en.wikipedia.org/wiki/Conway%27s\\_Game\\_of\\_Life](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life), 2015

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

## ProPrac

Vlad Bădoiu, Vlad Teodorescu

Teacher Coordinator: Mirela Mlisan, Antoanela Fărcășanu

### Abstract

ProPrac is an educational platform, focused on programming, for pupils, students and teachers of all ages. It contains a collection of programming problems, tutorials and learning materials. A typical user can register with an account, learn from the tutorials and documentation, solve problems in a programming language, upload his solution to be verified on some pre-generated tests and then, in a matter of seconds, receive the score on his problem, score that is registered for his account. The more problems he solves, the more points he has. The site and the problems are only available in Romanian and the problems can be solved using C/C++, Python, Ruby or Java. The design of the educational platform is custom and made with the help of the front-end framework called Bootstrap.

*Key Words:* programming, educational platform, problems, tutorials, documentation.

## 1. Introduction

A simpler version of this presentation was originally published in Romanian in the volume of the PCID national conference 2015 ([http://conferences.ulbsibiu.ro/conf.pcid/2015/volum/Volum\\_Conf\\_Nat\\_Inf\\_Elevi\\_PCID\\_2015.pdf](http://conferences.ulbsibiu.ro/conf.pcid/2015/volum/Volum_Conf_Nat_Inf_Elevi_PCID_2015.pdf)) [8].

ProPrac is an educational platform created with only one purpose: to allow programming enthusiasts to learn and to develop programming abilities with the help of the problems, the tutorials and the documentation, which are available to the user at any given time.

ProPrac is not the only one of its kind. There are at least three other sites that work on this domain: pbinfo.ro, infoarena.ro and champion.edu.ro. From among these three, the first one is focused on problems, the theory needed being less if not at all present. The third one also deals with problems, but, on top of that, there are some contests that take place. The second site deals more with the theoretical part. The problems that are available to the user being heavily connected with the basic idea of the algorithms that are presented in the theory. ProPrac, in comparison to the other three can be defined as a combination between pbinfo.ro and infoarena.ro, the way of how the problems are solved being similar to the first one, while the theory is presented in a way close to the second one.

Even though the educational platform ProPrac can be defined as a combination between the two websites presented earlier, its design and its functionality is unique and represents the solution to the

needs of the children in the secondary school or high school, on their way to the desired performance in the domain of programming.`

The present piece of work is structured in sections. The first section will present the general development of the platform. The second section will deal with the Backend programming that was necessary for developing the platform. The third part will focus on the Frontend development. The fourth section will present some technical details. The fifth section will be about final ideas and the future development of the platform. The sixth and final section will deal with references and bibliography.

## 2. General development

In this section, the focus falls on presenting the general idea and an overview of the steps made in creating this educational platform.

The general idea came to life at the moment when we, as pupils, started solving problems on websites like pbinfo.ro and champion.edu.ro. However, in time, we noticed different drawbacks in these websites, drawbacks like the lack of problems grouped by the age of study or chapters, or the impossibility of learning from an official solution or unprocessed tests. Thus, we decided to build our own website from ground up and make it more of an educational platform. We wanted to fix, from our point of view, the drawbacks of the other websites and to introduce at the same time new and original ideas.

The first step taken by us in this endeavor was to build a list of all the functions that our educational platform had to have. After that, we sorted all of those from the most important ones(the core ones) to the most insignificant ones. Moving on, we had then to divide the work among ourselves. Being a team of two pupils and two teachers, we decided to make a simple division: one pupil will take the Frontend development and the other one will deal with the Backend development. We took this decision based on our knowledge and with the logistic help that we needed which our teacher coordinators gave us.

The second step of this project was finding a way for coordinating our work. For this, we use a source-control technology (Git) for storing and updating our project.

Last but not least, a problem we are still working on is deploying our educational platform on the web and making it available for everyone that wants to use it.

The idea for the name of the platform was simple. We took the core meaning of it which is “Programming Practice” and essentially cut the words until they had a certain ring but kept their meaning, leaving us with “ProPrac”.

## 3. Backend Development

The main objective of this section is to present the Backend side of the project.

A big part of the Backend development is the database of the platform. The database contains a big amount of the data in the platform, such as the problems, tests, sources that need to be or were tested, the usernames and the passwords of the users. The database is secured, the only permission that the users have being that of writing at the moment when a new account is created or a source is sent to be tested. The usernames and passwords are saved under the shape of hashes, thus being impossible to obtain even if, in the worst-case scenario, the database would be accessed by someone without authorization. The hashing key is a 256-bit key, the hashing algorithm that we use being SHA-2 (en.wikipedia.org/wiki/SHA-2) [4] developed in 2001 by NSA. On top of this, the database is defended by attacks that want to fill it with useless information, the users being limited to only adding sources and the number of sources that are memorized in the database for a problem for a user being limited to one. In other words, the only source taken in consideration for a certain problem at a certain

point is the last one sent by the user. When a new source is sent, the one existing is deleted and the new one takes its place.

Regarding the security of the sources received from the users, the educational platform block the majority of the system calls(en.wikipedia.org/wiki/System\_call)[2], permitting the use of only four, instead of over 300. The compiler works on Linux and is protected by a security module named seccomp(en.wikipedia.org/wiki/Seccomp)[3]. Seccomp intercepts the system calls when a source code is ran and gives permission of running to only those system calls that are necessary. An example of a blocked system call is `sys_fork`. This system call is used in the creation of a computer virus known as a “rabbit virus”, a “wabbit” or a “fork bomb”(en.wikipedia.org/wiki/Fork\_bomb)[1]. This kind of virus executes a DoS (denial of service) attack and is, basically, a piece of code that auto-replicates until the resources of the system that it runs on run out and the system crashes. Plus, a second security module of Linux, known as SeLinux, makes the use of processes, ports (TCP/UDP) and shared memory segments impossible by the user through the running of the source code.

The main functionalities of the site are: adding problems by an admin, reading problems by the user, sending of sources, testing of sources and receiving a certain score for the source sent and verified.

In the following part of this section, some parts of the source code will be detailed and explained accordingly.

```
def home(request):
    """
    Handles the GET requests for the home page
    :param request:
    :return:
    """

    template = "home.html"
    latest_question_list = Problem.objects.order_by('-pub_date')[:5]
    if not request.user.is_authenticated():
        context = {'user.is_authenticated': False}
        return render(request, template, context)
    else:
        useractiv = UserProfile.objects.get(user=request.user)
        context = {'latest_question_list': latest_question_list, 'user.is_authenticated':
True, 'username': request.user.username, 'userid': useractiv.ref_id }
        return render(request, template, context)
```

This part of the code deals with the GET type requests that the user makes towards the home page of the educational platform.

```
class Solution(models.Model):
    problem = models.ForeignKey(Problem)
    user = models.ForeignKey(User)
    code = models.TextField()
    score = models.IntegerField(default=0)
    Language_CHOICES = (
        ('cpp', 'C++'),
        ('c', 'C'),
        ('py3', 'Python 3'),
        ('py2', 'python 2'),
        ('java', 'Java'),
    )
    log = models.TextField()
    language = models.CharField(max_length=10, default="C++", choices=Language_CHOICES)
    debugOutput = models.CharField(max_length=4000, blank=True)
    def __str__(self):
        return str(self.id)
```



This part of the code represents the form for the solutions sent by the user and verifies that all the data is correct before sending the solution to the database to be tested.

```
from bash import bash
import time
import uuid
import os

def get_ref_id():
    ref_id = str(uuid.uuid4())[6].replace("-", "").lower()
    return ref_id

def RunCode(solution_id, problem_id, problem_name, code, datains, dataouts, language, time, memory):
    solution_id_provision = get_ref_id()
    cmd = ("sh prepare.sh %s %s %s" %(problem_name, solution_id_provision, language))
    b = bash(cmd)
    print(os.getcwd())
    f = open(os.path.join('./compiling/%s/%s/' %(problem_name, solution_id_provision), "%s.%s"
    %(solution_id_provision, language)), 'r+')
    f.write(code)
    f.close()
    i=1
    for datain in datains:
        cmd = ("touch ./compiling/%s/in/input%s.txt" %(problem_name, i))

        t = bash(cmd)

        f = open(os.path.join('./compiling/%s/in/' %(problem_name), "input%s.txt" %(i)), 'r+')
        f.write(datain)
        f.close()
```

This piece of code here deals with the preparation of the environment for compiling and running the solution sent by the user.

## 4. Frontend Development

This section of the presentation will deal with the Frontend development of this project, as in design and structure.

The platform is structured in a simple but modern way. The first page is composed of a slider and a navigation bar that is accessible at any moment, on any section of the platform. The navigation bar opens the possibility of fast change between the different sections of the platform, such as the solution archive, the archive of problems structured by grade and the tutorials. The bar updates itself at the moment of a login, the registration button being replaced by a profile button and the login button by a logout button.

The design, although simple at the moment, is modern and user friendly. We tried our best to avoid color schemes and designs that tire the eyes. Thus, we avoided neon colors and used a more subtle scheme of colors that focuses on nuances of blue and white, that don't put as much stress on the eyes on a long session of usage.

There are a few images on the platform. They are either created by us or used under an open-source license.

In the following part of the section, some pieces of code will be detailed and explained.

```
<li><a href="/register/">Register</a></li>
<li class = "dropdown">
```

```
<a class = "dropdown-toggle" href = "#" data-toggle = "dropdown">Log in</a>
<div id="Login" class="dropdown-menu" style = "padding: 26px; padding-bottom: 15px;">
  <form action="/login/" method="post" accept-charset="UTF-8">
    {% csrf_token %}
    <input id="username" placeholder="Username" style="margin-bottom: 15px;"
type="text" name="username" size="30" />
    <input id="password" placeholder="Password" style="margin-bottom: 15px;"
type="password" name="password" size="30" />
    <input class="btn btn-primary" style="margin-bottom: 15px; clear: left; width: 100%;
height: 32px; font-size: 16px;" type="submit" name="commit" value="Sign In" />
  </form>
</div>
</li>
{% endif %}
</ul>
<ul class="nav navbar-nav">
  {% if user.is_authenticated %}
  <li id="Solutii"><a href="/solutions/">Solutii</a></li>
  {% endif %}
</ul>
{% endblock %}
```

This piece of code is the small overlay/drop-menu from the main page used to log in to your account.

## 5. Technical Details

The following part of this presentation is about the technical details such as technologies used and the creation of the platform.

The platform was developed using a rather big amount of programming languages. For the visual part, we used mainly HTML code, JavaScript and CSS, as well as images created in Gimp. The HTML, JavaScript and CSS was all tied together by a framework named Bootstrap. Bootstrap is a framework that was initially developed by Twitter, with the main objective of producing websites that can adapt to any screen resolution, be it tablet, phone or PC. This framework is also used as a template for CMS systems like WordPress or Joomla.

The database was created and is managed using a relational database system known as PostgreSQL ([en.wikipedia.org/wiki/PostgreSQL#Security](http://en.wikipedia.org/wiki/PostgreSQL#Security))[5] which is open-source.

The Backend programming is realized, in a big proportion, in Python ([en.wikipedia.org/wiki/Python](http://en.wikipedia.org/wiki/Python))[7], a dynamic programming language which is more and more used in today's society. However, the base version of Python didn't have all that was necessary for our platform to work, so we used a system of package management for Python, known as PIP. With the help of PIP, we installed a framework for Python known as Django. Django is a high-level web framework which encourages fast and clean development of websites. Besides this framework, we used other packages like Pillow, Django Simple Captcha, Six and Bash.

The creation of this platform wasn't as simple as it seemed in the beginning. Being made by a small team of two people supervised by another two, we needed to divide our tasks in a fair way. In order to work as efficient as possible, we used a host for source code which opened to us the possibility of simultaneously working on different parts of the platform.

As for the way the platform functions, for compiling the sources, we use the newest version of GCC for sources sent in C/C++ languages. For Java, we use OpenJDK. Ruby has its own interpreter, as does Python with its official interpreter.

In the section dealing with the Backend development, we mentioned the security module for Linux named seccomp ([code.google.com/p/seccompsandbox/wiki/overview](http://code.google.com/p/seccompsandbox/wiki/overview))[6]. Even though we mentioned an exploit and the way it works, we didn't explain why such a module is so important. This security module is extremely important because of the way it interacts with the memory allocated by the user in a certain program. It considers all the memory as possibly dangerous for the system.

## 6. Final Ideas and Future Development

In conclusion, even though the educational platform is hosted on a free server that doesn't support a great deal of users, it is functional and available at any time, with the only exception being during server maintenance.

When it comes to future development, we have in plan the following ideas but we are not bound by them:

- The possibility for source codes to be sent in as many programming languages as possible
- Changing the name and the hosting of the platform
- Creating a forum for the user to socialize or discuss solutions for different problems on
- Creating a blog or a news section for official announcements
- Introducing competition rounds
- Making the educational platform available in English
- Creating a newsletter system

## 7. References

- [1] [en.wikipedia.org/wiki/Fork\\_bomb](http://en.wikipedia.org/wiki/Fork_bomb)
- [2] [en.wikipedia.org/wiki/System\\_call](http://en.wikipedia.org/wiki/System_call)
- [3] [en.wikipedia.org/wiki/Seccomp](http://en.wikipedia.org/wiki/Seccomp)
- [4] [en.wikipedia.org/wiki/SHA-2](http://en.wikipedia.org/wiki/SHA-2)
- [5] [en.wikipedia.org/wiki/PostgreSQL#Security](http://en.wikipedia.org/wiki/PostgreSQL#Security)
- [6] [code.google.com/p/seccompsandbox/wiki/overview](http://code.google.com/p/seccompsandbox/wiki/overview)
- [7] [en.wikipedia.org/wiki/Python](http://en.wikipedia.org/wiki/Python)
- [8] [conferences.ulbsibiu.ro/conf.pcid/2015/volum/Volum\\_Conf\\_Nat\\_Inf\\_Elevi\\_PCID\\_2015.pdf](http://conferences.ulbsibiu.ro/conf.pcid/2015/volum/Volum_Conf_Nat_Inf_Elevi_PCID_2015.pdf)

Vlad Teodorescu  
C. N. Mircea cel Batran  
Software Educațional  
Rm. Valcea, Valcea  
Romania  
[vladjohn2013@gmail.com](mailto:vladjohn2013@gmail.com)

Vlad Badoiu  
C. N. Mircea cel Batran  
Software Educațional  
Rm. Valcea, Valcea  
Romania  
[vlad\\_andrew@openmailbox.org](mailto:vlad_andrew@openmailbox.org)

Mirela Mlisan  
C. N. Mircea cel Batran  
Software Educațional  
Rm. Valcea, Valcea  
Romania  
[mirela\\_mlisan@yahoo.com](mailto:mirela_mlisan@yahoo.com)

Antoanela Fărcașanu  
C. N. Mircea cel Batran  
Software Educațional  
Rm. Valcea, Valcea  
Romania  
[antfarm@yahoo.com](mailto:antfarm@yahoo.com)

# Smart Home Open Framework

Adrian Alexandru Bărbos, Kim Răzvan Velker, Andrei Călbăjos

Teacher Coordinator: Camelia-M. Pinte

## Abstract

The aim of the paper is to present a new smart home technology using an Android device as the hub to control home appliances, as heaters, AC, fans, etc. via a profile-enabled cloud smart home open framework. The entire smart home system will be managed through a user friendly application.

## 1 Introduction

We don't like to manually tweak the heater/AC in the room. We also don't like the pseudo-smart proprietary thermostats that will never understand why sometimes we need to keep the door open when we sleep. Or why we sometimes like the cold air in the room. Other times we don't. But mostly we hate when we go somewhere and can't bring our cosy home profile with us. Or combine our profiles with others when we share a room.

We're imagining a home that learns our behaviour and controls the appliances, and is kind enough to accept and combine other behaviours so we can socially share our environment in a smart and meaningful way.

For this to happen we need our home to have an easy to manage, open framework that handles both the user and the home appliances. Using Android as the MVD, we can leverage this need using today's open source technologies.

## 2 Technologies used for the framework

### 2.1 Android

A mobile operating system developed by Google. The Android operating system is primarily used in touchscreen mobile devices, such as cell phones and tablets. Its design allows users to manipulate mobile devices intuitively, with phone interactions following common motions such as pinching, swiping, and tapping.

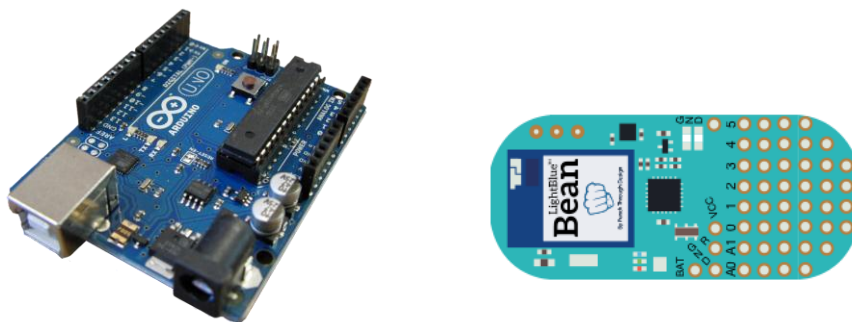
Android application development is done in Android Studio which is the official IDE for Android application development, based on IntelliJ IDEA

## 2.2 Arduino

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can communicate with software running on your computer (e.g. Flash, Processing, MaxMSP.) The boards can be assembled by hand or purchased preassembled the open-source IDE can be downloaded for free.

The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.



## 2.3 Light Blue Bean

The LightBlue Bean is a low energy Bluetooth Arduino microcontroller. Using Bluetooth 4.0, it is programmed wirelessly, runs on a coin cell battery, and is perfect for smartphone controlled projects.

## 2.4 PHP & MySQL

PHP and MySQL are Open Source, meaning that they are free development tools, and there is a large community of dedicated volunteer programmers who contribute to make improvements and are continuously adding features to it. The development tools and database servers that require licensing costs have limited programming resources compared to open source development tools, which have an enormous and fast growing dedicated and knowledgeable community that extends around the world.

We have found that, PHP and MySQL are the best development tools to be used when developing a cloud platform for collecting data from the Smart house sensors.

## 2.5 MQTT

MQTT, also known as Message Queue Telemetry Transport, is a publish-subscribe based “light-weight” messaging protocol for use on top of the TCP/IP protocol. It is designed for connections with remote locations where a “small code footprint” is required and/or network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker. The broker is responsible distributing messages to interested clients based on the topic of a message.

In software architecture, publish-subscribe is a messaging pattern where senders of messages, called publishers, do not program the messages to be sent directly to specific receivers, called subscribers. Instead, published messages are characterized into classes, without knowledge of what, if any, subscribers there may be. Similarly, subscribers express interest in one or more classes, and only receive messages that are of interest, without knowledge of what, if any, publishers there are.

### 3 The Smart home open framework

Smart Home Open Framework is new smart home technology using an Android device (MVD) as the hub to control home appliances (heaters, AC, fans, etc) via a profile-enabled cloud ( Figure 1 ) All the data will be gathered from the LBBs, also known as Light Blue Bean, which is a small Bluetooth device with sensors. This device is able to read and send an array of sensors and their values and also set the temperature of their according thermostat. The thermostat is, for the time being, a temporary LED, which will provide user feedback.

The data is gathered by the MVD, a Minimum Viable Device with Android software. This is the central hub of the framework that receives and sends data both to the cloud and to the LBBs. The device connects to the LBB via Bluetooth, it sends data to the cloud via a HTTP POST, and it receives data from MQTT via the publish-subscribe messaging system.

**Mobile App.** With the help of an Android and iOS application, the user can view his own profile and change any current settings he or she has. The application will interact with the MVD in order to access every endpoint of the framework (eg. thermostats, light switches, etc) and tweak them to get the required result.

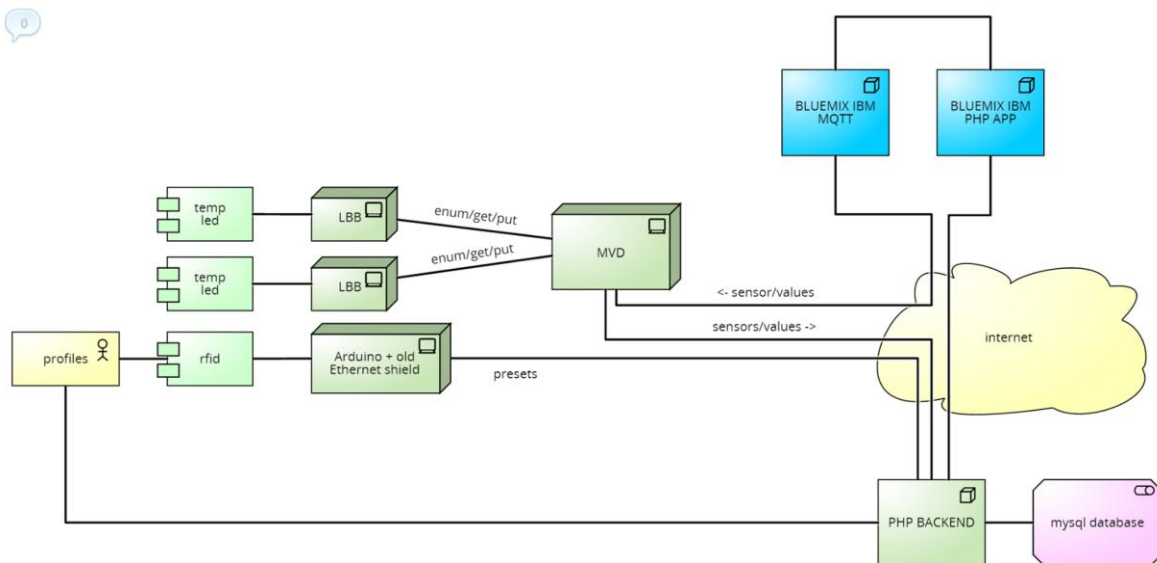


Figure 1 Smart Home Open Framework Infrastructure

## 4 Conclusion and Further Work

The current work is about introducing a new smart home technology with an Android device. The project could be further improved. All the data received from the MVD could be processed by a machine-learning algorithm, like Azure ML, so that, in time, the system learns, by itself, what our optimal living conditions and preferences are, thus setting the current profile accordingly. A profile could be activated using a RFID-module and a card, when the user enters his home. The RFID module connected with the MVD will set the profile of the user.

**Acknowledgement.** The authors would like to thank Professor Pop C. Petrica for his kind and continuous support. The current project qualified and was effectively implemented during a Heckaton at Lund University in April 2014.

## References

- [1] StackOverflow - question and answer site for programmers <http://stackoverflow.com>.
- [2] PHP NET-complete and hyperlinked manual documents and explains all elements of PHP. <http://php.net/manual/>
- [3] AdaFruit Industries - Unique & fun DIY electronics and kits. <https://www.adafruit.com>
- [4] Arduino official - boards, shields, kits and accessories. <http://www.arduino.cc>
- [5] Android Developer suite, <http://developer.android.com/>
- [6] Apple Developer suite, <http://developer.apple.com>

Kim VELKER  
Technical University  
Cluj-Napoca  
Faculty of Science  
76 Victoriei  
ROMÂNIA  
kim.velker@trecadis.ro

Adrian BĂRBOS  
Technical University  
Cluj-Napoca  
Faculty of Science  
76 Victoriei  
ROMÂNIA  
adrian.barbos@trecadis.ro

Andrei CĂLBĂJOS  
Technical University  
Cluj-Napoca  
Faculty of Science  
76 Victoriei  
ROMÂNIA  
andrei.calbajos@trecadis.ro

# Aviation HUD

**Cristian Beckert**  
**Teacher Coordinator: Delilah Florea**

## Abstract

A *Head Up Display* is a transparent display that presents data without requiring users to look away from their usual viewpoints [1]. They were initially developed for military aviation, but HUDs are now used in commercial aircraft, automobiles and other.

"**Aviation HUD**" is an application for android which brings the altitude, flying speed and coordinates in front of you, therefore giving you the advantage of a much quicker and comfortable way to check the data whenever you want to, without losing precious time nor considerably changing your field of view. Speed, altitude, longitude and latitude are taken from your device's GPS.

## 1. Introduction

### 1.1 Brief Description

This article focuses on the following application created for the Android Operating System (Android OS) entitled "**Aviation HUD**". The main goal of the created app is to bring more comfort in the lives of hard-working pilots and improve their efficiency. These could be achieved by making the cockpit's environment more users friendly through a well thought and organized process of gathering and displaying vital flight-related information. There are a small number of applications that have been developed with the same general idea, but certain features and utilities are either scattered among various apps or are simply non-existent. Furthermore they seem to be mostly intended for car/automobile drivers, which barely resemble meeting the needs of a pilot during his flight. Therefore, the AHUD will solve some of these inconveniences.

Through further reading of this document, the audience shall become familiar with the idea that sparkled and fueled this project. By doing so, the rough mechanics of its functionality would be easier to understand. A better comprehension should also make a clear connection between the program's actions, most of which are seen by the user, and its underlying code, understood by the executing machine and perfected by human programmers.

The core of this article consists of shedding some light upon the methods implemented up until this first functional release of the programs. Also various bugs or flaws may be discussed with the thought of solving as many of them as possible. By doing so, the AHUD's performance and the experience of using it are to be improved. The application was first developed and introduced at the pupils' conference PCID 2015 at the Mathematics and Informatics Department of the "Lucian Blaga" University from Sibiu, after this it was completed and developed further and presented here. [4]



## 1.2 Development History

Flying an airplane, as a beginner, could be overwhelming. The amount of information a pilot has to process is immense. The normal cockpit is still filled with numerous gadgets and all of them present different types of data. In simple terms, the height and speed of an airplane are equivalent to the gear and speed of a car, as long as the driven car uses a manual transmission. Beyond these major common features, the airplane takes off and engulfs the pilot's attention with additional information like a compass, pressure indicators and a pitch and roll indicator. Depending on the aircraft, there could also be found various lights acting as indicators, the most common being the one signaling the fact that the engine is active/turned on. Because of safety concerns, all of them have to be checked regularly by the person in charge of the flight.

This is the part that discourages most of the people willing to fly an airplane. It may sound easy that all that the pilot ought to do is keeping the aircraft steady, in the air and at the same time check the required information. Most of the people are familiar with the side effects of such actions and the best example are the fresh or soon-to-be car drivers and bicycle riders. There are few that are able to look in one direction without making the vehicle turn towards where their view is focused. On land, the movement takes place on a forward-backward axis and a left-right. This means, as long as, the vehicle or being is in contact with the ground, the movement is carried out in a two-dimensional space. Airplanes fly and, similarly to birds, they go up and down too. Also they are able to perform turns. It may not be the case for a plane to “fly in reverse”, if that is in any way possible, but helicopters do. Though still having the first two axes, a third one represents the up-down movement and defines the motion as being three dimensional (3D).

The lack of attention in a 2D space could be fatal and making errors is part of being human. On one hand is learning how to survive them that brought the civilization where it now is. On the other hand, forcing the attention to divide itself even more in order to survive the process of flying may not be easily described as a comfortable thought. It is tiring for the brain because paying attention is a very active on-going process, which drains great amounts of psychological energy. Also it is one of the main reasons why in order to become a good pilot, one is required to endure a long and thorough training.

Keeping in mind all the arguments, flying represents something special for me and all the commonly known dangers that would instill fright in numerous persons did not hold me back. I have been learning how to fly a small aircraft for a period of time and I was indeed overwhelmed at the beginning. Despite becoming familiar with the act of checking the cockpit's situation, there should be an easier method or at least safer. If all the information would be concentrated in one place that would mean less time spent switching focus from one gadget to another and more time left to actually fly. Moreover, as the thoughts keep flowing, somehow positioning that information as close as possible to the main area of a pilot's attention seems to make perfect sense: having to barely move the eyes from where they should be looking, would reduce the reaction time required in an emergency case. The flight would actually become easier and safer for everyone on board and on land.

The thought of using present technology to achieve such an accomplishment does not seem the same far-fetched idea it was a couple of decades ago. Science itself has seen an unimaginable progress in the past half-century. In addition to this, many of them were more or less pushed forward because of competition, be it diplomatic or military. However, the military competition is often the more active and keen on taking risks. This type of approach leads to an increased amount of experiments and therefore more innovation and development of different concepts.

Admitting, such an idea of helping pilots through making information easily available in front of their eyes resembles science-fiction movies, but it is the army that made it become reality. The technology devised bears the name of a Heads-Up Display (HUD) for obvious reasons: the classical gauges that present flight data to the pilot have the tendency of being located lower than the field of view on which the attention is actually focused. AHUD brings all that data to a higher viewpoint, preventing the pilot from looking down from the flight scene, which in turn means he will keep his head up without useless interruptions. The person flying the vehicle remains more focused on the flight and needs less time to react when switching between the focus points. This translates into a more relaxed and efficient pilot.

From an ordinary citizen point of view, anyone could tell that the budget would be limited. Despite the finite funding sources, the use of smartphones has become increasingly common and affordable nowadays. There is a great chance that you, as a reader may own one of these pocket-sized intelligent computers or even a real HUD if you happen to read this at least a decade later as of the time of this article's writing. Android OS is what helped smartphones gain momentum in the year 2012 [2] and makes use of Google's Play Store to enrich the users' experience with the help of countless apps, many of which are actually free of cost.

Since the projector technology has not progressed too much in the holographic area, a classic projector may actually impede the pilot from seeing something, if at all because of the bright light reflecting directly into the pilot's eyes. The windshield handles decently well the act of reflecting the top of the dashboard and is cheap (as long it is not broken and needs replacement), as it usually comes attached to the vehicle meant to be driven or flown. As a consequence, if smartphone with a bright enough screen lays on top of the dashboard, the information it displays would be reflected on the windshield, in the field of view of the person needing it. Last but not least, there would actually occur a mirroring effect, which flips the displayed information and could be counteracted by flipping the image in the beginning so it would appear normal to the one who sees it.

In conclusion a smartphone is acceptably affordable, depending on tastes and preferences. An application to run on its OS may even be free of charge and every airplane has a mirroring windshield. Besides buying an entire plane, a HUD for amateurs that rent one every now and then to fly is possible.

## 2. Implementation details

The interface of the Heads-up Display (HUD) present us with some of the most important informations for the pilots flight like position, altitude and speed and would appear as shown in the image below.

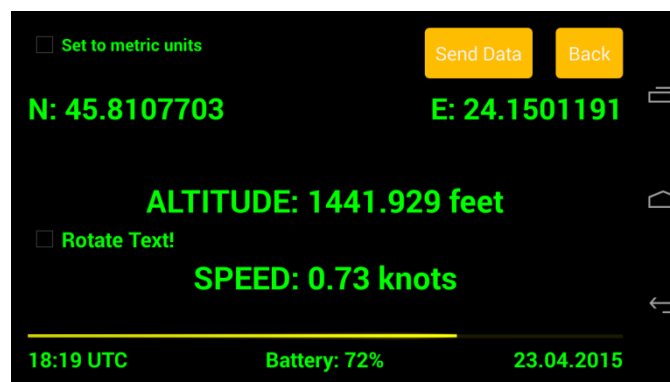


Fig. 1: Not rotated text, value expressed in aviation units

The application shows the position through coordinates N, E and allows the user to change between measurement units, i.e. feet to meter so the pilot can also report this during the flight.

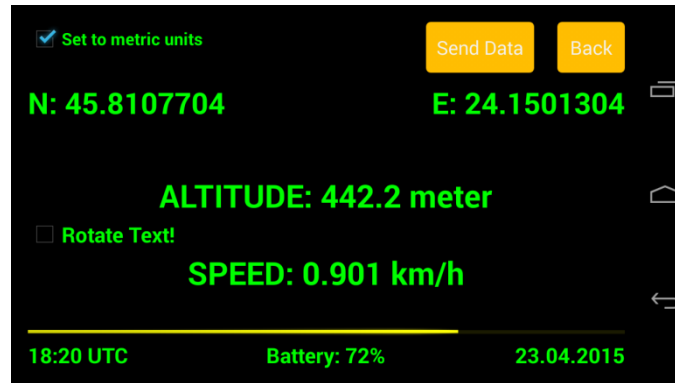


Fig. 2: Not rotated text, value expressed in metric units

The application has also some facilities like showing the time related to UTC, the current date and also the state of the battery, so the pilot knows which the state of his mobile device is.

## 2.1 Application interface

The application's interface has a simple design, which facilitates the data's interpretation. The colors are chosen to create a contrast between the screen and the text, so the pilot shall not meet any problems in reading the information, whether in bright sunlight, or in a cloudy day. In the future updates, the user may choose his favorite colors.

The main interface presents the text that is rotated 180° on its x-Axis so it can be reflected on the plane's windshield.

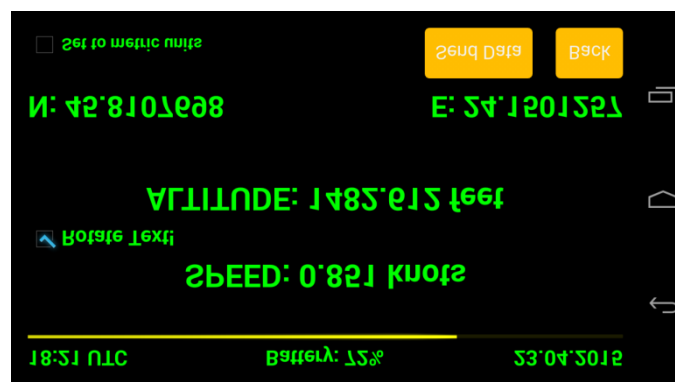


Fig. 3: Rotated text, values expressed in aviation units

The values for the altitude speed and coordinates are taken from the GPS and are continuously updated.

```
private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 1; //1 meter  
private static final long MIN_TIME_BW_UPDATES = 1000; //1 second
```

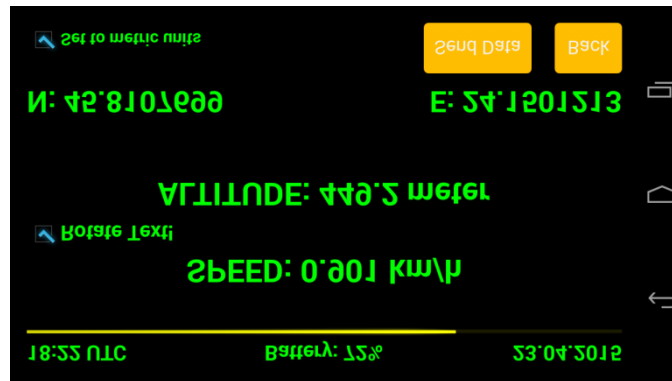


Fig. 4: Rotated text, values expressed in metric units

## 3. Development and Implementation

### 3.1 Programming Environment

The application was developed in the IDE Eclipse, using the object-oriented programming language Java. For compatibility reasons, the Android SDK was used in order to deploy the application for the operating system Android.

### 3.2 Section of Code

#### 3.2.1 The layout

The most important characteristic of the layout is that the text is rotated, that is a very useful characteristic for the pilot. If the text was not rotated, the pilot couldn't read it. The following code represents the layout of the text:

```
android:id="@+id/textSPD"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignParentLeft="true"  
android:layout_alignParentTop="true"  
android:layout_marginTop="20dp"  
android:text="@string/SPD"  
android:textAppearance="?android:attr/textAppearanceLarge"  
android:textColor="@color/green"  
android:textSize="30sp"  
android:textStyle="bold"  
android:rotationX="180"
```

#### 3.2.2 Calling the GPS

The entire concept of the application depends on the GPS of the phone.

```
LocationManager locationManager =
(LocationManager)this.getSystemService(Context.LOCATION_SERVICE);
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, this);

this.updateSpeed(null);
location.setUseMetricunits(this.useMetricUnits());
nCurrentSpeed = location.getSpeed();
nCurrentAltitude = (float) location.getAltitude();
nCurrentLongitude = (double) location.getLongitude();
nCurrentLatitude = (double) location.getLatitude();}
Formatter fmt = new Formatter(new StringBuilder());
Formatter fmt2 = new Formatter(new StringBuilder());
Formatter fmt3 = new Formatter(new StringBuilder());
Formatter fmt4 = new Formatter (new StringBuilder());
fmt.format(Locale.US, "%5.1f", nCurrentSpeed);
fmt2.format(Locale.US, "%5.1f", nCurrentAltitude);
fmt3.format(Locale.US, "%5.1f", nCurrentLongitude);
fmt4.format(Locale.US, "%5.1f", nCurrentLatitude);
String strCurrentSpeed = fmt.toString();
String strCurrentAltitude = fmt2.toString();
String strCurrentLongitude = fmt3.toString();
String strCurrentLatitude = fmt4.toString();
strCurrentSpeed = strCurrentSpeed.replace(' ', '0');
strCurrentAltitude = strCurrentAltitude.replace(' ', '0');
strCurrentLongitude = strCurrentLongitude.replace(' ', '0');
strCurrentLongitude = strCurrentLatitude.replace(' ', '0');
TextView txtCurrentSpeed = (TextView) this.findViewById(R.id.txtCurrentSpeed);
txtCurrentSpeed.setText(strCurrentSpeed + " " + strUnits);

TextView txtCurrentAltitude = (TextView) this.findViewById(R.id.txtCurrentAltitude);
txtCurrentAltitude.setText(strCurrentAltitude + " " + strUnits2);

TextView txtCurrentLongitude = (TextView) this.findViewById(R.id.txtE);
txtCurrentLongitude.setText(strUnits3 + " " + strCurrentLongitude);

TextView txtCurrentLatitude = (TextView) this.findViewById(R.id.txtN);
txtCurrentLatitude.setText(strUnits4 + " " + strCurrentLatitude);
```

The speed, altitude and coordinates variables are initialized and called from the GPS. Then, the variables are showed in the layout.

### 3.2.3 Conversion

The function `getSpeed()` calls the speed expressed in meters per seconds. This unit is not suitable for the horizontal speed and needs to be converted in kilometers per hour, or, as some pilots prefer, in knots. The same idea works with the altitude. Firstly, the distance between the sea level and the aircraft is expressed in meters, but some of the users may want the distance expressed in feet. The following part of code shows the conversion of the measurement units:

```
@Override
public double getAltitude() {
    double nAltitude = super.getAltitude();
    if(!this.getUseMetricUnits()) {
        nAltitude = nAltitude * 3.28083989501312d;
    }
}
```

```
        return nAltitude;
    }

    @Override
    public float getSpeed() {
        float nSpeed = super.getSpeed();
        if(!this.getUseMetricUnits()) {
            nSpeed = nSpeed * 2.2369362920544f;
        }
        return nSpeed;
    }
}
```

### 3.2.4 Storing the information

There is also the possibility to store the information that will be displayed, for the moment by sending it per email.

```
buttonSend = (Button) findViewById(R.id.btnSend);
buttonSend.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        String to = getString(R.string.adress);
        String subject = ("Flight Log");
        String message;
        if(chkUseMetricUnits.isChecked()){
            message = (gmtTime + " UTC" + "\n"
                + "- SPEED: " + strCurrentSpeed + " km/h" + "\n"
                + "\n"
                + "- ALTITUDE: " + strCurrentAltitude + " m" + "\n"
                + "\n"
                + "- Longitude: " + longitude + "\n"
                + "\n"
                + "- Latitude: " + latitude);}
        else { message = (gmtTime + " UTC" + "\n"
            + "-SPEED: " + strCurrentSpeed + " knots" + "\n"
            + "\n"
            + "-ALTITUDE: " + strCurrentAltitude + " feet" + "\n"
            + "\n"
            + "-Longitude: " + longitude + "\n"
            + "\n"
            + "-Latitude: " + latitude);}

        Intent email = new Intent(Intent.ACTION_SEND);

        email.putExtra(Intent.EXTRA_EMAIL, new String[]{ to});
        email.putExtra(Intent.EXTRA_SUBJECT, subject);
        email.putExtra(Intent.EXTRA_TEXT, message);

        email.setType("message/rfc822");
        startActivity(Intent.createChooser(email, "Choose an Email client:"));
    }
});
```

The application has also the facility of writing the information in a file on the device, so the information parts could be accessed also from here. This is called a log file and could be helpful to manage the flight.

```
try {
    File log = new File("/sdcard/AHUD/log.txt");
```

```
        if(!log.exists()){
            Toast.makeText(getApplicationContext(), "We had to make a new file.",
            Toast.LENGTH_SHORT).show();
            log.createNewFile();
        }

        FileWriter fileWriter = new FileWriter(log, true);

        BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);
        bufferedWriter.write("date: " + currentDateandTime + "\n");
        bufferedWriter.close();
        Toast.makeText(getApplicationContext(), "Done writing SD 'log.txt'", Toast.LENGTH_SHORT).show();
    } catch (Exception e) {
        Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
    }
    try {
        Thread.sleep(10000);
    } catch (InterruptedException e) {}
}
```

The safety system of the application sends this kind of e-mail if the phone has less than 5% battery, in case it dies and the pilot gets lost, so that other people can know the last location, speed and altitude.

The application will enable some warnings alerts for the pilot if flight parameters are wrong. If the plane has too much speed, or if the altitude is too high. These warnings will be only visual, as audio warnings are useless due to the loud noise from the motor.

### 3.2.5 Permissions

The application will ask the user for some permission in order to function properly and do its job. The permissions are written in AndroidManifest.xml file and look like this:

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

The permission “Access fine location” allows the app to access precise location from location sources such as GPS, cell towers, and Wi-Fi, while “Access coarse location” grants the permission to access approximate location derived from network location sources such as cell towers and Wi-Fi.

At last but not least, “write external storage” permission allows writing to the external storage and is used to write the log file and to “remember” the settings (email address, layout, measurement units and the airport from metar).

Also, the layout attribute `android:keepScreenOn="true"` prevents the phone's screen from turning off in order to avoid the uncomfortable situation of having to turn on the screen in what could be critical circumstances.

### 3.2.6 METAR

Another feature is the function of displaying the weather forecast, known in aviation as METAR. This can be useful if the pilot decides to make a longer flight and wants to check the weather conditions. The user has the possibility to choose the airport on which the METAR was issued by entering the ICAO code of the wanted airport.

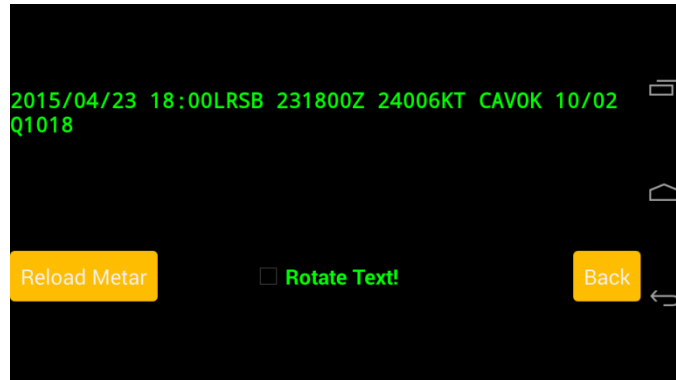


Fig. 5: METAR

## 4. Conclusion and further work

### 4.1 Conclusion

All in all, the Android application “Aviation HUD” could prove to be a useful component of aviation and has the potential to appeal to beginners, as well as to experienced pilots too. The interface has been built with a user-friendly design in mind allowing for timesaving and comprehensible reading.

After several tests, results have proven that besides its former use, the app may as well find its place in a regular vehicle, or anything else which has a windshield, providing the driver with the same useful and accurate data.

### 4.2 Further work

Aviation HUD has many ways to be improved. The following paragraphs describe possible options for the further development, which are not accessible in the application at the time being.

#### 4.2.1 The layout

At the moment, the layout is very minimalistic. The results based on testing the application in real conditions should help create a better design, also with the possibility to change the display colors so they could match best with the weather conditions.

In addition, the interface's customization may help to best meet the user's needs. As an example, individual colors may be chosen for background or text, as well as which data the HUD displays.

#### 4.2.2 Warnings

Another update of the application will enable some other warnings than the existing one and alert the pilot if flight parameters are wrong. For example if the climb/descend rate is too



fast, or the bank angle is too wide. These warnings will also be only visual, as audio warnings are useless due to the loud noise from the motor.

#### 4.2.4 Altitude

There is also the possibility of showing the altitude or the height on the screen. In some cases it's good to know the distance from the plane to the ground, not the distance from the aircraft to the sea level. These cases are mostly when taking off and landing on the same runway.

#### 4.2.5 GPS Accuracy

A regular GPS in commercial phones has an accuracy of about 30m at best. Taking into account combination of all parameters of GPS, Wi-Fi and cell tower triangulation, it goes up to 10m. In aviation, even 1 meter can make the difference between a smooth landing and a crash. The HUD is mostly used while taking off or landing, and providing a great accuracy makes the app more useful. Of course, a pilot shouldn't use only "Aviation HUD" while taking off or landing, he should use his eyes and guide himself with a landmark. For automatic landing, there is the ILS (Instrument Landing System) to be used.

## References

- [1] [http://en.wikipedia.org/wiki/Head-up\\_display](http://en.wikipedia.org/wiki/Head-up_display) – February 23, 2015
- [2] <http://blog.gsmarena.com/category/android/>
- [3] <http://www.javaexperience.com/measure-and-display-android-device-speed/> - February 27, 2015
- [4] Cristian Beckert, **Aviation HUD**, Volum PCID, 2015  
[http://conferences.ulbsibiu.ro/conf.pcid/2015/volum/Volum\\_Conf\\_Nat\\_Inf\\_Elevi\\_PCID\\_2015.pdf](http://conferences.ulbsibiu.ro/conf.pcid/2015/volum/Volum_Conf_Nat_Inf_Elevi_PCID_2015.pdf)

Cristian BECKERT  
Colegiul National „Samuel von Brukenthal”  
 Stiinte ale naturii  
Piata Huet nr.5  
Sibiu, Romania  
E-mail: beckertercristy@yahoo.com

# **SymoEngine 2.0**

**István Bialkó**  
**Teacher Coordinator: Grigore Albeanu**

## **Abstract**

This project is aiming to offer a graphical engine, with which anybody can create a virtual world, presentation without the need of any programming skill. Besides the many other engines, my concept differs in the common pipeline and drawing methods. Introducing a new feature that enhances the depth perception.

## **1 Introduction**

Despite the fact that many good graphical engines are available, they might not offer the best solutions to everybody. This engine is designed to work in a non-traditional way, in pursue of performance, easiness in creation and better visual experience.

The program can be used for live, interactive graphical presentations, 2D/3D games, tech demos. Its main strong point lays in the configuration files, so it becomes parametrical, editable without recompiling the program – not just screen resolution or mouse sensitivity, but the structure of scenes with their objects, lights and other components. One of the goals is memory efficiency – each material in the scene contains the data the objects use, while the objects hold the indices. If there are two vertices identical in a buffer, instead of a new insertion, the method will give back an index to that buffer element. This allows us to copy indices, if there are multiple objects using the same source.

In the current development state, it uses OpenGL 3.3 code [1], initial matrices defined in the configuration file for each object. The pipeline is divided in 3, in order not to use stacking and building the scene from the identity matrix for every frame. I used GLM [2] (OpenGL Mathematical Library) to get the matrix classes, transformations to avoid using the deprecated fix-function pipeline. The projection matrices (both 2D and 3D) are contained in the Pipeline, the viewMatrix is in the Camera, while the model matrices are in the objects and groups. This would result in a lot of matrix multiplications, but if some boolean variables are being implemented, unless the matrices are changed – in case of animation-, the matrices can be just passed forward to the shader.

If a taxonomy would be made regarding the engine, by its lighting model, it would be a forward type. It is possible that there is no exact choice for this, mainly because most of the calculations are run by the shaders.

## 1.1 Development history

The first version of SymoEngine was not a failure, but nothing more as a learning phase. I just started learning classes in C++, only knew how OpenGL 2.0 works and it resulted in a chaotic code. It lacked organization, most of the variables were global, and it wasn't easy to extend. Also, the final product would be a concrete game, not a more useful engine.

To overcome these problems, the 2.0 indicates total rework. From the beginning it was designed to be as object oriented as possible. Not one variable is in the global namespace, the classes have their variables, arrays as private. This can cause longer calls when accessing them, but there is no need for two functions (getter, setter), because of the possibility of using references as return value.

```
//In class Scene
vector<Group>& getGroups() { return groups; }

//In class Group
vector<Object>& getObjects() { return objects; }

//printing out object informations before the main loop
for(int i=0; i < scenes.size(); i++) //going through the scenes
{
    cout << "Scene name: " << scenes[i].getName() << endl;
    for(int j = 0; j < scenes[i].getGroups().size(); j++) //going through the groups
    {
        cout << "\tGroup name: " << scenes[i].getGroups()[j].getName() << endl;
        for(int k = 0; k < scenes[i].getGroups()[j].getObjects().size(); k++) //going through the objects
        {
            cout << "\t\tObject name: " << scenes[i].getGroups()[j].getObjects()[k].getName() << endl;
        }
    }
    cout << endl;
}
```

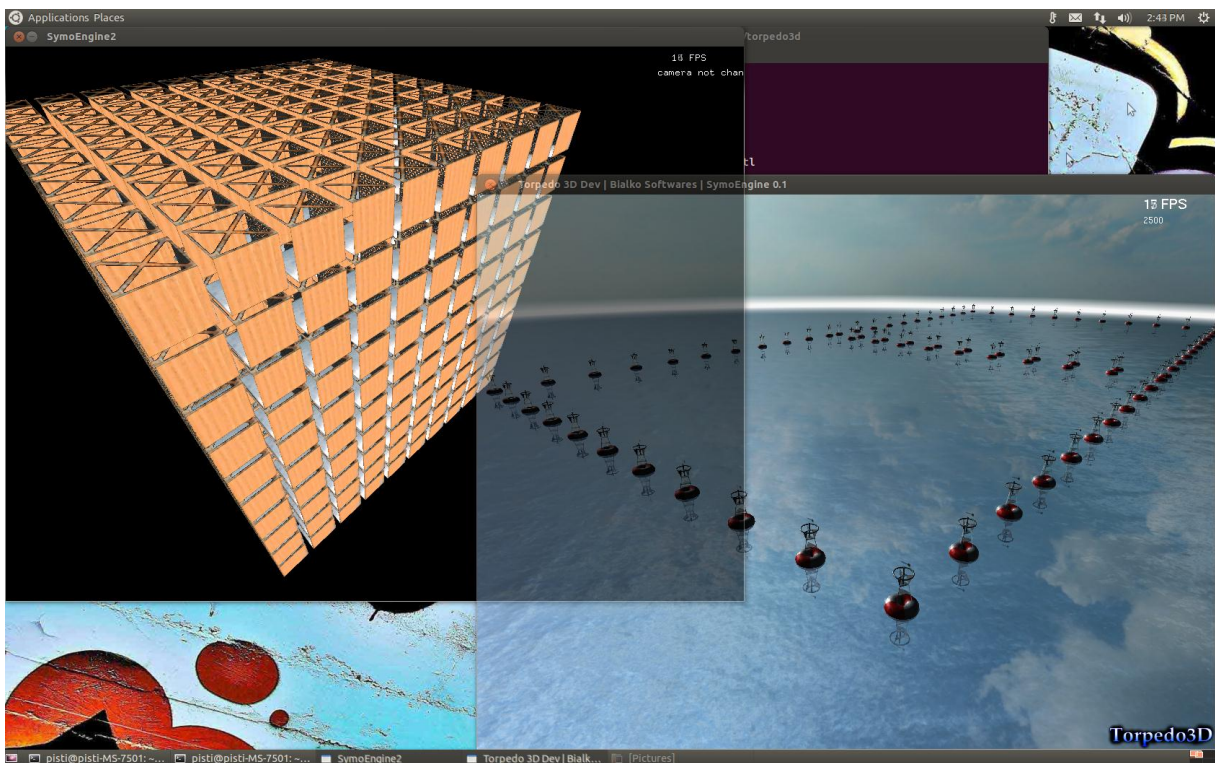
The references are also necessary in this case, because of the structure of classes. They are hidden, therefore not permitting creating local variables with these types.

With the last updates of the engine, it has around 2500 lines of code, which is much more efficient and flexible, easier to use, to expand, than the first version. That has implemented Framebuffers, GUI, but in both files totals a number of 4500 lines.

In the aspect of performance, I generated a simple scene of 1000 multi-material cube, which runs at the same FPS as 80 buoys, a water plane and a sky dome in the old program. (The test scene is built up by 100 groups. The first group loads one cube, from which the indices are copied to the other 9 objects. After the first group is set up, the rest of the objects are flagged "loaded" as well.)

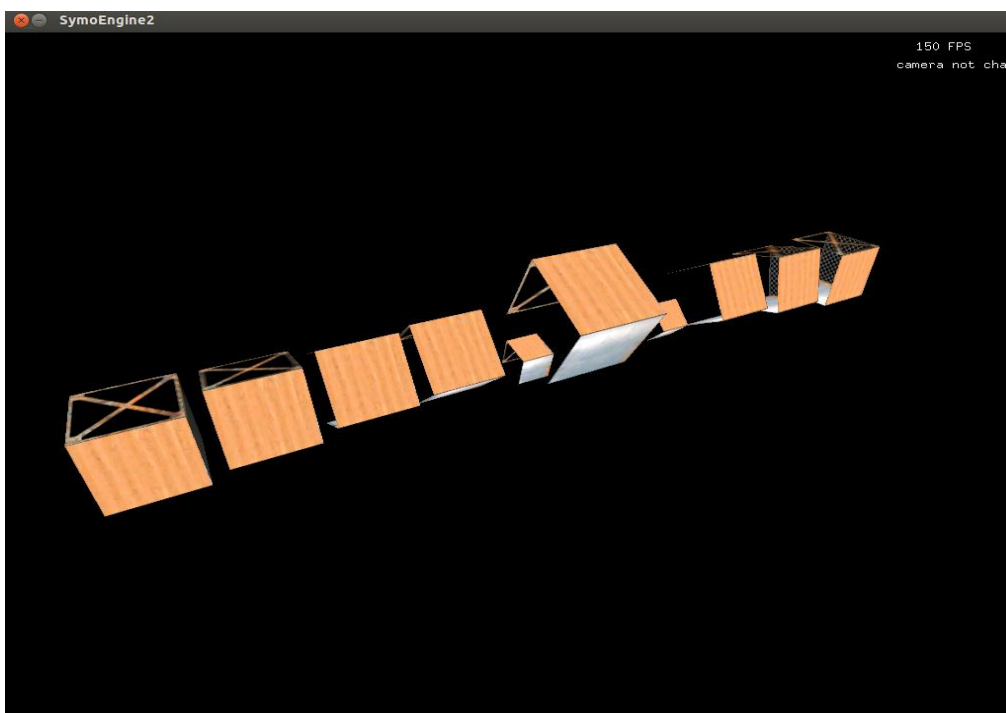
There is another advantage to the new version, which is flexibility in the making. As time goes on with the project, inevitable that problems will occur. Even if it is related to the idea, it can be adapted quickly. For example, the shaders were connected to the object, now they are attached to the materials. Most objects will be using the Composite shader (which is the main one), and this is loaded with the first objects – this will probably be moved somewhere else in the structure. Because of the configuration files, these kind of problems can be solved without restructuring the program, only the location of function calls will be different.

On the next image you can see two different scenes, in the two different versions. The common thing is the frame rate. Both are running at 15-18 FPS, but while on the left there are 1000 cubes, on the right there is only 80 buoys. Modifying the old engine to draw the same amount of objects would be a lot harder. Also, the old version is functioning with a lot of hard-coded variables, names, which is not flexible.



**Figure 0.a: Comparison between the old and the new version**

The flexibility is visible, if we change the scene a little. Let's take away all the groups except the first. Or we could comment out every second, third, what pattern we want. The first group has 10 cubes, each with its respective matrix. We can easily change their size, rotation.



**Figure 0.b: 10 cubes with different rotation angle and size**

If the animation would be implemented, the rotation of the group, as well as of the individual objects would be achieved just by modifying the configuration file.

## 2 Main structure

The engine is written in C++ [4], supported by Simple DirectMedia Layer [3] (SDL) and GLM. The goal was to write it in object oriented style, thus the classes are nested and all the variables are private. Instead of using the C style array declarations, everything is stored in vectors. For dynamic array, I used the new constructor and delete command. Where it is possible, the default character array is defined as string.

### 2.1 Main.cpp

The main.cpp file only creates an instance of the Engine class, the rest of the operations are flowing through the classes. If a child class detects an error, the program quits, or if the user closes the program, it will close the structure from inside.

```
1 #include "main.h"
2
3 int main(int argc, char** argv)
4 {
5     Engine* engine = new Engine();
6
7     engine->engineInit(argc, argv, engine);
8
9     return 0;
10 }
```

Figure 1: Main.cpp

As you notice, the pointer appears as parameter in the Init function. This is a side effect of the nested classes – even if a class is part of another, they are not related. The child class can not access functions or variables of the parent (without an instance), therefore a pointer is needed to be passed on.

### 2.2 Makefile

The program is not distributed in many little cpp files for each class, because the nesting would lose its purpose and the compilation would be more complicated. The project is developed in Ubuntu, but SDL is cross-platform, with minor modifications it can be ported to any supported OS. Because of the different platform, the performance will vary. Also the drivers are a huge factor.

```
1 all:
2     clang++ main.cpp -lSDL2 -lSDL2_image -lglut -lGL -lGLU -lGLEW
3
```

Figure 2: makefile

Glut is only included for 2D texts, for debugging, eventually TTF text will replace it. As the program grows, other libraries will be included for sound (OpenAL), threads (SDL\_thread), libJPEG to save the image.

I chose Clang++ [6] as compiler, because it offers more detail for errors.

## 2.3 Main classes

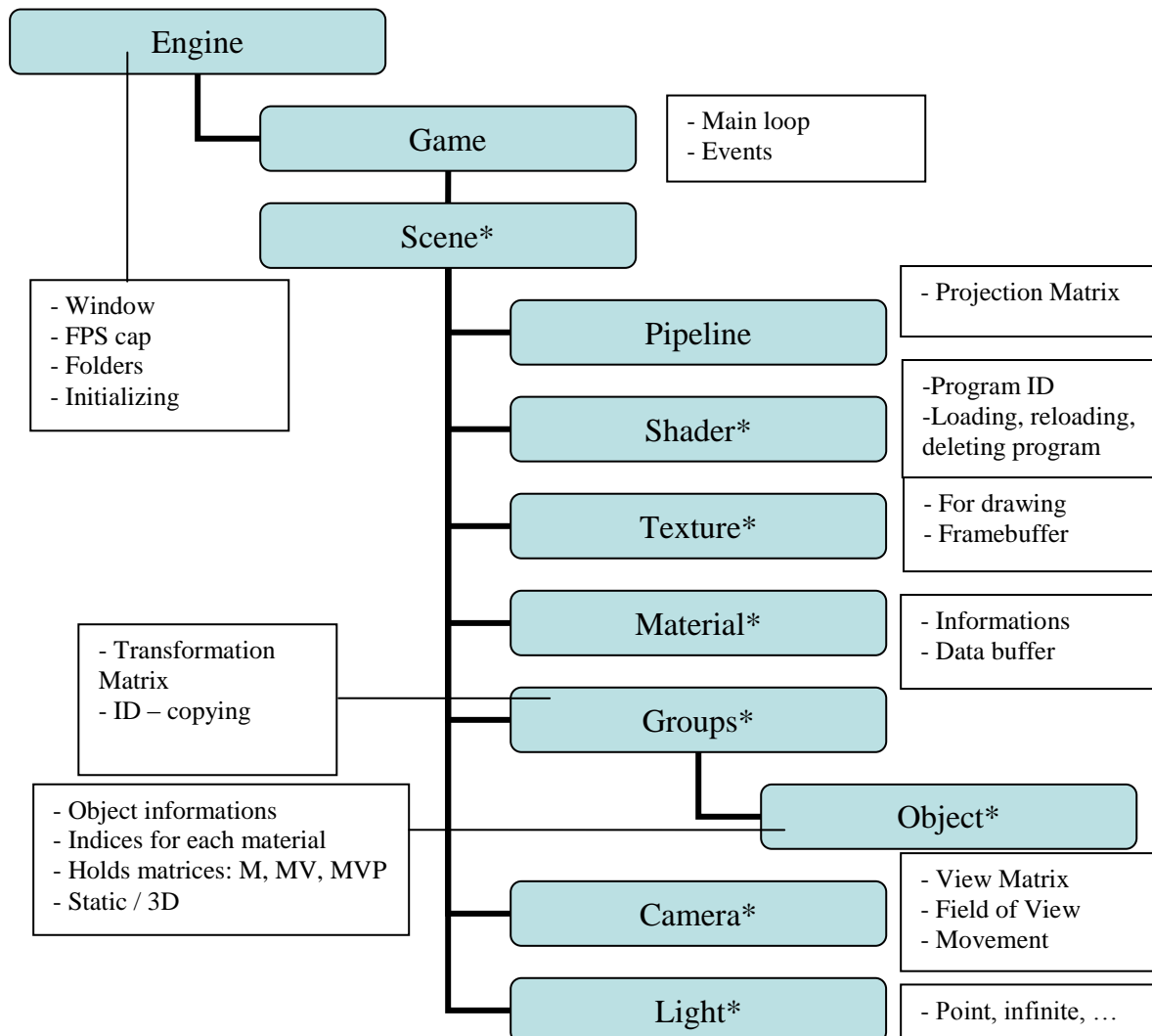


Figure 3: Class structure (\* multiple instances)

## 2.4 Scene.cfg

Currently the following elements are supported by the engine, with their respective properties:

- **S** Scene1: a new scene named Scene1. Everything is identified by a name, to easy debugging and distinguishing.
- **G** Group1 0 0 0 0 0 0 1 0 1 1 1: in order of name; ID; translate vector; rotation angle and axis; scaling. These 3 defines the groups' basic transformation matrix.
- **O** Object1 test/ static 0 0 0 0 1 0 0 1 1 1: in order of name; folder (subfolder of Engine->getObjectFolder() variable) containing the source; "static" indicates that the objects is not in the 3D feature (otherwise any string is accepted); elements of the transformation matrix as in the groups.
- **C** Camera1 25 50 50 0 0 -5 64: in order of name; position; lookAt vector; field of view.



- **L** Light1 **O** 0 1000 1000 0 0 0 1 1 1 1 0: in order of name; type; position; direction; color; intensity; influence. Possible light types: parallel (O), point (P), spot (S), directional (D), later it will be expanded. The influence is measured in units – if 0, it means infinite, like a sun type light source.

A scene can contain infinite number of these elements, but one group for an object is a must, and subgroups are not yet possible. The game can also contain any number of scenes (at least one), the activeScene variable will select which of them is drawing to the screen. After a graphical editor is provided, this configuration file will be generated, rendering its manually editing needless.

Loading of the game content is processed in a few passes. First is the loading the Scene.cfg file, to build a general structure. It is empty, no actual drawing data is loaded. This function is called after the engine and game instances are initialized. After the function finished, the default scene loads in the memory its structure – objects, materials, textures, shaders. If an object uses the same source, and it is previously loaded, the indices are copied. Otherwise the implemented Wavefront OBJ loader takes care of the materials first, then the vertices, normal vectors, UV coordinates and triangle informations. For each triangle, the 3 vertices are checked in the buffer to reduce memory usage. The buffers are static, so the animations are calculated by the shaders using the individual matrices.

### 3 Drawing method

In general, the drawing is done iterating through every object, using a data buffer for each vertex as position, normal and UV coordinates, without indices for the only material, or indexed for more materials on the object. Either case, the size of the data buffer is not optimized, since every object (even if the same) contains a larger buffer. This technique is good for landscapes (hills, water), because there is one material for all the polygons. For a forest, let's say, with similar trees (only differ in size, color), we need a better method to reduce the system overload. To save memory, each tree can be placed in one buffer, and for each instance we define a transformation matrix. This will help with the memory, but the number of draw calls is not changing. If the buffer would contain each tree as transformed, there would be no need to use indices, so the situation is inverse. Which solution is better? Knowing the performance of an average GPU, without considering the IGPs (integrated graphic processor), we can state that the main problem is copying data. There is enough computing power in most of the GPUs, even at the entry level. Previously mentioned, the pipeline is divided in 3 parts, to omit the stacking which slows down the system. In the drawing method, these 3 lines are coming together to finalize the MVP (modelViewProjection) matrix for each object, and these can be calculated by the shader as well, but that means per vertex, which is also slowing down the process. It is not a problem for the CPU to multiply some matrices.

On the other hand, there are the shaders. They are set by `glUseProgram()` before the call for drawing. This engine has a drawing order as the following:

```
for(int i=0; i<materials.size(); i++)
{
    //in the current state, the shaders are linked to the objects
    //but it will be changed
    glUseProgram(materials[i].getProgram());

    //set textures
    //set uniforms such as colors, lights

    //set vertex attrib arrays
```

```
for(int j=0; j<groups.size(); j++)
{
    for(int k=0; k<groups[j].getObjects().size(); k++)
    {
        //calculate MVP matrix
        //call draw function
    }
}
glUseProgram(0);
}
```

It is visible, for every material, we change the shader program once. This is a boost also to the performance. There can be a situation, where each material uses the same shader program. In that case, the `glUseProgram()` can be placed before the first loop, saving more time. Since the material contains informations, not just data, it can be implemented in the shader code, whether there is transparency, normal mapping or other effect on the material. A little more complicated code is not a problem for any modern GPU.

Another reason, why the drawing should be done in this fashion: there is a huge problem with transparency. If the polygon is transparent and closer to the camera, drawing behind this polygon will not have a good result. The transparent materials have to be drawn last (and the polygons ordered by the distance to the camera, if necessary). Ordering by the materials vector by this property is enough, the indices related to the materials in the objects are in a struct containing the `bufferID`, indices, material name.

### 3.1 Drawing in FrameBuffers

Every graphical application uses double buffering (or triple in OpenGL). That means the rendering target is a backbuffer, and at the end of each frame, it is swapped with the front buffer to send it to the screen. This process is good enough if there is no need for post process effects, such as bloom, black and white or gamma correction. For this reason the texture class is prepared to allocate memory for any framebuffer necessary. The framebuffers are used mostly for the 3D part of the scene, any 2D drawing and text is rendered onto the back buffer after the 3D section finished. In case of an implemented post process antialiasing, the texts won't be blurred.

### 3.2 Depth illusion

People started making GIFs, that offer a perspective, depth to the viewer. This is achieved by either some white lines, which hide elements that are behind, and the rest is visible, causing an illusion of having depth on the picture. This method could be easily implemented in the engine, but the presence of these "guidelines" are annoying, because they could hide important elements in the scene.

The other effect has a frame around the picture, with a neutral color (white or black). This is not disturbing the user, but offers an illusion just as good as the other. The calculations are run by the shader. Knowing that the frustum is defined on a  $[-1, 1] \times [-1, 1]$  coordinate system, giving a percentage rate will have the same effect on the screen space (e.g. 1024x768), therefore passing the transformed vertex position is enough for the function to decide whether the pixel should be drawn or discarded. For example, a 10% frame around the screen has the same proportion on both coordinate systems. The variable that contains this value can be a uniform in the shader, giving the user control of the effect. Also, this value can be different for the vertical and horizontal sides, so the frame has a different aspect ratio to the screen.



### 3.3 Other visual features

After the next bigger update, the engine will support normal mapping, environment mapping, blooming, contrast and gamma correction. The first two is part of the main “composite” shader, which is default for all materials that are loaded from an OBJ file. There are other features, that are calculated on the vertices, such as parallax mapping, shadow mapping, but these are for future development.

Post process effects are run on the rendered scene, on a frame buffer, and therefore the possibilities of manipulating the image are endless. Custom filters, antialias techniques, field of depth in case the framebuffer has Z-buffer. The implementation of these, as well as the customization is not available in the current state. The project will go on for a long time, constantly growing, improving.

## 4 Future development

There is much to implement. The current state only offers a basic renderer, no animations, no special effects. An editor is going to be made, to make it easier to put a scene together. This program will allow the user to animate the scene.

Graphical improvements will conclude in

- shadows,
- reflections, refractions for transparent objects, like glass or water,
- a parametric dynamic sky system with clouds, weather,
- GUI.

The engine is developed on Linux, but SDL is easily portable between operating systems. The next logical step is to provide on Windows and other supported, common platforms.

A game is not complete without sound. The main focus is the graphical subsystem, but eventually it will contain a sound system as well. The developers who will use this engine will have the choice of implementing other sound APIs.

Depending on the type of application or game this engine will be used for, an artificial intelligence system should be developed as well.

Not to mention a colliding system, which is necessary for most games.

A particle system will be part of the engine. In the same design as the rest of the objects, its data will be stored in a material statically, and will be animated in the shader (passing over time).

Many more ideas, techniques will be attached to this engine as time goes on.

## 5 Conclusion

The engine uses OpenGL, which is in the current situation of the upcoming new APIs (DX12, Vulkan) sounds unprofitable, but these are supported by a very little portion of graphics cards. Most of the users still have hardware from the times when DX10 was the hit.

Not thinking about the PC users, SDL is present in many other platforms, as well as OpenGL (ES), so a possible application has a bigger audience.

But who should take a look at the engine, who could use it? The user doesn't have to know programming, if the intention is to use the basic features. Until the editor is available, the file that describes the scenes has to be edited manually. Following the instructions a simple static scene requires just a few lines. For the beginning there will be a free camera mode to go around the scene. Basically anybody can use it after a few minutes of reading. The target users are

- students, who want to have something related to graphics very fast,
- artists (modelers), who want to see their work in a game engine,

- teachers, to show geometrical problems,
- architects, to see their construction in real time, without professional hardware.

## 6 References

For the OpenGL functionalities, I used this book:

[1] Dave Shreiner, Graham Sellers, John M. Kessenich, Bill M. Licea-Kane, *OpenGL Programming Guide (8th Edition)*, Michigan, 2013

For documentation, downloads, development status about the libraries used for this project, please visit their respective websites:

[2] GLM: <http://glm.g-truc.net/>

[3] SDL: <https://www.libsdl.org/>

[4] SDL\_image: [https://www.libsdl.org/projects/SDL\\_image/](https://www.libsdl.org/projects/SDL_image/)

[5] C++: <http://www.cplusplus.com/>

[6] Clang compiler: <http://clang.llvm.org/>

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](mailto:bialko.istvan@gmail.com)

## **“2D Skateboarding”**

**Paul Boldijar**

**Coordinators Teachers: Liliana Comarnic, Ciprian Ghișe**

### **Abstract**

The idea for this game came from a game that was in vogue 20 years ago. The character “came back to life”, but is different from the “old character”. He looks different, he uses a skate board and the technologies “he uses to move” are very different from those “he used” 20 years ago. This article is an extended and completed version of the article that I presented at PCID [1], held at "Lucian Blaga" University in 2015.

This game is for people of any age! The character has to be led on the board on complicated skating routes so to win as many points as possible.

The application is optimized for tablets and scheduled so the game will scale in an efficient manner on any device, so no black bars will appear on the screen and will not distort the screen. The application has been tested on phones with specifications rather weak and went without any problem because we can choose the desired asset. For example, if you have a Samsung Galaxy Fit with a 3.2 inch display, we choose the smallest package and the game will run with maximum FPS, about 60.

Besides all this, the game has a nice variety of sound effects and background melody.

## **1. Introduction**

The application is developed in Java, using the framework for 2D games libGDX that allows programming a game once and exporting of the android, ios, and desktop.

This framework has been used in parallel with the Android SDK, because some android native methods must be implemented by JAVA interfaces, such as messaging or messaging Toast dialog.

I used Box2D for physics. This is a physics engine written in C ++ by Erin Catto. Some of the most popular games that uses it are “Cut the Rope” and “Angry Birds”.

In this game I used the rigid body for each object in the game and player movement is done by applying a constant force to the left, right or top if necessary. Box2D only deals with physical equations behind objects, so you can make a separation between “the physical world” of the game

and the view, that is what we see, and we can implement MVC pattern (model - view - controller) to create an application as scalable and easily expandable in the future. Graphic was made in Inkscape and animated with Spriter program.

Images were resized for various categories of resolutions through a program created by me in Windows Forms.

Leap Motion sensor integration was possible through their SDK, so with only a few implementations of interfaces, we can move the player only by using the fingers in the air. The settings or the progress of the game are saved in the Shared Preferences on the android and on the desktop in an XML file that is in the logged user, in AppData file.

In the main menu of the game there are two “Easter eggs”. If you shake the device, the buttons will “fall off ”and will move depending on how tilted the phone or the tablet is, and if you press on the screen will appear the game character behind the logo.

## **2. The usability of the application**

### **2.1. Controls:**

- Swipe left - move left
  - Swipe right - move right
  - Swipe up - ollie (jump)
  - Swipe down – stop
- You can not move left or right when you're on the bar.
- Try to avoid spikes or falls on the level.
- Collect as many coins and upgrade your skateboard!

### **2.2. You can upgrade:**

- Bearing - You catch more speed.
- Board - You can jump higher.
- Wheels - With this upgrade you stop faster when you swipe down.
- Axes - If you want a more stable landing plate, Upgrade axes.

### **2.3. Features:**

- Graphics for different screen sizes (XHD, HD, MD, SD); You can select the desired resolution settings;
- Currently: 20 levels of super strong! If you want more, send us a message on Facebook. We will create the levels and we'll mention you.
- 2 wallpapers
- 2 Interactive secret
- Nice sound effects
- Pleasant graphics and animations

- You can ask any game's update on Facebook; if your idea will be considered, your name will be mentioned;

Did you like this game? Help us by purchasing the full version for just one dollar.

Additional features:

- Not ads
- Not dialogues asks if you want to press like on Facebook, rates
- Get twice as many pennies
- Yes you can zoom in / out during the game. It is very useful if the level is difficult.
- Many more levels.

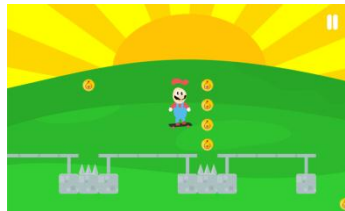


Fig. 1

- Do you like games with this character and skateboarding?
- Well, than the game is done for you!
- Try to finish each level with 3 stars collecting as many coins as possible.

## 2.4. The desktop version

For the Windows, Mac / Linux versions we can control the character with arrows or a motion sensor leap if we have one.



Fig. 2

We only use our hands in the air by rapid movements left, right, up, down. This experience is unique!

The application has the most downloads from the following countries:

1. United States
2. The UK
3. Canada
4. Russia
5. Brazil
6. Japan
7. Mexico
8. France

## 9. Germany

Paid version of Google Play was bought by 15 people and the free version has been downloaded about 5,000 people and assessed at 4/5 stars by 70 reviews.

Most users of this game are using android 4.4 (47%) and then android 4.2 (23%).

In the free version of the application were requested advertisements about 80,000 times and displayed 65,000 times.

The above statistics are taken from Google Play developer page and Admob.

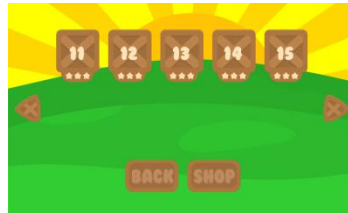


Fig. 3

## 3. Program Code explained

Let's see how do I add in the physical world of the game new blocks of land:

Chainshape is a Box2D class that contains information about object shape. The unit of length is the meter, and we must convert size in pixels, so I declared PPM constants. (Pixel per meter ).

```
public ChainShape createShape(float tileSize) {  
    ChainShape shape = new ChainShape();  
    Vector2[] v = new Vector2[5];  
    v[0] = new Vector2(-tileSize / 2 / B2D.PPM, -tileSize / 2 / B2D.PPM);  
    v[1] = new Vector2(-tileSize / 2 / B2D.PPM, tileSize / 2 / B2D.PPM);  
    v[2] = new Vector2(tileSize / 2 / B2D.PPM, tileSize / 2 / B2D.PPM);  
    v[3] = new Vector2(tileSize / 2 / B2D.PPM, -tileSize / 2 / B2D.PPM);  
    v[4] = new Vector2(-tileSize / 2 / B2D.PPM, -tileSize / 2 / B2D.PPM);  
    shape.createChain(v);  
    return shape;  
}
```

The following method loads in the memory a map of the game. They are saved in .TMX and can even be encrypted. TiledMap is the desired map and world is the physical world where we want to add it.

```
public void LoadGameTiles(TiledMap tileMap, World world) {  
    TiledMapTileLayer layer = (TiledMapTileLayer) tileMap.getLayers().get(  
        "Tiles");  
    // Here I shall prepare objects containing properties of the subsoil.  
    BodyDef def = new BodyDef();  
    FixtureDef fdef = new FixtureDef();  
    def.type = BodyType.StaticBody;  
    FixtureDef sensorDef = new FixtureDef();  
    sensorDef.isSensor = true;  
  
    int coinID = 0;  
    int starID = 0;  
    float tileSize = layer.getTileHeight();
```

```
// I save the map size in a static variable.
GameTileRenderer.width = layer.getWidth();
GameTileRenderer.height = layer.getHeight();
// Now I will fill an array that represents each item in the map by -1, to add information about the current map
later.
GameTileRenderer.tlz = new int[GameTileRenderer.width][GameTileRenderer.height];
for (int[] row: GameTileRenderer.tlz)
    Arrays.fill(row, -1);

// I access all elements in the map
for (int row = 0; row < layer.getHeight(); row++) {
    for (int col = 0; col < layer.getWidth(); col++) {
        Cell cell = layer.getCell(col, row);
        if (cell == null || cell.getTile() == null)
            continue;
// I found an empty cell, I move to the next iteration of the loop

//I prepare the physical body that represents a cell in the map
def.position.set((col + 0.5 f) * tileSize / B2D.PPM, (row + 0.5 f) * tileSize / B2D.PPM);
float x = col * tileSize / B2D.PPM;
float y = row * tileSize / B2D.PPM;

int tileType = cell.getTile().getId();
tileType--;

// Now depending on the cell type I add it on the map.

if (tileType == 11) {
    GameStats.GameSpawnPosition = new Vector2(x, y);
    Cell c = new Cell();
    layer.setCell(col, row, c);
}
if (tileType == 12) {
    // is final flag
    Cell c = new Cell();
    layer.setCell(col, row, c);
    GameMap.flagposition = new Vector2(x, y);
    def.position.set((col + 0.5 f) * tileSize / B2D.PPM, (row + 0.5 f) * tileSize / B2D.PPM);
    sensorDef.shape = createShape(tileSize);
    world.createBody(def).createFixture(sensorDef)
        .setUserData("finish");
}
if (tileType == 8) {
    // is spike
    int id = tileType;

    GameTileRenderer.tlz[col][row] = id;
    fdef.friction = 0;
    fdef.shape = createSpikeShape(tileSize);
    world.createBody(def).createFixture(fdef)
        .setUserData("die");
}
if (tileType == 0 || tileType == 4) {
    // It is a normal element
    int id = tileType;

    GameTileRenderer.tlz[col][row] = id;
    fdef.friction = 0;
    fdef.shape = createShape(tileSize);
    world.createBody(def).createFixture(fdef)
        .setUserData("tile");
}
```

```
    }  
    if ((tileType >= 1 && tileType <= 3) || (tileType >= 5 && tileType <= 7)) {  
        // It is the bar; 6 types are possible  
        int id = tileType;  
  
        GameTileRenderer.tlz[col][row] = id;  
        fdef.friction = 0;  
        fdef.shape = createShape(tileSize);  
        world.createBody(def).createFixture(fdef)  
            .setUserData("rail");  
    }  
    if (tileType == 10) {  
        // coin  
        Cell c = new Cell();  
        layer.setCell(col, row, c);  
        if (GameStats.AddTheCoin(coinID)) {  
  
            GameWorld.cr.coins.add(new Vector2(x, y));  
  
            def.position.set((col + 0.5 f) * tileSize / B2D.PPM, (row + 0.5 f) * tileSize / B2D.PPM);  
  
            sensorDef.shape = createShape(tileSize);  
  
            TileData tmp = new TileData(coinID, "coin");  
            coinID++;  
            world.createBody(def).createFixture(sensorDef)  
                .setUserData(tmp);  
  
        } else {  
            GameWorld.cr.coins.add(null);  
            coinID++;  
        }  
    }  
    if (tileType == 9) {  
        // coin  
  
        Cell c = new Cell();  
        layer.setCell(col, row, c);  
        GameWorld.cr.stars.add(new Vector2(x, y));  
  
        def.position.set((col + 0.5 f) * tileSize / B2D.PPM, (row + 0.5 f) * tileSize / B2D.PPM);  
  
        sensorDef.shape = createShape(tileSize);  
  
        TileData tmp = new TileData(starID, "star");  
        starID++;  
        world.createBody(def).createFixture(sensorDef)  
            .setUserData(tmp);  
    }  
} } }
```

To achieve buttons I wrote a class that will draw their texture, with a small animation and a sound effect. This class contains three variables:

1sprite – Sprite class from libGDX is instantiated

2isTouched – boolean, true if the button is touched and false if not

3scale – real number

Constructor class looks like this:



```
public CoolButton(TextureRegion texture, float width, float height) {  
    sprite = new Sprite(texture);  
    sprite.setPosition(0, 0);  
    sprite.setSize(width, height);  
    sprite.setOriginCenter();  
}
```

Next I set the texture, width and height. The class has three set methods, namely:

This method can change the texture of the button.

```
public void SetRegion(TextureRegion region) {  
    sprite.setRegion(region);  
}
```

This method can set the position of the button.

```
public void SetPosition(float x, float y) {  
    sprite.setPosition(x, y);  
}
```

This method can set the button rotation in radians.

```
public void SetRotation(float rad) {  
    sprite.setRotation(rad * MathUtils.radiansToDegrees);  
}
```

The Draw method handles the drawing of the button on a SpriteBatch.

Here I change the variable scale depending on how much time passed from the moment it was pressed, up to a maximum size of 1.2f.

```
public void Draw(SpriteBatch batch) {  
    if (isTouched) {  
        if (scale < 1.12f)  
            scale += Gdx.graphics.getDeltaTime();  
        else  
            scale = 1.12f;  
    } else {  
        if (scale > 1f)  
            scale -= Gdx.graphics.getDeltaTime();  
        else  
            scale = 1f;  
    }  
    sprite.setScale(scale);  
    sprite.draw(batch);  
}
```

This method will check if the button is touched, and if it is reached you will hear a sound effect.

```
public boolean TouchedDown(float x, float y) {  
    boolean oldTouched = isTouched;  
    isTouched = sprite.getBoundingBox().contains(x, y);  
}
```

```
    if (!oldTouched && isTouched) {  
        if (GameSettings.SoundON) {  
            if (SoundManager.ui_down.isPlaying() == false)  
                SoundManager.ui_down.play();  
        }  
    }  
    return isTouched;  
}
```

To add the skateboarder in the physical world and to control him, I wrote the class `Skater.java`. This includes the physical body of the skateboarder and his cap, these two are linked by a Prismatic Joint. The class has some variables that contain information about the physical properties of bodies, which will be changed depending on the level at which they were upgraded from the store.

The following method adds skater in the physical world. Here I make the initialization of the physical properties and the 3 parts of the body: one is the actual skateboarder, the second is a sensor that takes place of the legs, which does not react with the physical world but receives information related to the collisions (depending on this we check if we are on the ground or not) and the 3rd is the head, which is also a sensor.

```
private void CreatePlayer(World world) {  
  
    BodyDef playerDef = B2D.CreateBodyDef(GameStats.GameSpawnPosition.x,  
        GameStats.GameSpawnPosition.y, BodyType.DynamicBody);  
    playerDef.fixedRotation = true;  
    playerBody = world.createBody(playerDef);  
    PolygonShape playerShape = B2D.CreateBoxShape(13 / PPM, 29 / PPM);  
    FixtureDef playerFD = B2D.CreateFixtureDef(1, 0.3f, 1, playerShape);  
    playerBody.createFixture(playerFD).setUserData("skater");  
  
    // sensor  
    FixtureDef sensorDef = new FixtureDef();  
    Shape shape = B2D.CreateBoxShape(10 / PPM, 10 / PPM, new Vector2(0, -22  
        / PPM));  
  
    sensorDef.shape = shape;  
    sensorDef.isSensor = true;  
    playerBody.createFixture(sensorDef).setUserData("foot");  
  
    // head  
    shape = B2D.CreateBoxShape(12 / PPM, 5 / PPM, new Vector2(0, 25 / PPM));  
  
    sensorDef.shape = shape;  
    sensorDef.isSensor = true;  
    playerBody.createFixture(sensorDef).setUserData("cap");  
}
```

By the following method I will add the hat of the skateboarder. This can only move up and down and is connected by a join. Here I set the location and origin of each position of the join.

```
public void AddHat(World world) {  
  
    BodyDef hatbodyDef = new BodyDef();  
    hatbodyDef.type = BodyType.DynamicBody;  
    hatbodyDef.position.set(playerBody.getPosition().x,  
        playerBody.getPosition().y + 100 / 100f);  
}
```

```
hatBody = world.createBody(hatbodyDef);
FixtureDef def = B2D.CreateFixtureDef(.01f,
    0.3f * (1 - GameSettings.Stats4), 0);
System.out.println("Restitution: " + def.restitution);
def.shape = B2D.CreateBoxShape(10 / PPM, 10 / PPM, new Vector2(0, -22
    / PPM));
hatBody.createFixture(def);

PrismaticJointDef pdef = new PrismaticJointDef();
pdef.bodyA = playerBody;
pdef.bodyB = hatBody;
pdef.collideConnected = true;

pdef.localAnchorA.y += 50 / 100f;
pdef.localAxisA.set(0, 1);
pdef.enableLimit = true;
joint = (PrismaticJoint) world.createJoint(pdef);
joint.setLimits(0, .15f);

}
```

## 4. Conclusion

The application is unique because of the game algorithm that allows to reach win situation, the graphics and the used technologies. In future I will work to the game, so that the land on which the character is moving with the board to be more varied.

## References:

- [1] Author Paul Boldijar, Coordinators Teachers Liliana Comarnic and Ciprian Ghișe, “*Mario Skateboarding*”, PCID 2015, “Lucian Blaga” University, Sibiu

Paul Boldijar  
Computer Science  
The National College of Computer Science “Grigore Moisil”  
Brașov/Brașov  
E-mail: paul.bv96@yahoo.com

## **Virtual reality game**

**Petrica Bota, Bogdan Grecu, Darius Hategan**

**Teacher Coordinator: Dana Simian**

### **Abstract**

The aim of the article is to present an original game named Forrest runner. Forrest runner is an interactive, 3D, virtual game with the purpose of making people explore the 3D realm while having fun. The goal of the game is to get to the finish line while avoiding certain obstacles. Forrest runner is programmed in Unity with the help of the Zeiss VR One SDK. Also, in order to play the game in 3D, a virtual reality set is required. As of now the game is rather simplistic, but we strive to implement more features.

## **1. Introduction**

We live in a world where gaming is not only a part of our society but of us as well. The memories of playing hide and seek are being replaced with the ones of playing your favorite FPS(First Person Shooter) or RTS(Real Time Strategy) game. Actually, there isn't even a correct definition for the word „gamer”.

As programmers we are always on the lookout for new devices and technologies. Thus, we have been paying close attention to the recent rise in popularity of the virtual reality headsets and their corresponding games.

At the moment there is a race in-between companies in order to be at the front of this new emerging technology. Though approaches differ from firm to firm. For instance we have at the lower end Google Cardboard, a cheap and easy way to experience 3D. At the other end we have Oculus Rift, the best at the moment in matters of quality and immersion. Also there are companies that don't try to create a one hundred percent simulated environment, but to augment the already existent one. Such a company is Microsoft with its HoloLens project.

Though there are many headsets from which to choose from, we chose Zeiss VR One. We personally believe that it offers a good price to quality ratio in contrast with the other virtual reality headsets on the market(for the same quality VR One is more affordable).

Also, we discovered that they are able and willing to help the developers create apps that use their VR. Furthermore, the possibility of upgrading the VR without having to buy newer models is an added bonus.

## 2. General presentation of the game

Forrest Runner is a thrilling, immersive game in which you are required to get to the finish line while avoiding obstacles; such as fallen trees or big rocks that block the way. Your character finds himself in rather bad situation in which a rogue wolf is chasing after him. As a player you have to help your character to get to safety by avoiding the obstacles that block his path.

The trick is that instead of using your mouse or controller to move you are actually going to use your own body to complete the action from a first person view. If there is a fallen tree to your left you have to tilt your head to your right and the character will change path accordingly. Also while wearing the headset you are experiencing a real 3D environment; what this means is that if you were to look up, you would actually see the sky, or if you were to look around, you would see the forest. We hope that by providing these controls we will create a not only realistic but also immersive game.

In order to create such a game newer technologies and devices were required. Therefore we required a game engine with which to create the map, the character and in which to program the controls. We also required a virtual headset that would fulfill the 3D aspect of the game. We selected Unity as the game engine because it presents programmers with so many in-engine assets and in-game written scripts. As for the VR, we selected the VR One due to the fact that it uses your smartphone as a screen for your eyes thus, by simply upgrading your smartphone you would also upgrade your VR, which we think of as a huge advantage.

### 2.1 Project inspiration and goals

We chose game design as a topic due to the fact that we are „gamers”.

We grew up with games around us and we wanted to discover how they are made along with creating our own. We started learning Pascal, C, C++ and C# in order to achieve this goal. This project is yet another challenge, we hope to provide the gaming community with a new not only interactive but also immersive game.

#### 2.1.1 The three main goals of the project:

1. Creating a simple game that appeals to all types of people, indifferent of age.
2. Presenting people the existence of virtual reality headsets
3. Learning more about game development .

#### 2.1.2 Secondary objectives of the above:

- a) Learn how to create a simple 3D environment.
- b) A better understanding of the C# language.
- c) Learn how to program in Unity
- d) To prove ourselves that we can.
- e) Promote the finished product.

### 3 Used technologies



**Fig. 1** Unity's logo [6]

#### 3.1 Unity

In order to develop the game (with its map and character) we required a powerful and easy to use game engine. We chose to use Unity's due to its sturdy framework and friendly interface. Also the fact that there are many tutorials on the internet that explain, if not all, aspects of the engine pushed us in this direction.

Unity is a flexible and powerful development platform for creating multiplatform 3D and 2D games and interactive experiences. It contains everything necessary for programmers to create high-end graphics, physics or in-game mechanics. The engine can work on the following APIs: Direct3D on Windows and Xbox 360; Open GL on Mac, Windows and Linux.

One of its greatest advantages are its in game assets (textures, sounds, images) that can be downloaded for free, or payed for at the assest store. If you require certain styles or art you don't have to draw or make it yourself, you can find in on the store. Furthermore, Unity has its own community around it which is more than eager to help you whenever you find yourself in an unsolvable situation.

Another strong point of Unity is that of its versitability. You can develop a game in whatever manner you please and then you simply have to export it to whatever operating system or device you desire. For instance, you create your game in Unity with its assets and scripts and when you are finished you simply have to export it to Android or IOS with no added changes(though for IOS you need a developer's license).

#### 3.2 Zeiss VR One headset

As stated in the General presentation section, we chose Zeiss VR One for our virtual reality headset. What really made us choose this VR was their involvment with not only the users, but also the developers that use their device. We discovered that they provide an SDK (Software Development Kit) for Unity. Therefore, we were able to create frameworks in Unity that would be usable with their device. The best part about the SDK and the guidelines we found on their website was that they were all free of charge.

Zeiss has taken a very interesting approach to the virtual reality scene by incorporating in their device another device that most of us already posses. The VR has plastic trays in which you set your smartphone (with the app installed) then simply slide everything inside the headset and you are ready to experience the 3D world. The headset accepts smarphones with

display sizes between 4.7 and 5.2 inches. Specific trays are tailor-made for each smartphone and if you don't want to buy your tray you can download its 3D model and print it on a 3D printer.

We are currently using a Samsung Galaxy S5 smartphone to install an test the app. The smartphone tracks the users movements with its built in gyroscope and sensors then transmits that data to the app which with the help of Unity Scripts and the VR SDK transforms it into actual movements and actions inside the game.

The biggest advantage of using the VR One headset is that if we are to upgrade our smartphone we would automatically upgrade our VR headset as well.



**Fig. 2** VR One headset with its tray in which the smartphone is introduced [7]

### **3.3 Zeiss VR One SDK**

VR One SDK is a VR SDK for Unity that helps developers to create immersive virtual reality experiences for the VR One headset. At the moment it supports mobile apps made in Unity for IOS and Android.

The VR One SDK features many scripts that inside the Unity engine provide a split screen or a stereo two camera setup and applies a configurable radial and chromatic pre-distortion effect to prepare the image for optimal display in the VR ONE for various smartphone models.

Beside the VR One SDK, Zeiss provides a multi-sensor head tracking plugin. The plugin is activated automatically in the SDK, but if you desire you can deactivate it and develop your own head-tracking. The plugin uses all of your phone's sensors in order to reduce latency between you actions and their effects on the display.



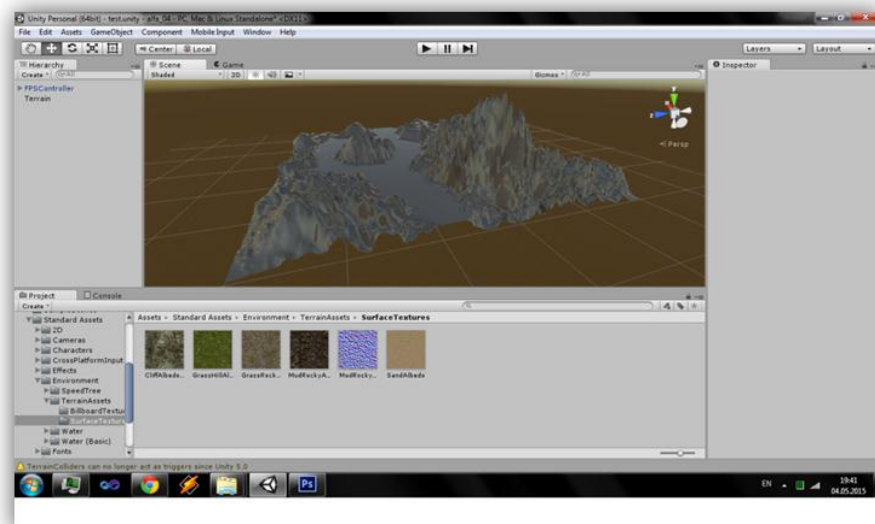


**Fig. 3** The SDK is splitting and distorting the original image in two so that there are two images (one for each eye) in order to create a 3D effect [8]

## 4. Game Design

The game is rather simplistic in its nature, there is a long curving pathway which is surrounded by a forest. Your character finds himself running alongside this pathway and you have to guide him to safety.

We created the map with the help of Unity's built-in terrain creator. First we added a simple terrain base, to which we later added mountains and rivers (fig. 4). Later we imported some textures and assets from the asset store in order to give the terrain its grassy green look. Lastly but not least we populated the terrain with the forest and added obstacles that would block our character's path (fig. 5).



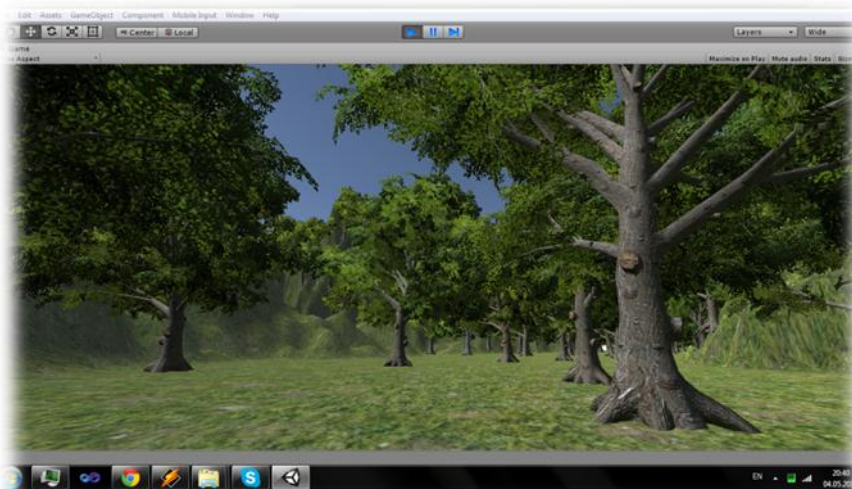
**Fig. 4** Early map Modelling (creating the mountains)





**Fig. 5** Implementing grass, sky, lighting.

What really sets our game apart is the immersion level. Once you put your headset on you have the impression that you are in a real forest. The game envelops your point of view, so you can look around and all that you would see would be in-game environment (fig. 6).



**Fig. 6** A first person view from inside the game

The following piece of code exists in the VR One SDK and is responsible with the head-tracking of the player. The app constantly communicates with the phone's sensors and updates its in-game camera view:

```
public class HeadBob :  
MonoBehaviour // a small example of code  
{  
    public Camera Camera;
```

```
public CurveControlledBob motionBob = new CurveControlledBob();
public LerpControlledBob jumpAndLandingBob = new LerpControlledBob();
public RigidbodyFirstPersonController rigidbodyFirstPersonController;
public float StrideInterval;
[Range(0f, 1f)] public float RunningStrideLengthen;

// private CameraRefocus m_CameraRefocus;
private bool m_PreviouslyGrounded;
private Vector3 m_OriginalCameraPosition;
}
```

## 5. Conclusions and future development

To sum up, Forrest Runner is our contribution to the game industry, small as it may be in comparison. Though we are the beginning of our yellow brick road, we strive to make the best programs that we are capable of. As future developments, we would like to implement the following:

- Tougher obstacles
- Smaller time delay from player to game
- More levels
- A wolf that runs after you

## References

- [1] Zeiss VR One homepage - <http://zeissvrone.tumblr.com/>
- [2] Infomation about Zeiss VR One - [https://vronecontest.zeiss.com/index.php?controller=pages&view=about\\_vr](https://vronecontest.zeiss.com/index.php?controller=pages&view=about_vr)
- [3] Unity homepage - <https://unity3d.com/>
- [4] Information about unity - [http://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](http://en.wikipedia.org/wiki/Unity_(game_engine))
- [5] VR One SDK  
<https://bitbucket.org/vrone/unity3d/src/3c7395473a371ff755f89715a36ee532427c780d#markdown-header-what-is-the-vr-one-sdk>
- [6] <http://blog.aledorn.com/wp-content/uploads/unity-logo-black.png>
- [7] <http://o.aolcdn.com/hss/storage/midas/b1e600229180424ec310c832c159d847/200904794/VR-One.jpg>
- [8] <https://bitbucket.org/vrone/unity3d/src/3c7395473a371ff755f89715a36ee532427c780d#markdown-header-what-is-the-vr-one-sdk>

Petrica Bota  
„Lucian Blaga” University  
Faculty of Sciences  
Str. Dr. I. Rațiu, Nr.5-7, Sibiu  
Romania  
E-mail: [bota.petrica2@yahoo.com](mailto:bota.petrica2@yahoo.com)

Greco Bogdan  
„Lucian Blaga” University  
Faculty of Sciences  
Str. Dr. I. Rațiu, Nr.5-7, Sibiu  
Romania  
Email: [bogdy\\_greco@yahoo.com](mailto:bogdy_greco@yahoo.com)

Darius Hategan  
„Lucian Blaga” University  
Faculty of Sciences  
Str. Dr. I. Rațiu, Nr.5-7, Sibiu  
Romania  
E-mail: [hategandarius71@gmail.com](mailto:hategandarius71@gmail.com)

# **GFastPay**

**Georgi Chavdarov**

**Teacher Coordinator: Katalina Grigorova**

## **Abstract**

Millions of people use their computers, mobile phones and tablets as a tool for shopping online. Hundreds of thousands of people use debit or credit cards as payment tool in various outlets. Now it's time for something new, something different, something attractive and something quickly - one-click way to pay. GFastPay is an alternative which completely replaced the need for a bank card.

## **1 Introduction**

Before entering the technology in our everyday life, the only way to pay for a given amount has been - manually, i.e. banknotes and coins. For the purpose of convenience of the customer, cards are created with the help of which were possible purchases of fuel and goods on the stations [1]. Later, they are followed by the department stores.

With the bank cards payment became faster. It is not necessary person to search coins and bills to make payments, but he or she must have own card. And what happens if the customer has forgotten it? What employee should do, if Point of sale (POS) terminal device doesn't work, in the shop? If the card is present and POS terminal works, but there is a failure on the chip card, how the customer can pay the product with the money that he HAS, but cannot use as intended?

With GFastPay life becomes easier. The employee enter the information about purchased items and it's enough the customer to take in his or her hands own smartphone or tablet, load the QR Code reader and with one click is able to make the payment.

Actually, GFastPay is simple software that allows introducing one or more products and their price values. This makes it possible to generate a link to 'Pay Now' functionality of PayPal, by converting it into QR Code, and ready for scan from the client. Depending of requirement mode of the payment, every product may be static (in restaurants, bars) or dynamic (gas stations, private parking).

The application is written using Java and is intended for all of the most used operating systems like Windows, Mac and Linux. For storing products, page & printer settings and the private credential information are used binary files. The API used for QR Code images generation is QrGen which is based on ZXING – one of the most popular barcode-reader libraries.

## 2 Pay easy and fast

### 2.1 Main functionality

Digital shopping has never been so easy before. Adding products, paying and verifying after that has never been done with such ease and speed.

The presented GFastPay application allows operation of many products with a sense of paying with real credit card and ease of online buying.

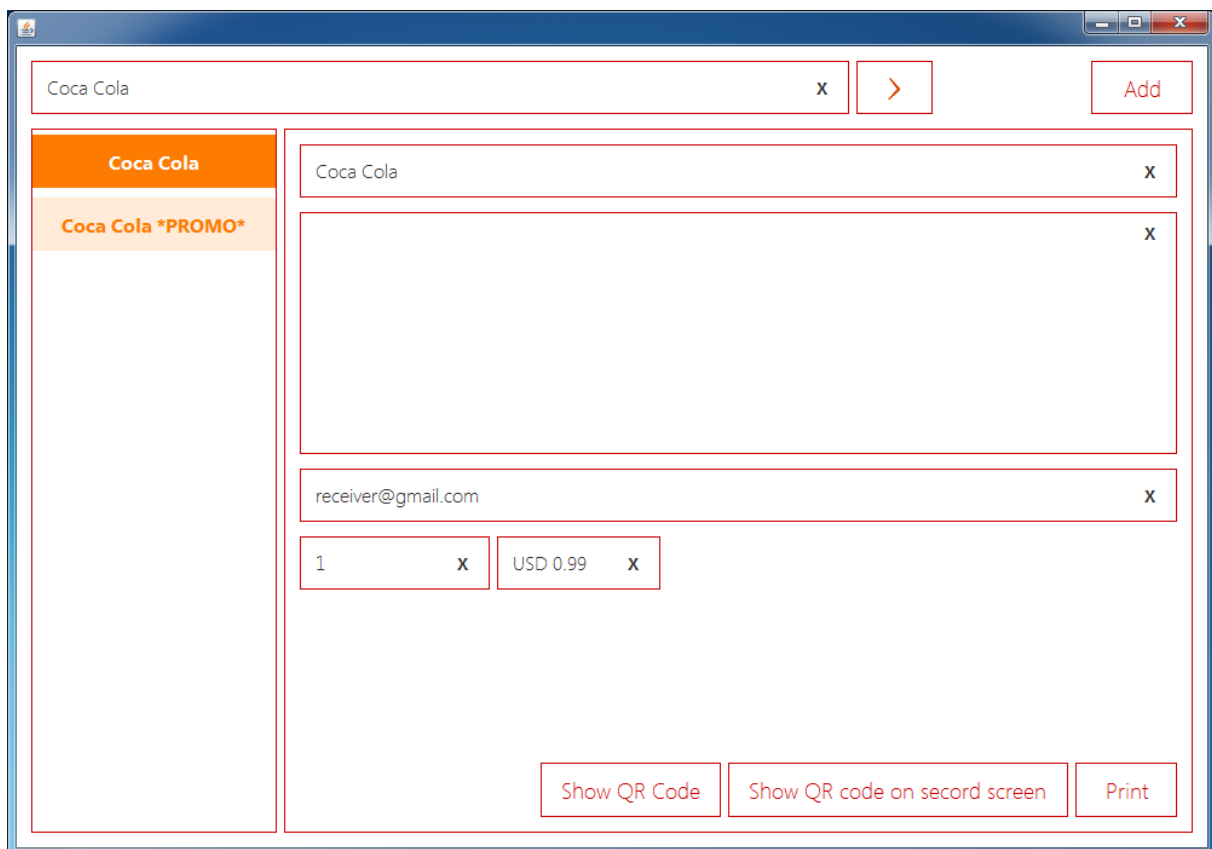


Fig. 1. Main window

#### 2.1.1 Loading application

When the application loads, the main window is shown, an empty order is selected and 'New product' action is executed. If the user wants to select existing product, it can be selected from 'Recent products' list. If the user wants to add new products, it can be done with 'Add' button located on the top right corner (Fig. 1).

#### 2.1.2 Manipulating products

Excluding the option to add products data, each one can be edited or deleted instantaneous with double-click over desired item. For each product, the employee have the opportunity to define different fields like title, body, product quantity, price per one item and the receiver email

registered on PayPal web site. If the user uses more than one email, an autocomplete function can be used for fast switching between them.

As we said there are two product types - static and dynamic. Static items are able to be created once and then to be saved for repeatedly using. Dynamic items also can be used repeatedly, but the only variable unit is the one that the user has defined. Examples of variable units:

- Current date - option to generate the day, month and year of shopping
- Current time - an option that generates hour, minute and second of shopping
- Date (plus / minus) (some) (days / months / years) - a feature that can be used to pay for rent, monthly contracts and others.
- Hour (plus / minus) (some) (seconds / minutes / hours) - a feature that can be used for prepaid parking, subscribing for any service, reserving a table in restaurant or bar for a certain period of time and etc.

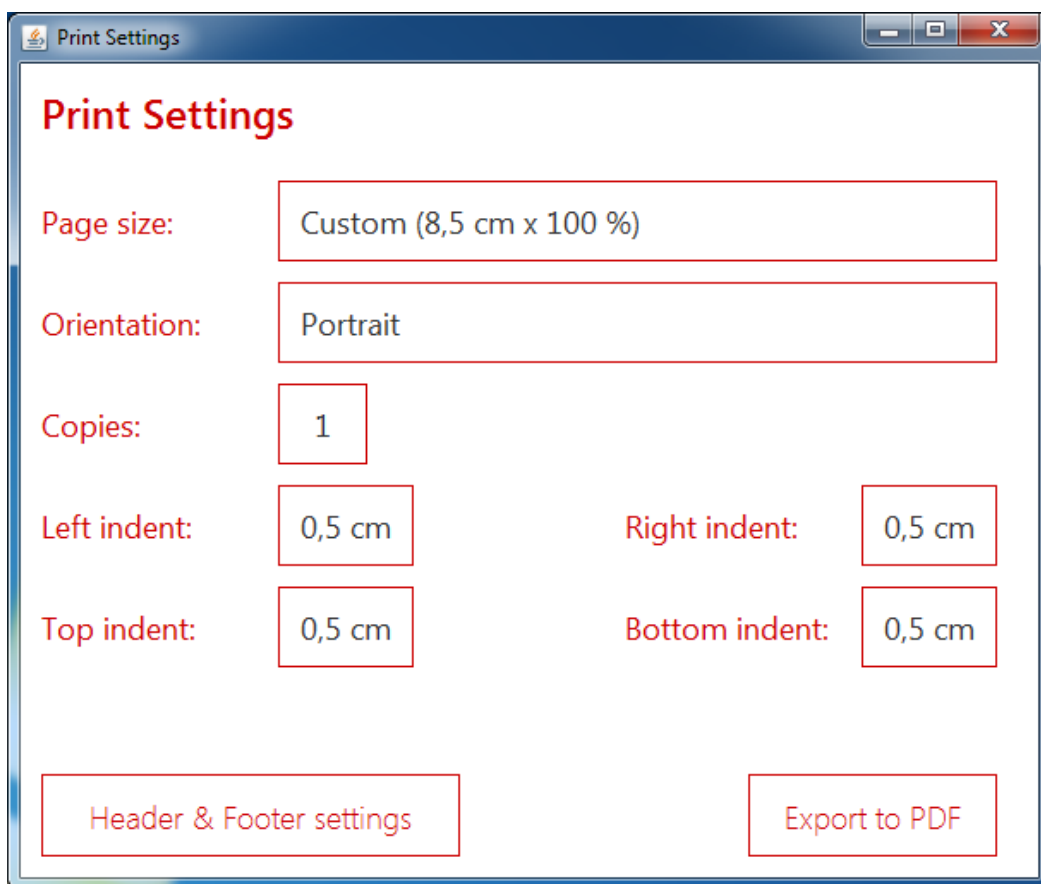


Fig. 2. Print Settings

### 2.1.3 Processing products

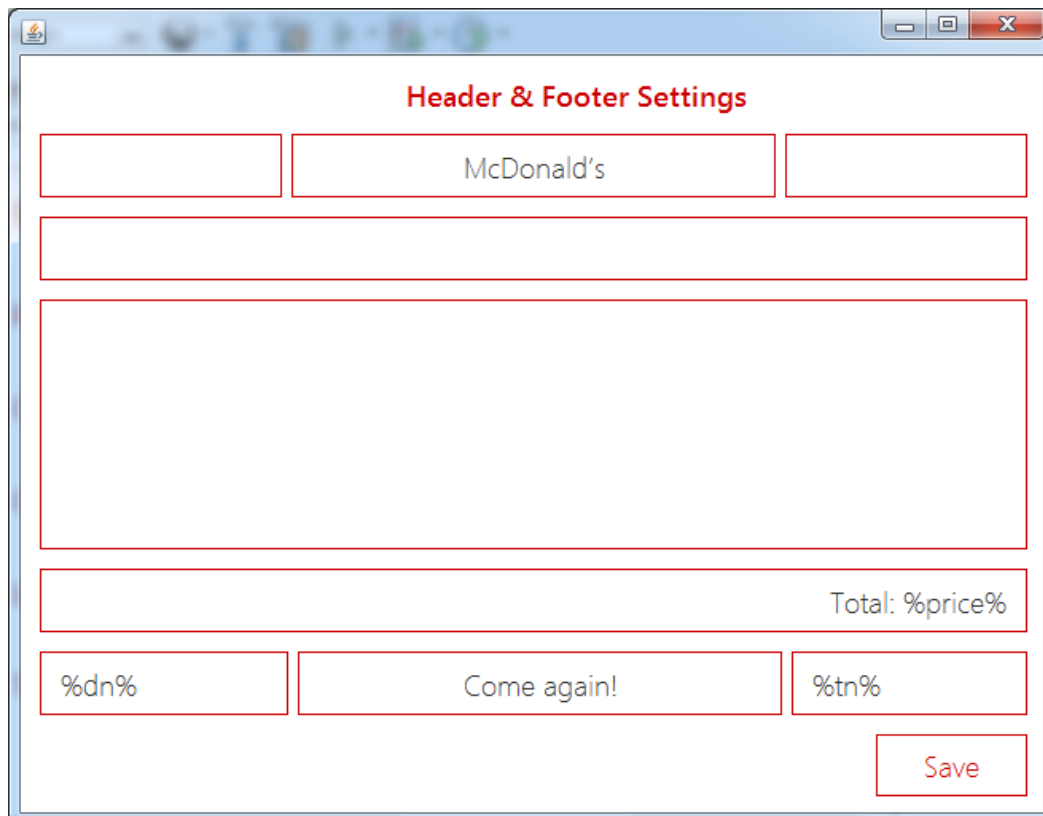
Once the employee has chosen an existing product or added a new one, he or she immediately has the ability to prepare a buy link as QR Code for product purchasing, by pressing the 'Show QR Code' button. Customer loads its QR Code Reader Application and go to load generated PayPal 'Pay now' link. User confirms the payment into PayPal system by entering his personal information.

If at the site is not provided work of employees (private parking, self-check bus / metro / train tickets), the manager can use the 'Print QR Code'. A typical example would be a request such as:

'*One Hour subscription from %tn% -% tn + 1h%*', which will generate a new PayPal link titled '*Hour subscription from 12:00 to 13:00*'.

#### 2.1.4 Printing product details

When the user choose to print a cash receipt, a new print settings dialog is shown. It allows to export generated content to PDF format, the light of greater functionality of Adobe Reader for printing in different paper size and various printer devices (Fig. 2).



The screenshot shows a window titled "Header & Footer Settings". The window contains several text input fields. The top row has three fields, with the middle one containing "McDonald's". Below this is a single wide field. The next row has a single wide field containing "Total: %price%". The bottom row has three fields: the first contains "%dn%", the second contains "Come again!", and the third contains "%tn%". A "Save" button is located in the bottom right corner of the dialog.

Fig. 3. Header & Footer Settings

The options for printing are:

- Determining the size of paper in international ISO standard (A2, A3, A4, A5, etc.) [3]
- Determining the size of paper set precise dimensions (cm, mm, px / dpi, etc.);
- User defined color print - print QR Codes with different color combinations to fit on brochures, magazines (such as donation campaigns);
- Setup page margins;
- Definition the number of copies (eg. when two copies are needed - one for the client and one for the employee, the lessor or the owner).

### 2.1.5 Header & Footer Templates

'Header & Footer' dialog makes it possible to position the static or dynamic text to be used on every future generated printout. In this case the company name, title, current date and time and other additional information can be reused every time.

### 2.1.6 Review of existing solutions

There is a wide range of methods allows the generation of QR codes [2]. One of them is Scansfer. It allows dynamically QR Code generation, but it has no options for saving and reusing entered information, again. QuaR.me is another simple project with same functionality, but there are not provided options for cash receipt printing, it needs the employee to load the browser always.

### 2.1.7 System requirements

For a good and complete operation the application needs in addition a good internet connection and the large display. The minimum resolution for the proper visualization of all the menus and dialogs is 1280x600 pixels.

Although these minimum requirements, the user may need to install additional printer driver(s).

## 3 Conclusion

The designed application proposes convenient and easy to use features. But as expected the application can not be perfect and therefore the author plans expanding the capabilities in the following areas:

- Option for adding logos and pictures;
- Ability to customize the font for printing;
- Login option for more security;
- Keyboard mapping for fast working;
- Direct print support;
- Integration with Windows 8/10 Metro UI;
- Sound equalizer;
- QR Code picture correction tools.

## References

- [1] Wikipedia. *Payment card*. 2008, [http://en.wikipedia.org/wiki/Payment\\_card](http://en.wikipedia.org/wiki/Payment_card)
- [2] Mobile Payments Today, 2013, <http://www.mobilepaymentstoday.com/articles/this-rise-of-hce-based-mobile-payments/>
- [3] Wikipedia. *Paper size*. 2009, [http://en.wikipedia.org/wiki/Paper\\_size](http://en.wikipedia.org/wiki/Paper_size)

GEORGI CHAVDAROV  
University of Ruse  
Computer Sciences  
Studentska St. 8  
Bulgaria  
E-mail: powerbsd@abv.bg

# **Groups Tracker**

## **Don't get lost from your friends**

**Bogdan – Mihai Fieraru, Cristian Oancea, Georgiana Secarea,**  
**Cristian – Ioneal Aldulea**

**Teacher Coordinator: Anca Vasilescu**

### **Abstract**

This Android application is designed for finding other users. Developed for people who wants to connect each other, our software allows the users to create groups, track friends from their groups and send messages.

Groups Tracker allows the user to locate each of his friends from its groups on the map, and send them messages for meeting.

## **1 Why we made our choice?**

The idea came to us last summer, when we were in Milano, Italy, visiting the Duomo and was very crowded. We lost each other and some of us didn't have the roaming activated, therefore it was hard and time consuming to gather together again.

When we met, we thought that it would be nice if there would be an application to locate all of us on the map, without being connected to a network. Then we thought it would be nice even using a network connection.

## **2 About the application**

### **2.1 Technology that we used**

On the server side we used Apache Maven. Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information. [3]

On the client side we used Gradle. Gradle is an advanced build toolkit that manages dependencies and allows you to define custom build logic. Android Studio uses a Gradle wrapper to fully integrate the

Android plugin for Gradle. The Android plugin for Gradle also runs independent of Android Studio. This means that you can build your Android apps from Android Studio and from the command line on your machine or on machines where Android Studio is not installed. [4]



We also used for the client OkHttp, which is an HTTP client that's efficient by default. OkHttp perseveres when the network is troublesome: it will silently recover from common connection problems. If your service has multiple IP addresses OkHttp will attempt alternate addresses if the first connect fails. [5]

The application has Google Maps API integrated.

Version control is essential to the success of any software project. It provides the ability for multiple developers to work on the same codebase simultaneously and allows projects to be versioned for release. We used as version control system Git.

Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. As with most other distributed revision control systems, and unlike most client-server systems, every Git working directory is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server. Like the Linux kernel, Git is free software distributed under the terms of the GNU General Public License. [6]

To work more easily with Git we used TortoiseGit. TortoiseGit is a Git revision control client, implemented as a Microsoft Windows shell extension and based on TortoiseSVN. It is free software, also released under the GNU General Public License.

On the server side, we also used WildFly (old JBoss), which is a , lightweight, managed flexible application runtime that help us deploy the application. The data persistency is made by Hibernate ORM. The server use as well Jersey RESTfull Web Services, a framework is open source, production quality, framework for developing RESTful Web Services in Java that provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339) Reference Implementation. [7]

The communication between client and server is made using JSON (JavaScript Object Notation). JSON is a lightweight data-interchange format. It is based on a subset of the JavaScript Programming Language. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language. [8]

The server is online 24/24, thanks to OpenShift, which is a Red Hat's public cloud application development and hosting platform that automates the provisioning, management and scaling of applications. [9]

## **2.2 System requirements**

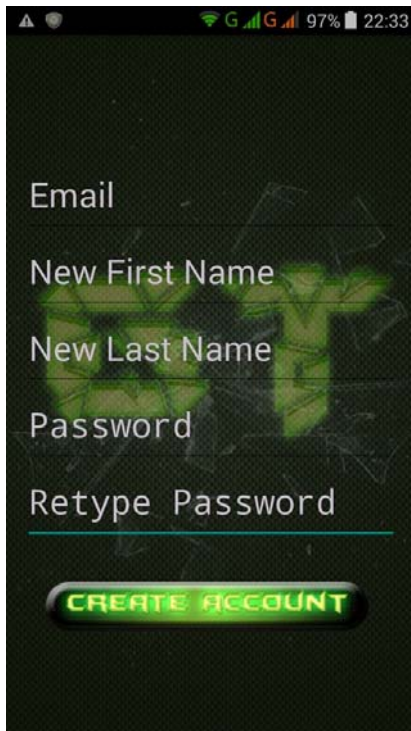
The system requirements for this software are low cost:

- smartphone with Android 4.1.2 JellyBean
- internet connection
- GPS

## **2.3 How the application works**

### **2.3.1 Create Account Activity**

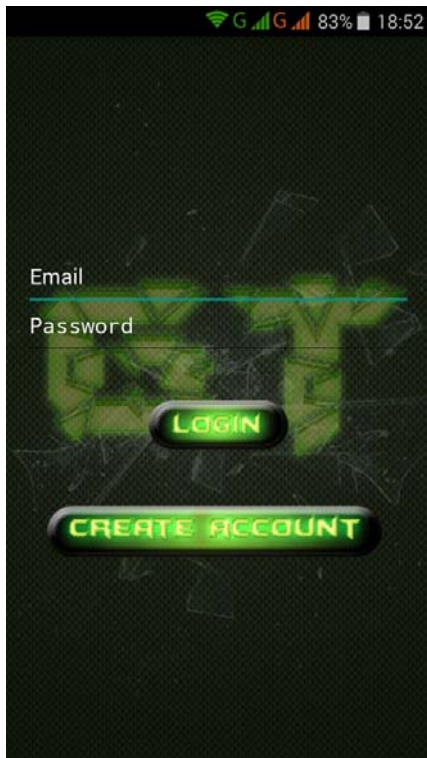
First of all, when the user starts the application, he will be asked if he has an account. In case he doesn't own one, he can press the button that opens the register form. The account name will be the user's email. After creating the account, the user is able to login and start using the application.



### 2.3.2 Login Activity

Login is a mandatory step, and the user must be authenticated with the email for being localized by Google. Otherwise, Google don't grant the user the permission to use the map.

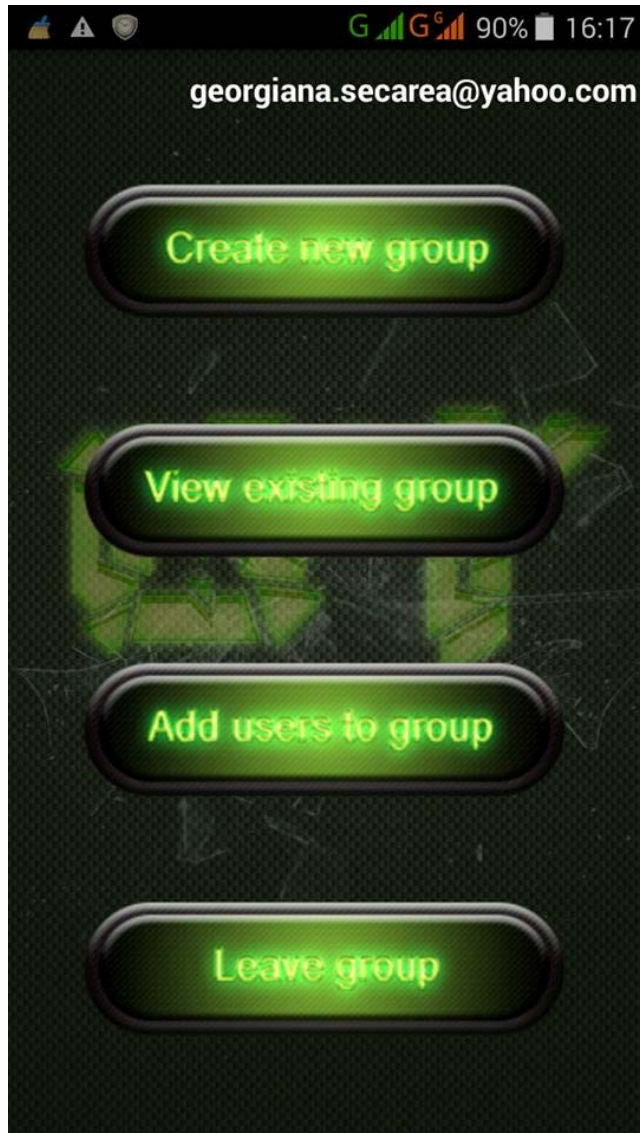
A "login with facebook" feature is in progress to allow the user to skip the Create Account and Login steps.



### 2.3.3 Main Menu Activity

This is the place from where the user can navigate through the application.

He is allowed to create new groups, use the existing ones, add more users to the existing ones or leave a group where he think he should be no more.



### 2.3.4 Create New Group

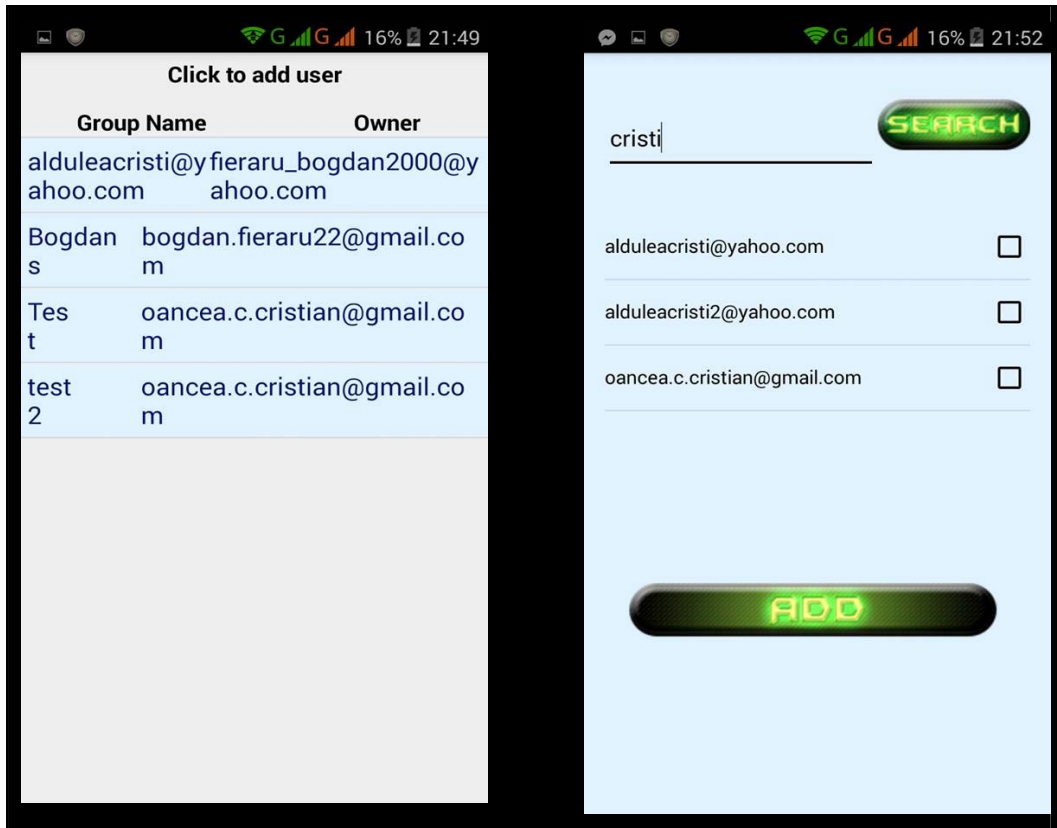
After he is logged in, he is sent to the main menu, where he can create a "buddy group" and add his friends in this group by their email. There is a smart search mechanism (based on Hibernate query language) that allows the user to find his friend even if he did not write the entire email address.

When a friend receive a group invitation, he is able to decline or accept it (first, the invited friend is added in a pending group, and only if he accepts the invitation, he will be added to the real group), and then, both friends can follow each other on the map, and any other friend from that group.



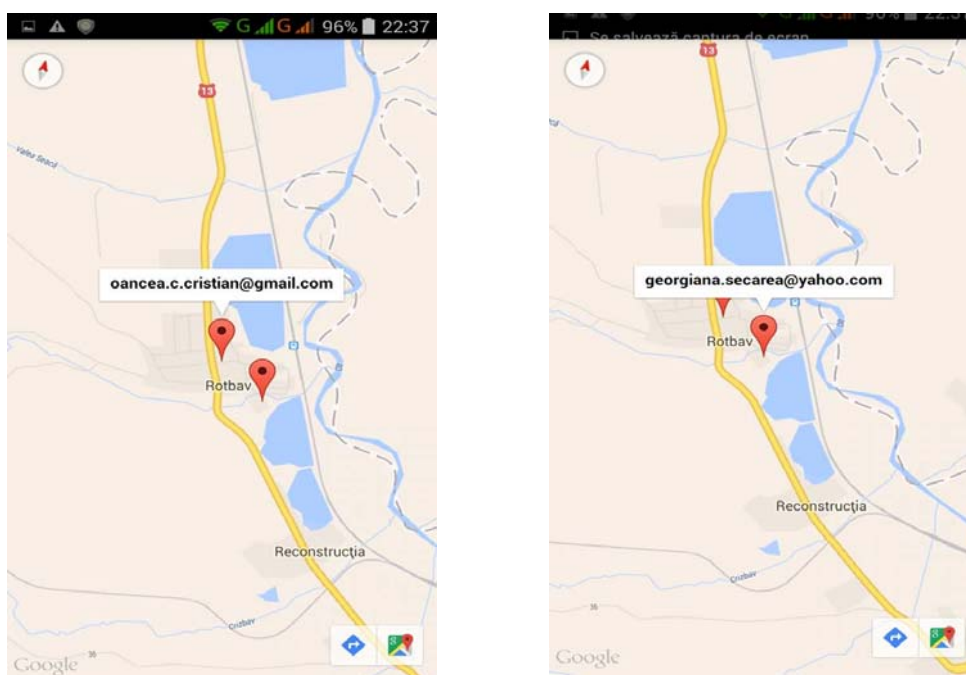
### 2.3.5 Add users to group

First, the user has to select the group where he wants to add his friends, then, he is able to enter his friends email, like is shown in pictures blow:



### 2.3.6 View Groups

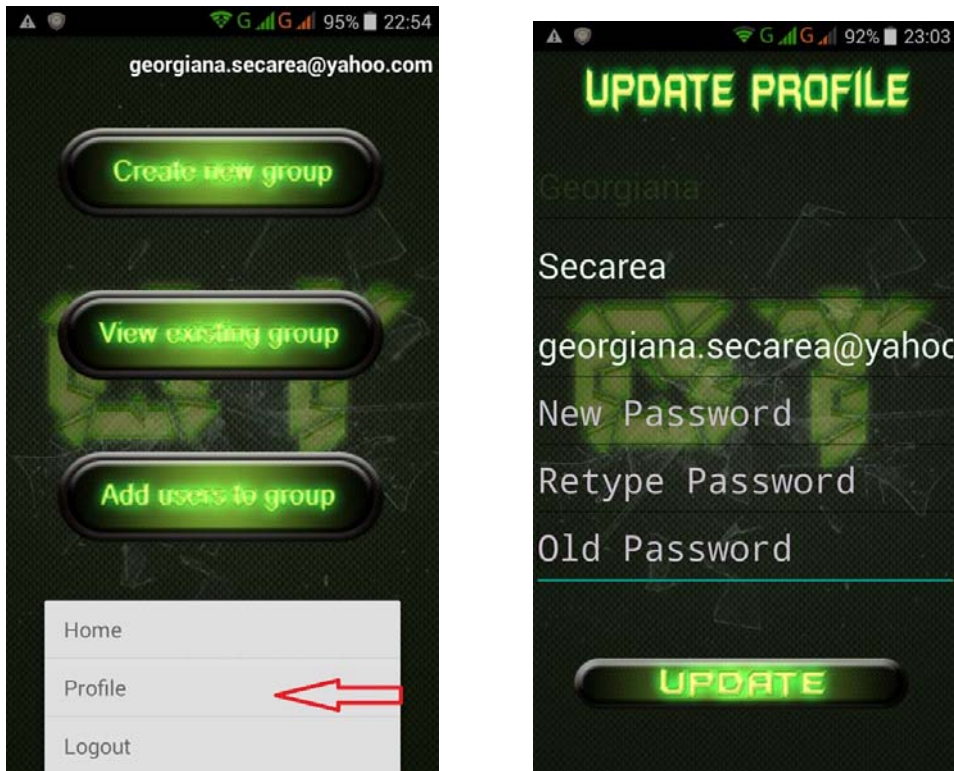
This activity allow the user to visualize the users from a selected group.





### 2.3.7 Update profile

This menu option allow the user to update his profile.



### 2.3.8 Send Messages – future development

Because it is not enough to see on the map where is the lost friend, we need to add the option to send messages between users to establish a meeting place, or just for chat.

## 3 References

- [1] <http://developer.android.com/training/index.html>
- [2] <http://tools.android.com/tech-docs/new-build-system/user-guide>
- [3] <https://maven.apache.org/>
- [4] <https://developer.android.com/tools/building/plugin-for-gradle.html>
- [5] <http://square.github.io/okhttp/>
- [6] <http://en.wikipedia.org/wiki/TortoiseGit>
- [7] <https://jersey.java.net/>
- [8] <http://www.json.org/>
- [9] <https://developers.openshift.com/en/overview-what-is-openshift.html>

Bogdan-Mihai Fieraru  
Transylvania University  
of Braşov  
Department of Mathematics and  
Informatics  
Str.Iuliu Maniu Nr. 50 Brasov ,  
500091  
Romania  
E-mail:  
fieraru\_bogdan2000@yahoo.com

Cristian Oancea  
Transylvania University  
of Braşov  
Department of Mathematics  
and Informatics  
Str.Iuliu Maniu Nr. 50 Brasov  
, 500091  
Romania  
E-mail:  
oancea.c.cristian@gmail.com

Georgiana Secarea  
Transylvania University  
of Braşov  
Department of Mathematics  
and Informatics  
Str.Iuliu Maniu Nr. 50 Brasov ,  
500091  
Romania  
E-mail:  
georgiana.secarea@gmail.com

Cristian Aldulea  
Transylvania University  
of Braşov  
Department of Mathematics  
and Informatics  
Str.Iuliu Maniu Nr. 50 Brasov  
, 500091  
Romania  
E-mail:  
alduleacristi31@gmail.com

## Public transport management in Sibiu

Ionuț Hodină

Teacher Coordinator: Daniel Hunyadi

### Abstract

When we speak about public transport, comes to mind several benefits like: personal and economic opportunities, reduction of carbon footprint and traffic congestion.

In 2013, the local public transport company from Sibiu, received „Growth with Public Transport – Central and Eastern Europe Edition” award at contest “Grow innovative with public transport” during 60th Congress UITP at Geneva. These efforts must be harnessed in order to achieve the full potential of city.

As it's well known, smartphones usage increased substantially in last years whether it's about entertainment, business or everyday activities. Having a smartphone application which offers quick browsing offline timetables and information about stations and routes would be really helpful for local citizens. It would be also helpful for tourists from other cities or even from other countries.

## 1 Introduction

Currently there are few possibilities to find details about public transport in Sibiu. The well known service “Google Maps” is, probably, the most useful when thinking about level of details, but the reality is that information is not updated and it requires a constant internet connection. I found that on low-end hardware smartphones the user experience is not good. Also there are native applications which use Google Maps API, but the results are the same.

Next chapter describes, summary, technologies and concepts that have been used to develop interface and business logic of pc and phone applications. Each part has screenshots followed by explanations. Also, some technical details are mentioned. Follows a chapter with the final conclusions and future planning and the last one with references used as theoretical and study component of this paper.

## 2 Design and implementation

### 2.1 Technologies used

Visual Studio represents a collection of tools and services for developing applications for windows, web, devices and cloud. Offers a development environment for building apps targeting the full range of Microsoft platforms, Windows Phone apps and others. Microsoft Visual Studio has an editor, a compiler and a designer for many programming languages. The programming

Languages included in Visual Studio are: Microsoft Visual C++, Microsoft Visual C#, Microsoft Visual Basic, Microsoft Visual Web Developer and Team Foundation Server. [1]

C# is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It uses the Microsoft .Net Framework, a collection of classes the users can use many data structures.[2]

Windows Presentation Foundation (or WPF) is a graphical subsystem for rendering user interfaces in Windows-based applications by Microsoft. WPF employs XAML, an XML-based language, to define and link various interface elements. WPF applications can also be deployed as standalone desktop programs, or hosted as an embedded object in a website. WPF has a built-in set of data services to enable application developers to bind and manipulate data within applications. In WPF you can define the look of an element directly, via its properties, or indirectly with a Template or Style. At its simplest, a style is a combination of property settings that can be applied to a UI element with a single property attribute. Templates are a mechanism for defining alternate UI for portions of your WPF application. There are several template types available in WPF (ControlTemplate, DataTemplate). WPF supports time-based animations, in contrast to the frame-based approach. This decouples the speed of the animation from how the system is performing.[3]

Extensible Application Markup Language, or XAML, is a declarative markup language. As applied to the .NET Framework programming model, XAML simplifies creating a UI. It can initialize objects, set properties of objects and defines the visual appearance.

JSON or JavaScript Object Notation, is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is used primarily to transmit data between a server and web application, as an alternative to XML. Although originally derived from the JavaScript scripting language, JSON is a language-independent data format. Code for parsing and generating JSON data is available in many programming languages.[4]

Model-View-ViewModel (MVVM) is an architectural pattern for software development. MVVM is a variation of Martin Fowler's Presentation Model design pattern. Like Fowler's Presentation Model, MVVM abstracts a view's state and behavior. Model refers either to a domain model, which represents the real state content (an object-oriented approach), or to the data access layer that represents that content (a data-centric approach). As in the MVC and MVP patterns, the view is the user interface (UI). The view-model is an abstraction of the view that exposes public properties and commands. Instead of the controller of the MVC pattern, or the presenter of the MVP pattern, MVVM has a binder. In the view-model, this binder mediates communication between the view and the data binder. The view-model has been described as a state of the data in the model.[5]

## **2.2 Solution proposed**

Solution that I've worked on, integrates two separate projects. First project handles data management and structure. Second project is a Windows Phone 8.1 application that consumes and reshapes data created by first project. Data is serialized as JSON file on local computer. Later, this file can be loaded in memory using deserialization.

The JSON file created is then uploaded on a web hosting service. From there, the smartphone application downloads it as a long string. This string is deserialized and then data is loaded into



memory. Memory economy and data flexibility are the close-up view of this software architecture so the file contains only raw information.

### 2.3 Data management

This is the first part of the solution which helps to manage data regarding routes and stations of local public transport service. It is developed as WPF project for Windows desktop. Here are defined data models for manipulating objects that represents real life concepts(Route, Station, Time and so on). There are 2 main data models: routes and stations.

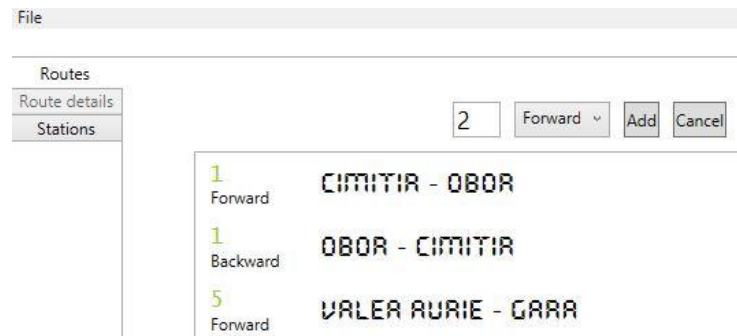


Figure 1. Main Page

In Figure 1 is the main page where routes added can be viewed. Also, other routes can be added after is specified a route number and a direction different from what already exists. Routes list is basically a ListView (WPF control) that binds to code-behind saved routes using a defined template. This template indicates for each route a number, direction and starting/ending points.

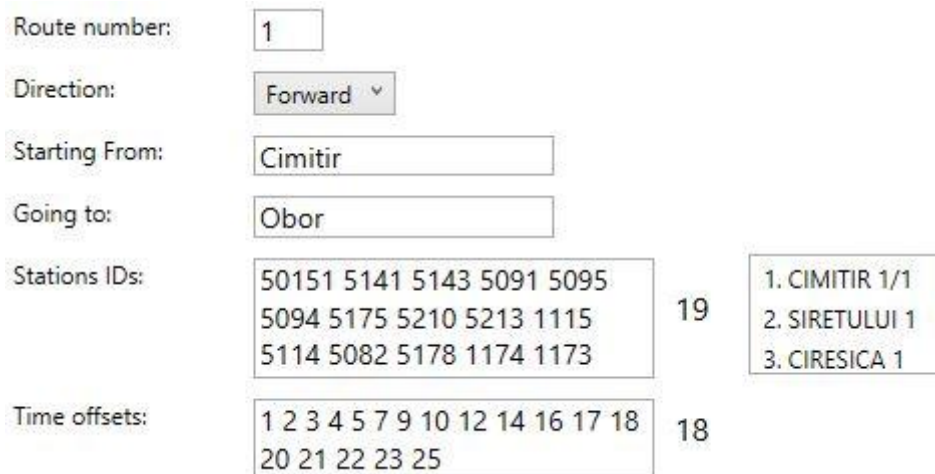


Figure 2. Route Details Page

The picture above is a snapshot from a secondary page where route details can be viewed and edited. Additionally from first page, here can be specified the stations that a route contains. As it can be seen a station is identified by its id. There are three different timetables for a route: Week, Saturday, Sunday. A route saves only the timetables of the first station. To find a specific station

timetables, an offset in minutes must be added from first station timetables. A route contains  $n$  stations and  $n-1$  time offsets.

## 2.4 Tursib for Windows Phone

Represents the Windows Phone 8.1 application needed for the ordinary user to find information about routes and stations. Below are some representative snapshots of this application.



Figure 3. First Page



Figure 4. Route page

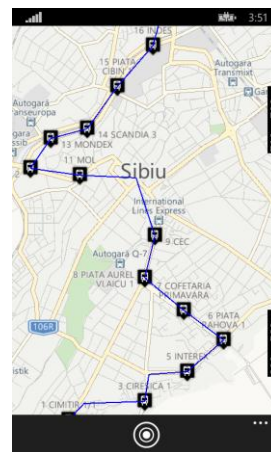


Figure 5. Route map

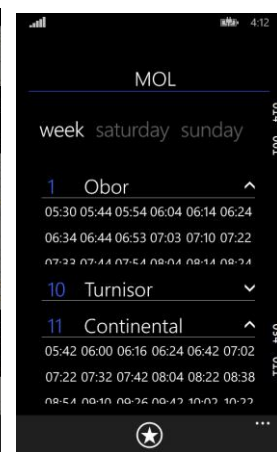


Figure 6. Station timetables

Here is a brief description of the pages involved in this smartphone application:

1. **Main Page** - Here are placed just 3 buttons that link to 3 different pages(Figure 3).
2. **Settings Page** - User can download latest data available. It can also customize the way the application looks.
3. **Favorites Page** - User has a page with shortcuts to favourite stations.
4. **Route Page** – Here are displayed the stations that a route contains. Links to other routes are calculated for a station(Figure 4).
5. **Route Map Page** – Stations of a route are marked on a map and straight lines interconnect them(Figure 5).
6. **Station Timetables Page** – On the top you find the name of the station. Below comes the timetables for selected stations, structured for each route that contains it and by day of the week(Figure 6).

## 2.5 Technical details

The smartphone application is built using Model-View-ViewModel architecture. This means it has great separation of concerns. Properties from View pages bind to specific View-Model classes that implement interface *INotifyPropertyChanged* which allows to notify back the properties when something changed. It is structured such a manner that a lot of code can be reused. Only by providing new layouts of pages and replacing phone specific features(e.g.:location capability) it can be built its Windows 8.1 application correspondent.

Every WP 8.1 application has an App.xaml page which is the main entry point. Here are put data and behavior that can be accessed globally. My application use this feature. Every time it starts, tries to retrieve local saved data. If data is loaded successfully, then builds an object that can access all data related to routes and stations. On the local storage are saved only 2 lists(routes and stations), but the links between them are very limited. A route has only IDs of stations, while a station doesn't know about routes that use it. So it builds kind of a graph object with circular references. Here comes 2 new data entities: RouteNode(contains information about that route and a list of StationNodes) and StationNode(contains information about that station and a list of RouteNodes). By doing this, on the selected station page can be viewed all routes that use it, only by binding properties to that StationNode object. Also, it will not be hard to implement the path finding algorithm between 2 map points that use local bus transport.

Several features are implemented. Application has the ability to load the specific language on the UI based on phone language. Theme color changes based on phone accent color. Timetables are loaded based on day of the week supplied by phone. Costly CPU operations are implemented as asynchronous methods.

### 3 Conclusion and future features

Solution created includes both data management and a smartphone applications for quickly finding information about stations and routes of public transport in Sibiu.

Several features should be added in order to maximize user experience and application effectiveness. Pathfinding, live tiles and sharing options are at the top of priorities. As Windows 10 will be released this year, this application needs an upgrade to the new platform.

### 4 References

- [1] Official Visual Studio webpage, <http://www.visualstudio.com>
- [2] C# programming language, [http://en.wikipedia.org/wiki/C\\_Sharp\\_%28programming\\_language%29](http://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29)
- [3] WPF technology, [http://en.wikipedia.org/wiki/Windows\\_Presentation\\_Foundation](http://en.wikipedia.org/wiki/Windows_Presentation_Foundation)
- [4] JSON data format, <http://en.wikipedia.org/wiki/JSON>
- [5] Model-View-ViewModel, [http://en.wikipedia.org/wiki/Model\\_View\\_ViewModel](http://en.wikipedia.org/wiki/Model_View_ViewModel)

Hodină Ionuț  
Faculty of Science "Lucian Blaga"  
Department of Informatics  
Str. Dr. I. Ratiu, No. 5-7, Sibiu, 550012, România  
Email: ionut.hodina@gmail.com

## **Modeling the population evolution in Romania in the period 1960 -2012**

**Mihaela Itu, Toporaş Andreea**  
**Teacher Coordinator: Alina Bărbulescu**

### **Abstract**

Knowledge about the population evolution is important for forecasting the human resources that are between the most important production factors. Therefore, this study aim to model the population evolution in Romania, using two types of models: the linear econometric one and the ARMA one, for the period before on after the change point of the series (that was detected to be 1991, when the maximum population of Romania has been registered). It is proved that the models fit well the data and can be used for prediction.

## **1 Introduction**

The demographic transition of Romania has become a complex process, as an integrant part of economic and social development and of societal modernization. The extent of this process is reflected in his behavior: the fertility and mortality transition, as well as the age structure, the marriage, family and urbanization transitions.

Transformations occurring after 1989 in the political system, economy, social life and peoples' mentality influenced the demographic behavior of the population. Current demographic situation of the country is the cumulative result of the evolution of birth, mortality, and external migration. Examining the dynamics of these components it can be seen that the birthrate decrease has major implications compared to others phenomena in the actual demographic decline, especially on long and very long term because it contributes to the population decreasing and to the damage of its age structure.

In the period 1960 - 1990 the evolution of population in Romania registered and incresing trend, that was followed by a decreasing one. In 1990, the population was of 23.206.720 inhabitants. In 2012 the Romanian population decreased of over 56.000 persons due the negative natural birth rate [6]. Romanian`s population will continue to decrease in the next year, such that in 2030 the population number will be of 19 million persons, and in 2050 it will reach 15.5 million inhabitants [5].

We choose to study the evolution of the population between 1960 and 2012 since the issue of population decline is of great interest, being accompanied by an alert ageing rhythm. After 1990 the number of young population decreased and automatically the weight of elderly population increased, augmenting the degree of demographic ageing. From a population of 1.7 million aged over 60 years before 1989, it arrived at a population of 4.3 million over the age of 60, in 2010. Therefore, one active person supports 4 retired employees.

The article has the following structure: Section 2 contains the study methodology, Section 3 contains the modeling results and Section 4 contains the pseudocode. The final section contains the conclusions.

## 2 Methodology

### 2.1 Unifactorial linear model

An unifactorial model is a model that has the equation:

$$Y = f(X) + u, \quad (1)$$

where:

$Y$  is the endogenous variable, whose registered values are  $(y_1, \dots, y_n)$ ;

$X$  is the exogenous variable, whose registered values are  $(x_1, \dots, x_n)$ ;

$u$  is the stochastic variable, whose registered values are  $(u_1, \dots, u_n)$ .

The model identification consists of choosing a mathematical function that describes the values of endogenous variables as well as possible function of exogenous variable.

The equation of the econometric unifactorial linear model is:

$$Y = a + bX + \varepsilon, \quad (2)$$

or

$$y_i = a + bx_i + \varepsilon_i. \quad (3)$$

The parameters  $a$  and  $b$  are unknown and will be estimated by the least squares method, that consists of minimizing the function:

$$F(a, b) = \sum_{i=1}^n (y_i - a - b * x_i)^2. \quad (4)$$

The hypotheses on the econometric unifactorial model are:

1.  $\varepsilon_t$  are normally distributed,
2.  $E(\varepsilon_t) = 0$ ,
3.  $E(\varepsilon_t^2) = \sigma_\varepsilon^2$ ,
4.  $E(\varepsilon_t, \varepsilon_{t'}) = 0, t \neq t'$ ,
5.  $Cov(x_t, \varepsilon_t) = 0$ ,

where  $E$  denotes the mean and  $Cov$  denotes the covariance and  $\sigma_\varepsilon^2$  is the variance of the residual  $\varepsilon_t$ .

These hypotheses are checked using the *Levene* test (for homoscedasticity) [3], the Shapiro-Wilk test (for normality) [4] and studying the autocorrelation function.

In the Levene test the null hypothesis is:  $H_0: \sigma_1^2 = \sigma_2^2 = \dots = \sigma_k^2$ , and its alternative is  $H_1: \sigma_i^2 \neq \sigma_j^2$ , for at least one pair  $(i, j)$ . For applying it, the statistics

$$W = \frac{n-k}{k-1} \cdot \frac{\sum_{i=1}^k n_i (\bar{Z}_i - \bar{Z}_{..})^2}{\sum_{i=1}^k \sum_{j=1}^{n_i} (Z_{ij} - \bar{Z}_i)^2}, \quad (5)$$

is built, where:  $k$  is the number of groups in which the sample is divided,  $n$  is the sample volume,  $n_i$  is the sample size of each group,  $X_{ij}$  is the element  $j$  in the  $i$ -th group,  $\bar{X}_i$  is the mean of the  $i$ -th group,

$$Z_{ij} = |X_{ij} - \bar{X}_i|, \quad (6)$$

$\bar{Z}_i$  is the mean group of  $Z_{ij}$  and  $\bar{Z}_{..}$  is the overall mean of  $Z_{ij}$ .

The null hypothesis is rejected at the significance level  $\alpha$  if  $W > F_{\alpha, k-1, n-k}$ , where  $F_{\alpha, k-1, n-k}$  is the upper critical value of the Fisher – Snedecor distribution with  $k - 1$  and  $n - k$  degrees of freedom at  $\alpha$  significance level [1].

In the the Shapiro - Wilk test the null hypothesis is that the series is normally distributed (i.e. Gaussian) and its alternative is that the series is not normally distributed. The statistic test is defined as:

$$W = \frac{\left( \sum_{i=1}^n w_i u'_i \right)^2}{\sum_{i=1}^n (u_i - \bar{x})^2}, \quad (7)$$

where  $n$  is the sample volume,  $u'_1, u'_2, \dots, u'_n$  are the increasingly ordered data,  $\bar{x}$  is the sample mean,  $w' = (w_1, w_2, \dots, w_n)$  or

$$w' = MV^{-1} \left[ (M^{-1})(V^{-1}M) \right]^{-1/2}, \quad (8)$$

where  $M$  denotes the expected values of standard normal order statistics for a sample of size  $n$  and  $V$  is the corresponding covariance matrix. Small values of  $W$  indicate the normality absence [1].

The *autocorrelation function* (ACF) of a time series  $(X_t)$  at lag  $h$  ( $h \in \mathbf{N}^*$ ) is defined by:

$$\rho(h) = \frac{\text{Cov}(X_t, X_{t+h})}{D^2(X_t)}, \quad h \in \mathbf{N}^*. \quad (9)$$

$h$  is called *lag*.

To verify if the values of a time series are correlated, they are compared with the limits of the confidence interval, at a confidence level (95% in our case). If the ACF values are inside the corresponding confidence interval, we can conclude the data are not correlated.

To validate a linear econometric model, each parameter must be significant, and the model must be significant as a whole.

Checking the significance of each parameter  $a, b$  is done by the test  $t$  Student whose hypotheses are:  $H_0: a = 0, H_1: a \neq 0$ , respectively  $H_0: b = 0, H_1: b \neq 0$ .

Checking the significance of the model as whole is done using the test  $F$  Fisher-Snedecor test, which has the null hypothesis:  $H_0: a = 0$  or  $b = 0$ , and the alternative one:  $H_1: a \neq 0$  and  $b \neq 0$ .

All tests are carried out at the level of significance  $\alpha = 0.05$ . If the p-value associated to the tests is less than the 0.05, then null hypothesis is rejected.

An indicator of the performance of the linear model unifactorial is the determination coefficient ( $R^2$ ). The closer  $R^2$  to 1, better the model is.

## 2.2 ARMA models

A discrete process  $(X_t)$  is said to be autoregressive of  $p$  order (AR( $p$ )) – if:

$$X_t = \sum_{i=1}^p \varphi_i X_{t-i} + \varepsilon_t, \quad \forall t \in \mathbf{N}^*, \varphi_p \neq 0, \quad (10)$$

where  $(\varepsilon_t)$  is a white noise.

A discrete process  $(X_t)$  is said to be autoregressive moving average process of  $p$  and  $q$  orders (ARMA( $p, q$ )) if:

$$X_t - \sum_{i=1}^p \varphi_i X_{t-i} = \varepsilon_t - \sum_{j=1}^q \theta_j \varepsilon_{t-j}, \quad \forall t \in \mathbf{N}^*, \varphi_p \neq 0, \theta_q \neq 0, \quad (11)$$

where  $(\varepsilon_t)$  is a white noise [2].

Particularly, ARMA( $p, 0$ ) = AR( $p$ ).

Between the possible models of ARMA type best one was selected, based on the Akaike Information Criterion (AIC).

For modeling and testing different hypotheses n models, the R software has been used.

### 3 Results

The data modelled is formed by the series of annual population, as whole and on genders (masculine and feminine) registered in the period 1960 - 2012 (Figure 1). The data are taken from the Statistical Yearbook of Romania. Each data series was divided in two subseries: the first one contains the first 31 data, and the second one, the rest. The partition has been done taking into account that the 31<sup>st</sup> data was the point of maximum. For the first subseries a linear model is proposed. For the second sub-series an ARMA model has been determined.

#### 3.1 Linear econometric models

In this model, the independent variable is the time, whose values are numbered from 1 to 31, corresponding the years from 1960 to 1990, and the dependent variable is the population number.

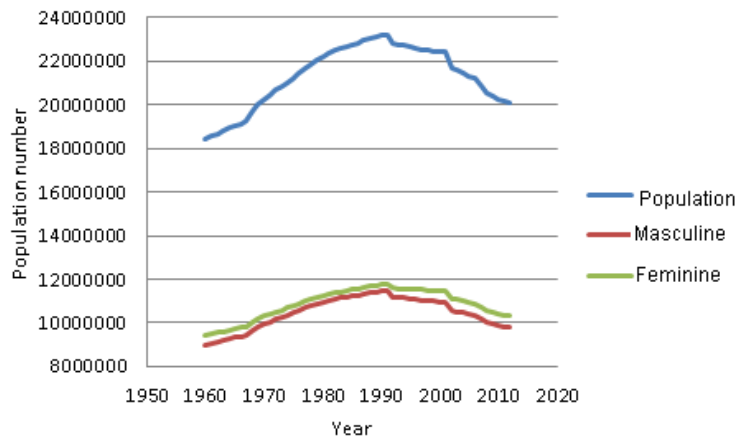


Figure 1. The evolution of the total population and by sex during the period 1960-2012

The results of the models are presented in Tables 1 and 2.

The columns of Table 1 contain respectively:

1. the second column: the estimated coefficients of the model (2);
2. the 3<sup>rd</sup> and the 5<sup>th</sup> columns: the values of the  $t$  and  $F$  statistics;
3. the 4<sup>th</sup> and the 6<sup>th</sup> columns: the p-values corresponding to the  $t$  and  $F$  tests;
4. the 7<sup>th</sup> column: the determination coefficient.

Table 1. Results of the tests on the coefficients of the linear model

	Coefficients	t-stat	p-val (for t)	F-stat.	p-val (for F)	$R^2$ (%)
Population	a=18231497.193	219.839	0.000	1503.128	0.000	98.10
	b=175406	38.770	0.000			
Masculine	a=8905323.677	199.902	0.000	1376.257	0.000	99.93
	b=90159.387	37.097	0.000			
Feminine	a=9326173.574	242.617	0.000	1652.458	0.000	98.27
	b=85246.633	40.650	0.000			

Since the p - values are less than 0.05, the  $t$  and  $F$  tests rejected the null hypotheses, so the coefficients of the models and the models, as whole, are significant.

Table 2 contains the results of the Levene and Shapiro –Wilk tests on the random variable,  $u$ . Since the p-values are greater than 0.05, we couldn't reject the hypotheses that the series are homoskedastic and Gaussian. Analysis of ACF proved that the autocorrelation hypothesis was rejected. So, the models are correct viewpoint of statistics.

Table 2. Results of the homoscedasticity and normality tests for residuals

	Levene test		Shapiro- Wilk test	
	Statistics value	p-value	Statistics value	p-value
Population	2.06	0.162	0.980	>0.100
Masculine	2.53	0.122	0.980	>0.100
Feminine	1.55	0.223	0.978	>0.100

### 3.2 ARMA models

For the sub-series registered in the period 1991-2012, the following models of type (11) have been found, after the mean subtraction:

- for the total population:

$$X_t = 0.6721X_{t-1} + 0.2239X_{t-2} + \varepsilon_t + 0.1764 \varepsilon_{t-1},$$

where  $(\varepsilon_t)$  is a white noise;

- for the masculine population:

$$X_t = 1.415 X_{t-1} - 0.4290 X_{t-2} + \varepsilon_t,$$

where  $(\varepsilon_t)$  is a white noise;

- for the feminine population (Figure 2):

$$X_t = 1.974 X_{t-1} - 0.9885 X_{t-2} + \varepsilon_t - 0.9951\varepsilon_{t-1}$$

where  $(\varepsilon_t)$  is a white noise.

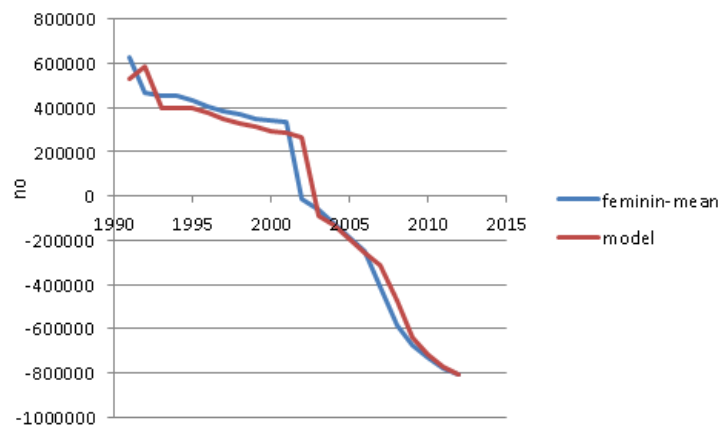


Figure 2. Model for the series of feminine population after the mean subtraction – period 1991-2012

## 4 Program

The program has been realized using the R software.

In order to realize our models, below are shown the steps that have been followed:

- Install the R packages 'acf', 'aic', 'ar', 'arima', 'lawstat', 'lm', 'shapiro.test';



- Fit the linear model for the series registered in the period 1960 - 1990;
- Check the hypotheses on the significance of the series coefficients and on the models' significance;
- Check the hypotheses that the residual forms an white noise;
- Built the ARMA models for the sub-series registered in the period 1990 – 2012;
- Choose the best models based on AIC.

## 5 Conclusions

The present study proved the existence of two periods of population evolution in Romania, after 2012. We provided here two types of models: linear, for the first period, and ARMA, for the second period. It was proved that the models fit well the data. Therefore, the ARMA models can be used for forecasting the population evolution for the next period.

## References

- [1] A. Bărbulescu, C. Șerban (Gherghina), C. Maței, Statistical Analysis and Evaluation of Hurst Coefficient for Annual and Monthly Precipitation Time Series, *WSEAS Transactions on Mathematics*, 10(9): 791 – 800, 2010.
- [2] P. Brockwell, R.A. Davis, *Introduction to time series and forecasting*, Springer, 2002.
- [3] H. Levene, Robust tests for equality of variances, *contributions to Probability and statistics: essays in Honor of Harold Hotelling*, I. Olkin et al. Eds, Stanford University Press, 278 – 292, 1960.
- [4] S. S. Shapiro, M. B. Wilk, An Analysis of Variance Test for Normality (Complete Samples), *Biometrika*, 52 (3/4): 591-611, 1965.
- [5] <http://www.wall-street.ro/articol/Economie/27686/Chinezii-isi-cauta-visul-occidental-in-estul-Romaniei.html>
- [6] <http://www.wall-street.ro/articol/Economie/89095/Populatia-Romaniei-in-continua-scadere.html>

ITU MIHAELA  
Ovidius University of Constanta  
Faculty of Mathematics and Computer Science  
124, Mamaia Blvd. Constanta  
ROMANIA  
E-mail: [itu\\_mihaela\\_93@yahoo.com](mailto:itu_mihaela_93@yahoo.com)

TOPORAS ANDREEA MONICA  
Ovidius University of Constanta  
Faculty of Mathematics and Computer Science  
124, Mamaia Blvd. Constanta  
ROMANIA  
E-mail: [andreeamonica88@yahoo.com](mailto:andreeamonica88@yahoo.com)

# Graphic Engine

Arpad Kerestely  
Teacher Coordinator: Adrian Deaconu

## Abstract

This paper presents a 3D graphic engine made in C++ with the help of a few open-source, cross-platform libraries like OpenGL, SDL, Assimp. It's meant to cover the graphical necessities of a 3D game, offering high quality over low CPU performance, taking most of the computation to the GPU.

The engine offers terrain rendering, dynamic sky shading based on time of the day, sun with sun rays, water rendering, loading and animating of 3D models, scene lighting based on sun position, etc.

## 1 Introduction

In the industry of informatics and programming, game industry has a huge word to say. *"Modern personal computers owe many advancements and innovations to the game industry: sound cards, graphics cards and 3D graphic accelerators, faster CPUs, and dedicated co-processors like Physx are a few of the more notable improvements. Modern games are among the most demanding of applications on PC resources. Many of the high-powered personal computers are purchased by gamers who want the fastest equipment to power the latest cutting-edge games."*[3]

*"Considered by some as a curiosity in the mid-1970s, the computer and video game industries have grown from focused markets to mainstream. They took in about US\$ 9.5 billion in the US in 2007, 11.7 billion in 2008, and 25.1 billion in 2010"*[3]. For 2015 worldwide, the game industry market is estimated to grow to US\$ 115 billion.[4] With the game industry evolving, higher quality video games are required, which have better graphics, and simulate the real life more accurately.

Therefore, the application presented aims to bring a cross-platform graphic engine that delivers high quality image at relatively low cost using mainly the GPU for complex computations, leaving the CPU for other jobs, such as game logic and physics engine.

### 1.1 Other existing engines

There are quite a few existing engines out there, most popular being Unity3D[1] and Unreal Engine[2]. In comparison to my engine, they offer a lot more: collision detection, configurable game engine, many post processing effects, sound and UI management through a user friendly

environment. The disadvantage is that they are heavy-weighted; Unity3d costs money if you want to publish your game, and Unreal Engine takes 5% royalty on games and application your release.

## 1.2 Project inspiration and goals

The idea of making a graphic engine came from a previously made game which lacked a graphic engine. The resulting game obviously did not gave a very good first impression, although it had a pretty good handler for game logic and physics. The conclusion was simple: catch the viewers eyes [first] than worry about the rest [later]. The idea of building it in C++ with OpenGL, was to make it fast and cross platform.

### 1.2.1 Main goals of the project

1. Building a good looking graphic engine of my own.
2. Learn modern OpenGL (with computations on GPU).

### 1.2.2 Secondary goals of the project

Needed to accomplish the main goals.

1. Learn GLSL (OpenGL Shading Language).
2. Build terrain based on height map.
3. Load 3d models and armature.
4. Implement model animation.
5. Make it load fast.
6. Have dynamic sky, sun and sun rays.
7. Make semitransparent waving water.
8. Implement ambient and diffuse scene lighting.
9. Implement particle system, for optimized rendering.
10. Have a minimal UI for scene organization.

## 2 Used Libraries

All of the used libraries are cross-platform, which makes the application easily portable to other operating systems, and are built in C/C++ which makes them fast.

## 2.1 OpenGL

*“OpenGL (Open Graphics Library) is a cross-language, multi-platform application programming interface (API) for rendering 2D and 3D vector graphics. The API is typically used to interact with a graphics processing unit (GPU), to achieve hardware-accelerated rendering. The API is defined as a number of functions which may be called by the client program, alongside a number of named integer constants.”*[5]

The first version of OpenGL, version 1.0, was released in 1992. This version used a fixed functionality pipeline, and was mainly CPU consumer. With version 2.0, which came out in 2004, introduced a programmable pipeline, where programmers could code their own shader in a C-style language GLSL (OpenGL Shading Language). The latest version of OpenGL was released in 2014.

The engine uses OpenGL version 2.1 and GLSL version 1.20 which makes it run on most of today's hardware and makes it OpenGL ES (a lighter version of OpenGL used on Android and iOS devices) compatible.

The choice to use OpenGL instead of Microsoft's Direct3D was to make the engine run on every existing platform, not only on Microsoft's platforms (Windows, Windows Phone, Xbox360).

## 2.2 GLEW

Given the high workload involved in identifying and loading OpenGL extensions, a few libraries have been designed which load all available extensions and functions automatically. One of the most used library of this kind is OpenGL Extension Wrangler Library (GLEW).

*“GLEW is a cross-platform open-source C/C++ extension loading library. GLEW provides efficient run-time mechanisms for determining which OpenGL extensions are supported on the target platform. OpenGL core and extension functionality is exposed in a single header file, which is machine-generated from the official extension list. GLEW is available for a variety of operating systems.”*[6]

There are a few other extension loader libraries, but GLEW seems to be the most used and most reliable, although it loads all extensions on program start (even those that the program does not use), and does not check for errors (ex. you can use `glGenBuffers(GL_FLOAT)` which is not a valid operation).

## 2.3 SDL 2

Given that creating an OpenGL context is quite a complex process, and given that it varies between operating systems, automatic OpenGL context creation has become a common feature of several game-development and user-interface libraries such as Simple Directmedia Layer (SDL).

*“SDL is a cross-platform development library designed to provide low level access to audio, keyboard, mouse, joystick, and graphics hardware via OpenGL and Direct3D. It is used by video playback software, emulators, and popular games including Valve's award winning catalog and many Humble Bundle games. Officially supports Windows, Mac OS X, Linux, iOS, and Android. It is written in C, works natively with C++.”*[7].

## 2.4 Assimp

*“Open Asset Import Library (Assimp) is a cross-platform 3D model import library which aims to provide a common application programming interface (API) for different 3D asset file formats. Written in C++, it offers interfaces for both C and C++. The imported data is provided in a*

*straightforward, hierarchical data structure. Assimp currently supports 41 different file formats for reading, including COLLADA (.dae), 3DS, DirectX X, Wavefront OBJ and Blender 3D (.blend).*”[8]

## 2.5 GLM

*”OpenGL Mathematics (GLM) is a header only C++ mathematics library for graphics software based on the OpenGL Shading Language (GLSL) specifications.*

*GLM provides classes and functions designed and implemented with the same naming conventions and functionalities than GLSL so that anyone who knows GLSL, can use GLM as well in C++.*

*This project isn’t limited to GLSL features. An extension system, based on the GLSL extension conventions, provides extended capabilities: matrix transformations, quaternions, data packing, random numbers, noise, etc.*

*This library works perfectly with OpenGL but it also ensures interoperability with other third party libraries and SDK. It is a good candidate for software rendering (raytracing / rasterisation), image processing, physic simulations and any development context that requires a simple and convenient mathematics library.*

*GLM is written in C++98 but can take advantage of C++11 when supported by the compiler. It is a platform independent library with no dependence.*”[9]

## 3 A peek inside

### 3.1 ’Initializing...’

The engine starts with context and window creation, camera initialization and loading of all the data required for the terrain (height map) and models(vertex data, bone data, textures). In the init function, the sky dome, sun, and water models are generated. Basically this is where the scene building is accomplished.

```
void init()
{
    window.Create("Engine", screenWidth, screenHeight, Engine::FULLSCREEN);
    SDL_SetRelativeMouseMode(SDL_TRUE);

    camera.Init(screenWidth, screenHeight);

    level = new Level("Resources/Map/imgn45w114_1");
    levelRenderer = new LevelRenderer(level);
    modelManager = new ModelManager(level);

    sky = new Engine::SkyDome;
    sun = new Engine::Sun(screenWidth, screenHeight);
    water = new Engine::Water;
}
```

### 3.1.1 Camera

The camera stores and computes the transformation matrices needed to get the object from view space to projection space and than to screen coordinates. In the *Init* function the camera builds the projection matrix with the use of screen width and height, and sets the model matrix to be the identity matrix.

```
void Camera::Init(const int screenWidth, const int screeHeight)
{
    this->screenWidth = screenWidth;
    this->screeHeight = screeHeight;
    projectionMatrix = glm::perspective(VIEW_ANGLE, (float)screenWidth/screeHeight,
                                        NEAR_CLIP_PLANE, FAR_CLIP_PLANE);
}
```

### 3.1.2 Terrain

The terrain generating starts with the construction of a new *Level* object. The terrain's height map is stored in a binary file containing  $nRows * nCols$  floats, representing the height data. The  $nRows$  and  $nCols$  are the first two *unsigned int* data from the binary file. With the height data loaded, the *LevelRenderer* generates the vertex positions, normals and texture coordinates. Note: vertex normals don't exist in geometry, but they do exist in graphic programming, and represent:  $vertexNormal = normalize(\sum_{i=0}^k N_i)$ , where  $k$  is the number of adjacent triangles and  $N_i$  is the normal of the  $i$ -th triangle. The last step of the terrain generating is to make triangles out of these vertexes, by generating the indexes array and to upload all the data to the GPU.

```
void LevelRenderer::buildModel(const Level *level)
{
    const float* levelData = level->GetData();

    //construct vertex position based on level data
    Engine::Vertex *vertices = new Engine::Vertex[ nRows * nCols ];
    for(int z=0; z<nRows; z++)
        for (int x=0; x<nCols; x++)
        {
            float y = levelData[z*nCols + x];
            vertices[z*nCols + x].SetPosition(x*CELL_SIZE, y, z*CELL_SIZE);
            vertices[z*nCols + x].SetUV(x,z);
        }
    //calculate vertex normals based on triangles formed with adjacent vertexes
    for(int z=1; z<nRows-1; z++)
        for (int x=1; x<nCols-1; x++)
        {
            Position leftUpper = Position::Normal(vertices[z*nRows + x].position,
            vertices[(z-1)*nRows + x].position, vertices[z*nRows + x-1].position);
            Position centerUpper = Position::Normal(vertices[z*nRows + x].position,
            vertices[(z-1)*nRows + x+1].position, vertices[(z-1)*nRows + x].position);
            Position rightUpper = Engine::Position::Normal(vertices[z*nRows + x].position,
```

```
        vertices[z*nRows + x+1].position, vertices[(z-1)*nRows + x+1].position);
    Position rightLower = Engine::Position::Normal(vertices[z*nRows + x].position,
        vertices[(z+1)*nRows + x].position, vertices[z*nRows + x+1].position);
    Position centerLower = Position::Normal(vertices[z*nRows + x].position,
        vertices[(z+1)*nRows + x-1].position, vertices[(z+1)*nRows + x].position);
    Position leftLower = Position::Normal(vertices[z*nRows + x].position,
        vertices[z*nRows + x-1].position, vertices[(z+1)*nRows + x-1].position);
    vertices[z*nRows + x].normal = (leftUpper + centerUpper + rightUpper +
        rightLower + centerLower + leftLower);
    vertices[z*nRows + x].normal.Normalize();
}
//assign triangle indices. Two triangles at once, that form a rectangle.
unsigned int *indices = new unsigned int[(nCols-1)*(nRows-1)*2*3];
for(int i=0,k=0; i<nRows-1; i++)
    for(int j=0; j<nCols-1; j++)
    {
        indices[k++] = i*nCols + j;
        indices[k++] = (i+1)*nCols + j;
        indices[k++] = i*nCols + j+1;

        indices[k++] = i*nCols + j+1;
        indices[k++] = (i+1)*nCols + j;
        indices[k++] = (i+1)*nCols + j+1;
    }
//upload to vertices to GPU
glGenBuffers(1, &vboId);
glBindBuffer(GL_ARRAY_BUFFER, vboId);
glBufferData(GL_ARRAY_BUFFER, sizeof(Engine::Vertex) * nRows * nCols,
    vertices, GL_STATIC_DRAW);
glBindBuffer(GL_ARRAY_BUFFER, 0);
delete[] vertices;
//upload to indices to GPU
glGenBuffers(1, &iboId);
glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, iboId);
glBufferData(GL_ELEMENT_ARRAY_BUFFER, sizeof(unsigned int)*(nCols-1)*(nRows-1)*2*3,
    indices, GL_STATIC_DRAW);
glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, 0);
delete[] indices;
}
```

### 3.1.3 Sky Dome

A Sky Dome is a sphere cut in two, with a circle, as base. This object's vertexes and indexes are generated and uploaded to GPU in the *Init* function. Note: the vertexes generated here have only position; they don't have normals nor texture coordinates. The Sky Dome will represent much of the background of a 3D game.

### 3.1.4 Sun

The Sun is basically just a circle, although it appears to be a sphere. To trick the user into believing that it's a sphere, but making much less computations by representing it as a circle is done using a trick called billboarding. A billboard is a 2D object in a 3D world that rotates, moves or scales according to the position and rotation of the camera. In this application, the sun rotates as the camera rotates, and moves with the camera, so the user can never see the sun from behind. The vertexes generated in the *Init* function have only position.

```
//generate vertices of circle
const float slice = 2*PI/nVertices;
for (unsigned int i=1, iVert=1; i<=nVertices; i++, iVert++)
{
    float crtSlice = slice*i;
    vertices[iVert].SetPosition(center.position.x + cos(crtSlice) * radius,
        center.position.y + sin(crtSlice) * radius, center.position.z);
}
```

### 3.1.5 Ocean

The Ocean is a simple flat plane split in many triangles into the *Init* function. The weaving effect will be computed in the vertex shader every time we render it. The vertexes present here have position, normal and texture coordinates.

### 3.1.6 Model Manager

The Model Manager handles the work with every imported models present while the engine runs, also handles the work with the Particle System. In the *Init* function the Model Manager loads all the available models to an array of pointers once, then loads data referring to models positions, rotation, scale and finally builds particle systems with trees, rocks, grass.

### 3.1.7 Particle System

Calling many OpenGL calls a frame produces the phenomena known as '*bottleneck*'. The rendering of a single model (ex. a tree) repetitively (500-1000 times) with different transformation matrices produces this problem. To avoid this, a solution was needed to draw all those models with a single OpenGL call. Thus came the idea of putting a model repetitively in a buffer, with the transformation matrices already applied, and rendering the models (the whole buffer) as a single object. The class managing these actions is the Particle System.

## 3.2 First Rendering. Lazy Init.

The shaders are loaded during runtime, at the moment they are needed (lazy init).

```
Render(...)
{
    if (shaderProgram == nullptr)
        initShaders();
    ...
}
```



### 3.3 Rendering

After everything needed loaded, the rendering begins.

#### 3.3.1 Main render function

The order of rendering does matter. The sky needs to be rendered first, because it has depth buffer disabled, and sun needs to be rendered last, because the effect of the sun rays is in fact a blurring (post processing) and a custom depth testing of the resulted final image. The rendering of all the other objects come in between.

```
void MainGame::renderScene()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    sky->Render(camera, sun->GetSunPosition());
    levelRenderer->Render(camera, sun);

    modelManager->Render(camera, sun);

    water->Render(camera, time);
    sun->Render(camera);

    window.SwappBuffer();
}
```

#### 3.3.2 Model Manager render function

The *ModelManager* has a list of *particle systems*, *3D models* and *model elements*. Model elements have a pointer to a model, and information about it like rotation, position, scale. Every frame the *ModelManager* renders the particle systems and the model elements.

```
void ModelManager::Render(const Camera &camera, const Sun *sun)
{
    //render models with specific position, rotation and scale
    //update animation frame based on current time
    for (auto it = modelElements.begin(); it<modelElements.end(); it++)
    {
        it->model->Update(time);
        it->model->Position = it->position;
        it->model->RotateY = it->rotationY;
        it->model->Scale = it->scale;
        it->model->Render(camera, sun);
    }
    //render particles
    for(auto it = particleModels.begin(); it != particleModels.end(); it++)
        (*it)->Render(camera, sun);
}
```

### 3.3.3 Model rendering

A model is built from many meshes. A mesh represents a number of triangles that use the same texture (material) for rendering. For example the rendering of a tree which is model consist of rendering of its trunk and leafs which are separate meshes, using different textures. To render a model we need to send its model matrix, the view-projection matrix and light position to the shader which will render it on the right spot, with the right rotation and scale, and with the right illumination. But before we render anything via the *glDrawElements* call we need to tell the shader, at which offsets will it find the vertex attributes, and we also need to tell it which texture to use.

```
void Model::Render(const CameraSpectator &camera, const Sun *sun)
{
    program->Use();
    glUniform3fv(program->GetUniformLocation("lightPos"), 1, &sunPos[0]);
    glUniform3fv(program->GetUniformLocation("lightColor"), 1, &sun->GetSunColor()[0]);
    glm::mat4 m = computeModelMatrix();
    glm::mat4 mvp = camera.GetCameraMatrix() * m;
    glUniformMatrix4fv(program->GetUniformLocation("MVP"), 1, GL_FALSE, &mvp[0][0]);

    glBindBuffer(GL_ARRAY_BUFFER, vboId);
    glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, sizeof(Engine::Vertex),
        (void*)offsetof(Engine::Vertex, position));
    glVertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE, sizeof(Engine::Vertex),
        (void*)offsetof(Engine::Vertex, normal));
    glVertexAttribPointer(2, 2, GL_FLOAT, GL_FALSE, sizeof(Engine::Vertex),
        (void*)offsetof(Engine::Vertex, uv));

    glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, iboId);

    glActiveTexture(GL_TEXTURE0);
    for (auto it = meshes.begin(); it != meshes.end(); ++it)
    {
        glBindTexture(GL_TEXTURE_2D, materials[it->materialIndex]);
        glDrawElements(GL_TRIANGLES, it->nIndices, GL_UNSIGNED_INT,
            (void*)(sizeof(unsigned int) * it->baseIndex));
    }
    program->UnUse();
}
```

### 3.3.4 Sun, sun rays rendering

The rendering of the sun and sun rays involves multiple steps that include altering the graphics pipeline, and rendering to different color/depth textures.

```
void Sun::Render(const Camera &camera)
{
    //copy existing depth buffer to depth texture
    glBindTexture(GL_TEXTURE_2D, texIdDepth[0]);
```

```
glCopyTexSubImage2D(GL_TEXTURE_2D, 0, 0, 0, 0, 0,
    camera.GetScreenWidth(), camera.GetScreenHeight());
//draw sun
glBindFramebuffer(GL_FRAMEBUFFER, fboId);
glFramebufferTexture2D(GL_FRAMEBUFFER, GL_COLOR_ATTACHMENT0,
GL_TEXTURE_2D, texIdColor[0], 0);
glFramebufferTexture2D(GL_FRAMEBUFFER, GL_DEPTH_ATTACHMENT,
GL_TEXTURE_2D, texIdDepth[1], 0);
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
renderSun(camera);
//apply costum depth test
glFramebufferTexture2D(GL_FRAMEBUFFER, GL_COLOR_ATTACHMENT0,
GL_TEXTURE_2D, texIdColor[1], 0);
glFramebufferTexture2D(GL_FRAMEBUFFER, GL_DEPTH_ATTACHMENT,
GL_TEXTURE_2D, 0, 0);
glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, texIdColor[0]);
glActiveTexture(GL_TEXTURE1); glBindTexture(GL_TEXTURE_2D, texIdDepth[1]);
glActiveTexture(GL_TEXTURE2); glBindTexture(GL_TEXTURE_2D, texIdDepth[0]);
depthTestProgram->Use();
renderRect();
depthTestProgram->UnUse();
//apply radial blur and render to screen
glBindFramebuffer(GL_FRAMEBUFFER, 0);
glBlendFunc(GL_ONE, GL_ONE_MINUS_SRC_COLOR);
glEnable(GL_BLEND);
glActiveTexture(GL_TEXTURE0); glBindTexture(GL_TEXTURE_2D, texIdColor[1]);
radialBlurProgram->Use();
glUniform2f(radialBlurProgram->GetUniformLocation("sourcePos"),
    sunPosOnScreen.x, sunPosOnScreen.y);
renderRect();
radialBlurProgram->UnUse();
glDisable(GL_BLEND);
}
```

The custom depth test and radial blur are done on the GPU via shaders (written in OpenGL Shading Language).

```
//custom depth test fragment shader
#version 120
varying vec2 fragmentUV;
uniform sampler2D texture, textureDepth, sceneDepth;

void main()
{
    if(texture2D(sceneDepth, fragmentUV).r < texture2D(textureDepth, fragmentUV).r)
    {
        gl_FragColor = vec4(vec3(0.0), 1.0);
    }
    else
```

```
    {  
        gl_FragColor = vec4(texture2D(texture, fragmentUV).rgb, 1.0);  
    }  
}  
  
//radial blur fragment shader  
#version 120  
varying vec2 fragmentUV;  
uniform vec2 sourcePos;  
uniform sampler2D image;  
  
void main()  
{  
    vec3 radialBlur = vec3(0.0);  
    vec2 texCoord = fragmentUV;  
    int samples = 128;  
    vec2 radialBlurVector = (sourcePos - texCoord) / samples;  
    for(int i = 0; i < samples; i++)  
    {  
        radialBlur += texture2D(image, texCoord).rgb;  
        texCoord += radialBlurVector;  
    }  
    gl_FragColor = vec4(radialBlur / samples, 1.0);  
}
```

### 3.3.5 Final results (screen shots)



(a) The sun rises!



(b) Light the campfire, night is here!



(c) Selfie with my buddy Bob.

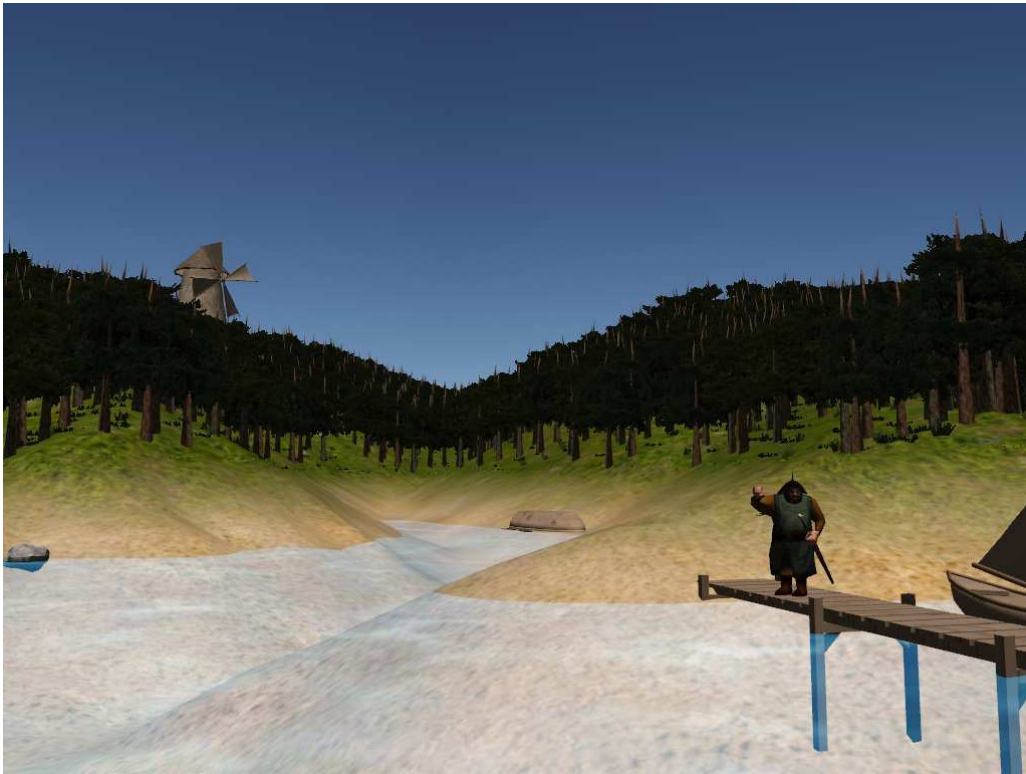


Figure 2: New land ahead. Land of opportunities.



Figure 3: From the eyes of a hawk.

## 4 Conclusions and future development

The engine loads and generates the scene in 9 s on average (fast loading), has dynamic sky, sun and sun rays, ocean, textured terrain, imported 3d models (some of them animated), particle system for mass rendering, ambient and diffuse lighting on the scene objects (sun and sky excluded), and the user can customize the scene (place, move, scale, rotate the 3D models, and save the changes) with a combination of keys and the mouse. From the second frame onwards (first frame containing some shader initializations), the engine runs at 55 FPS (frames per second) and uses 6% of the CPU on average. *Test were made on a Samsung Laptop having: Intel Core i3 (2.10 GHz), 4 GB RAM, nVidia GeForce GT 520MX (1 GB memory) and Windows 8.1 OS.*

All of the proposed secondary objects were achieved, and the two main objectives are nearly done. All in all, this graphic engine achieved its goal, producing high quality image and at relatively low CPU cost.

Future development of the engine would include shadow mapping and random terrain generating based on noise. Shadow mapping would require a second rendering of the scene from the sun's point of view, and some calculations on the depth buffer. Terrain generating would require implementing Perlin noise to generate a 2D gray-scale image that would apply for the height map. These features would require at most 3 weeks to be added to the engine.

## 5 References

### References

- [1] *Unity 3d* - <https://unity3d.com/>
- [2] *Unreal Engine* - <https://www.unrealengine.com/>
- [3] *Video Game Industry* - [http://en.wikipedia.org/wiki/Video\\_game\\_industry](http://en.wikipedia.org/wiki/Video_game_industry)
- [4] *Video Game Industry revenue estimation for 2015* - <http://www.statista.com/statistics/278181/video-games-revenue-worldwide-from-2012-to-2015-by-source/>
- [5] *OpenGL* - <http://en.wikipedia.org/wiki/OpenGL>
- [6] *GLEW from Wikipedia* - [http://en.wikipedia.org/wiki/OpenGL\\_Extension\\_Wrangler\\_Library](http://en.wikipedia.org/wiki/OpenGL_Extension_Wrangler_Library)
- [7] *SDL* - <https://www.libsdl.org/>
- [8] *Assimp* - [http://en.wikipedia.org/wiki/Open\\_Asset\\_Import\\_Library](http://en.wikipedia.org/wiki/Open_Asset_Import_Library)
- [9] *GLM* - <http://glm.g-truc.net/0.9.6/index.html>

Kerestely Arpad  
Universitatea Transilvania  
Strada Iuliu Maniu 50  
Brasov 500091  
Romania  
E-mail: [k\\_arpi2004@yahoo.co.uk](mailto:k_arpi2004@yahoo.co.uk)



# **Smart Hotel**

**Mitrica Alexandru, Sperila Alecsandru**  
**Teacher Coordinator: Sorin Ilie**

## **Abstract**

This paper presents a smart home web-mobile hybrid application designed for hotels. The goal of this paper is to present a method to control access, lights and power sockets from a mobile device. For this purpose all functionality runs on a server in the form of web services. Users with different levels of access are managed using a database. High level users like hotel employees, will be able to grant users access to a room for a given amount of time. Low level users like hotel guests and maintenance staff, are able to open the door using their mobile device, change lighting options, turn power sockets on and off. High level users will also be able to monitor room access, presence and power consumption of the room power sockets.

The mobile devices communicate with the server via internet connection. The server communicates via wired Ethernet with Arduino modules in hotel rooms. Each Arduino controls an electric door, lights, power sockets but also measures power consumption and determines human presence in the room using a motion sensor. To accomplish this, four distributed software applications will interact: the Arduino Ethernet server, the web server, the database server and the android mobile application client.

## **1 Introduction**

Hotels and accommodation-based businesses are increasingly relying on technology for their day to day operations. Computing devices are now cheap and small enough to allow rooms to have a local embedded controller reporting to/ commanded by a server.

Furthermore, the personal smart mobile device boom offers new opportunities for improvement. Presuming that every individual has a personal device we can create a virtual alter-ego of that person. Their device can communicate with the server and thusly be given access to the room controllers.

This work envisions a new way of interacting with hotel rooms by simply interacting with a mobile app that can control door locks, lighting, plugs etc.. Access can be given to guest-users

and maintenance staff to a room for a given amount of time, even at predefined intervals. At all times the server can log all door accesses, presence, power consumption, and so on for inspection. Similar approaches have been made by a multitude of companies, such as Philips, Belkin etc.:

Philips's Hue Connected Bulb[1] uses a central module that plugs directly into a wireless router and translates signals between a Wi-Fi-connected smartphone and the bulbs. The bulbs cost from \$79 to \$89 a piece and a kit containing one hub and three bulbs costs \$199.

The Belkin WeMo Switch + Motion[2] is a relatively low price device which has a plug-and-play configuration that will appeal to many budding home automations. Belkin's WeMo Switch+ controls your plugs via wifi.

SmartThings[3] is another approach to a smart environment, they use a hub connected to a router which can control multiple modules. The modules connect to the hub via wi-fi and are controlled using an Android or iOS device.

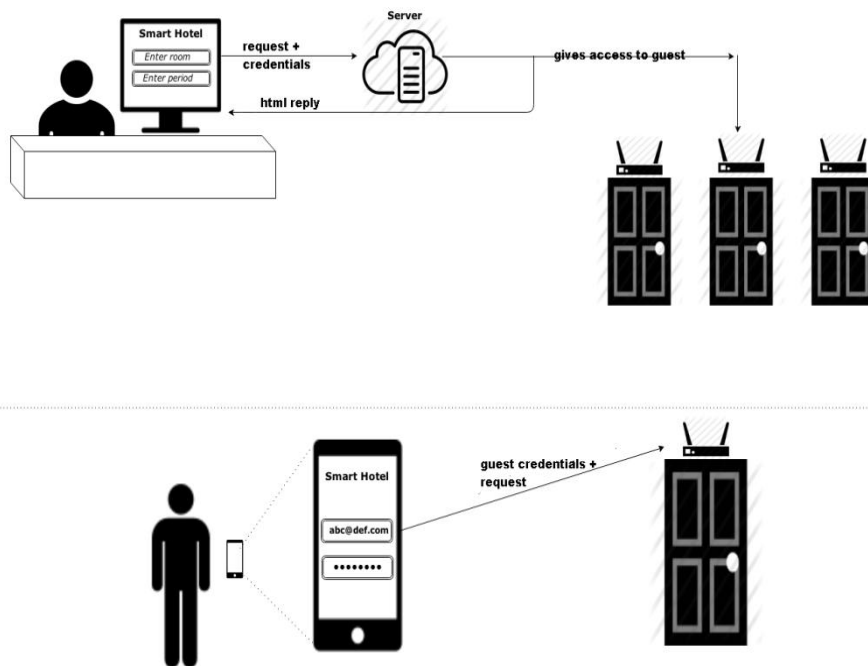
Smart Room Control Solution[4] from Distech Controls is a different approach to the smart room problem. They use different modules which necessitate a complicated installation compared to other solutions. The wiring is complicated and modules have to be carefully mounted on the fuse box.

Control4[5] from Exzel Smart Home, is an award-winning product that can help homeowners manage their home in a smart and integrated way. From turning music on, choosing movies, to controlling the lighting, setting room temperature, controlling sprinklers, monitoring the front door camera to see if anybody is there, setting and monitoring fire alarms and security alarm systems. All of this can be done by one single user interface; be it a touch screen panel, iPhone, iPad, iPod, or universal remote control.

## **2 Design and implementation**

The high end user functionality must be available as a website for full compatibility with any device (see the high level functionality image in Figure 1). This allows the hotel staff to grant access and control of a room to a guest and manage the room statistics and logs. Access can only be granted to registered users. Registration is free and open to anyone who downloads the mobile application. When granted access, a guest can use any and all room functionality until the end of his/her stay directly from the mobile device.





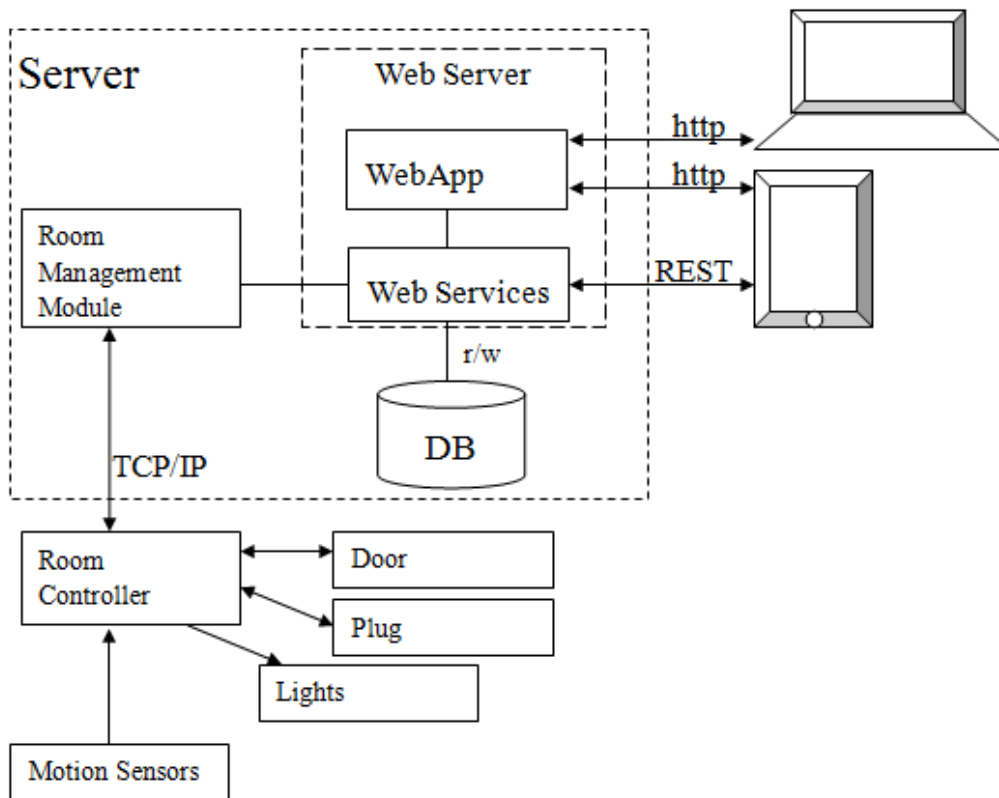
*Fig. 1. Giving control of a room to a registered user*

## 2.1 Architecture

The idea we propose is composed of a central server, one embedded controller per hotel room, one mobile device for each staff member or guest and interconnecting Ethernet or WiFi networks.

The embedded hotel room controller implements all interactions with the physical elements of the room: power plugs, lighting, door lock, motion sensor (see figure 2 for the block diagram). The hotel server communicates via Ethernet with all controllers using basic socket communication, but also waits for requests from mobile devices and reports from room controllers. This is done via the Room Management Module. All functionality of this module as well as administrative functionality and user profile management is offered to other applications using REST web services [16].

Mobile devices offer a user-friendly graphical user interface (GUI). Depending on user interactions, functionality from the server is invoked via WiFi using web services.



*Fig. 2. System architecture*

We will now detail each individual hardware and software module in the project in a top-down fashion starting with the human-machine interaction.

### **2.1.1 Android mobile device**

The personal mobile device is used to identify the user and exercise control over the assigned room in a meaningful way to the guest or staff member. We chose to implement this initial version of the project using Android mobile devices [11] and the Java programming language [14]. The application connects to the internet and invokes functionalities from the server using web Services [16].

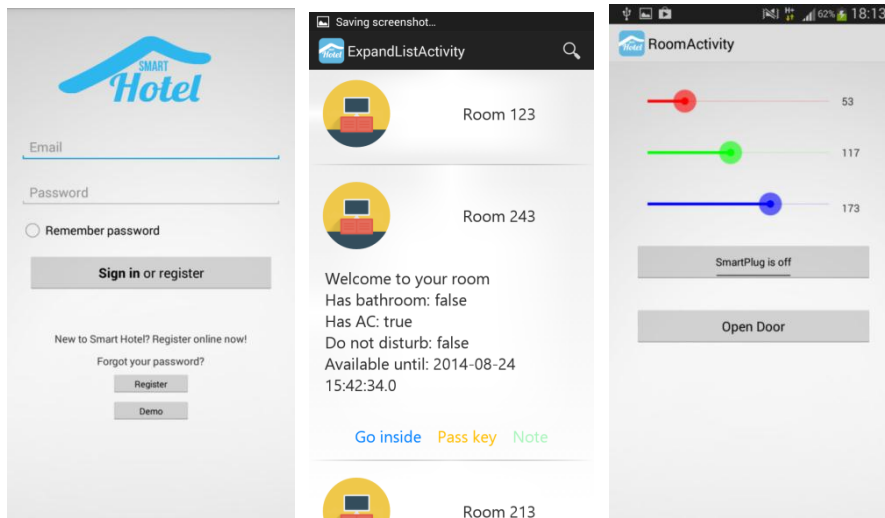


Fig. 3. The graphical user interface of the android application

When opening the app, users are greeted with a basic login screen in which they can type their credentials (see figure 3). If a guest does not possess such credentials, he/she must use the Register button down below. This will register a new account into the Smart Hotel Database, and will store the credentials into the phone's account manager. Logging in will also store the credentials into the account manager. The Demo button skips the login and goes straight to the interface, with some dummy credentials, for demo purposes.

The graphical interface of the room contains RGB Sliders for light color control, smart plug On/Off button for a single appliance, Open door button to open the door of the room assigned to the guest account.

One of the issues we had to tackle was the security of login over a WiFi connection. We approached this challenge by implementing an access token with an expiration date. The token  $t$  obfuscates the actual text password  $p$  using the SHA1 algorithm[12], and incorporates the time of the access request  $t_0$  using an internet time service. In the following equation an access token  $t$  is generated from  $p$  and  $t_0$ .

$$t = SHA1 (SHA1(p), t_0) \quad (1)$$

Access token  $t$  is only valid  $x$  seconds after creation after which the server will not recognize it any longer. The server can determine if the token is expired because  $t_0$  is attached to the command along with the token, username and command. The server can then use the hash of the user's password from the database and the time to generate a token This type of token eliminates sniffing attacks.

### 2.1.2 Server

This software module runs on a physical server and is the central entity of the system. The software baseline on which we built this module is: windows 8 operating system [7] , a Tomcat [10] webserver and a MySQL database server [6] all running on the same machine.

The rooms and users are entries in the database, and the Rooms and Rent servlets are used to create the web interface needed to assign control of a room to a guest. DbManager class controls and populates the database access object classes RoomsDAO and UserDAO. UserLogin is the login page for the web application. It uses class PassCheck to verify the validity of a password. Services Login and Register, implemented as servlets, are necessary for the android application to invoke functionality.

The HotelDriver class implements the protocol used to interact with the room controller (Intel galileo module[8]). Every method of the HotelDriver is offered as a service to the mobile apps after adding a user authentication layer.

```
Service url: /openDoor
input: String user, String passwordToken
output: Boolean acknowledgement
Service url: /doorStatus
input: String user, String passwordToken
Service url: /setLight
input: String user, String passwordToken, int red(0..255),
int green(0..255),
int blue(0..255)
output: int RGBConfirmation
Service url: /smartplugOn
input: String user, String passwordToken
Service url: /smartplugOff
input: String user, String passwordToken
output: Boolean isOn
Service url: /smartplugStatus
input: String user, String passwordToken
output: Boolean isOn
Service url: /smartplugMonitor
input: String user, String passwordToken
output: float instantaneousPower
```

We also added a service for the motion sensor allowing the room controller to provide notifications when movement is detected in the room.

```
Service url: /alarm
Input: null
Output: Boolean acknowledgement
```

### **2.1.3 Room Cotroller**

The room controller is implemented on an Intel Galileo (Arduino-compatible) module [8]. The software baseline on which we built this module is Yocto linux[9]. We built an ethernet server that communicates with the server using an Ethernet. Each room must be assigned a static IP.

The ports on the Galileo board are used as follows:

**Ports 6,7,8** digital ports used to open door

**Port 2** digital port for movement sensor

**Port 13** digital output for plug on/off control

**Ports 9,10,11** analog output ports for rgb lighting

**Port A0** analog input port for current intensity

**Port 12** digital input for door open sensor

External commands for the Room Controller received over a Socket and the functionality they provide:

**Sending 1-** Opens door

**Sending 2-** Asks Door Status

**Sending 3-** Sends Light Control (expects 3 integer values for red green and blue)

**Sending 4-** Turns on the Smart Plug

**Sending 5-** Turns of the Smart Plug

**Sending 6-** Checks Smart Plug Status

**Sending 7-** Reads Current Sensor

We can now outline a schematic of the interaction protocol between the mobile device, the server and the room controller (see figure 4 for the sequence diagram[13]). We identify three types of interactions: i) mobile->server interactions, ii) mobile->room controller interactions via server and iii) room controller -> server interactions.

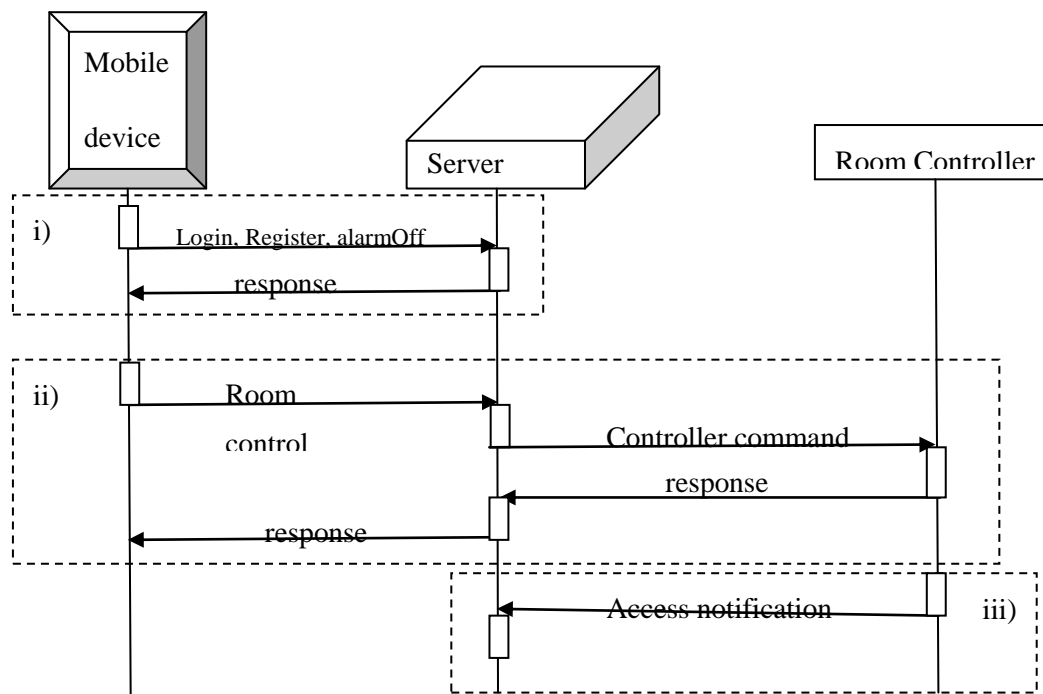


Fig. 4. Interactions Schematic

## 2.2 Hardware room controls

In this section we present the individual hardware designed and implemented in order to provide the required functionality.

### 2.2.4 Door

Pressing the "Open Door" button in the mobile app invokes to the central server via wifi which in turn sends an Ethernet packet to the room controller that opens the door.

We built an automatic door that also has a normal metal key in case of blackouts (see figure 6). We used a relay to operate the door. We also connected a 5V output port on the door lock and an input port on the door frame so we can detect when the door is shut.

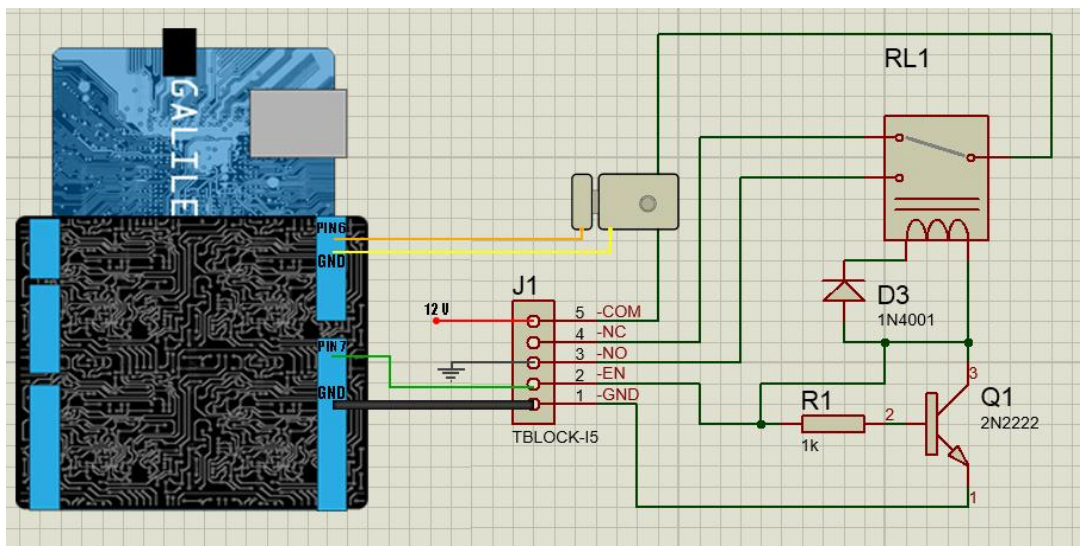


Fig. 5 Schematic of the relay

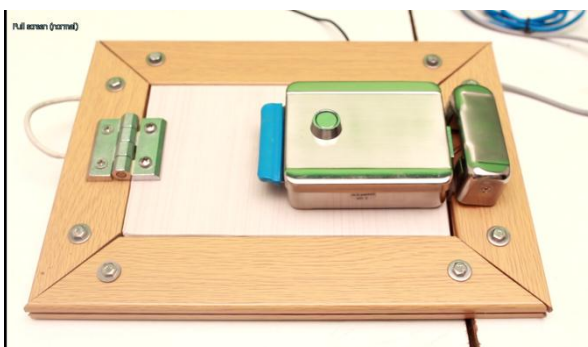


Fig.6 Demo door

In the figure above: COM goes to one of the leads from the lock, NO/NC stands for normally open/closed and requires a 12V DC 2A source of power, EN is connected to a port on the Galileo board and GND is connected to a ground port on the Galileo.

### 2.2.5 RGB Lighting

RGB LEDs can emit any color of light the users desire depending on three analog 0-5V command circuits. This allows the user to set the mood in the hotel room as he desires. The user can choose any color using the android app where he sets 3 different values for the Red, Green and Blue hues. The preferred colors can also be saved on the Server. This command can be given via internet from virtually anywhere. The RGB LED strip, we used requires a 12V DC 2A power source so we had to build a command circuit for it.

The RGB LEDs are controlled by the 3 analog pins from the Galileo board (0-5V) using 3 transistors as shown in figure 7. The analog output pins 9, 10, 11 are connected to the Pulse-Width Modulation for Red(PWMR), Pulse-Width Modulation for Green(PWMG) and Pulse-Width Modulation for Blue(PWMB) ports of the circuit. The optocoupler is necessary because it protects the Galileo 5v command circuit from 12V surges from the RGB LED strip.

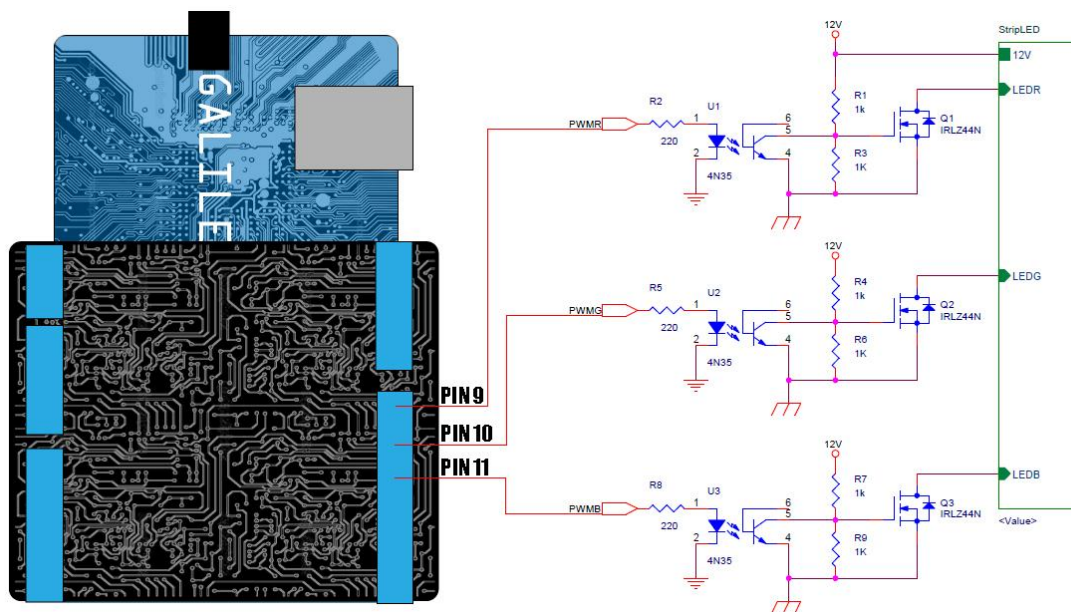


Fig. 7 Command circuit for RGB LED strip

### 2.2.6 Plug

The smart plug functionality allows the user to turn a device connected to it on or off by pressing the SmartPlug button on the android app (see figure 3). The app will send a message to the central server via WiFi which will send a signal to the Galileo and activate or deactivate the relay (see figure 8). By using a current sensor the SmartPlug can read how much current is being used. We use the ACS712 5A current sensor which uses the Hall-Effect to measure how much current passes through it. In order to control the power plug we use a relay much like the ones used for the door. Both the relay and the sensor are connected in series with the power plug.



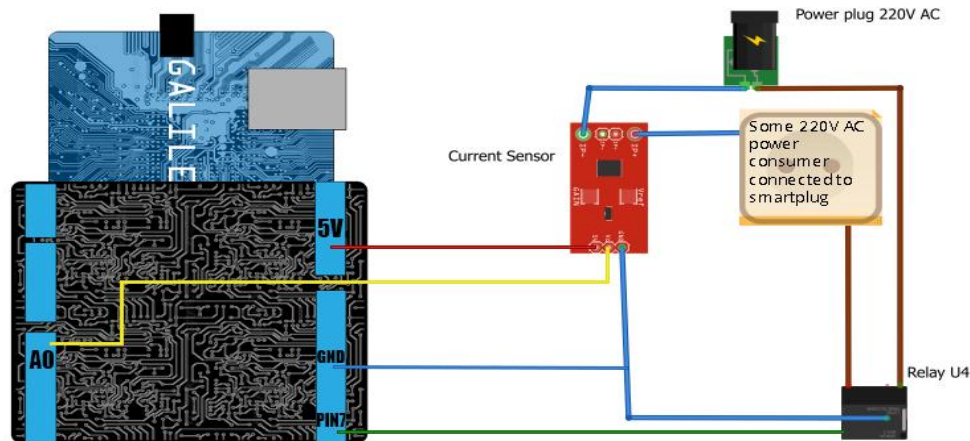


Fig. 8 Smart plug schematic

### 2.2.7 Motion Sensor

Movement detection sets off a notification, it is up to the user to decide if it is an emergency. This functionality is provided by the server which decides if the PIR going off is a bad thing and the staff needs to be notified. The room controller will read input from the PIR and forward it to the server via the alarm service when presence is detected regardless of server settings so that the server can log it. The PIR sensor needs 12V and 1A to be powered on and a 3.5 V 1A for the inbuilt relay that sends the information to the server.

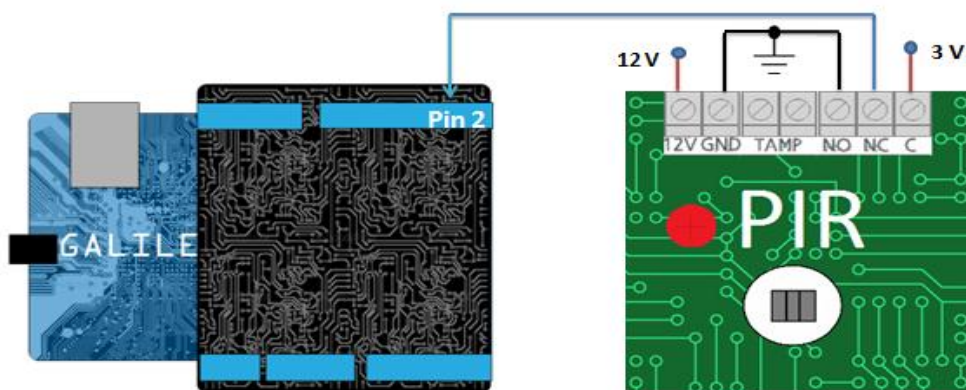


Fig. 9 The PIR sensor



## References

- [1] Philips's Hue reference, accessed February 2015  
<http://www.usa.philips.com/e/hue/hue.html>
- [2] The Belkin WeMo Switch + Motion reference , accessed February 2015 <http://www.belkin.com/us/p/P-F5Z0340/>
- [3] SmartThings reference, February 2015  
<http://www.smartthings.com/>
- [4] Smart Room Control Solution by Distech Controls reference, accessed February 2015  
<http://www.distech-controls.com/en/csa/products/smart-room-control-solution/>
- [5] Control4 from Exzel Smart Home reference, accessed February 2015  
<http://www.exzelsmarthome.com/products/homecontrol/index.html>
- [6] MySql reference manual <http://dev.mysql.com/doc/refman/5.7/en/> accessed June 2014
- [7] Windows 8 User manual , accessed June 2014, <http://www.microsoft.com/en-us/download/details.aspx?id=39055>
- [8] Intel Galileo online manual, accessed June 2014, <http://www.intel.com/content/www/us/en/do-it-yourself/galileo-maker-quark-board.html>
- [9] Yocto Linux documentation, accessed June 2014 <https://www.yoctoproject.org/documentation>
- [10] Apache Tomcat Server documentation <http://tomcat.apache.org/tomcat-7.0-doc/>
- [11] Android Developer Manual <http://developer.android.com/> , accessed June 2014
- [12] Schneier on Security: Cryptanalysis of SHA-1, accessrd June 2014  
[https://www.schneier.com/blog/archives/2005/02/cryptanalysis\\_o.html](https://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html)
- [13] Unified Modeling Language User Guide, The (2 ed.). Addison-Wesley. 2005. p. 496. ISBN 0321267974.
- [14] Ian F. Darwin : Java Cookbook, 3rd Edition, ISBN 978-1-4493-3704-9, publisher O'Reilly Media, 2014
- [15] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. 2001.Introduction to Algorithms (2nd ed.). McGraw-Hill Higher Education.
- [16] Leonard Richardson, Sam Ruby: RESTful Web Services Web services for the real world, ISBN: 978-0-596-52926-0, publisher O'Reilly Media, 2007

Mitrica Alexandru  
University of Craiova  
Department of computers and  
information technology  
Blvd. Decebal nr. 107, Craiova  
Romania  
E-mail: alexmitrica@yahoo.com

Sperila Alecsandru  
University of Craiova  
Department of computers and  
information technology  
Blvd. Decebal nr. 107, Craiova  
Romania  
E-mail:sperila.alecsandru@gmail.com

Sorin Ilie  
University of Craiova  
Department of computers and  
information technology  
Blvd. Decebal nr. 107, Craiova  
Romania  
E-mail:ilievioresorin@gmail.com

# Playing Hex with Monte Carlo simulation

Nikola Radovic, Ljubomir Raicevic  
Teacher Coordinator: Milan Tuba

## Abstract

The goal of this paper is to show that Monte Carlo simulation can be used to play zero-sum board games, with application to Hex. Additionally, we wish to measure two properties of this approach: what is the minimum number of Monte Carlo repetitions after which benefits are negligible, and second, does making a game tree with Monte Carlo simulation as evaluation function perform better than the naive approach, and if it does, by how much.

## 1 Introduction

In section 1.1 we discuss the motivation for using Monte Carlo simulation in the game of Hex. In section 1.2 rules of the game and some history are presented. In section 1.3 we give an overview of Monte Carlo in general. Section 2 explains some of the choices we made during implementation and experimental phases. Section 3 and the sub-sections present the quality of the Monte Carlo Hex player, reasonable values for Monte Carlo repetitions and benefits of using a game tree, through experimental data. Section 4 discusses limitations of this approach and future work.

### 1.1 Motivation

Monte Carlo has been used with great success in many applications. The main advantage of using Monte Carlo in this setting is that besides the basic dynamics of the game (i.e. players take turns, only one tile can be marked per move etc) basically no domain knowledge about the game is necessary, and therefore it could easily be implemented for other similar games, or any decision making process which can be modeled in a similar fashion.

### 1.2 About Hex Board Game

Hex is a zero-sum board game, in which two players take turns in making moves. It was invented in 1942 by Piet Hein and it was independently reinvented and studied in 1947 by John Nash. [2] Board on which the game is played is rhomboid in shape and is composed of hexagonal tiles with the same number of rows and columns. Traditionally, Hex is played on boards of size 11x11, 13x13 or larger.

At each turn, player marks a single empty tile by placing a mark, or more traditionally a stone, making the tile occupied. After a tile gets occupied, the mark cannot be removed and no additional marks can be put on the tile. This limits the maximum number of moves in a game to

the number of tiles on the board. First player's goal is to connect the two vertical edges of the board by a path of tiles marked with his marks. The second player has the same goal with the two horizontal edges. When one of the players achieves this, the game ends.'

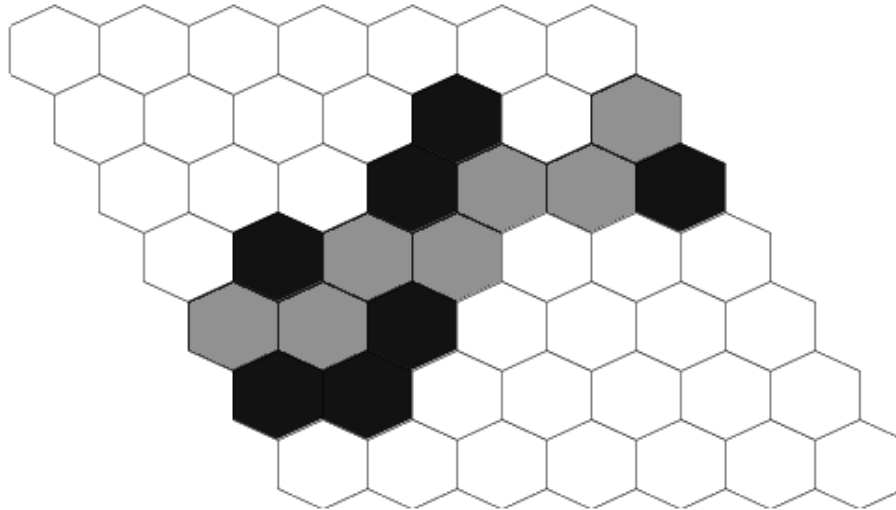


Figure 1. Game won by the second player.

Since the first player has great advantage to the second player [2], pie rule is used. According to the pie rule, the second player can choose to switch positions after the first move has been played. This removes the first player's advantage.

One of the properties of the game of Hex is that there are no ties [1]. The game can end before the maximum number of moves has been played if one of the players connects their respective board edges by a path of marks. The alternative is to play the maximum number of moves, in which case a single winning path - from the top to the bottom, or from the left to the right - is formed. More generally, any random permutation with appropriate proportion of marks for the first and the second player that covers the entire board will result in a winning path. This property will be important in the implementation of Monte Carlo simulation Hex player, since it greatly simplifies the process of evaluation of potential moves.

### 1.3 Monte Carlo Simulation Method

Monte Carlo simulation method is model evaluation (simulation) method based on random sets of inputs. Two main aspects of usage are: simulation of models dependent on random factors and simulation of complex mathematical problems not dependent on random factors, but that could be simplified via probabilistic estimation. One of most famous usage example is estimating area of figure A within unit square, by randomly selecting N points, as ratio number of points within A to N.

Idea of using randomness in deterministic models first appeared in 18th century, as part of "Buffon's needle" method for evaluating PI number. [3] Officially, Monte Carlo Simulation method is developed in 1940s by John von Neumann and Stanislaw Ulam in Manhattan project. Appearance of computers led to large usage in variety of fields such as risk evaluation, weathercast prediction, random walk prediction, fluid dynamics and thermodynamics [5].

Monte Carlo simulation has three steps: create random input sets, evaluate model for each set and compute statistic of interest. Providing uniform distribution of input sets is essential part of simulation and often most challenging for implementation. Main advantages of this approach are

its simplicity and speed as creating large number of pseudo random input sets is easily done by computers, and evaluating function is trivial in most cases.

In two player deterministic games Monte Carlo Simulation can be as follows: find all possible moves for player, evaluate all of them using Monte Carlo simulation method and choose best evaluated. This approach can be extended with optimal move “search tree” where Monte Carlo simulation is used for evaluation of leaves. This version of algorithm is called Monte Carlo Search algorithm and is successfully used in games such as GO, Backgammon and Scrabble. [4]

## 2 Implementation Choices and Simplifications

The main simplification that was made was not implementing the pie rule. Rationale behind this decision was that the pie rule only applies to the first move, and this decision - to switch or not - is a completely different problem from the rest of the game, and the two should be analyzed separately. Pie rule could be implemented easily, but it would add complexity that is not necessary for the purposes of this paper.

Naive Monte Carlo approach for deciding the best move for the player is given by the following simple algorithm:

```
procedure MonteCarloNaive
  For each empty tile on the board do
    Place a temporary mark on the tile
    For i := 1 to N do
      Fill the rest of the board randomly
      If current player won then
        wins = wins + 1;
    Remove the random marks
    Remove the temporary mark from the tile
  end
  return tile with the highest number of wins
```

This algorithm uses the fact that a completely filled board has exactly one winner. [1] It checks each of the next possible moves by generating N random outcomes of the game for each of them and chooses the one with the highest number of winning positions for the given player. Although these outcomes are random, and unlike games with real players, are not produced by intelligent decision making, they are a sufficient approximation of player’s chances of winning on a given board.

Finally, all experimental data presented in this paper was gathered on 7x7 boards. This was done for simplicity and easy comparison, as well as shorter execution times, since execution time is superlinear with regard to board size.

## 3 Experimental results

### 3.1 Quality of Monte Carlo Simulation Hex Player

The quality of the player was tested in two ways: by playing games against human players, and by playing games with other Monte Carlo players, with varying number of repetitions for the opponents.

When playing against human players, if the number of repetitions is set to 2500 or more, Monte Carlo player always wins. It is worth mentioning that none of the human players are expert Hex players, but this is definitely indicative of the quality of the Monte Carlo player.

Games involving two Monte Carlo players also show expected behaviour. Since pie rule was not implemented, first player has the advantage. Figure 2 shows the results of games played between different Monte Carlo players.

first player	second player	# of games	1st player won	ratio
500	500	200	152	0.76
1000	1000	200	165	0.825
3000	3000	200	185	0.925
5000	5000	200	194	0.97
7000	7000	200	195	0.975
10000	10000	200	199	0.995
13000	13000	200	199	0.995

Figure 2. Results of games between two Monte Carlo players

It is apparent that as the number of Monte Carlo repetitions increases, so does the ratio of first player's wins to the total number of games played.

Also, inspecting games move by move shows reasonable choices from both Monte Carlo players.

### 3.2 Diminishing Returns in Monte Carlo Simulation Repetitions

Naive Monte Carlo search algorithm described in Section 2 will perform better when  $N$ , the number of repetitions, is higher. This is due to the fact that adding more repetitions is equivalent to sampling more future moves of both players. If every possible permutation could be explored, the algorithm would be completely deterministic. Of course, this is prohibitively expensive in terms of CPU time, and therefore the number of possible outcomes per move that gets sampled by the algorithm is much smaller than the number of all possible outcomes. However, the previous section shows that even this very limited sampling yields good results, and the following question can be posed: how many Monte Carlo repetitions are necessary so that increasing their number produces negligible benefits in terms of decision quality?

Previous section shows that increasing the number of repetitions after 10000 does not improve player's performance. Also choosing a number of repetitions below 500 leads to poor quality of moves chosen by Monte Carlo.

When determining optimal number of Monte Carlo repetitions, the following procedure was used. Number of Monte Carlo simulation repetitions started from 500 and at each step was increased by 5 percent. There were 62 such steps, and the greatest number of repetitions observed was 9813. In each step one hundred randomly generated boards were evaluated. These boards are chosen randomly for given number of moves played i.e. how many first and second player's marks are there on the board. Same boards are evaluated at each step. Boards are evaluated as probability that player that made last move on board wins game.

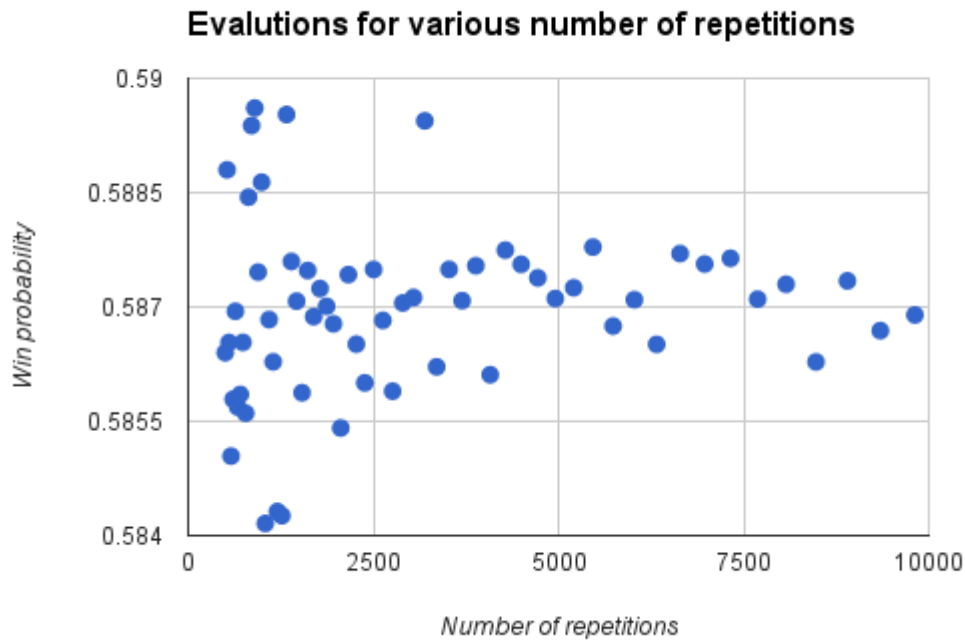


Figure 3. Single board evaluated by different Monte Carlo players

Figure 3. shows chart for a typical board. Even with number of repetitions below 1000 all probabilities are within 0.5% range. As number of repetitions grows variance of results decreases. For number of repetitions 3000 and greater probabilities are within range of 0.02%.

# of Monte Carlo repetitions				1000			3000			5000		
# of move	goal prob	min prob	max prob	stdev	mean	ratio	stdev	mean	ratio	stdev	mean	ratio
3	0.56902	0.56624	0.57270	0.00116	0.56988	0.15150	0.00077	0.56942	0.07051	0.00061	0.56892	0.01740
3	0.60952	0.60531	0.61216	0.00105	0.60774	0.29325	0.00078	0.60916	0.05991	0.00072	0.60928	0.04054
5	0.56644	0.56357	0.56864	0.00083	0.56632	0.02110	0.00059	0.56640	0.00742	0.00047	0.56589	0.09761
5	0.56086	0.55797	0.56455	0.00110	0.56045	0.07284	0.00090	0.56212	0.22362	0.00082	0.56013	0.12981
13	0.47075	0.46773	0.47198	0.00089	0.47065	0.02085	0.00083	0.46983	0.19503	0.00049	0.47058	0.03586
13	0.50621	0.50377	0.51005	0.00107	0.50656	0.07046	0.00081	0.50666	0.09014	0.00044	0.50687	0.13196
17	0.51385	0.50940	0.51741	0.00087	0.51378	0.01288	0.00070	0.51405	0.03982	0.00072	0.51350	0.06722
23	0.40901	0.40467	0.41204	0.00083	0.40814	0.21161	0.00057	0.40896	0.01281	0.00057	0.40882	0.04499
29	0.69701	0.69438	0.70005	0.00105	0.69745	0.06284	0.00077	0.69739	0.05508	0.00045	0.69679	0.03118
29	0.37759	0.37337	0.38058	0.00076	0.37568	0.50538	0.00063	0.37737	0.05775	0.00033	0.37692	0.17744

Figure 4. Statistical parameters for different Monte Carlo players

Cumulative statistical parameters for 10 boards are given in Figure 4. First column shows number of moves played on board, second column is average probability of three Monte Carlo players with greatest number of repetitions, i.e. probability that will be used for comparison. Two succeeding columns show minimum and maximum probability on the whole interval. Difference between these two probabilities is not greater than 1% for any board.

Following three groups show statistical parameters for three interesting cases, where number of repetitions is 1000, 3000 and 5000. First column shows standard deviation for all repetitions greater than observed one. Second column gives average probability around observed point, and third one shows ratio of difference of this average and goal probability in percents. Ratio average value for 1000 repetitions is 0.142%, for 3000 0.081% and for 5000 0.077%.

Variance decreases as number of repetitions grows in the following manner: average standard deviation for 1000 repetitions is 0.00096, for 3000 0.00073 and for 5000 0.00056.

first player	second player	# of games	1st player won	ratio
1000	3000	100	85	0.85
1000	5000	100	81	0.81
3000	5000	100	94	0.94

Figure 5. Results of games between two different Monte Carlo players

Figure 5 shows that number of repetitions close to 3000 is where improvements in quality become negligent. This conclusion is reached by analyzing data in Figure 4, and is confirmed by data in Figure 5, which show that Monte Carlo player with 3000 repetitions will win even against a stronger player in 94% of the cases.

### 3.3 Benefits of Using a Decision Tree Alongside Monte Carlo Simulation

Although the Naive Monte Carlo algorithm is a viable method for playing Hex, it is reasonable to suppose that better results could be obtained by using a game tree, with Naive Monte Carlo as evaluation function. Since Monte Carlo simulation is expensive in terms of CPU, nothing close to complete game tree is feasible for boards of interesting sizes. Therefore, the following procedure for pruning the tree is used: do a Naive Monte Carlo evaluation, sort the moves by probability of win, take some small number of top moves, and do branching only on those; next, use Naive Monte Carlo in the same way to obtain the opponents best move, and continue branching. Repeating this process for a chosen number of plies will result in a tree of significantly reduced size compared to the complete game tree and it will not waste resources on evaluating low quality moves. This is a variation of the technique known as Monte Carlo Tree Search. [2]

Best moves	Plies	# of repetitions	% of different choices
2	2	400	5.5
2	3	400	1.75
3	2	400	2.5
3	3	400	0.5

Figure 6. Percentage of different moves chosen by Naive and Monte Carlo with game tree

We chose three characteristic boards from the start of the game and compared results of Naive algorithm and Monte Carlo Tree Search. Figure 6 shows that Monte Carlo Tree Search plays the

same moves as Naive algorithm in vast majority of cases. Additionally, 20 full games were played between Monte Carlo Tree Search as the first player against Naive Monte Carlo as the second player. Monte Carlo Tree Search lost every one of them, in spite of the first player advantage, which shows that method described here does not work as intended.

Considering the exponential growth in running time, it is evident that this variant of Monte Carlo Tree Search is too expensive and brings very small potential gain, and in cases presented here produces even worse results than the Naive algorithm.

## 4 Conclusion

Using Monte Carlo simulation, with no domain knowledge, we implemented a Hex player of good quality. Naive algorithm described in this paper produces very good results when considering running time. The same algorithm with a small game tree gives the same or worse results, with exponentially growing running time with the number of plies explored. This result is unexpected, and most likely due to aggressive pruning of the game tree. The exact causes of this result require further examination.

## References

- [1] D. Gale, The game of Hex and the Brouwer fixed-point theorem, *Amer. Math. Monthly* 86, 818–827, 1979.
- [2] Arneson, B.; Hayward, R. B.; and Henderson, P. 2010. Monte Carlo tree search in hex. *Computational Intelligence and AI in Games, IEEE Transactions on* 2(4):251–258.
- [3] Harrison RL. Introduction To Monte Carlo Simulation. AIP conference proceedings. 2010;1204:17-21. doi:10.1063/1.3295638.
- [4] David Silver , Gerald Tesauro, Monte-Carlo Simulation Balancing, Proceedings of the 26th International Conference on Machine Learning (ICML-09) - 2009
- [5] Sun-Chong Wang, Interdisciplinary Computing in Java Programming, The Springer International Series in Engineering and Computer Science Volume 743, 2003, pp 117-131

Radović Nikola  
Megatrend University  
Faculty Of Computer Science  
Bulevar Umetnosti 29, Beograd  
SERBIA  
E-mail:  
nikolaradovic85@gmail.com

Raičević Ljubomir  
Megatrend University  
Faculty Of Computer Science  
Bulevar Umetnosti 29, Beograd  
SERBIA  
E-mail:  
rljubo90@gmail.com



## Web design techniques with applications

Petru Rogojanu

Tutors Coordinators: Doina Solomon, Marius Amza

### Abstract:

This project is a presentation of the web pages of two educational institutions in Sibiu Municipality, respectively *Palatul Copiilor Sibiu* <http://palatulcopiilorsibiu.ro/> and *Samuel von Brukenthal High School* <http://colegiulbrukenthal.ro/>, which I created by using the specific *web design* and *programming* instruments. The websites were designed in Wordpress by using: HTML, CSS, JavaScript, Photoshop, PHP. I used a special application for displaying the students' schedule. This application was created by using the data bases and can be used by any person that has knowledge of *web design* or *programming*.

*Key words: web design, programming, Palatul Copiilor Sibiu, Samuel von Brukenthal High School*

## 1 Introduction

Some data about the project author, Petru Rogojanu: I am a student in the 7<sup>th</sup> grade at the „Samuel von Brukenthal” High School in Sibiu. I began my activity in the computer field by the informatics circle at Palatul Copiilor Sibiu, as early as the 3<sup>rd</sup> grade, assisted by the tutor Doina Solomon. I won the 1<sup>st</sup> place at the regional informatics contest „Top Info”. I started to attend web design courses since the V<sup>th</sup> grade in a private environment and I had Marius Amza as tutor. The first website designed by me was that of Palatul Copiilor Sibiu, by which I obtained a web designer licence from „Haute école Arc”, Neuchâtel, Switzerland, in June 2013. I intended this website to be not just an abstract exercise but a useful, functional one. Afterwards I attended the web programmer course, I designed the website of „Samuel von Brukenthal” High School and I was awarded the second licence of web developer from „Haute école Arc”, Neuchâtel, Switzerland, in June 2014.

I am currently attending web expert courses and I intend to improve my skills in the web design field, a field that is rapidly evolving.

The project presented is aimed at designing the web sites of two educational institutions, which are of interest for the local community and allow for a better information of students and parents regarding the activities performed by these.

The first website <http://palatulcopiilorsibiu.ro/> is designed for Palatul Copiilor in Sibiu. Palatul Copiilor Sibiu is the only institution with extra-school profile in Sibiu and the

provider of informal education that gathers children from all the schools in the city but also from the neighboring localities. This year Palatul Copiilor in Sibiu completes 62 years of activity and currently a number of 2509 students are registered for activities. The author of the website is a student of Palatul Copiilor and the website was designed by the author in the V<sup>th</sup> grade when it became the official webpage of the educational institution.

The intended field is that of complementary education, performed outside the usual school framework, where children come due to passion for certain fields.

The second website <http://colegiulbrukenthal.ro/> is for Samuel von Brukenthal High School in Sibiu, an educational unit with high school and gymnasium grades, with German language teaching. It is the oldest German school in Romania and it has a total of approximately **850** students. The website was designed only in German because at the time when it was designed there was no German version of the High School web page. The author is a student of this High School and the website was designed in the 6<sup>th</sup> grade. This article is an extended and completed version, of the PCID article. [1]

## 2 Project goals

The purpose of the project is to comparatively present the two websites: Palatul Copiilor Sibiu <http://palatulcopiilorsibiu.ro/> and *Samuel von Brukenthal High School* <http://colegiulbrukenthal.ro/>:

- the web design techniques necessary for designing these websites
- design and functionality

The web page of Palatul Copiilor Sibiu must promote the circles activities, the courses that can be attended, the staff, the schedule, the contests that will be organized and details about these, as well as the results obtained by the students from PCSb. The second website <http://colegiulbrukenthal.ro/> was designed for Samuel von Brukenthal High School in Sibiu, an educational unit with high school and gymnasium grades with German teaching.

## 3 How the website was designed

The first website created was for Palatul Copiilor Sibiu. I started from a wordpress theme and afterwards several changes were performed because a large quantity of information needed to be systematized, to offer legibility, an adequate design but also a certain level of security, in other words functionality and a pleasant aspect. [2]

I started by collecting the information to be posted on the website and then I systematized them, respectively I created eight pages as follows:

- **Home** page (Fig.1) that presents a slider with 75 photos with aspects from the activities of all the circles existing within the institution. Below these the Vision and Mission of Palatul Copiilor Sibiu are presented, followed by a welcome note. On the right side novelties are presented (contest posters, shows, magazine, etc.), partners and affiliated Children Clubs, by links;



Fig. 1 Palatul Copiilor Home page



Fig. 2 Home page/slider

- **Circles** page displays both links towards each circle presented by a suggestive drawing and also a menu in cascade that leads directly to the intended circle. The final destination of these links is a presentation page of each circle, where there are images of the specific activity, the name of the tutor that runs the circle, the weekly schedule and some words describing the respective circle activity;



Fig 3 Circles page

- **Projects** page presents the underway projects, partners, objectives, activities and project products;



Fig. 4 Projects page

- **Contests** page presents the contests organized by each circle with details by accessing the related links;
- **Tutors** page presents all the tutors that run the circle within Palatul Copiilor, with a short personal presentation, together with the photo that represents a link towards the circle taught by the respective tutor;



Fig 5 Tutors page

- **Schedule** page displays the weekly schedule by circles for the underway school year

CERCUL	LUNI	MARTI	MIERCURI	JOI	VINERI	SAMBATA
KERAMICELE	12.30 - 14.30	14.30 - 16.30	14.00 - 16.00		8.00 - 10.00	8.00 - 10.00
RACHETOMODELE	14.30 - 16.30	16.30 - 18.30	16.00 - 18.00		11.00 - 13.00	10.00 - 12.00
	16.30 - 18.30				12.00 - 14.00	
BALET	13.00 - 15.00	13.00 - 15.00	9.00 - 11.00	14.30 - 16.30	14.30 - 16.30	
BANI MICHILIAN	16.00 - 17.00	16.00 - 17.00			16.30 - 18.30	
CERAMICA	16.00 - 17.00	14.30 - 16.30	15.30 - 17.30	13.30 - 15.30	14.30 - 16.30	
		16.30 - 18.30	15.30 - 17.30	15.30 - 17.30	16.30 - 18.30	
CHEMI EXPERIMENTALA	16.00 - 18.00	9.00 - 11.00	10.00 - 12.00	15.00 - 17.00		9.30 - 11.30
		16.00 - 18.00	14.00 - 16.00			11.30 - 13.30
		16.00 - 18.00	16.00 - 18.00			
CULTURA SI CIVILIZATIE ROMANEASCA	12.00 - 14.00	12.00 - 14.00	16.00 - 18.00		14.30 - 16.30	10.00 - 12.00
	14.00 - 16.00	RS LIC NEKA			16.30 - 18.30	12.00 - 14.00
		RS LIC NEKA				
		16.00 - 18.00				
LANZILUR POPULARE	17.00 - 20.00	14.00 - 16.00	14.00 - 16.00	16.00 - 18.00	17.00 - 20.00	9.00 - 11.00
	Casa Artalet	16.00 - 18.00	16.00 - 18.00	16.00 - 18.00	Casa Artalet	11.00 - 13.00
		16.00 - 18.00	16.00 - 18.00	16.00 - 18.00	16.00 - 18.00	
DESEN-PICTURA	14.00 - 16.00	14.00 - 16.00	11.30 - 13.30	14.00 - 16.00	14.00 - 16.00	9.00 - 11.00
	16.00 - 18.00	16.00 - 18.00	14.00 - 16.00	14.00 - 16.00	16.00 - 18.00	
			16.00 - 18.00			
ECOTURISM	14.30 - 16.30	16.30 - 18.30		14.30 - 16.30	14.30 - 16.30	8.00 - 10.00
	16.30 - 18.30		16.30 - 18.30	16.30 - 18.30	16.30 - 18.30	10.00 - 12.00
						12.00 - 14.00
ELECTRONICA	16.00 - 17.00	15.00 - 17.00	15.00 - 17.00	8.00 - 10.00	8.00 - 10.00	
	17.00 - 19.00	17.00 - 19.00	17.00 - 19.00			

Fig 6 Schedule page

- **Registration** page contains an online registration form at the beginning of the school year. In order to facilitate the registration, a column with the educational offer was also introduced, so that the selection should be easier.
- **Contact** page offers the contact data of the institution together with a map that helps the orientation. Also e-mails can be sent from here to the webmaster.

On the website, symmetrically with the logo of Palatul Copiilor, a link towards Games is included. [3]

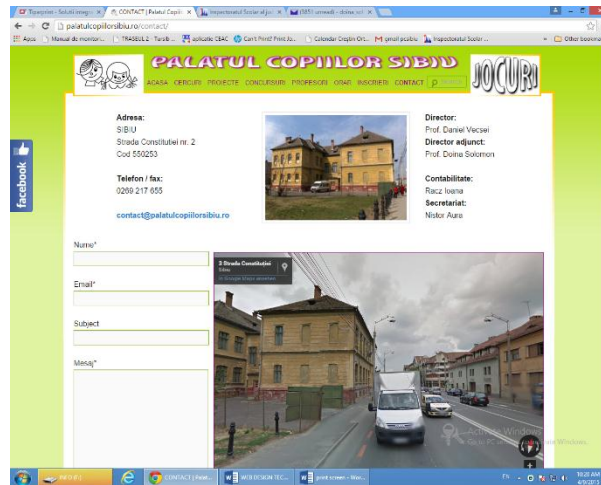


Fig 7 Contact page

## 4 Means of implementation

The website <http://palatulcopiilorsibiu.ro/> was designed in WordPress, the theme used is Twenty Eleven, where I created two new Templates: 'homepage' and 'circles'. I used the grid system 960gs on the first page. HTML was used for supplementing all the menu pages: for tables, texts, etc. My own contribution to CSS:

- background: gradient and color;
- the color contrast rule was used <http://colorshemadesigner.com/csd-3.5/>;
- the design of the div's in the main page, icons on the circles of Palatul Copiilor Sibiu, tutors' page etc.;

- plug in **Dropifi Contact Widget** custom color and text;

- In the Header section I introduced:

1. logo of Palatul Copiilor Sibiu arranged in Photoshop;
2. the title of the website created in Photoshop;
3. logo of games created in Photoshop;
4. menu: Plug in **WP Menu JQuery Effect**, 1s transition effect.[4]

- In the page I used:

1. Plug in **Blaze Slideshow** with custom settings: colors, text, effects;
2. Photo gallery: plug in **Easy FancyBox, Gallery Plugin**;
3. For registration to Palatul Copiilor Sibiu I used plug in **Contact Form 7** custom, but for the school year 2014-2015 the online registration was no longer used;
4. For Contact Palatul Copiilor Sibiu I used plug in **Contact Form Manager**;
5. IFrames used: contact map - google maps; Social media Links - Facebook; games for children - KIZI;
6. CSS was used for special effects for circles and tutors.

The Footer section is personalized:

1. The sponsors of Palatul Copiilor: Carousel created with Plug in Wordpress;



2. The signature of the page author was created in Photoshop.

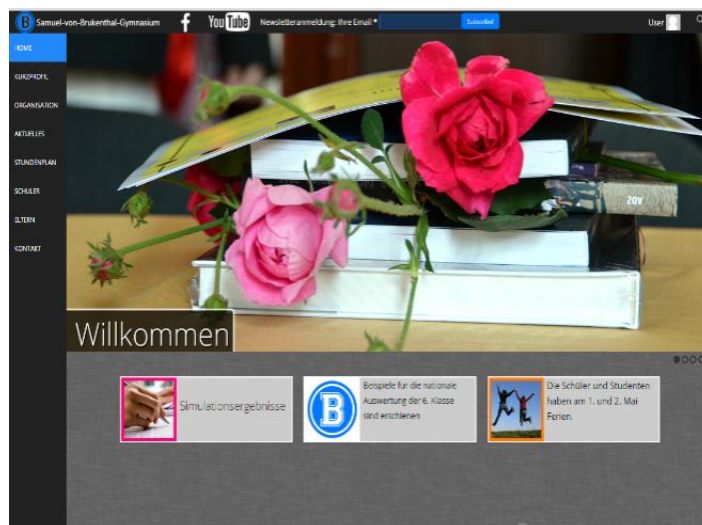
The login section on the administrator's page was personalized by using plug in **Custom Login 2.0** <http://palatulcopiilorsibiu.ro/login000/>.

The website <http://colegiulbrukenthal.ro/> was designed in WordPress, the theme used is Twenty Ten, where I created some new Templates. I used a custom grid system, created by me. HTML was used for supplementing all the menu pages: for tables, text, etc. My own contribution to WordPress:

- Personalized CSS: menu, template design, design page, etc.;
- Personalized photo gallery in CSS and „responsive” Lightbox plugin;
- Two image sliders, sliders created in PHP plus personalized settings in these pages;
- The footer section is personalized: personal signature;
- Personalized WordPress plugins: Options Framework, Contact Form 7, Slider Mata Mare JS, Slider Mata Mare JS Altheit;
- Custom page templates for WordPress: Custom HTML page templates with design in CSS (Stundenplan, Home, Admin Schedule etc.);
- Custom category template for WordPress: New Standart, Stundenplan, New Standart + Sub-Menu;
- Use of a plugin for newsletter: personalized Wysija Newsletters Plugin and with images developed in Photoshop; [16]
- Data bases: I used data bases to create tables with MySQL. I used the obtained tables together with PHP to create a new page for the easy change of the grades schedules. This change can be currently performed by a secretary of the school or by a student responsible from each grade;[5] [6]
- IFrame used: contact map – Google Maps;
- Social media links - Facebook, Youtube;
- JavaScript was used for the educational offer;
- Photoshop was used for: logo, organizational chart and photos (sliders, photo galleries, tutors).

The website is made up of the following sections:

- 1) **Home** – main page: here is a slider designed by me with representative images for the school and some frameworks with Links to news.



*Fig 8 Colegiul Brukenthal Home page*

2) **About us** with the following subchapters: greetings of the school principal, our mission, educational offer, facts and figures, history – where there is a new slider created by me, as well as photos and videos.

3) **Organization** has the following subchapters: organizational chart created by me in Photoshop, school management, tutors – where I enumerated all the tutors of the taught subjects for the school year 2014-2015, personnel and career, plus the internal regulation of the school. [15]

4) **News** with the following subchapters: structure of the school year, novelties, calendar for the school scheduled events, Brukenthal in mass-media, useful links, school yearbooks, partners where the partner schools from other countries are listed and finally the archive.

5) **Schedule**: a complete schedule of grades, organized by days and hours, easy to follow.

Stundenplan II. Semester 2013/2014

5. A | Klassenlehrer: RADU CIPRIAN

UHRZEIT	MONTAG	DIENSTAG	MITTWOCH	DONNERSTAG	FREITAG
13:00-13:45					Klassenstunde
14:00-14:45	Deutsch	Rumänisch	Deutsch	Englisch	Rumanisch
14:45-15:30	Deutsch	Rumänisch	Deutsch	Englisch	Musik
15:45-16:30	Rumänisch	Mathematik	Geschichte	Deutsch	Mathematik
16:30-17:15	Rumänisch	Mathematik	Erdkunde	Biologie	Mathematik
17:30-18:15	Sport	Zeichnen/I	Religion	Design	sächsische Folkuttenanze
18:15-19:00	Sport	EDT2	Religion		
19:05-19:50	-	-	-	-	-

UNTERRICHTSFACHER	LEHRER	ARBEITSSPRACHE
Rumänisch	RADU CIPRIAN IOAN	Rumänisch
Deutsch	SECAS CIANA	Deutsch
Englisch	SAVA CRISTINA	Deutsch
Mathematik	BONCIU ALEXANDRINA	Deutsch
Biologie	MUNTEAN CLAUDIA	Deutsch
Geschichte	FROHN ALEXANDER HUGO	Deutsch
Erdkunde	MISACHEVICI GERHARD	Deutsch
Religion	JUNESCH HANS GEORG	Deutsch
	MIHAI DANIELA	
Musik	PHILIPPI FRIEDRICH GEORG PH???	
Zeichnen	PARAIAN SORANA BIANCA	Deutsch

Fig 9 Schedule page

6) **Students**: a section that includes anything interesting for the students: registration, exams, awards, contests, scholarships, clubs: Reading club and Mindlab, library (including a schedule of the school library), reading books, bibliography for the gymnasium grades. [12], [13]

7) **Parents**: this section facilitates the relationship between the parents and the school and tutors and includes: the contact for the parents association and donations, meetings with parents and consultation sessions with the school tutors when the parents can check the results of the students' activities.

8) **Contact**: contact data of Samuel von Brukenthal High School, together with a map for orientation purposes. Moreover, e-mails can be sent from here to the secretariat. [8].

## 5 Actual description of the project

A notable difference between the two websites created is that the second one is dynamically designed. A dynamic website allows the change respectively update of its pages like in the case of a Word document, depending on the respective needs (see the schedule at the beginning of the school year), [7], [9], [10], the changes being performed to the website in real time. This fact renders the website <http://colegiulbrukenthal.ro/> not just a collection of pages that include texts and image, but an interactive interface between tutors and students (for ex. schedule page), that assures a flexible communication means. [11], [14], I present below some pages from the both websites designed by me and the program codes related to these.

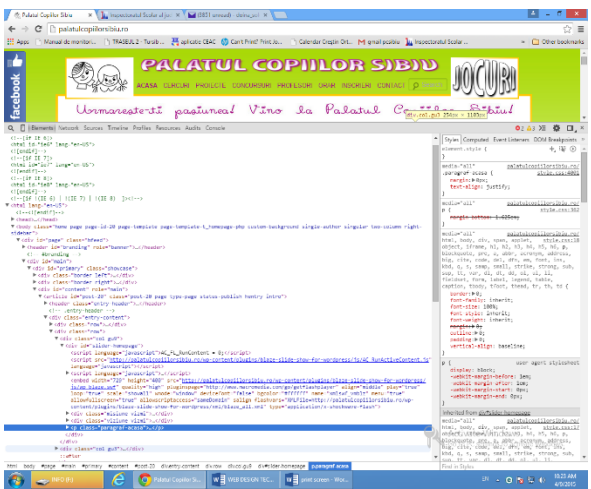


Fig 10 Palatul Copiilor Home page

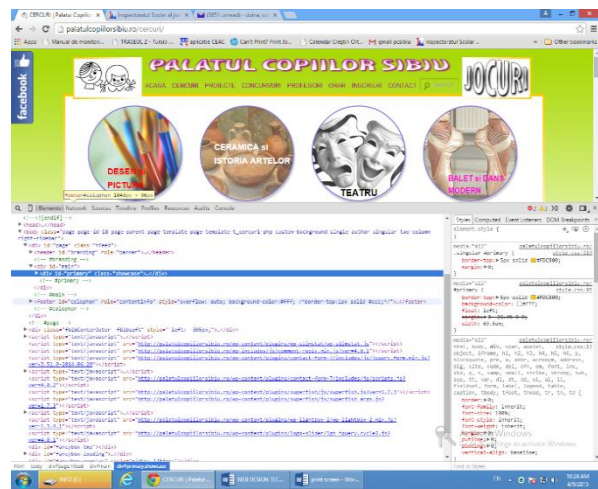


Fig 11 Circles page

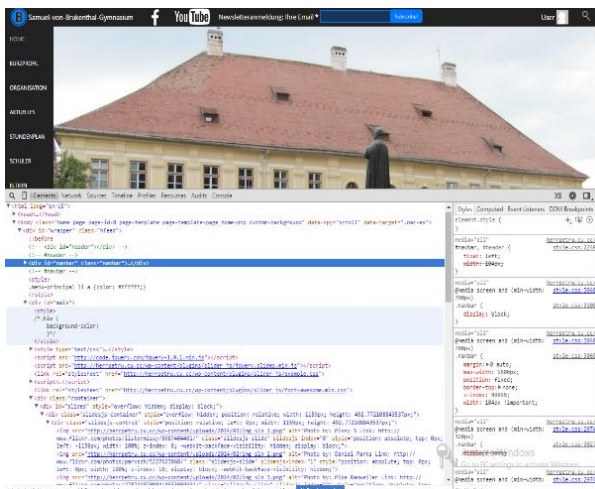


Fig. 12 Colegiul Brukenthal Home page

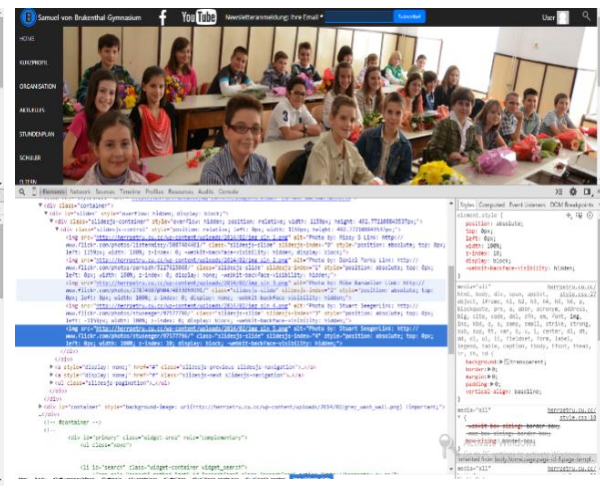


Fig. 13 Home page /slider



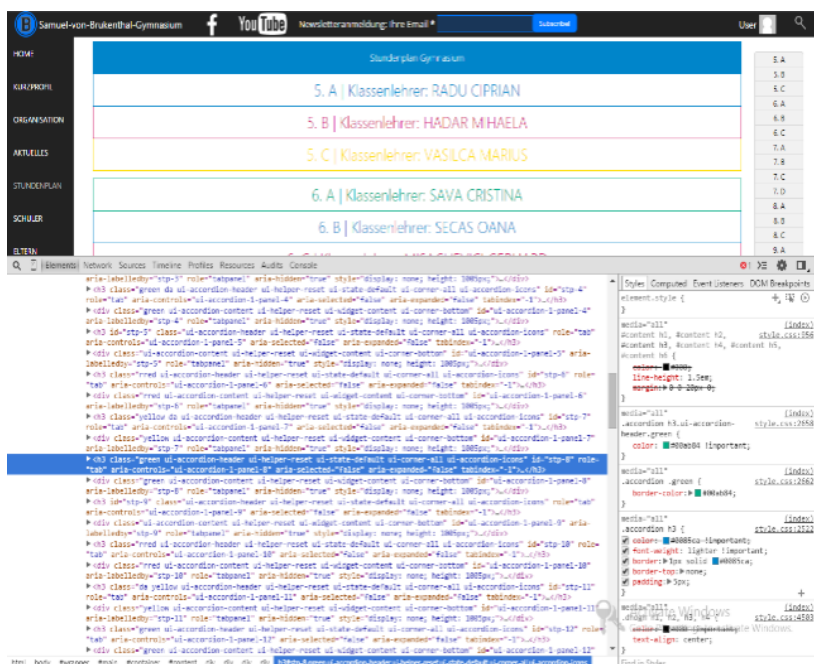


Fig 14 Schedule page

## 6 Conclusions and future development

After the completion of the first website, it was a challenge for me to improve the website both in terms of design and functionality, but especially to design a dynamic website for *Samuel von Brukenthal High School*. Thus I created an educational website which is not only dynamic but can be applied and developed for the general management of schools and high schools.

In the near future I intend to improve the website of *Palatul Copiilor* by designing the “responsive” type website, so that the pages should be useful, attractive and thus sufficiently accessed.

## 7 References

- [1] Rogojanu Petru - Design și implementare de pagini de internet în educație, Volum PCID, Sibiu 2015
- [2] <https://wordpress.org/>
- [3] <http://colorshemesdesigner.com/csd-3.5/>
- [4] <https://wordpress.org/plugins/>
- [5] <http://getbootstrap.com>
- [6] <https://codex.wordpress.org/>
- [7] [https://codex.wordpress.org/Theme\\_Development](https://codex.wordpress.org/Theme_Development)
- [8] [https://codex.wordpress.org/Conditional\\_Tags](https://codex.wordpress.org/Conditional_Tags)
- [9] [https://codex.wordpress.org/Child\\_Themes](https://codex.wordpress.org/Child_Themes)
- [10] [http://codex.wordpress.org/Plugin\\_API/Hooks](http://codex.wordpress.org/Plugin_API/Hooks)
- [11] [http://adambrown.info/wp\\_hooks/hook](http://adambrown.info/wp_hooks/hook)

- [12] <https://gist.github.com/luetkemi/2007374>
- [13] <https://roots.io/>
- [14] <http://premium.wpmudev.org/blog/wordpress-plugin-development-guide/>
- [15] [http://phpxref.ftwr.co.uk/wordpress/nav.html? functions/index.html](http://phpxref.ftwr.co.uk/wordpress/nav.html?functions/index.html)
- [16] <http://underscores.me/>

Petru Rogojanu  
Palatul Copiilor Sibiu  
7<sup>th</sup> grade  
Sibiu, Romania  
E-mail: herrpetru01@gmail.com

# **AIM**

## **Kit for app development**

**Robert Sandica, Marius Toma**

**Teacher Coordinator: Simian Dana**

### **Abstract**

Gaming nowadays is a huge industry which gives huge amounts of fun. But what would this industry be without keyboard and mouse or controllers and many other input devices, those devices are the connection between player and the game creating a fun experience . The aim of this article is to design a device capable to improve the players interactive experiences and to create games which use this device. To this end we developed a library for creating apps. The proposed system is built not only for player use but also for developers use who can use the library to design other games.

## **1 Introduction**

Our project named AIM is intended to bring fun simulating a real experience replacing the mouse whit a device capable of aiming at a target on the screen.

AIM is a project designed for creation and for use. It uses an arduino board as the brain, that control all the necessary components.

We put together a couple of sensor and other components such as buttons used for acting like keyboard buttons. Our device acts like a real gun simulating the aiming and shooting acts.

The principle of our device is different of similar existing devices.

Our final purpose is to create games using this device to do this we offer the game developers possibility to capture sensors and buttons status this is what the C# library is able to do.

In the rest of the article we will detail the two parts of the project

## 2 AIM Rifle



Fig.1 1.Arduino board, 2.IUM sensor, 3.Proximity sensor, 4.Power supply, 5.Breadboard (for wire connecting ), 6.Button A



Fig.2 7.Button B, 8.Button C, 9.Button D, 10.Button E,

The AIM Rifle is the device purposed for aiming on the target, Fig 1 and Fig 2 presents the components of AIM Rifle in early stages of construction. In this section we will present more of the rifle principle and equipment.

The rifle can work in two ways one analogic and one digital, the analogic one move the target on the screen at the exact position and the digital one only says in which direction the rifle is moved.

### 2.1 AIM principle

As said above AIM is not just a device AIM is whole system with two parts

The first part AIM Rifle is intended more for the user because you don't have to be a programmer to use it, it's more like plug and play type.

The rifle aim principle stays in the IMU sensor which gives back to arduino two axes the X axis given by magnetometer reading the North pole and Y given by accelerometer using these

two we have developed a formula which converts the position of the IMU into the Mouse cursor onto screen.

The proximity sensor it is put there to mimic the existence of a scope when you break the infrared beam it is like you would click the right mouse button and in some games this activates the scope.

The buttons default are like this :

A is for left click in some games this is the trigger

B is S in games this is used for back

The first step in using AIM rifle is to calibrate it using this image and selecting the distance to the screen

You set the cursor in the center of the image and aim the rifle then fire once and the calibration is over if you want to recalibrate it you simply press the reset button on the arduino board and redo the above step.

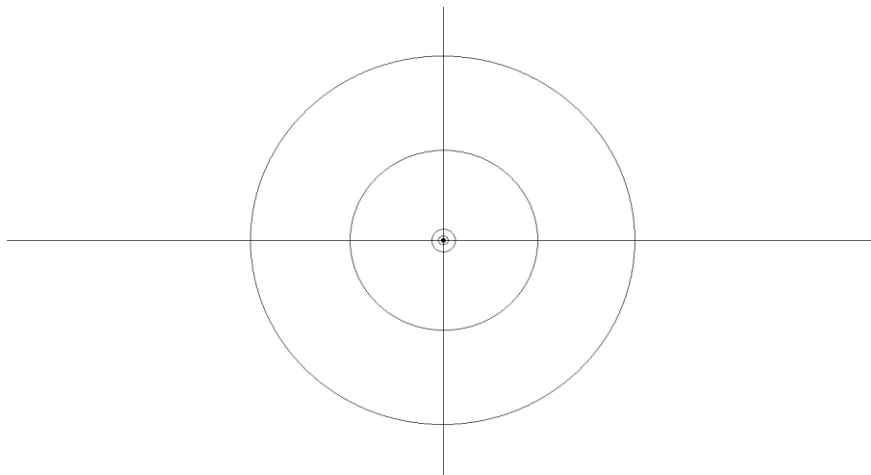


Fig 3 calibration picture

## 2.2 Arduino

(1)Arduino is an open source kit composed of a program called Arduino and a single board microcontroller it is use for many projects such as school projects home projects, it has a large variety of sensors and other components.

The board it self it's based on AT Mega family of microcontrollers which can be programed in the Arduino platform based on the C and C++ programing language. In our project we are using the Arduino Leonardo board based on the ATMega microcontroller we chose this board because it is capable of controlling the mouse and keyboard whit a set of libraries called Mouse and Keyboard.



Fig.4 Arduino board

## 2.3 Sensors and other components



Fig.5 IMU



Fig.6 proximity sensor

(2)IMU: The Pololu MinIMU-9 is 9 degree of freedom device which give back to user the exact orientation of the device it is a 3 in 1 device because it is equipped with a accelerometer a magnetometer and a gyroscope all three have 3 axes of orientation giving the name of 9 degree of freedom. In our project we are using the magnetometer and the accelerometer to move our sight on the screen.

(3)Proximity sensor it is a infrared based device it detects if an object is bloking the ray of inrared light at distances of 2 to 10 cm this acts in our project as a scope.



Fig.7 power supply



Fig.8 breadboard



Fig.9 brick button

(4) Power supply this piece of equipment mounted on the AIM rifle ensures that all the energy is provided in the range of 3 to 5 Volts

(5) Breadboard: is a wire connection pad

(6) Buttons this buttons have a brick from, meaning that they are some sort of plug and play ready to use. B replace S for back, C is W used for forward, D is used for A keyboard button also called left, E is used for D called also right, But using the AIM library you can change the buttons and proximity sensor configuration giving them any value that you want.

### 3 AIM Library

The AIM Library is the second part of the AIM project this library it is written in C# code. This part of project is intended more for developers because it is the connection between arduino and the code that you are writing to create the application who will use the AIM Rifle. This library has two main uses, one is intended to receive values from the IMU sensor, proximity sensor and the buttons and the other one is intended to set the keyboard buttons to A B C D E buttons located on the rifle.

For each component there are methods and classes that give you access to the specific part. The class responsible for receiving sensor and buttons status it is called SignalReceived and has some derived classes IMU and Proximity which IMU give back the accelerometer gyroscope and magnetometer values and the Proximity give back the true value if the ray of infrared from sensor is blocked.

The class responsible for buttons is called Buttons and every button has it own method of transmission the true value if the buttons is pressed. The second part is called SetButtonKey and for every button there is assigned a character find on the keyboard using this class and it methods you can change the default character of the buttons placed on the rifle.

### 4 Conclusion, Future developments and improvements

AIM project is a system which gives user possibility to get a fun experience by using this device. On our knowledge the device principle is new.

The library that give the access to the sensor and buttons can be customized by game developers. This whole system AIM was tested by creating a game called Scoping. In the top position of future improvements is the way of finding a better formula for converting the IMU onto the cursor position to improve accuracy and mobility.

The next step in development area is to create a game special designed for AIM Rifle capabilities And an add-on to the rifle we are looking to create a stand to increase stability and accuracy.

### References

- [1] *arduino description* <http://www.arduino.cc/>
- [2] *pololu IMU description and picture* <https://www.pololu.com/product/1268>
- [3] *pololu proximity sensor description and picture* <https://www.pololu.com/product/1134>

- [4] *power supply picture* <http://www.robofun.ro/breadboard-power-5v-3-3v?search=power>  
[5] *power supply description* <http://www.robotshop.com/en/sfe-breadboard-power-supply-kit.html>  
[6] *arduino Leonardo board picture and description*  
<http://www.arduino.cc/en/Main/ArduinoBoardLeonardo>

Sandica Robert  
ULBS  
Computer Science  
Adress: str.Dr.I.Ratiu, Nr.5-7, Sibiu,  
550012,Romania  
Sandica.robert@gmail.com

Toma Marius  
ULBS  
Computer Science  
Adress: str.Dr.I.Ratiu, Nr.5-7, Sibiu,  
550012,Romania  
kid.morvius@gmail.com



# Modeling of decision-making process by means of a method of a multicriteria alternative choice ELECTRA

Maria Sokolova  
Teacher Coordinator: Vladimir Chernov

## Abstract

The paper illustrates advantages and disadvantages of business process modeling and the analysis of notations and tools of business process modeling.

Modeling of decision-making process by means of a method of a multicriteria alternative choice ELECTRA was made on the basis of the carried-out analysis. Modeling was made in parallel with the help of the Business Studio program in two notations: IDEF0 (Integration Definition for Function Modeling) and EPC (Event-driven Process Chain).

## 1 Introduction

People should make decisions almost every day and in different situations: during military operations, in policy, at business management, when choosing the car or the apartment and in thousands of other cases. The scheme of decision-making process doesn't depend on the area in which the decision is made. In other words, laws of decision-making are uniform for all problem domains.

Decision-making process by means of a method of a multicriteria alternative choice ELECTRA is considered in this work. This method is directed on the solution of tasks with specified multicriteria alternatives. The quality indicator of each alternative isn't defined quantitatively in this method, but only the superiority condition of one alternative over another is established.

The problem of the formalized graphical representation of the ELECTRE method stages for simplification of its understanding is actual. A main goal of this work is modeling of decision-making process by means of a method of a multicriteria alternative choice ELECTRA [1].

This work consists of the basic concepts of business process modeling; the analysis of notations and tools of business process modeling; the description and modeling of the ELECTRE method stages by means of the Business Studio program in IDEF0 and EPC notations; and plan of further researches.

## 2 The basic concepts of business process modeling

*A business process or business method is a collection of related, structured activities or tasks*

*that produce a specific service or product (serve a particular goal) for a particular customer or customers.* It may often be visualized as a flowchart of a sequence of activities with interleaving decision points or as a Process Matrix of a sequence of activities with relevance rules based on data in the process.

The **term modeling** has two major meanings. Firstly, it is the process of model creation as certain object representation of the original reflecting its most important features and properties. If the model is already constructed, then modeling — is the research (analysis) process of system functioning. The final goal of business process modeling consists in achieving operational enhancement [2].

**Business process model** *is the formalized description (graphic, tabular, text, symbolical) of real-life or expected activity of the enterprise.*

It is possible to use various methods for business process modeling. **The modeling method, or the methodology**, includes sequence of actions which should be executed for model creation, i.e. modeling procedure, and the applied notation (language).

### **3 The review of notations and tools of business process modeling**

It is senselessly to speak about advantages of this or that system or notation until the main objectives which this project has to solve aren't defined. In this work I compare **the most popular notations used for the description of business processes** (EPC and IDEF).

One of the most important aspects of the description of business process model is the reflection of the operating influences, feedbacks on control and control procedure control on the model. Control procedure can be reflected in the ARIS EPC notation only by means of the incoming documents indication. Unlike ARIS, each procedure in the IDEF0 notation must have at least one control response. If we show only sequence of procedures performance at creation of the model without caring of reflection of the operating documents and information, these models will have low value from the point of view of the analysis and further use.

If we try to reflect all conditions and restrictions defining performance of functions, it will be required to describe a large number of events and the input information, and the model will become difficult and badly readable. The notation of IDEF0 hasn't got these disadvantages.

Thus, IDEF0 notation is useful for small on scales (small and medium-sized enterprises, 2-5 people in group of consultants) and duration (2-3 months) projects. ARIS EPC notation is useful for large and/or long projects [3].

Within this work the one-time project of the decision-making process description by means of an ELECTRA method will be considered. **Both notations (IDEF0 and ARIS EPC) have significant advantages and disadvantages, also they complement each other. It is decided to model the process with the use of both notations for the purpose of their comparison and finding the best of them in this situation.**

Today there are a lot of various program tools for working with business process models from simple graphic editors to full-fledged multipart BPM-systems. In this work the choice was made in favor of Business Studio system. Advantage of this system consists in possibility of use of the most popular notations: IDEF0, Process (Basic Flowchart), Procedure (Cross Functional Flowchart), BPMN 2.0, EPC. Thus, this system will be suitable for the comparison of IDEF0 and ARIS EPC notations best of all.

## 4 The description of a method of a multicriteria alternative choice ELECTRA

The ELECTRE method is directed on the solution of tasks with specified multicriteria alternatives. The quality indicator of each alternative isn't defined quantitatively in this method, but only the superiority condition of one alternative over another is established. The problem formulation usually has the following type:

**Statement of a problem:** N criteria with rating scales (usually quantitative), the weight of criteria (usually integers), alternatives with estimates by criteria.

**It is required:** to allocate group of the best alternatives.

**Main stages of the ELECTRE method:**

1) two indexes (agreement and disagreement) are counted on the basis of the set estimates of two alternatives values. These indexes define agreement and disagreement with a hypothesis that the alternative A dominates alternative B;

2) levels of agreement and disagreement (to which the counted indexes for each couple of alternatives are compared) are set. If the index of agreement is higher than the set level, and a disagreement index is lower, one of alternatives dominates another. Otherwise alternatives are incomparable;

3) the dominated alternatives are removed from a set of alternatives. Remained alternatives form the first base. The alternatives entering a base can be either equivalent or incomparable;

4) "weaker" values of levels of agreement and disagreement (smaller on value level of agreement, and bigger level of disagreement) are entered at which bases with smaller amount of alternatives are allocated;

5) the last base includes the best alternatives. The sequence of bases determines the orderliness of alternatives by quality.

The important advantage of the ELECTRE method is the phasing of preferences allocation of the person making the decision (decision-maker). The detailed analysis allows the decision-maker to form the preferences, to define compromises between criteria [4].

## 5 Modeling of decision-making process by means of a method of a multicriteria alternative choice ELECTRA

Modeling process with the use of the IDEF0notation is visually presented on Fig. 1-6. There is a contextual chart of the process „solution of the task by ELECTRA method” on Fig. 1.

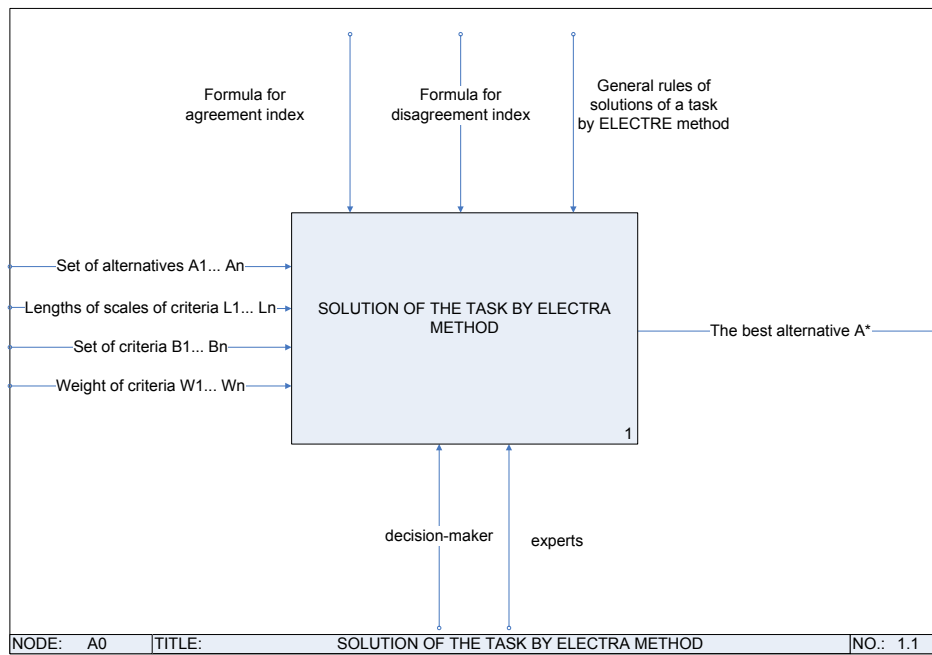


Figure 1 - Contextual chart of process A0

On Fig. 2 you can see the main six stages of the ELECTRA method:

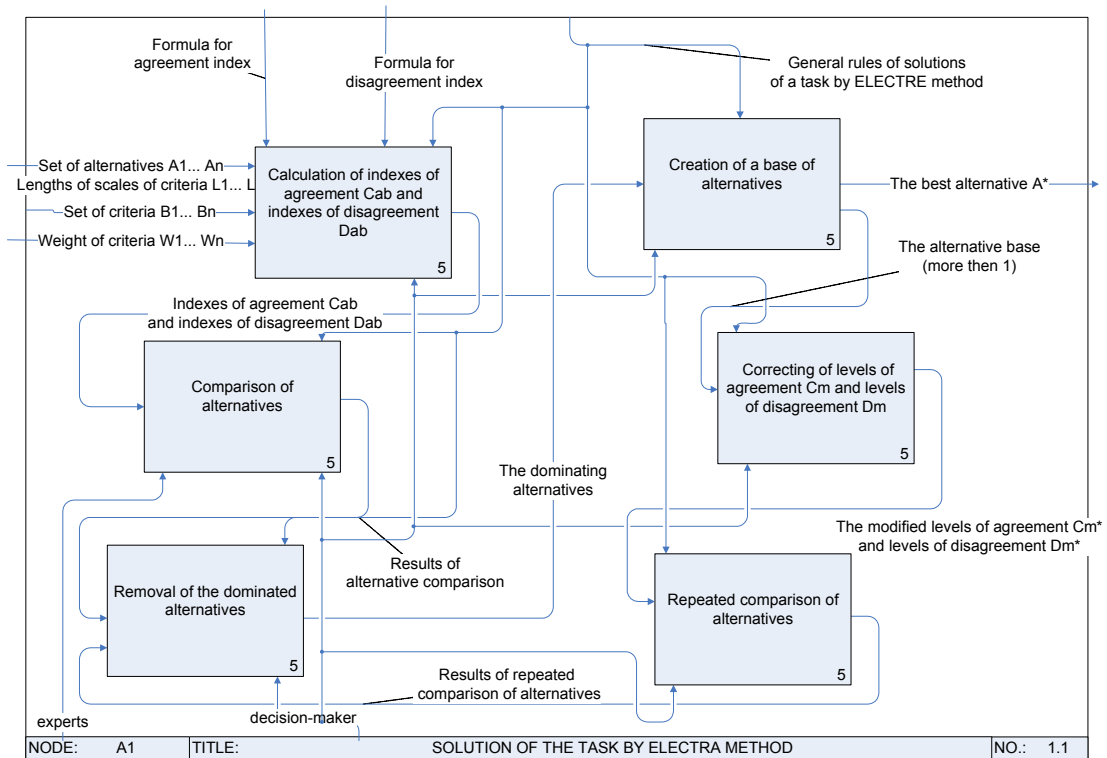


Figure 2 – Chart of the top level A1

On the first stage two indexes (agreement and disagreement) are counted on the basis of the set estimates of two alternatives values (Fig. 3):

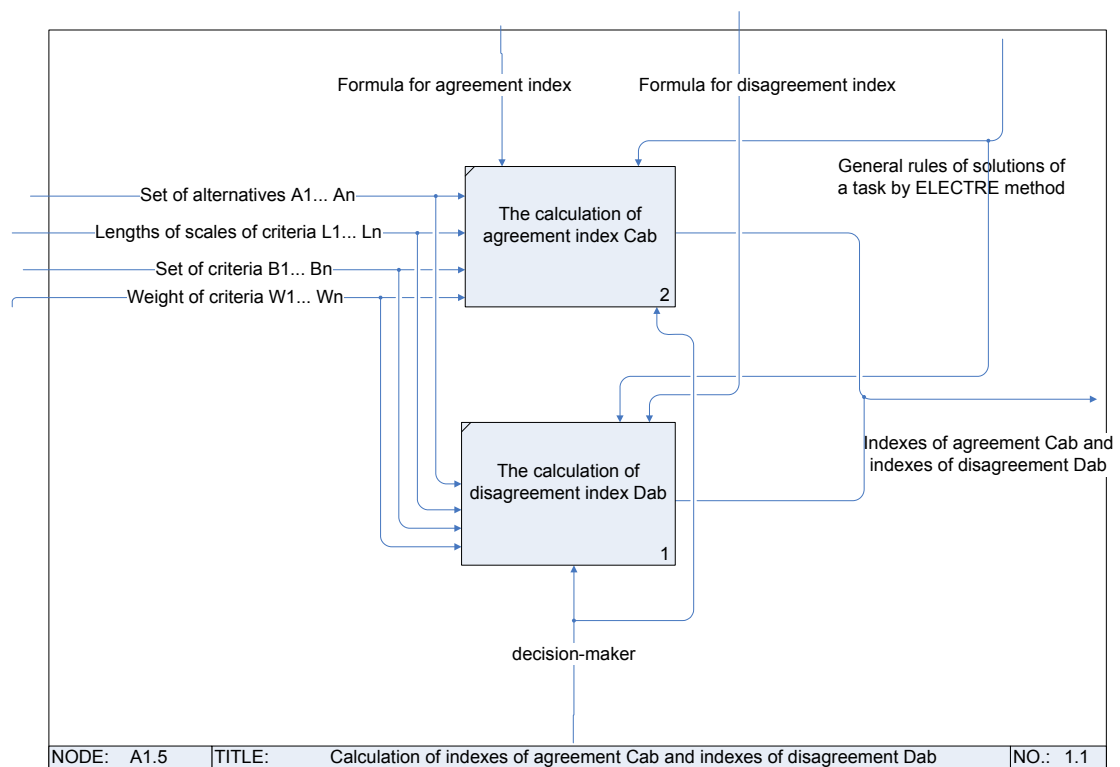


Figure 3 – Chart 2

On the Fig. 4 we can see that the dominated alternatives are removed from a set of alternatives.

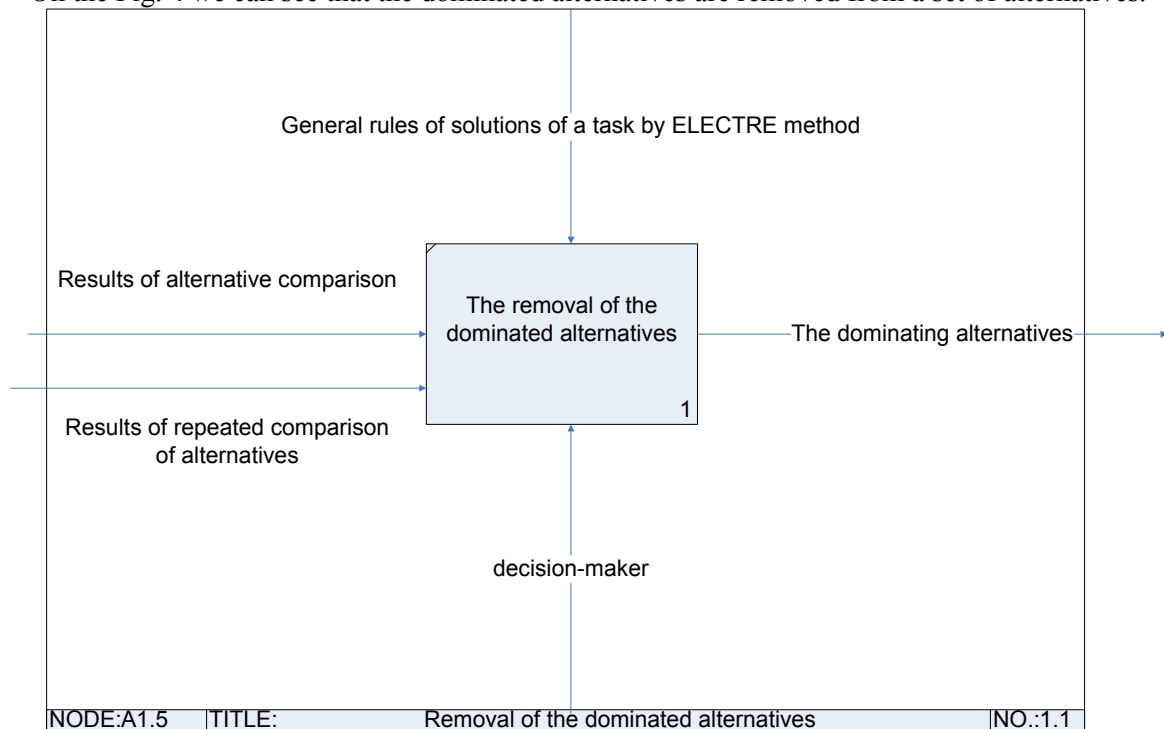


Figure 4 – Chart 3

On Fig. 5 we can see the correcting of levels of agreement and disagreement.

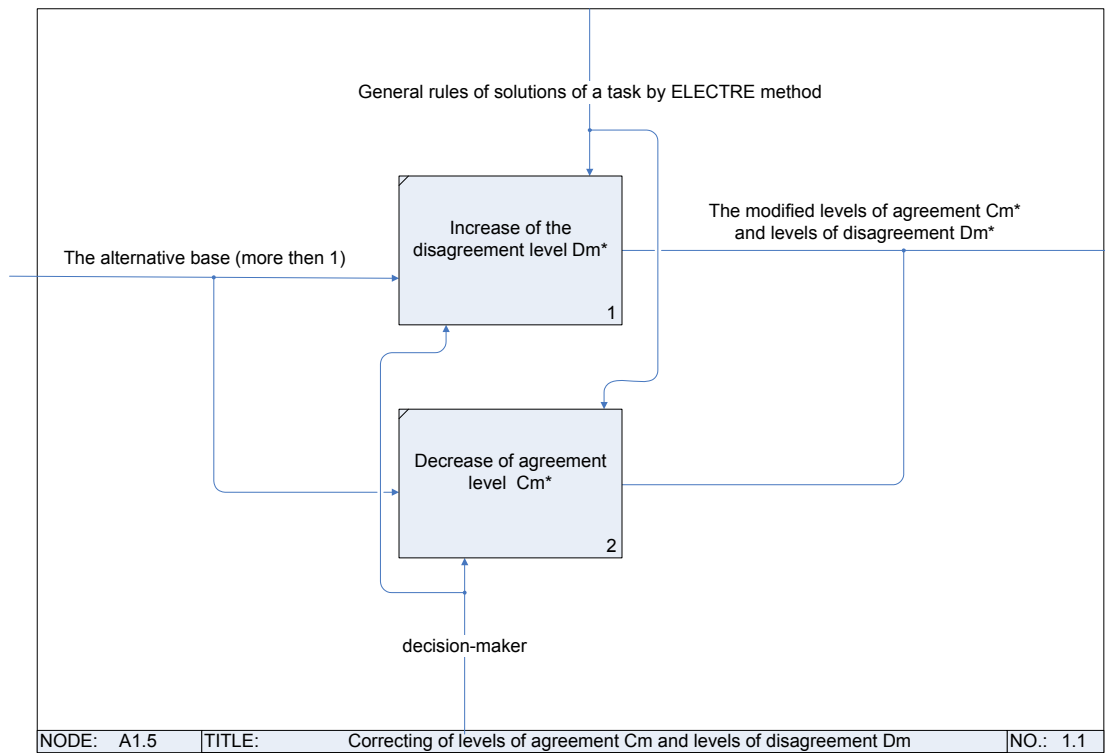


Figure 5 – Chart 4

There is a repeated comparison of the alternatives after removal of the dominated alternatives and correcting of levels of agreement and disagreement (Fig. 6).

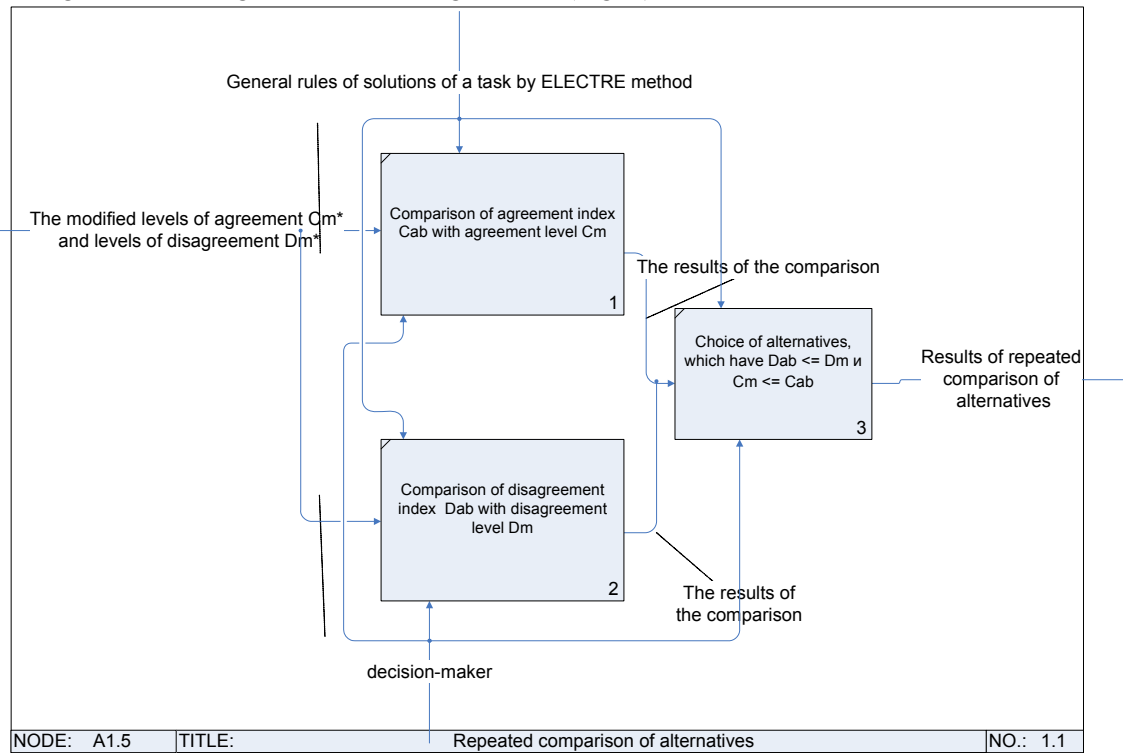


Figure 6 – Chart 5

Modeling process with the use of the EPC notation is visually presented on Fig. 7.

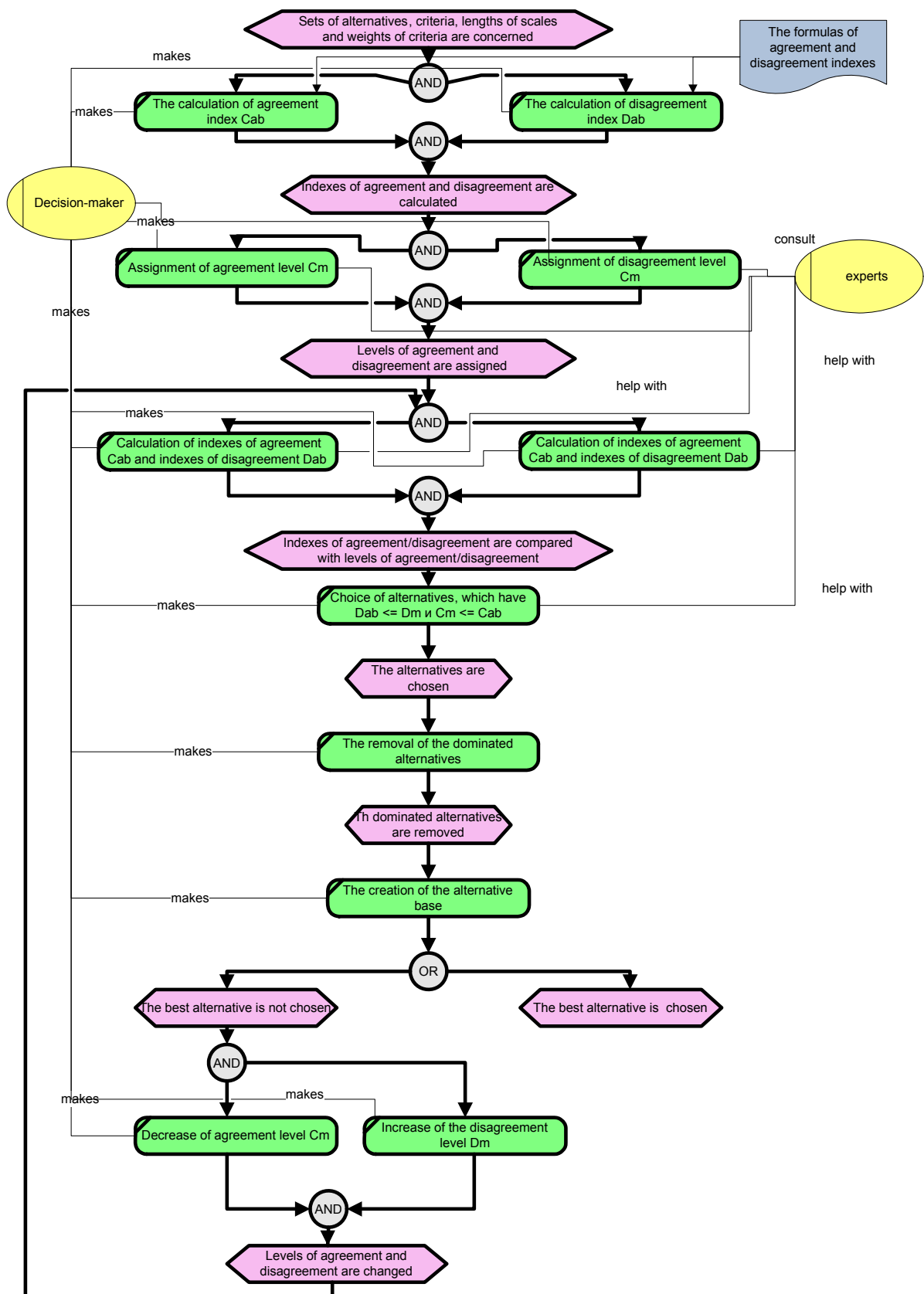


Figure 7 – Modeling of process in the EPC notation

## 6 Conclusions

Modeling of decision-making process by means of a method of a multicriteria alternative choice ELECTRA was made with the use of two notations **IDEF0** and **EPC** (Table 1).

Table 1  
Advantages and disadvantages of notations

	<b>Advantages</b>	<b>Disadvantages</b>
<b>IDEF0</b>	<ul style="list-style-type: none"><li>- use of functional approach,</li><li>- visibility of charts,</li><li>- accurate syntax and semantics of charts,</li><li>- possibility of decomposition of charts and modeling of hierarchy of processes</li></ul>	<ul style="list-style-type: none"><li>- not expressed temporary sequence,</li><li>- absence of specifications of processes (including restrictions on an input and output)</li></ul>
<b>EPC</b>	<ul style="list-style-type: none"><li>- use of process approach,</li><li>- visibility of charts,</li><li>- use of logical operators</li></ul>	<ul style="list-style-type: none"><li>- weak formalization of syntax and semantics of charts,</li><li>- complexity of modeling process (lack of decomposition)</li></ul>

**The main advantage of the IDEF0 notation is in decomposition principle which is applied when partitioning difficult process into its components.** Thus the level of specification of the process is defined directly by the developer of the model. **The main advantage of the EPC notation is in possibility of use of the logical operators allowing to describe process branchings** [5].

**Thus, both notations have the essential advantages and disadvantages, but, in my opinion, for modeling of decision-making process by means of a method of a multicriteria alternative choice ELECTRA, the notation of EPC is better because of the visibility of model.**

## References

- [1] Repin V.V. *Process approach to management. Modeling of Business processes*, Moscow, Standards and quality, 2004.
- [2] Repin V.V. *Business processes of the company: construction, analysis, regulation*, Moscow, Standards and quality, 2007.
- [3] Semenov I.O., Rychkin V.S. *Structural analysis of systems: IDEF technologies*, Moscow, Finance and statistics, 2001.
- [4] Sheer A.V. *Business processes: basic concepts, theories, methods*, Moscow, Educator, 1999.
- [5] Repin V.V. *Business reengineering: reengineering of the organization and information technologies*, Moscow, Standards and quality, 2006.

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



# GoStudent

Cosmin Stoian, Ioana Martin  
Teacher Coordinator: Elena-Liliana Martin

## Abstract

GoStudent is an educational software created to satisfy all students' needs regarding their school assignments. It includes 4 apps: Calculator, HTML Editor, Notepad and Portfolio. The applications do not resemble each other, so that each one is based on a different concept: Calculator – Mathematics Tool, Notepad – ICT Tool, HTML Editor – Programming Tool and Portfolio – Management Tool.

All apps bring some innovative graphical elements in order to attract and motivate students with their design, as well as with additional functions, which cannot be found in Windows.



Fig.1: GoStudent Logo

## 1 Introduction

These four tools were initially created separately and their order in GoStudent is actually the order of their creation and development. In the first instance, we wanted to create some of the basic programmes we had seen in Windows: Calculator and Notepad. Not only we managed to create these two, but we also improved some of their aspects, such as design and functions, based on our needs as students.

Secondly, we saw how HTML is taught in schools and we thought that our work could be significantly eased by introducing a fully functional HTML Editor.

Afterwards, we realized that none of the existing project manager apps were designed for our needs. Each software could only work with a certain type of files and most of them were completely dedicated to online use. This is what determined us to develop a suitable software for those who want to organize their files into categories offline and later to be able to share them with a Group: Portfolio.

Visual Basic uses Object Oriented Programming (OOP) which allows working with windows, buttons and other facilities. Therefore, all of our software are interactive, highly dynamic, with personalized interfaces. GoStudent is integrated on a Web page edited online (blog type) so that any user can download the desired applications quickly and safely. There is an entire category dedicated to this project. (<https://softwarebyics.wordpress.com/category/gostudent/>). A more simplified version of this papers was presented at the PCID Conference [4].

## 2 GoStudent's apps

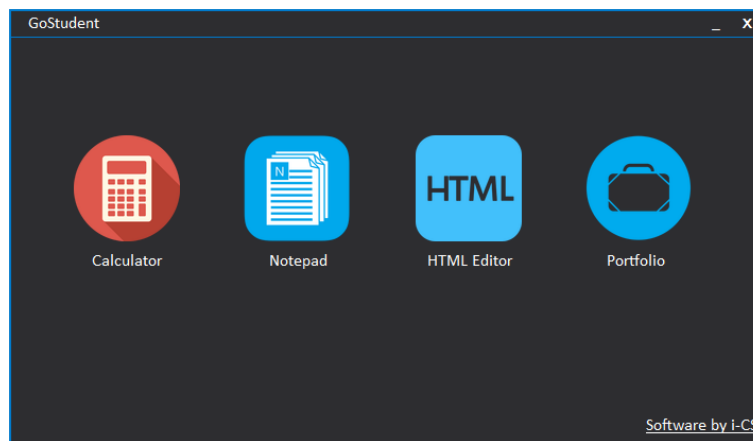


Fig. 2: GoStudent Menu

The use of facilities is interactive and dynamic. When accessing an option, the application provides the appearance of a ToolTip to confirm the action chosen by the user.

The visual effect related to discrete color change for various operations is perceived by the user who can recognize if the operation he is going to perform is really the one desired. (eg. closing the window is shaded with a red contour).

For each of the following apps, we have conceived Help functions where users can find the basic information for common use. These instructions for use were designed to be simple and clear, in order to make it easier for the users to adapt to the new form of the well-known apps (Calculator, Notepad) and also to get to know our original apps and use them to their fullest.

### 2.1 Calculator

This app is quite similar to the one from Windows, as it allows work with usual elementary arithmetic functions. The graphical layout makes Calculator more attractive than the one accessed at ICT classes when the teacher presents tools provided by Windows.

To help students who would like to use it, in addition to the standard features of an usual calculator, this one has buttons for the calculation of linear and quadratic equation's roots.

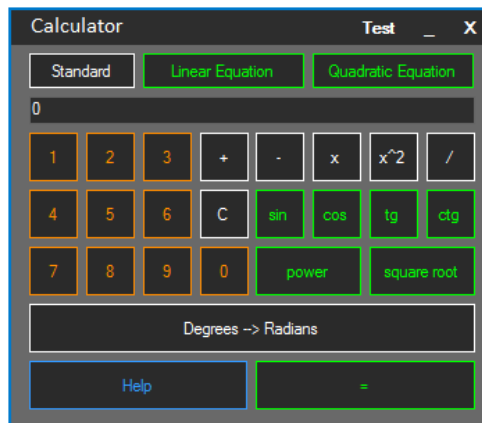


Fig. 3: Calculator Standard Window

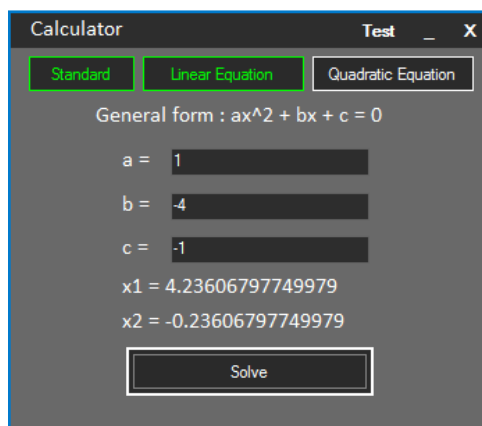


Fig. 4: Calculator Quadratic Equation Window

## 2.2 Notepad

Notepad is a text editor like the one provided by Windows. The menu allows file operations (New, Save, Open), specific operations of text editing (Redo, Undo, Select All, Copy, Paste), specific operations of text formatting (Font, Color) and access to the Help function.

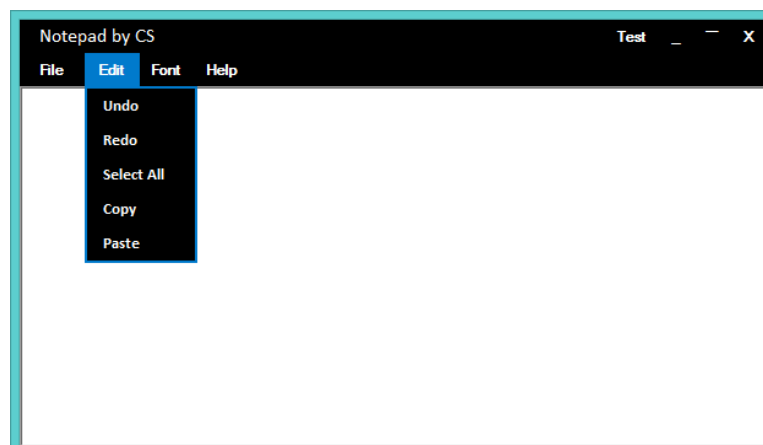


Fig. 5: Notepad Edit Functions

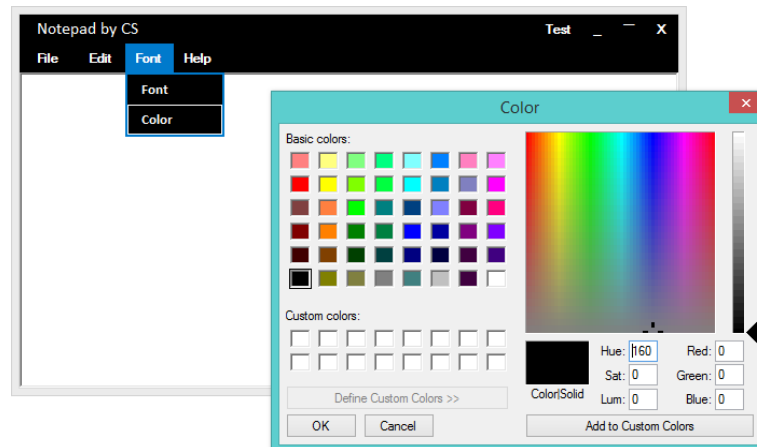


Fig. 6: Notepad Color Selection

Its simple and light design can help students get started with file operations, text formatting, etc.

## 2.3 HTML Editor

HTML Editor allows the user to write HTML code and see the effect obtained by loading the Preview function which works based on an internet browser (Internet Explorer). In this way, students will no longer have to switch between the two forms of the file, eliminating a major source of error (saving the file with the right extension, refreshing the page after every change in the code etc.).

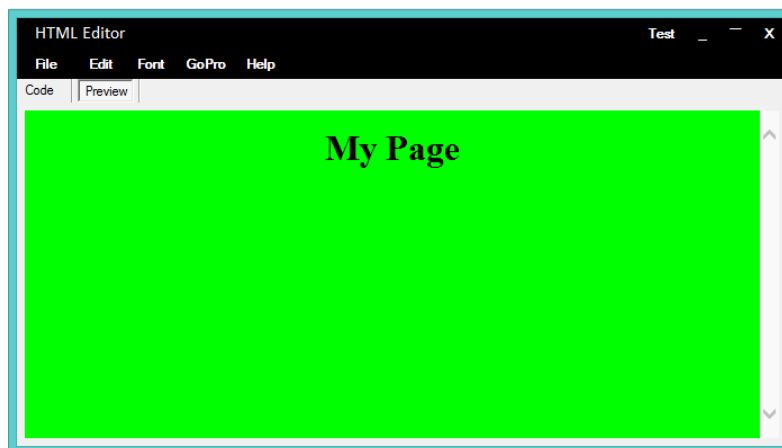


Fig. 7: HTML Editor Preview function

The editing is done in the Code window and the student is able to preview in real time the effect of the labels and the tags he used, thereby easing the work and eliminating time switching between the edit window and the view in the browser.

### 2.3.1 HTML Editor Pro

If users want to create complex webpages and organize projects, we have created a GoPro function in the top menu to launch an HTML Editor Pro version. This function allows users to create projects and work with several files at the same time (Tabs).

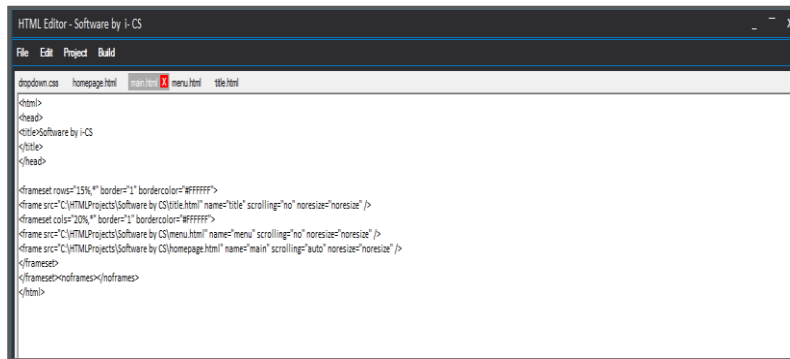


Fig 8: Example of a project with Tabs

In order to preview a site, we have created two ways to „Build”: Fast Build and Pro Build. Fast Build shows the progress in the main window, whereas Pro Build connects the code to the default browser, if the user prefers working with it.

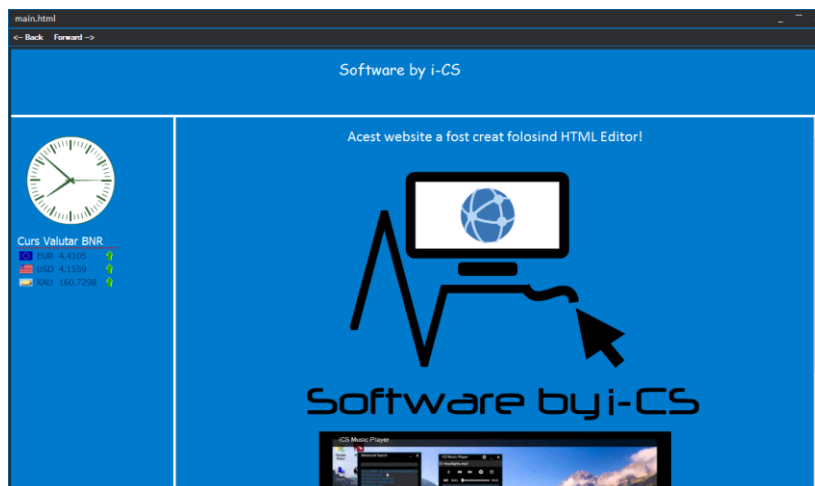


Fig. 9: Fast Build



Fig. 10: Pro Build

Moreover, HTML Editor Pro comes with a prediction function, for those who are very passionate about HTML or CSS, but not very interested in learning all tags to the letter.

### 2.3.2 i-CS Editor

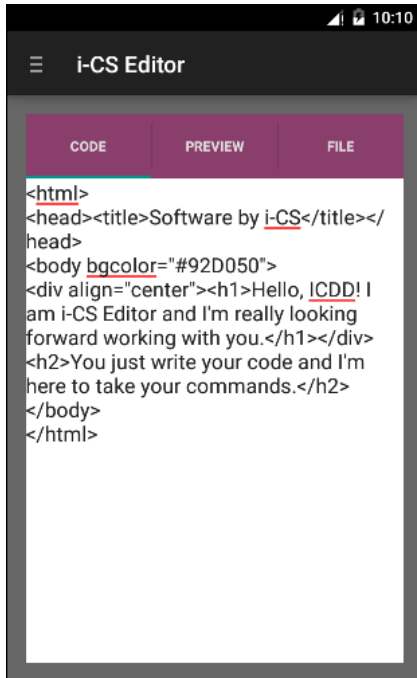


Fig. 11: Example of HTML code

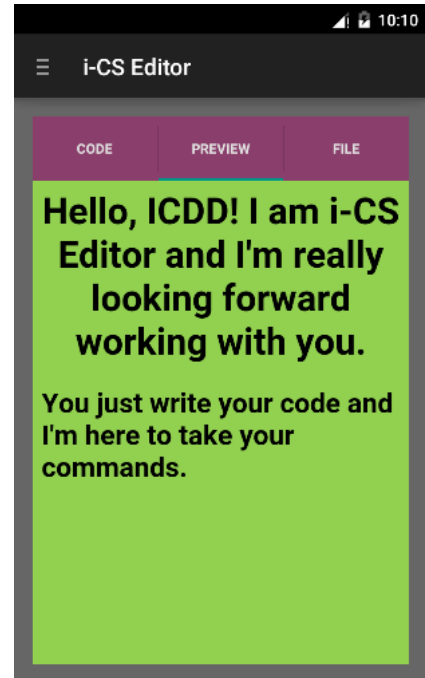


Fig. 12: Preview function

i-CS Editor is a very simple editor of HTML code, dedicated not only to people who are passionate about HTML and want to work even when not around a computer, but to students as well.

It works on all Android devices (smartphones, tablets). This app has a light, yet alluring design, which makes it a quick and pleasant tool for anybody. As tablets industry is under constant development, this application comes as an adaptation of the HTML Editor for Windows to support people who would rather work on a tablet than on a computer.

What is more, in many countries people are working hard to modernize educational systems so that in a few years it is possible to witness the placing of tablets in schools.

## 2.4 Portfolio

Portfolio is a tool which can be used to organize documents in school activities and more. The application was developed using the facilities of Visual Basic and allows multiple file operations organized by categories.

Users may define the hierarchical structure of the portfolio (categories → subcategories → files). and they can always change the structure by adding, deleting etc. In addition, the main window along with the menu window (in the left) can be resized –drag- if necessary (for example, if the name of one category/file is too long).

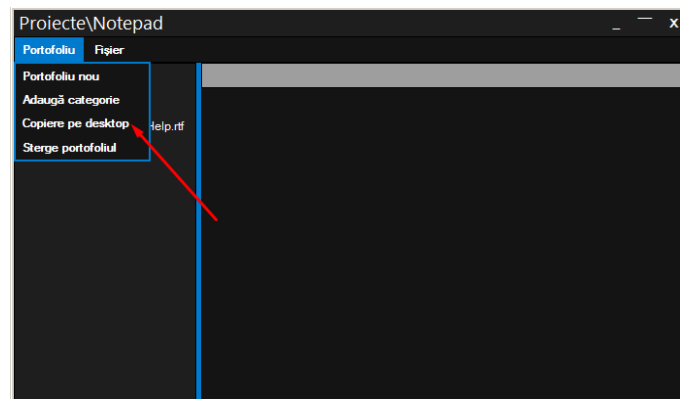


Fig. 13: Send to desktop function

For easier access to the portfolio, this app allows users to send it to the desktop with just one click. What is more, Portfolio comes with additional functions, namely creating an account, logging into it and creating/joining a group, in order to share some of the files for easier communication.

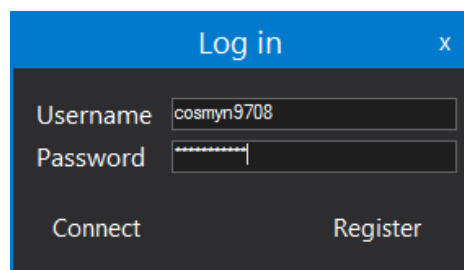


Fig. 14: Log in function

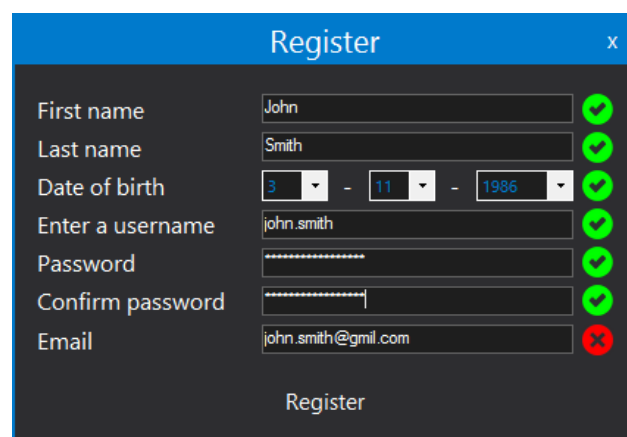


Fig. 15: Register form – wrong e-mail structure

The Register form is controlled by some restrictions like validation function for the date of birth, the length of the password and the structure of the e-mail address.

## 3 Program Code

### 3.1 Calculator

```
If a2 = 0 Then
    If b2 = 0 Then
        If c = 0 Then
            MsgBox("There is an infinite number of solutions!!", , "Calculator")
        Else
            MsgBox("This equation is undefined", , "Calculator")
        End If
    Else
        MsgBox("There is only one solution for this equation!", , "Calculator")
        solutie1 = -b2 / c
        solutie2 = solutie1
        x1_label.Text = "x1 = " & solutie1
    End If
Else
    delta = b2 * b2 - 4 * a2 * c
    If delta >= 0 Then
        solutie1 = (-b2 + Math.Sqrt(delta)) / (2 * a2)
        x1_label.Text = "x1 = " & solutie1
        solutie2 = (-b2 - Math.Sqrt(delta)) / (2 * a2)
        x2_label.Text = "x2 = " & solutie2
    Else
        MsgBox("There are no real solutions!", , "Calculator")
    End If
End If
```

### 3.2 Notepad

```
Try
    Dim dlg As SaveFileDialog = New SaveFileDialog
    dlg.Title = "Save"
    dlg.Filter = "Rich Text Files(*.rtf)|*.rtf"
    If dlg.ShowDialog() = System.Windows.Forms.DialogResult.OK Then
        RichTextBox1.SaveFile(dlg.FileName, RichTextBoxStreamType.RichText)
        sem = 1
    End If
Catch ex As Exception :
End Try
```

### 3.3 HTML Editor

```
Try
    Dim dlg As SaveFileDialog = New SaveFileDialog
    dlg.Title = "Save"
    dlg.Filter = "HTML Files(*.html)|*.html"
    If dlg.ShowDialog() = System.Windows.Forms.DialogResult.OK Then
        RichTextBox1.SaveFile(dlg.FileName, RichTextBoxStreamType.PlainText)
        sem = 1
        File_btn.Enabled = True
        Edit_btn.Enabled = True
        Font_btn.Enabled = True
        Help_Btn.Enabled = True
        Minimize_btn.Enabled = True
        Maximize_btn.Enabled = True
        Close_btn.Enabled = True
        RichTextBox1.Visible = True
        Save_warning_panel.Visible = False
    End If
Catch ex As Exception :
End Try
```



### 3.4 Portfolio

```
If fullscreen = False Then
    Dim p As Point
    p.X = Screen.PrimaryScreen.WorkingArea.Left
    p.Y = Screen.PrimaryScreen.WorkingArea.Top
    sx = Me.Width
    sy = Me.Height
    old_loc = Me.Location
    Me.Location = p
    Me.Size = Screen.PrimaryScreen.WorkingArea.Size
    fullscreen = True
Else
    fullscreen = False
    Me.Location = old_loc
    Me.Width = sx
    Me.Height = sy
End If
```

## 4 Conclusion

Calculator and Notepad are two apps which can also be found in Windows, but the ones from GoStudent were adapted and improved with other useful functions (Conversion Degrees → Radians, Calculation of linear and quadratic equation's roots) and a more attractive design.

HTML Editor was initially created for students, therefore it comes with a theoretical support, always there for students. The Pro version is focused on the use of more advanced users, so that it allows working with more files simultaneous (Tabs) and organizing activities and projects for further use.

Portfolio is an original app, meant to be an ease for users who have to create lots of projects and want to organize them into categories.

Each of these apps was created from scratch by the authors with Visual Basic code. The entire concept is simple, dynamic and original, designed especially for students, as it was conceived by students. The graphical elements were also made by the authors one by one, to bring more originality to the project.

## 5 References

- [1] Sandor Kovacs, *Visual Basic*, Editura albastră, 2002
- [2] Pavel Florin Moraru, *Informatică pentru liceu și Bacalaureat*, Editura Donaris, 2012
- [3] <http://invatahtmlake.3x.ro/>
- [4] Cosmin Stoian, Ioana Martin, *GoStudent*, National Conference on Informatics PCID 2015, Editura ULBS, pp. 89-95, Sibiu  
[http://conferences.ulbsibiu.ro/conf.pcid/2015/volum/Volum\\_Conf\\_Nat\\_Inf\\_Elevi\\_PCID\\_2015.pdf](http://conferences.ulbsibiu.ro/conf.pcid/2015/volum/Volum_Conf_Nat_Inf_Elevi_PCID_2015.pdf)

COSMIN STOIAN  
The "B.P.Hasdeu" National College  
Mathematics and Computer Science Area  
No.1, Bd. Garii, Buzau  
ROMANIA  
E-mail: cosmyn9708@gmail.com

IOANA MARTIN  
The "B.P.Hasdeu" National College  
Mathematics and Computer Science Area  
No.1, Bd. Garii, Buzau  
ROMANIA  
E-mail: ioana.martin98@yahoo.com

## Interactive learning using LabVIEW – simple transformations of the ideal gas

Marius-Nicolae Țoc  
Coordinating Teacher: Ramona Humeniuc

### Abstract

Today, multimedia have major implications in learning, particularly in Physics where real-time modeling experiments completes students' knowledge. It is important that laboratory experiments be combined with virtual experiments, thereby deepening thorough transdisciplinary knowledge and skills acquired by students. In this project I used as software of graphical applications in modular programming, the LabVIEW medium in order to simulate, from the topics studied in Physics in the tenth grade, the simple transformations of the ideal gas: isochoric, isobaric and isothermal transformation. I chose LabVIEW because it is a highly productive development environment for creating custom applications that interact with real-world data or signals in fields such as science and engineering.

## 1 Introduction

A simplified version of this paper was published in the proceeding of PCID Conference [7].

Simulation of physical phenomena is useful for studying them or, as teaching material, enabling a better understanding of the phenomena. It has been scientifically proven that supplement the information with visual images helps retain information for a long period of time.

The act of simulating something generally entails knowing the characteristics of the simulated phenomenon, the theory necessary to produce simulation models corresponding to a real system or a process, data collection and analysis, choice of methods and techniques for simulation, using programs for simulation and interpretation of results.

LabVIEW is one of the most powerful and most used softwares based on graphics core language G (graphic language), especially for building applications for control and data acquisition, their analysis and presentation of results.

At the implementation of a LabVIEW program, a programmer has designed its two main components:

- *front panel*, through which the program receives input data and displays the output;
- *block diagram*, component containing the algorithm will then perform the necessary calculations and reasoning for specific information processing program.

LabVIEW is a programming environment used primarily for making measurements and monitoring of automated processes. LabVIEW contains several libraries of predefined functions for acquisition, processing, display and transmission of datas. LabVIEW programs are called virtual instruments (Visual Instruments -VIs) rely on updating concepts of modularity and tree hierarchy. [1]

## 2 Theory - physical and mathematical description of the types of simple transformations of the ideal gas and thermodynamic work

A thermodynamic system is a whole body or a finite macroscopic bodies consist of a large number of microscopic particles (molecules, atoms, ions, electrons, photons, etc.). The simple thermodynamic systems are gaseous because molecules - at long distance from each other - interact very weakly. To study the gas can be used models, the simplest of which is the ideal gas model.

The ideal gas is defined by assigning the following features:

1. It consists of a large number of molecules;
2. The molecules are considered material points;
3. The molecules are constantly in a state of thermal agitation;
4. The molecules do not interact with each other;
5. The collisions of molecules with the walls of the container are perfectly elastic.

Properties 1, 3 and 5 coincide with those of a real gas system, while properties 2 and 4 are approximations of the real system (property 4 is not respected by a real gas if its temperature is low and the pressure is high).

Using kinetic-molecular theory, determine the equation of state of ideal gas heat (also called Clapeyron-Mendeleev equation), and this is the link between state parameters pressure ( $p$ ), volume ( $V$ ) and temperature ( $T$ ) in a state of thermodynamic equilibrium:

$$pV = \nu RT \quad (1)$$

where  $R$  is a universal constant called the ideal gas constant and has the value  $R = 8,314 J \cdot mol^{-1} \cdot K^{-1}$ .

The simple transformation of the ideal gas means a transformation in which one of the state parameters  $p$ ,  $V$  and  $T$  remains constant. In view of this definition, the simple transformations are: isothermal, isobaric and isochoric. [3]

In thermodynamics, work performed by a system represents the energy transferred by the system to another that is accounted for by changes in the external generalized mechanical constraints on the system. As such, thermodynamic work is a generalization of the concept of mechanical work in physics. [6]

Thermodynamic work value ( $L$ ) is greater than 0 if the thermodynamic system performs work to another system and lower than 0 if the thermodynamic system receives work from another system.

### 2.1 Isothermal transformation

The isothermal transformation is the transformation of an ideal gas closed ( $\nu = const.$ ) whose temperature remains constant ( $T = const.$ ). The law of the isothermal transformation is Boyle-Mariotte law: in the isothermal transformation of an ideal gas, its pressure varies inversely with its volume. Mathematical form of Boyle-Mariotte law is:

$$pV = const. \quad (2)$$

This law is plotted in the three coordinate systems  $p$ - $V$  (Clapeyron coordinates),  $p$ - $T$  and  $V$ - $T$ , like in figure no. 1.

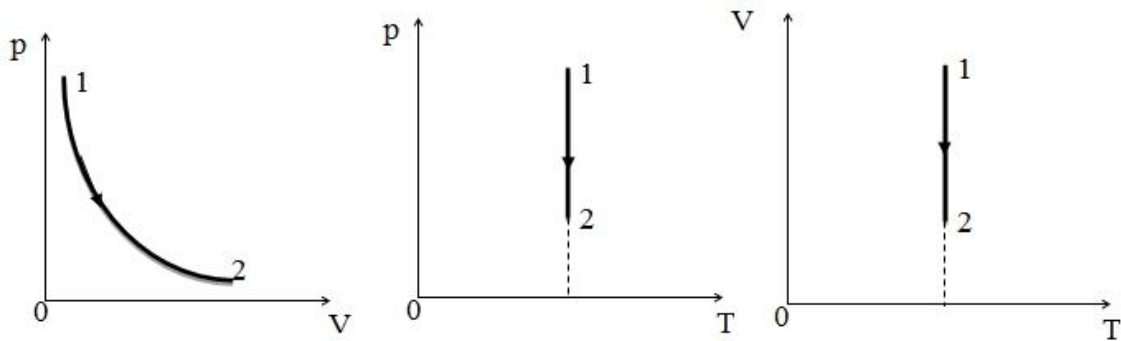


Fig.1: Graphical representation of Boyle-Mariotte law

The outside thermodynamic work (the transition from state 1 to state 2):

$$L_{12} = \int_{V_1}^{V_2} p \cdot \Delta V = p_1 \cdot V_1 \cdot \ln \frac{p_1}{p_2} = \nu \cdot R \cdot T \cdot \ln \frac{p_1}{p_2} \quad (3)$$

## 2.2 Isobaric transformation

The isobaric transformation is the transformation of an ideal gas closed ( $\nu = const.$ ) whose pressure remains constant ( $p = const.$ ). The law of the isobaric transformation is Gay-Lussac law: in the isobaric transformation of an ideal gas, the ratio between its volume and its temperature remains constant. Mathematical form of Gay-Lussac law is:

$$\frac{V}{T} = const. \quad (4)$$

This law is plotted in the three coordinate systems p-V, p-T and V-T, like in figure no. 2.

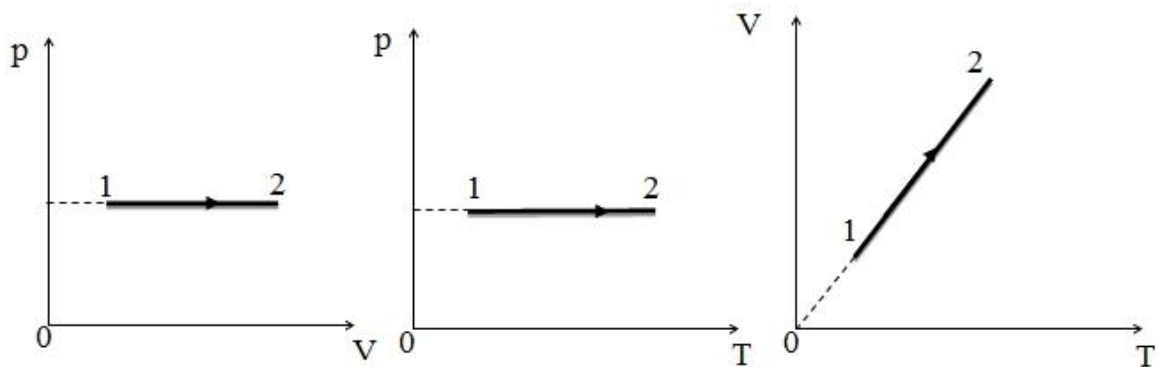


Fig.2: Graphical representation of Gay-Lussac law

The thermodynamic work performed in an isobaric transformation:

$$L = p \cdot \Delta V = p \cdot (V_2 - V_1) = \nu \cdot R \cdot \Delta T \quad (5)$$

## 2.3 Isochoric transformation

The isochoric transformation is the transformation of an ideal gas closed ( $\nu = const.$ ) whose volume remains constant ( $V = const.$ ). The law of the isochoric transformation is Charles' law: in

the isochoric transformation of an ideal gas, the ratio between its pressure and its temperature remains constant. Mathematical form of Charles' law is:

$$\frac{p}{T} = \text{const.} \quad (6)$$

This law is plotted in the three coordinate systems p-V, p-T and V-T, like in figure no. 3.

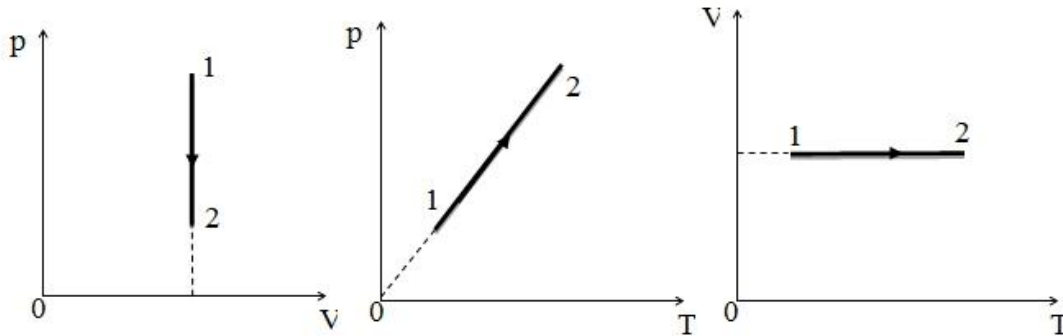


Fig.3: Graphical representation of Charles' law

The thermodynamic work performed in an isochoric transformation is zero because the volume remains constant, so is impossible to produce a variation of volume.

$$L=0 \quad (7)$$

### 3 LabVIEW 2013 simulations solutions

#### 3.1 Isothermal transformation

##### 3.1.1 Graphical user interface

The virtual instrument *Isothermal transformation.vi*.

The Front Panel of this instrument is included in figure no. 4. It includes a numeric control (for determining the amount of substance), a Vertical Pointer Slide control (for determining the temperature), a Tank control (for determining the volume of gas), an indicator Gauge (for pressure displaying), an indicator XY Graph (for viewing the graphical representation of Boyle-Mariotte law) and a Table Control object (for displaying the pressure, the volume and the pV product).

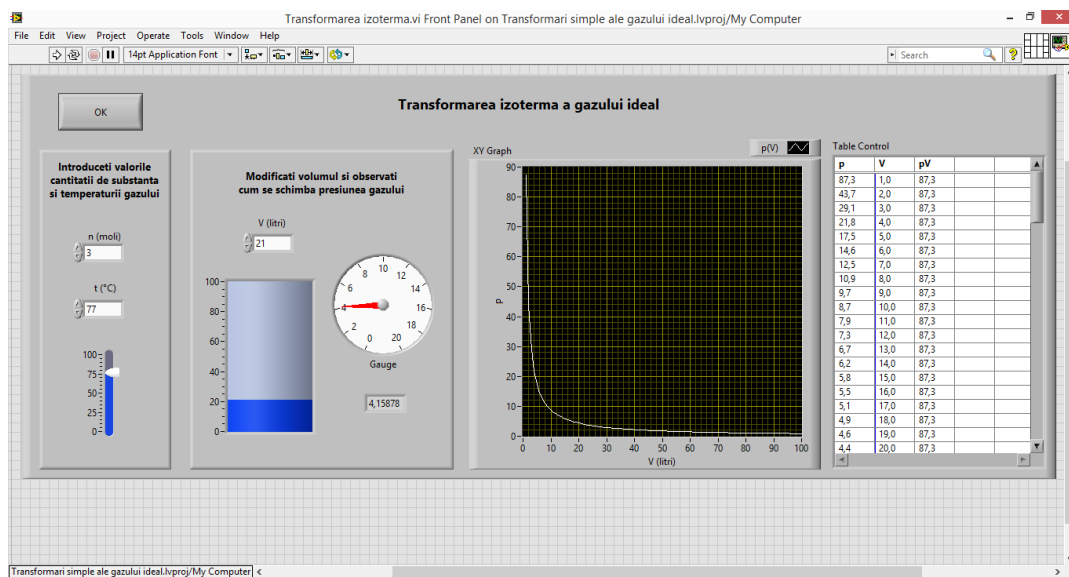


Fig. 4: The front panel of the virtual instrument. Isothermal transformation.vi

### 3.1.2 How to use it

Fix the amount of substance and temperature: change the volume, Gauge indicator showing the variation of the pressure.

Having the quantity of substance and the temperature fixed, it can be seen the graphical representation of Boyle-Mariotte law and the values of pressure, volume and their product, included in the table.

### 3.1.3 Block Diagram

The block diagram of this virtual instrument is included in figure no. 5. It contains a While Loop which allows continuous running of application as long as a boolean value of true is sent to the conditional terminal (Continue if True).

The diagram contains two Formula Nodes and one For Loop. The first Formula Node transforms the value of temperature from Celsius degrees to kelvins (adding 273,15) and calculates the value of current pressure (adequate to the amount of substance, the temperature and the volume fixed by the user).

The second Formula Node with the For Loop are used for generating three 1D arrays which contain the values of volume (between 1 and 100 l), pressure and pV product. These arrays are assembled in a 2D array that, after transposition (using Transpose 2D Array function) and formatting (using Number to Fractional String function), is connected –for displaying– to the Table Control object. Also, the first two enumerated 1D arrays are assembled into a cluster using the Bundle function, it's output being connected to the XY Graph indicator.

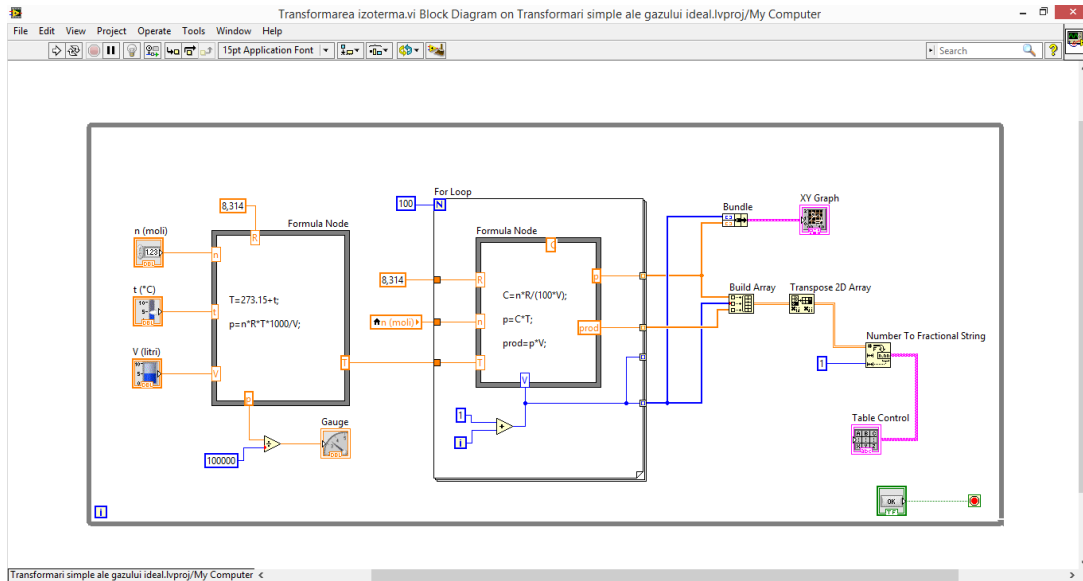


Fig. 5: The block diagram of the virtual instrument. Isothermal transformation.vi

## 3.2 Isobaric transformation

### 3.2.1 Graphical user interface

The virtual instrument *Isobaric transformation.vi*.

The Front Panel of this instrument is included in figure no. 6. It contains a numeric control (for determining the amount of substance), a Gauge control (for determining the pressure), a Tank control (for determining the volume of gas), a Vertical Pointer Slide indicator (for temperature displaying), an indicator XY Graph (for viewing the graphical representation of Gay-Lussac law) and a Table Control object (for displaying the pressure, the volume and the V/T ratio).

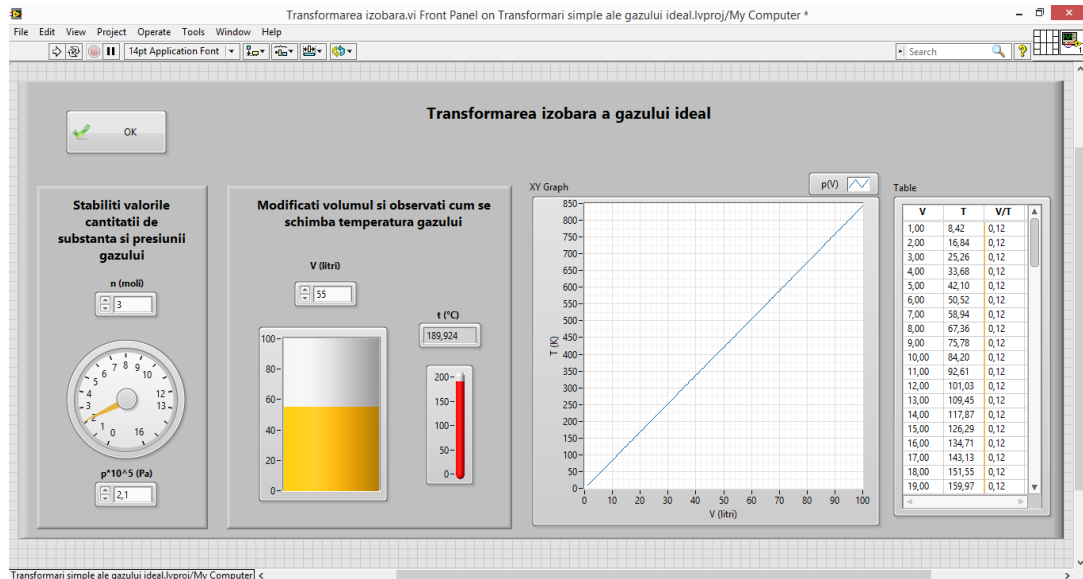


Fig. 6: The front panel of the virtual instrument. Isobaric transformation.vi

### 3.2.2 How to use it

Fix the amount of substance and pressure: change the volume, Vertical Pointer Slide indicator showing the variation of the temperature.

Having the quantity of substance and the pressure fixed, it can be seen the graphical representation of Gay-Lussac law and the values of volume, temperature and their ratio, included in the table.

### 3.2.3 Block Diagram

The block diagram of this virtual instrument is included in figure no. 7. It contains a While Loop which allows continuous running of application as long as a boolean value of true is sent to the conditional terminal (Continue if True).

The diagram contains two Formula Nodes and one For Loop. The first Formula Node transforms the value of pressure in Pa (Pascals), multiplying the value introduced by the user with  $10^5$  and determines the value of current temperature of the gas according to the amount of substance, pressure and volume.

The second Formula Node with the For Loop are used for generating three 1D arrays which contain the values of volume, temperature and V/T ratio. These three 1D arrays are assembled in a 2D array (using Build Array function) that, after transposition (using Transpose 2D Array function) and formatting (using Number to Fractional String function), is displayed with Table Control object. Also, the first two mentioned 1D arrays are assembled into a cluster using the Bundle function, it's output being connected to the XY Graph indicator for viewing the graphical representation of Gay-Lussac law.

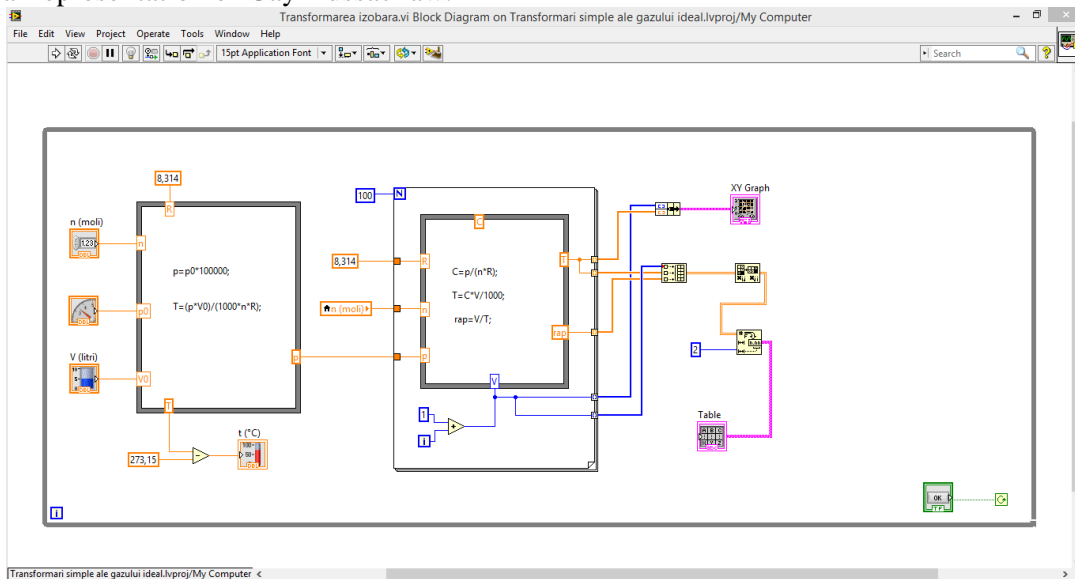


Fig. 7: The block diagram of the virtual instrument. *Isobaric transformation.vi*

## 3.3 Isochoric transformation

### 3.3.1 Graphical user interface

The virtual instrument *Isochoric transformation.vi*.

The Front Panel of this instrument is included in figure no. 8. It contains a numeric control (for determining the amount of substance), a Tank control (for determining the volume of gas), a Vertical Pointer Slide control (for determining the temperature), a Gauge indicator (for displaying current pressure), an XY Graph indicator (for viewing the graphical representation of



Charles' law) and a Table Control indicator (for displaying the pressure, the volume and the ratio between p and T).

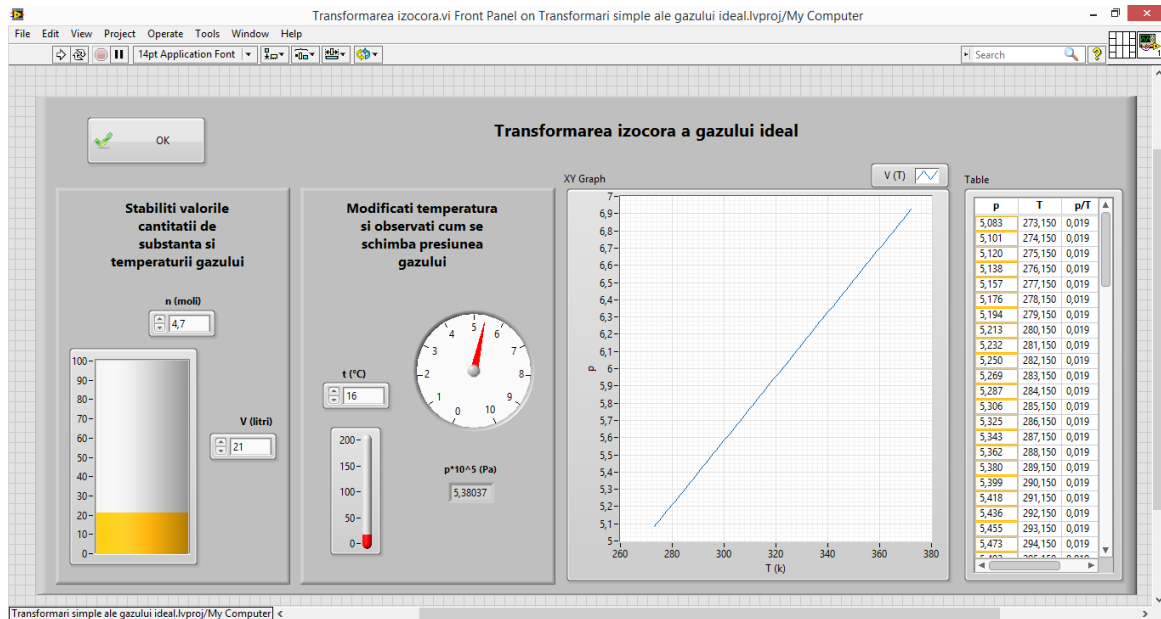


Fig. 8: The front panel of the virtual instrument. Isochoric transformation.vi

### 3.3.2 How to use it

Fix the amount of substance and volume: change the temperature, Gauge indicator showing the variation of the pressure.

Having the quantity of substance and the volume fixed, it can be seen the graphical representation of Charles' law and the values of pressure, temperature and their ratio, included in the table.

### 3.3.3 Block Diagram

The block diagram of this virtual instrument is included in figure no. 9. It contains a While Loop which allows continuous running of application as long as a boolean value of true is sent to the conditional terminal (Continue if True).

The diagram contains two Formula Nodes and one For Loop. The first Formula Node transforms the value of temperature from Celsius degrees to kelvins and determines the value of current pressure of the gas according to the amount of substance, volume and temperature.

The second Formula Node with the For Loop are used for generating three 1D arrays which contain the values of temperature, pressure and p/T ratio. These three 1D arrays are assembled in a 2D array using Build Array function. The new array, after transposition (using Transpose 2D Array function) and formatting (using Number to Fractional String function), is displayed with Table Control object. Also, the first two mentioned 1D arrays are assembled into a cluster using the Bundle function. It's output is connected to the XY Graph indicator for viewing the graphical representation of Charles' law.

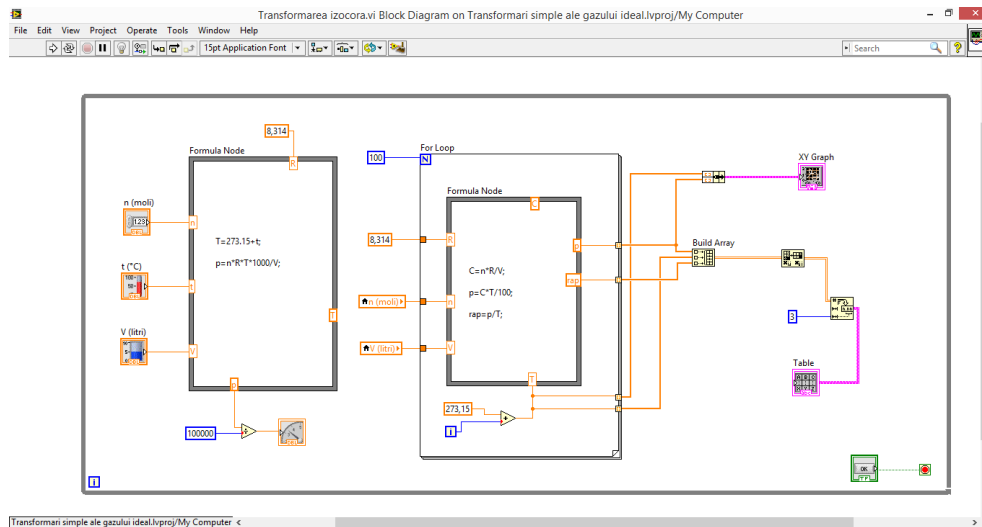


Fig. 9: The block diagram of the virtual instrument. Isochoric transformation.vi

### 3.3 Calculation of the thermodynamic work performed during processing

#### 3.3.1 Graphical user interface

The virtual instrument *Thermodynamic works of the transformations.vi*.

The Front Panel of this instrument is included in figure no. 10. It includes 2 clusters, for isothermal transformation and isobaric transformation (thermodynamic work for isochoric transformation is 0). The first cluster (for isothermal transformation) contains a numeric control (for determining the amount of substance), a Vertical Pointer Slide control (for determining the temperature), a Tank control (for determining the volume of gas), and two Gauge controls (for determining the initial and the final pressure). The second cluster (for isobaric transformation) contains a Gauge control (for determining the pressure of gas) and two Tank controls (for determining the initial and the final volume of the gas). Also, there are two numeric indicators for displaying the thermodynamic works of the transformations.

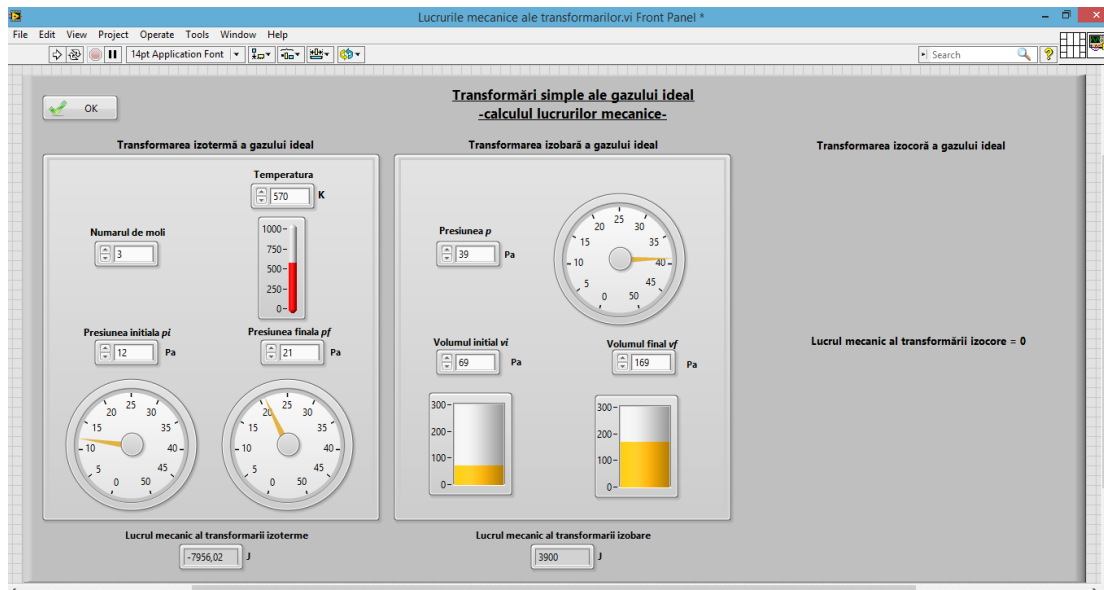


Fig. 10: The front panel of the virtual instrument. Thermodynamic works of the transformations.vi

### 3.3.2 How to use it

Fix the values and see the thermodynamic work calculated for each transformation.

### 3.3.3 Block Diagram

The block diagram of this virtual instrument is included in figure no. 9. It contains a While Loop which allows continuous running of application as long as a boolean value of true is sent to the conditional terminal (Continue if True).

The diagram contains two Unbundle functions which unbundle the datas from clusters, a Division function which calculates the ratio between  $p_i$  and  $p_f$ , a Natural Logarithm function which computes the base e natural logarithm of that ratio and a Formula Node which calculates  $L$ . It also contains a Substraction and a Multiplication function. Two numeric indicators display the results.

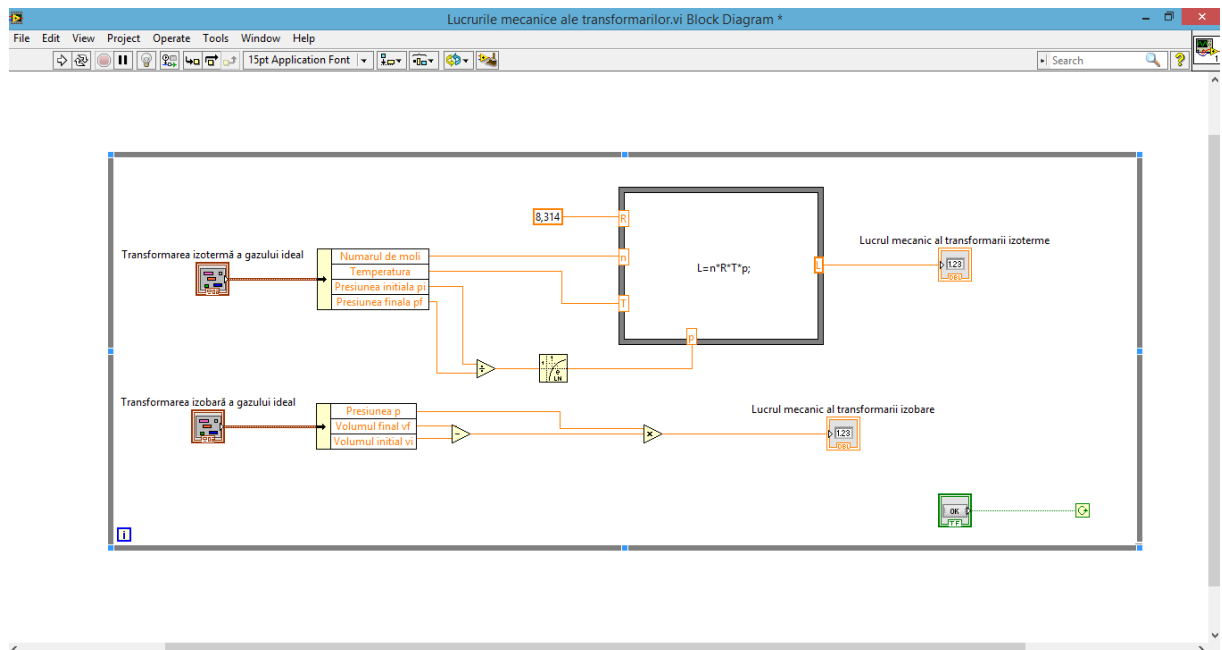


Fig. 11: The block diagram of the virtual instrument. Thermodynamic works of the transformations.vi

## 4 Conclusions

I chose to use LabVIEW Virtual Instrumentation for simple transformation of the ideal gas for several reasons: enhanced flexibility of the configurations of measuring systems, high reliability of the components, increased accuracy of information processing, the capability to perform more complex processings and a greatly reduced cost compared with conventional devices.

Regarding future developments, I am working on the virtual instruments for the adiabatic transformation ( $Q=0$ , where  $Q$  represents the heat) and the graphical interpretations of this transformation's thermodynamic work, using mathematical modelling of Poisson's equation.

Also, in developing is the implementation of Remote Panels function of this software for these virtual instruments. This function allows to operate the front panel on a machine that is separate from where the VI resides and executes, within a web page for example.

## References

- [1] T. Anghel, *LabVIEW Interactive simulations with applications in physics*, Blue Publishing, Cluj-Napoca, 2010.
- [2] L. Arsenoiu and T. Savu, *LabVIEW Programming fundamentals*, PrinTech Publishing, Bucharest, 1998.
- [3] C. Gherbanovschi and N. Gherbanovschi, *Physics: Manual for 10<sup>th</sup> grade*, Niculescu ABC Publishing, Bucharest, 2004.
- [4] Y. Yang, *LabVIEW Graphical Programming Cookbook*, Ebook Pack Publishing, 2013.
- [5] <http://www.ni.com/labview/>
- [6] <http://www.wikipedia.org>
- [7] M. N. Țoc, *Simple transformations of the ideal gas – LabVIEW simulations*, The papers of National conference of informatics for students “Programming, Communication, Imagination, Design” (PCID), Ist Edition, Sibiu, 2015

ȚOC MARIUS-NICOLAE  
”Horea, Cloșca și Crișan” National College of Alba Iulia  
Mathematics-Informatics intensively Informatics  
No. 11, 1 Decembrie 1918 Avenue, Alba Iulia  
ROMANIA  
E-mail: toc.marius@gmail.com

## Handwritten digit recognition algorithm

Eva Tuba

Teacher Coordinator: Milan Tuba

### Abstract

Higher level of image processing usually contain some kind of recognition. Digit recognition is common in applications and handwritten digit recognition is an important subfield. Handwritten digits are characterised by large variations so template matching, in general not very efficient, is even less appropriate. In this paper we describe an algorithm and application for handwritten digit recognition based on projections histograms. Classification is facilitated by artificial neural network.

## 1 Introduction

Digital images have been introduced recently but very quickly become present almost anywhere. Digital image processing is nowadays one of the most used techniques [1]. It consists of many different algorithms, some of them for low level image conditioning and some of them for higher level image understanding. One of the common goals of higher level processing is to recognize different objects in the image. Handwritten digit recognition is one such application used for various purposes. It is used, for example, in post offices for mail sorting, in banks for check processing, form data entry etc. Handwritten digit recognition requires accuracy and speed. Many algorithms have developed for solving this problem, but it is still interesting enough topic for researchers to search for algorithms that provide better results.

This paper is organised as follows. Section 2 presents two main theories for handwritten digit recognition. After that, in Section 3, problem of classification and usage of artificial neural networks for classification will be explained. At the end, in Section 4 there is a detailed description of proposed algorithm followed by experimental results.

## 2 Handwritten digit recognition

Digit recognition has many purposes. One of more used purposes is recognition of licence plates on cars [2], [3]. Licence plate digits are all of same shape and with clearly written symbols. That means that we could use template matching technique, but only under condition that licence plate is always recorded under same angle or, at least, under limited number of angles. Handwritten digit recognition is much bigger problem. There is huge number of different writing styles, angles, pencil thicknesses etc. For their recognition we need different algorithms.

## 2.1 Template matching

The first idea of the algorithm for object recognition is to simulate the process of humans for this task. Humans use for object recognition and their identification a process that is called template matching [4]. When we try to identify some object or a person, we automatically compare it with stored information based on our past experiences. This process is very quick for humans and often we are not even aware that knowing what we are looking at is the result of any process at all. Even though this is a very easy and natural process for us, it has various problems when it comes to simulating it.

In order to recognize a handwritten digit using template matching technique, handwritten digit is compared with templates stored in memory. If the good match is found, digit is recognized based on that match. For example, we are looking at pattern that has two circles, one above the other connected in one point. Looking for a good match by comparing this pattern with various templates should be done when the pattern is compared to the template for digit eight. Pattern would be recognized and identified as digit eight. However, template matching has problem with templates variability. In Fig. 1 digit eight written by different people are presented. Every digit from Fig. 1 needs a matching template. Similarly, numerous variations in size and orientation require their templates. The conclusion is that for just one digit we need a very large number of templates (templates for variations in size, orientation, form, style etc.) and that would certainly lead to crossing to templates for some other digit.



Fig. 1: Examples of handwritten digit eight

## 2.1 Feature detection

In order to accommodate variability of patterns new theory was developed, theory of feature detection. Each type of pattern is stored as set of its features. Number of features is determined and object that we want to recognize and identify is broken down into its component features. These features are compared with stored patterns features and is identified according to the best match, the greatest feature overlap. For example, for digit four stored set of features can be two vertical lines, one horizontal line.

Performance of algorithms that use feature detection theory depend on the choices of features. In this paper will present and compare results for few different set of features.

## 3 Classification

The problem of classification consists of classifying unknown instance into one of the offered categories – classes. Each instance (in our case handwritten digit) is represented by some selected set of its features. Also, each instance can have a feature that represents the class to which the instance belongs. Classification consists of determining the value of the feature for the class based on the remaining features of the instance. Classification is one of the most common task of machine learning.

Commonly used models for classification are support vector machines (SVMs) [5], [6], [7]. SVMs are supervised learning models. Learning algorithm is associated with SVM and then this algorithm analyzes data and recognizes patterns.

### 3.1 Neural networks

Artificial neural networks (ANN) are one of the algorithms for machine learning. Machine learning is a field of artificial intelligence. One definition could be that it is a scientific discipline that deals with the construction of adaptive computing systems that are capable of improving their performance using information from the experience. Machine learning can be also defined as the discipline that studies the generalization and construction and analysis algorithms that generalize. Artificial neural networks, commonly referred to as neural network (NN), are a mathematical model or computer developed model. First idea for ANN was to try to mimic the structure and functional aspects of biological neural networks, but with time this idea was abandoned.

Artificial neural network consists of interconnected artificial neurons and processes information using computing approach called connectionism. In most cases, ANN is a flexible system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern NNs are nonlinear modelling tools for statistical data, usually used for modelling complex relations between inputs and outputs or to find patterns in the data.

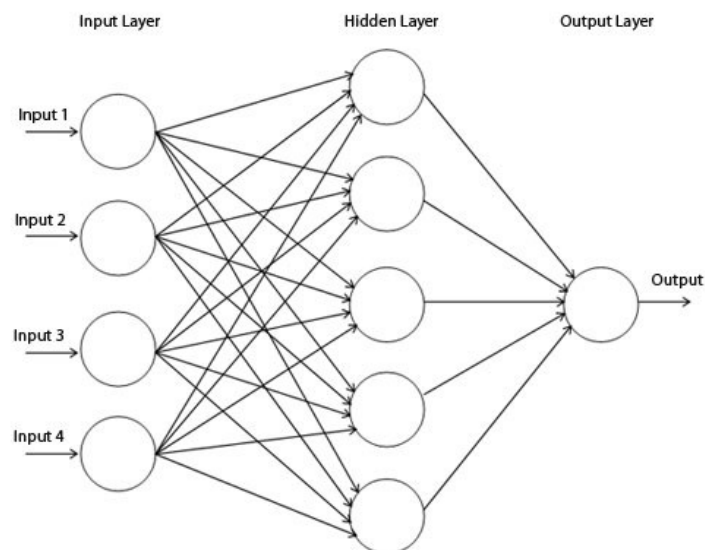


Fig.2: Artificial neural network

## 4 Our proposed algorithm

In this paper images will be represented as few different sets of features. Results of accuracy for different sets will be presented. Software needed for extracting features is written in C#. For input image, software generate output file that contains features. This output file is input file for ANN.

Algorithms for handwritten digit recognition (and object character recognition) have three steps. First step is to prepare image for processing. This pre-processing faze is very important for algorithm. This step should remove irrelative data and data that could have negative influence on recognition. Usual steps in this faze are binarization, normalization, smoothing and denoising. Binarization is the process that convert image into black and white image (pixels can have one of two values, 0 or 1). Normalization is scaling technique. The aim of smoothing is to suppress image noise. Smoothing removes salt and pepper, but also smooth the edges. Denoising is usually done by media filter. It completely remove noise and it does not blur the edges.

Handwritten digit recognition is very important problem in object character recognition because of earlier named reasons, but also it is used as test case for different theories for pattern recognition and algorithms for machine learning. For these reasons, a few standard databases of handwritten digits images are made for testing. In those databases images are pre-processed. One of well-known database is CENPARMI, MNIST and CEDAR [8], [9].

MNIST database is used for testing the proposal algorithm, so the algorithm does not include pre-processing step [10], [11]. Today, MNIST is standard database for testing machine learning algorithms. MNIST database was made from NIST special databases SD3 and SD7. Those two databases were made for purpose of training and testing for the First Census OCR Systems Conference. Those two databases were collected from completely different sources and that was problem for machine learning. Results of very good algorithm could be very poor because of that reason. MNIST database is created in order to overcome this problem. MNIST database contains 60,000 images for training and 10,000 images for testing. Images in this database are white digit with black background.

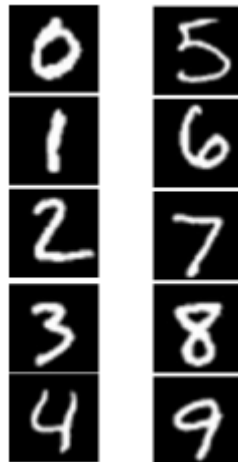


Fig. 3: Example of MNIST database

## 4.2 Feature extraction

Feature extraction is the second step in algorithm for handwritten digit recognition. This step is crucial for performance of the algorithm. The accuracy of classification will depend on how well the features are selected and extracted. As result images are represented as set of features.

In our algorithm we will represent every digit as a projection histogram. Projection histogram represents number of pixels that belong to digit per axis. As first axis we took axis  $x=0$ . On Fig. 4 are shown histograms for all 10 digits.



Fig. 4: Projection histogram ( $x=0$ )

From histograms we can see that some histograms have more elements (which means that original picture has more white pixels) than others. For example, number one has less elements than all other numbers, while numbers five, eight and nine has the most elements. This characteristic can be used as feature of the number we are trying to recognise, but in that case the thickness of a pencil would play a major role. Aside total number of elements we can see that digits are differentiated by number of pixels in columns too. We can take digits zero, three and eight as examples. On the Fig. 5 are presented histograms for those numbers. Zero has less pixels in the middle than at the sides. For number eight histogram is reversed, it has more pixels at the sides than in the middle. Number three is like the number eight which was cut part of the left side. Histograms for this two numbers are similar, but number three has less pixels at the left side.





Fig. 5: Histograms of projection on x-axis



Fig. 6: Histogram of projection on y-axis

If these data take as input for ANN, we will get results shown in Table 1. Our algorithm was tested under limited number of images and classification is done by artificial neural networks. For this task we are using JustNN software. Results that we get are not good enough. From ten digits only 6 are correctly classified. Next step is to calculate histograms for  $y$  axis as seen on Figure 7.

Table 1: Results of classification ( $x=0$ )

Picture	0	1	2	3	4	5	6	7	8	9
Output	6	1	2	8	4	5	5	7	8	3

Like in the previous case, some of the histograms are different, but there are some that are similar. Histograms for digits three and eight that were very similar are now clearly differentiated, but digits eight and zero still look alike. Also histograms for digits five and three are similar. They have three peaks, two on the ends and one in the middle, but in case of digit 5 difference between peak and bottom is bigger. If we use histogram like this for input in ANN we get results shown in Table 2. Classification like this did not yield better results than last one.

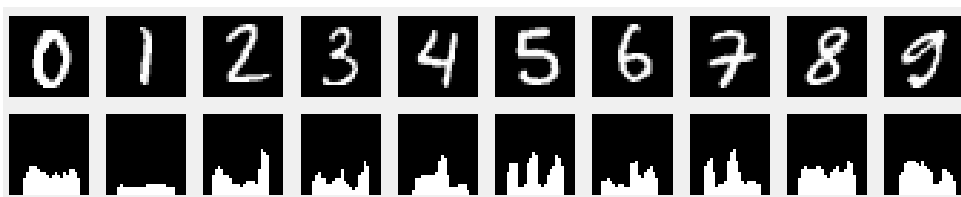


Fig. 7: Projection histogram ( $y=0$ )

Table 2: Results of classification ( $y=0$ )

Picture	0	1	2	3	4	5	6	7	8	9
Output	4	1	2	2	4	3	6	1	8	9

Next idea is to feed both histograms in input of ANN. Results are given in Table 3. Combination of both histograms gave us much better results.

Table 3: Results of classification ( $x=0$  and  $y=0$ )

Picture	0	1	2	3	4	5	6	7	8	9
Output	0	1	2	8	4	5	6	1	8	9

## 5 Conclusion

Projections on other axis are easy to implement. It is possible that better results could be produced if we do projection on some other axis,  $y=-x$  for example or with combination of some other projections. In future work we will test those options. Some parameters of histogram had more influence on learning process than other. If we add weight coefficients to the parameters, it is very likely that we will get much better results during classification. In this algorithm we ignored parameters that are the same for all digits (usually beginning or ending parameters). Also it is possible to ignore some of the parameters less important, or even have negative impact on learning process. Like it is shown, this method can be used for starting phase in handwritten digit recognition algorithm, so it is possible to build on this method with further process of recognition.

## References

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, New Jersey, 2002.
- [2] H. Caner and H. S. Gecim. *Efficient Embedded Neural-Network-Based License Plate Recognition System*. IEEE Transactions on Vehicular Technology, Vol. 57, No.5, 2675-2683, 2008.
- [3] R. Juntanasub, N. Sureerattanan. *Car license plate recognition through Hausdorff distance technique*. 17th IEEE International Conference on Tools with Artificial Intelligence ICTAI 05, 2005.
- [4] S. Abdleazeem and E. El-Sherif. Arabic handwritten digit recognition. *International Journal of Document Analysis and Recognition (IJDAR)*, Vol. 11, No. 3, 126-141, 2008.
- [5] T. Loo-Nin and L. Kia-Foxk. Robust vision-based features and classification schemes for off-line handwritten digit recognition. *Pattern Recognition*, Vol. 35, No. 11, 2344-2364, 2002.
- [6] T. Yamaguchi, M. Maruyama, H. Miyao and Y. Nakano. Digit recognition in a natural scene with skew and slant normalization. *International Journal of Document Analysis and Recognition (IJDAR)*, Vol. 7, No. 2-3, 168-177, 2005.
- [7] J. Heikkonen and M. Mntynen. A computer vision approach to digit recognition on pulp bales. *Pattern Recognition Letters*. Vol. 17, No. 4, 413-419, 1996.
- [8] C.Y. Suen, K. Liu and N.W. Strathy. *Sorting and recognizing cheques and financial documents*, Document Analysis Systems: Theory and Practice, Springer, Berlin, 173–187, 1999.
- [9] S. Belongie, J. Malik and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, Vol. 4, 509–522, 2002.
- [10] Jukka H J. X. Dong, A. Krzyzak and C. Y. Suen. A multi-net learning framework for pattern recognition. *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, Seattle, 328–332, 2001.
- [11] M. Kang and D. Palmer-Brown. *A model learning adaptive function neural network applied to handwritten digit recognition*. Information Sciences, Vol. 178, No. 20, 3415-3429, 2007.

Eva TUBA  
 University of Belgrade  
 Faculty of Mathematics  
 Studentski trg 16, Belgrade  
 SERBIA  
 E-mail: eva.tuba@gmail.com

# JPEG algorithm compression adjustment

Ira Tuba

Teacher Coordinator: Milan Tuba

## Abstract

This paper describes JPEG algorithm with a focus on quantization. JPEG algorithm uses discrete cosine transform on 8x8 blocks of the image to transform light intensity values to frequency coefficients. Main compression is done by discarding less important coefficients. This is enabled by integer division of DCT coefficients with corresponding values from the quantization matrix. After that process many coefficients are rounded to zero. We have developed application that allows to manipulate with a level of compression by choosing the values in quantization table.

## 1 Introduction

Digital images are part of our everyday life. They are used in journalism, medicine, in police investigations, archaeology etc. Storing them have many benefits such as faster search through database and easy processing [1], [2], [3]. An important feature of digital images is that they can be easily processed by mathematical methods [5], [6]. However, one of the problems with digital images is memory needed for saving them. A representation of digital image can require even tens of megabytes. The solution for this problem is compression. Lossy compression algorithms provide very high degree of compression and cause minimal quality loss.

Well known and widely used lossy compression algorithm is JPEG (Joint photographic experts group). Degree of compression as well as quality of digital image is determined with quantization matrix.

The software for JPEG algorithm compression adjustment is developed in C#. Application has graphic user interface and it is very easy to work with it.

The next two section of this paper describe digital images and JPEG algorithm for compression. In section 4 will be described application and shown some results of different compression rates.

## 2 Digital images

Digital image represents the projection of 3D world into a 2D rectangle. The rectangle is double discretized, horizontally and vertically, and the result is the rectangle divided on small rectangles, usually squares. This squares are named pixels. Resolution is the number of horizontal and vertical pixels. Digitized images are also called raster images.

Raster image saves information about brightness and colour (for colour image) for each pixel. For black and white images, one bit is needed for each pixel (0 – white, 1 – black). With  $n$  bits,  $2^n$  shades can be described. RGB is common color model for digital images. Since human eye has receptors for red, green and blue, RGB model corresponds to human eye naturally. In

this color model, each pixel is represented with three numbers (shades of red, green and blue component). Color depth is the number of bits needed for describing the color per pixel. Main characteristic that make compression possible is fact that image is not random written numbers. Adjacent pixels have similar values. Compression is possible because of redundancy of the images.

### 3 JPEG algorithm

In the JPEG algorithm we can choose the level of compression. The higher the compression is, the lower the image quality is, and conversely.

The first step of the algorithm is block preparation. RGB model is the most often used color model, but for image processing there are more appropriate models, such as, for example, YCbCr. Component Y represents intensity, while Cb and Cr are hue (chrome) components.

As we mention earlier, human eye is less sensitive on shades of colors, so we will reduce color components. Blocks of four pixels will be replaced with their average value. This will reduce the size of the matrices for those components. We get a 50% reduced file size. This step is irreversible, we can not reconstruct the original image.

Next step is dividing the image into 8x8 matrices. Then, two-dimension discrete cosine transformation (DCT) is applied on each block of the image. DCT is a Fourier-related transform appropriate for image processing.

Definition of DCT is described with Eq. 1:

$$D(i, j) = \frac{C(i) * C(j)}{4} \left[ \sum_{x=0}^7 \sum_{y=0}^7 d(x, y) * \cos \frac{(2x+1) * i * \pi}{16} * \cos \frac{(2y+1) * j * \pi}{16} \right] \quad (1)$$

Invers transform (IDCT) is given by Eq. 2:

$$d(x, y) = \frac{C(x) * C(y)}{4} \left[ \sum_{i=0}^7 \sum_{j=0}^7 D(i, j) * \cos \frac{(2x+1) * x * \pi}{16} * \cos \frac{(2y+1) * y * \pi}{16} \right] \quad (2)$$

where C(u) are constants:

$$C(u) = \frac{1}{\sqrt{2}} \text{ for } u=0, \text{ and } C(u)=1 \text{ for } u>0.$$

DCT coefficients contain information about the composition of the image frequency. The first coefficient is named DC component. It corresponds to average intensity values of 8x8 matrices and also contains the most of the image information. The other 63 coefficients are called AC components. DC component has the lowest frequency and coefficients closer to lower-right corner has higher frequencies which are usually close to zero.

Matrix form of Eq. 2 is obtained from Eq.3:

$$T_{i, j} = \begin{cases} \frac{1}{\sqrt{N}}, & \text{if } i = 0 \\ \sqrt{\frac{2}{N} \cos \left[ \frac{(2j+1) * i * \pi}{2N} \right]}, & \text{if } i > 0 \end{cases} \quad (3)$$

For N=8, it results matrix T:

$$T = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4175 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 \\ .2778 & -.4904 & .0975 & .4175 & -.4175 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix}$$

Matrix T is an orthogonal matrix which means that its inverse matrix is T'. Now we can apply DCT on matrix M (block of the image previously reduced on range [-128, 127]).

$$D = TMT' \tag{4}$$

This step is reversible, i.e. applying the invers DCT obtain a starting matrix M. Some errors would appear as a result of the rounding of a real value to the nearest integer.

### 3.1 Quantization

Quantization is the most important part of compression. It is conducted by quantization tables. Values in this tables are pre-defined and determined according to human visual system. Quantization is defined as the integer division of each DCT coefficient with the corresponding coefficient of the quantization table (Eq. 5).

$$B(i, j) = \text{round}\left(\frac{D(i, j)}{Q(i, j)}\right) \text{ for } i = 0, 1, \dots, 7; k = 0, 1, \dots, 7 \tag{5}$$

Experiment based on human visual system have resulted in the JPEG standard quantization matrix which has a quality level 50 [4].

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 130 & 99 \end{bmatrix}$$

After quantization, procedure follows a zigzag analysing of 63 AC coefficients for each block. At the output of the quantization are obtained matrices with nonzero elements in the upper left corner, while the other elements are equal to zero. Using zigzag analyses we get one-dimensional matrix or vector. The first element is always the DC component, followed by AC components different from zero. At the end of the series appears AC coefficients equal to zero.

Entropy coding is a special form of lossless compression, and the last step of the JPEG compression. This is encoding based on static values of quantized coefficients. The JPEG standard specifies two ways of coding: Huffman's coding and arithmetical coding. The idea of Huffman's coding is to numbers that appear often represent with shorter code than those that appear less often.

Arithmetic coding gives a 5-10% better compression than Huffman's coding, but it is much more complex .

To decode some image we need to implement all the previously mentioned steps in the reverse order. As a result of decoding we get decompressed JPEG version of the image.

The first step is converting Huffman codes in the sequence. Then symbols from sequence are extended to one 64 element long array of DCT coefficients for each pixel 8x8 blocks. Every quantized DCT coefficient needs to be multiplied with corresponding value from quantization table. After that coefficients are compiled from the "zigzag" sequence to original sequence. Last step is applying inverse DCT which will reconstruct image samples.

## 4 Compression adjustment

Degree of compression and the level of losses is determined by the quantization matrix. As the result of dividing each coefficient in the frequency domain by a corresponding value in the quantization table many coefficients are rounded to zero and many of the others are small numbers, so we can represent 8x8 matrix with many fewer bits. By choosing quantization table we can manipulate with the level of compression.

If a number in quantization matrix is bigger than 128, the result after quantization can be 0 or 1, so we need only one bit to encode that coefficient. If a number in quantization matrix is between 64 and 127, the result after quantization can be 0, 1, 2 or 3, i.e. we need only two bits to encode that coefficient, etc.

In this application under the original image are two differently compressed images (Fig. 1). Above them are shown quantization matrices which can be changed by user. This matrices are used for compression. Images are in bitmap format size 512x512.

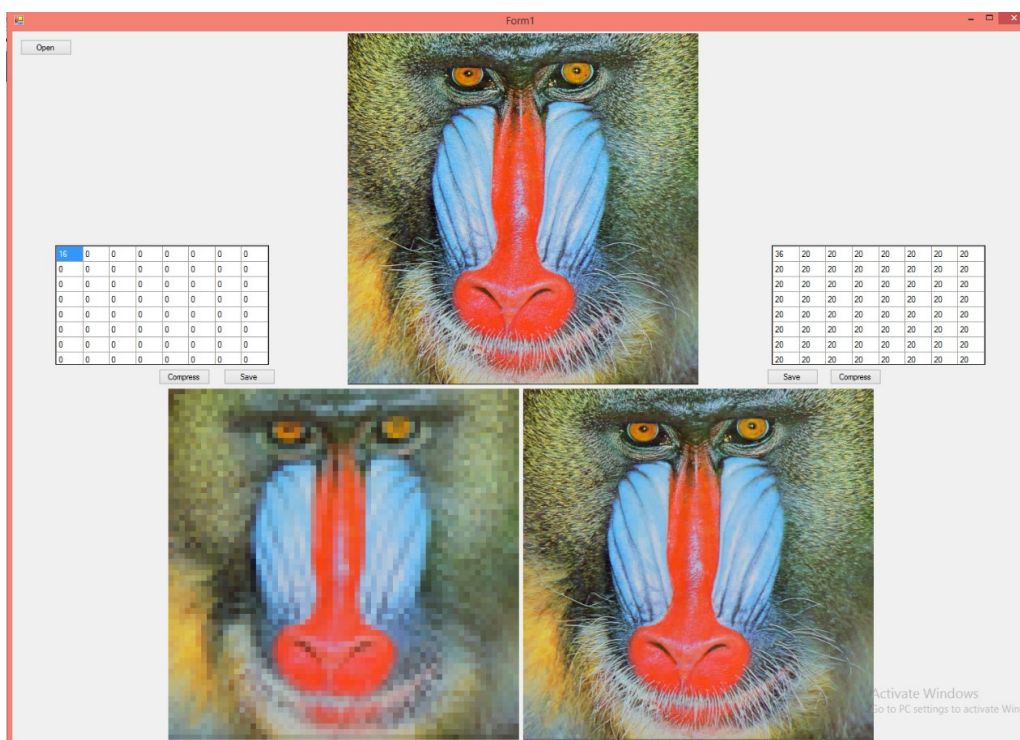


Fig. 1: Screen shot of the application

Interesting cases for our experiments are the ones with very high degree of compression. High degree of compression is reached by excluding the majority of the coefficients and to exclude a coefficient its corresponding value in the quantization matrix must be 255. On the other



hand, if a coefficient needs to be preserved its quantization value will be 1. We can also reduce the number of bits required to record one coefficient by dividing it with a number between 1 and 255.

For the first example we will take an extreme case of compression. Quantization table is set so after dividing each DCT coefficient with corresponding value from quantization matrix DC component stays the same and the others are rounded to zero. Now, the image will be recorded with 8 bits. The result is 64:1 compression. Despite of high level of compression the image is still recognizable, but the resolution is lower.



Fig. 2: The original

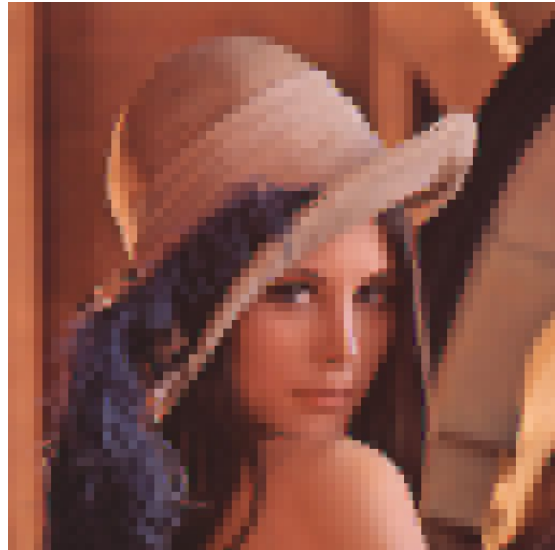


Fig. 3: Compression 64:1

By modifying previous example we can get even higher degree of compression. The first coefficient in quantization table will not be 1, but, for example, a power of 2. If DC component is divided with  $2^5$ , it can be represented with 3 bits and level of compression is 128:1 (Figure 4). If we choose  $2^7$  for corresponding quantization value, first coefficient will require only 1 bit and we get compression in ratio 512:1 (Figure 5). With this levels of compression image quality is poor but still acceptable.

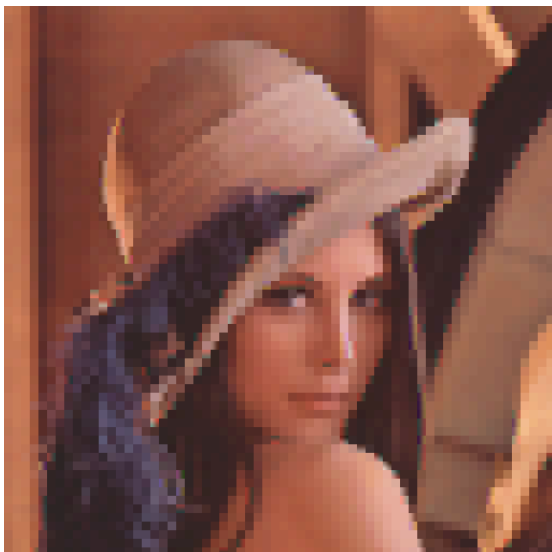


Fig. 4: Compression 128:1

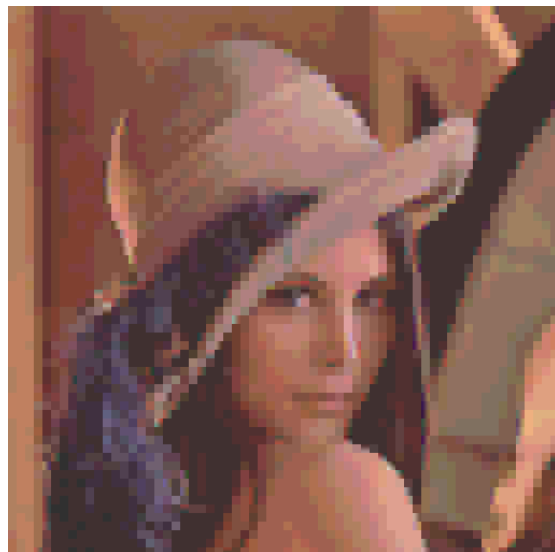


Fig. 5: Compression 512:1

It would be interesting to show a picture that is also recorded with 8 bits, but they are differently arranged. Lower frequencies contain important information for the image and they are placed in upper-left corner, so our 8 bits will be in that area. Each coefficient in  $2 \times 2$  matrix located in mentioned corner will be divided with 128. The result is much better image quality for the same level of compression (Figure 6).

Now we will represent image with first 4 frequency components kept as they are (divided with 1) and compare that image with previous image. Degree of compression is lower but the resulting image is much better and very close to the original image (Figure 7).

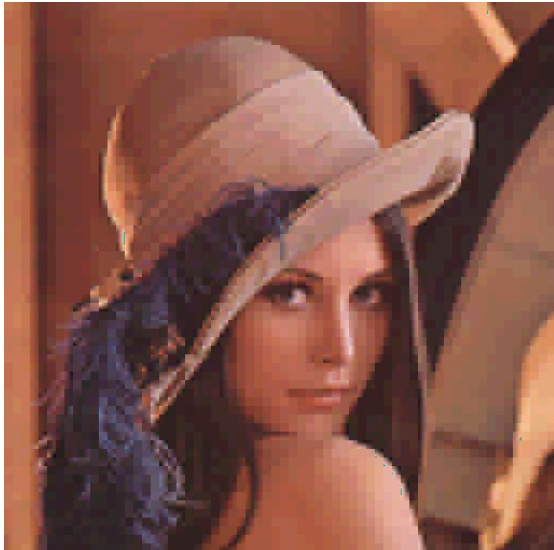


Fig. 6: Compression 64:1

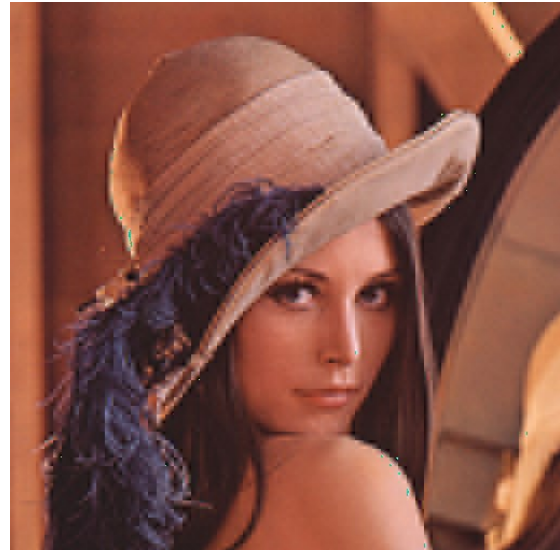


Fig. 7: Compression 16:1

For the next experiment we will keep matrix  $3 \times 3$  located in upper-left corner but coefficients will be divided on following way: DC component is divided with 32 and the others with 128, so the total number of bits required for this image is 16. The quality of the resulting image is minor than with compression from the preceding example (Figure 8). If we try to alleviate this compression we will keep entire first coefficient. Now, 24 bits is needed (8 more) but image quality is no better (Figure 9).

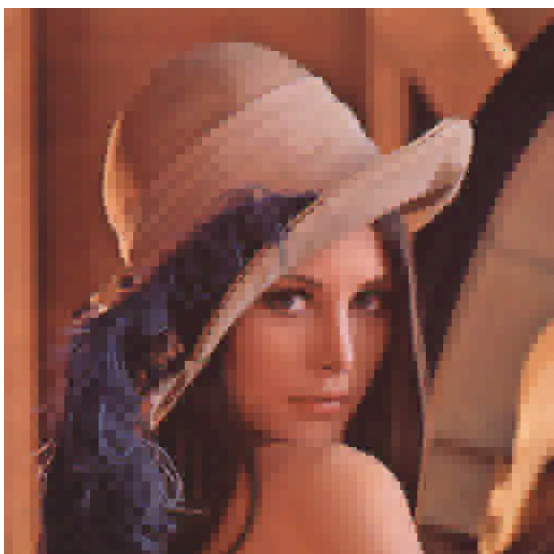


Fig. 8: Compression 32:1

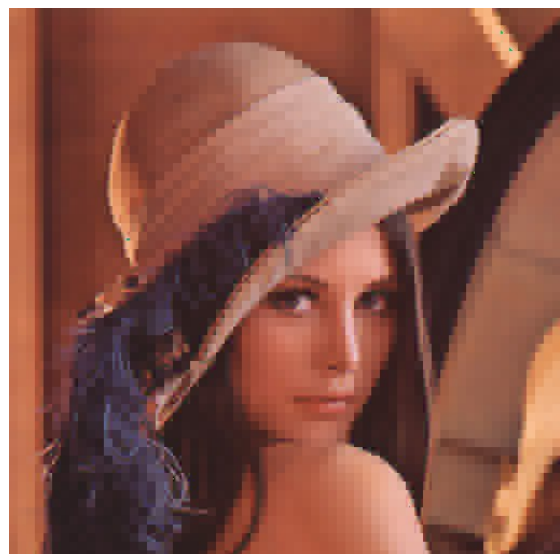


Fig. 9: Compression 64:3



## 5 Conclusion

This application is useful for comparison of two images compressed with different quantization matrix. Depending on how we change quantization values we can see various changes on the image and draw conclusions.

In further development we can extract quantization matrices that are proved as “good”. Quantization matrix is considered as “good” if it has high level of compression but not much losses on image quality. To estimate how good the matrix is we can develop different metrics as, for example, relation between numbers of bits required for compressed image and deviations of that image from the original.

## References

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, New Jersey, 2002.
- [2] W.K. Pratt, *Digital Image Processing*, PixelSoft, Los Altos, CA, 2007.
- [3] S. Jayaraman, S. Esakkirajan and T. Veerakumar. *Digital Image Processing*. TMH, New Delhi, 2010.
- [4] C. Y. Wang, S. M. Lee and L. W. Chang. *Designing JPEG quantization tables based on human visual system*, Signal Processing: Image Communication 16, 501-506, 2001.
- [5] R. Szelinski, *Computer Vision Algorithms and Applications*, Springer-Verlag, London, 812, 2011.
- [6] W. Burger, M. J. Burge, *Principles of Digital Image Processing*, Springer-Verlag, London, 261, 2009.

Ira TUBA  
Megatrend University  
Faculty of Computer Science  
Bulevar umetnosti 29, Belgrade  
SERBIA  
E-mail: ira.tuba@gmail.com

## Blur detection in digital images

Viktor Tuba

Teacher Coordinator: Milan Tuba

### Abstract

One of the common irregularities in digital images is blur. Usually it is caused by the motion or out of focus. In this paper we present an algorithm and application for detecting images with blur or blurred regions within an image. Blur is detected by different algorithms for edge detection or with high pass filters in frequency domain. Additional classification is introduced to determine more precisely blurred regions.

## 1 Introduction

For decades after invention of camera, making quality images was very difficult and time consuming. It required high skill levels out of people who made them and lot of expensive equipment to properly develop image from film to paper. Things started to change rapidly with invention of digital photo sensors. After photo sensors and computer science for image processing developed enough, digital cameras took over almost completely. Now we can see results of taking image almost immediately. It makes a lot easier to make better images. Even more it is possible to process images after taking them and remove various imperfections afterwards. There are many algorithms being developed for various areas [1], [2], [3]. One of such areas is blur. Common use of algorithms for blur detection is for discovering forged documents [4], [5].

We say that image is blurred when it does not have sharp edges on at least some part of the image. Sometimes blur is deliberately created by photographer for artistic purposes, and sometimes it is created by mistake. In this paper we are concerned with blur that is made by mistake.

## 2 Blur

There are two causes for blur in images, motion blur and out of focus. Motion blur can occur due to subject movement or because of camera shaking. When during the recording of one frame capture changes due to rapid movement occurs motion blur. Also, motion blur may occur due to shaking of the camera while shooting pictures. Fig. 1 are examples of images with motion blur. Blurred regions can also appear because they are out of focus. Today, this very rarely blur the whole image. Usually one part of the image is clear and a few regions are out of focus. In Fig. 2 we can see that the left side of image, flowers, is in focus and the other parts of image is blurry.



Fig. 1: Motion blur



Fig. 2: Out of focus

### 3 Blur detection

Sharp images have clear edges, while blurred regions have softer edges. There are papers that use this feature to develop algorithms for blur detection [5], [6], [7]. In this paper we will present different algorithms for blur detection based on edge detection.

#### 3.1 Edge detection

There are two ways to represent image in computers: spatial domain representation and frequency domain representation.

Spatial domain representation keeps information about every pixel on the image. Each pixel is represented by several bytes of data that represent its colour (usually red, green and blue intensity of the colour) and transparency. This representation is easy to understand and works well for displaying images on screen or printing and even for storing. But for image processing, spatial domain representation is not useful.

Contrary to spatial domain representation that treats image as set of pixels and filters are applied on each pixels, frequency domain representation make it easier to detect various features of the image, such as edges.

### 3.1.1 Spatial domain

There are many algorithms in spatial domain for edge detection. The simplest way to detect edges is to subtract softened image from original image. Result is black image with white edges. Edges are representing high difference in intensity of neighbour pixels. Second method for edge detection is using derivatives. Edges would be represented by high values of first derivative. By detecting zero-crossing of second derivative we can detect edges.

One of the most used algorithm for edge detection is Canny edge detection. This algorithm have five steps. First step is to apply Gaussian filter. This filter smooth the picture and removes the noise. Edge detection algorithms are very sensitive on the noise, so this is important step. Second step is finding intensity gradients of the image. Purpose of this step is to find all edges with different direction (vertical, horizontal and diagonal). In this step are used four different filters. Third step is applying non-maximum suppression. It is edge thinning technique. Four step is applying double threshold. The aim of this step is to find potentially edges. Last step is track edge by hysteresis.

### 3.1.2 Frequency domain

Between spatial and frequency domain exists the following connection: slower grey levels change correspond to the lower frequencies. Vice versa, higher frequencies represent fast changes in grey levels. As we know, edges (and noise also) in spatial domain are recognized as big difference between neighbours pixels. Similarly in frequency domain we can recognize the edges as high frequency. For detecting the edges in frequency domain it is used high-pass filter. High-pass filter suppresses low frequencies and leaves high frequencies. We can experiment with leaving different number of high frequencies.

## 4 Our propose algorithms

Blur detection in our algorithm is done by edge detection. There are many different algorithms for detecting edges on the images and we can do precision analysis for each algorithm.

First step is to prepare image. For the sake of simplicity and for clearer edges, we will first transform image into black and white image. First step is to color image turn into grayscale image. It is done by averaging. Pixel value is set to average value of all color components. Next we turn 256 shades of grey into only 2 shades, pure white and pure black. For threshold we take average value of all pixels.

First classification we will do is in two groups: blurred images and not blurred images. If we determine that image is blurred, we will process it further to detect parts of the images that are blurred.

As we stated before, sharp parts of the image have clear edges, and blurred parts do not have clear transitions between colors. Based on that, next step is edge detection. For edge detection in spatial domain are implemented two algorithms. First, the primitive one, subtracting smoothed image of the original image. Second one is the most popular Canny edge detection algorithm. In frequency domain we tested high-pass filter with different parameter. We will introduce measurement for relation of edges versus flat parts of image. It is calculated as relation of number of edge pixels and number of non edge pixels. We need to experimentally determine threshold for this relation that will give us best results in blur detection.

After we determined that image contains blur, we split image in  $X*Y$  regions and apply algorithm on them. For the image size  $n*m$ , size of regions are  $(n/X)*(m/Y)$ . Choice of number and size of regions is very important for precision of algorithm. If we split the image on too many regions, there is a possibility that part of the image between two edges be recognised as blur. On the other hand if we take too little regions there is possibility that sharp parts of image be split in multiple regions and entire image be detected as blurred in spite of existence of sharp parts.

After we split the image to regions and detect blurred regions, we can do classification of the blur. In this algorithm we split blur in two types: motion blur and out of focus blur. Main feature on



images that are out of focus is existence of at least one sharp object and blurred rest of image. There for there are one or several parts of the image that are sharp. So if we have one part of the image that is sharp and rest is blurred, we will assume that it is out of focus blur. On the other hand, motion blur is created by movement of the object in moment of taking the image. In that case we will have single blurred part of the image and the rest will be sharp and we will assume it is motion blur. There is an exception in motion blur. It is possible that motion blur is created not by movement of the object, but by movement of the camera. In that case entire image will be blurred. In that case we will also assume that it is motion blur. There for, our algorithm categorise images in three categories: non-blurred images, images with motion blur and images with out of focus blur.

Software for testing is developed in Visual Studio 2013, program language C#. Parameters for high-pass filter and number of regions can be set by user.

## 5 Experimental results

Images used for testing are from standard database for testing image blur detection. For testing, aside of standard database, we used un-blurred images from photo sharing websites flickr.com. Image size have impact of size of the regions, but that problem is disregarded in this version of the program. We used images of the same sizes for testing. After algorithm execution image is labeled as sharp, motion blurred or out of focus blurred. Fig. 3 show impact of number of regions on precision of detection. Middle image show detection with optimal number of regions witch is found experimentally.

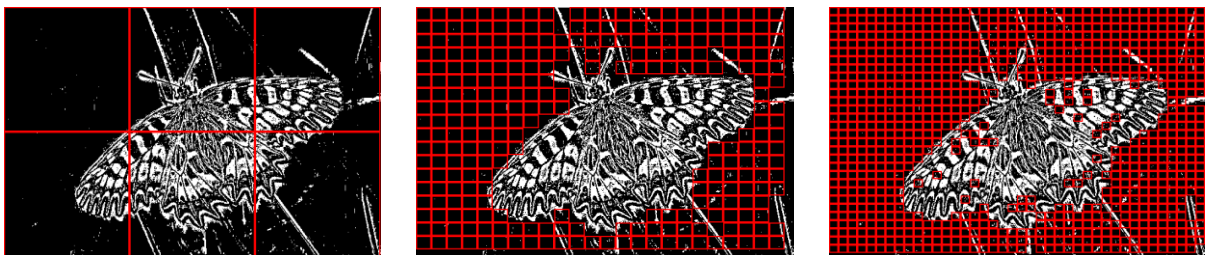
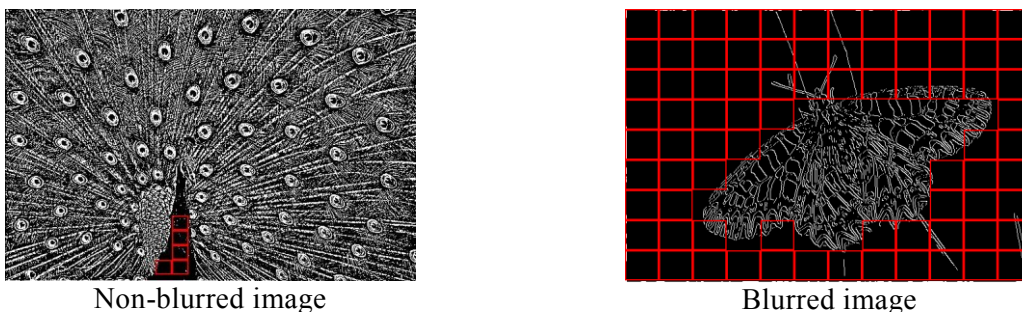


Fig. 3: Different number of regions

Determining threshold for edge saturation is the main task. Trough experiment we found that best value is between 0.5 and 0.7. Problems occur with images of sky or grassland or similar. Those images do not have lot of edges, but they are not blurry. On Fig. 4 are represented results of classification blurred and un-blurred images with saturation threshold 0.5 and on Fig. 5 with saturation threshold 0.7. Edges are detected with Canny algorithm.



Non-blurred image

Blurred image

Fig. 4: Classification with saturation threshold 0.5

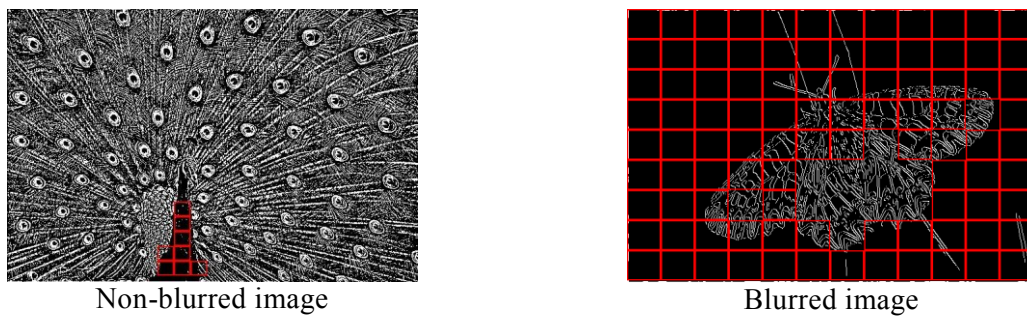


Fig. 5: Classification with saturation threshold 0.7

Images that are labeled blurred are further processed and on Fig. 6 are shown the results of detection of blurred regions and their classification. On Fig. 6 are images with motion blur (a), out of focus (b) and camera shake images (c) and their detected blurred regions respectively. For images where blur was created because of movement of the object, blurred regions are clustered on one place. On image (c) most of the image is detected as blurred, and therefor blur was create because camera shake.

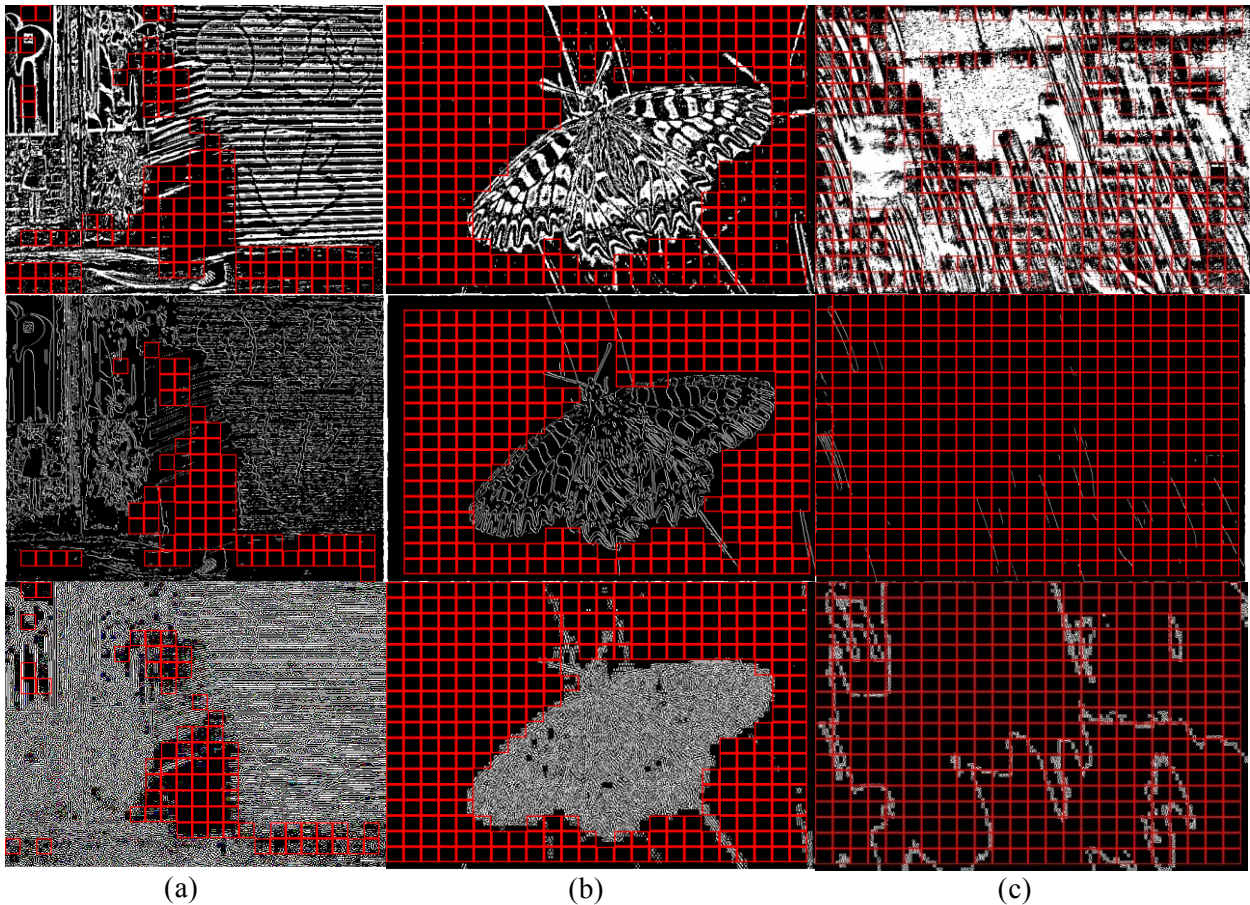


Fig. 6: Blur classification

On Fig. 6, first row of images are results of detection using primitive algorithm for edge detection. In second row is Canny algorithm and in third row are edges detected with high pass filter. Original pictures are shown on Fig. 7. Experiment show that primitive algorithm have problem with noise on image. For better results, we need to process image to remove noise. High pass algorithm is used with coefficient 5 witch is experimentally found to be the best. Canny



algorithm for edge detection gives very good results in our algorithm. Conclusion of these experiments is that based on needs, for processing in spatial domain can be used Canny algorithm, and for frequent domain high pass filter.

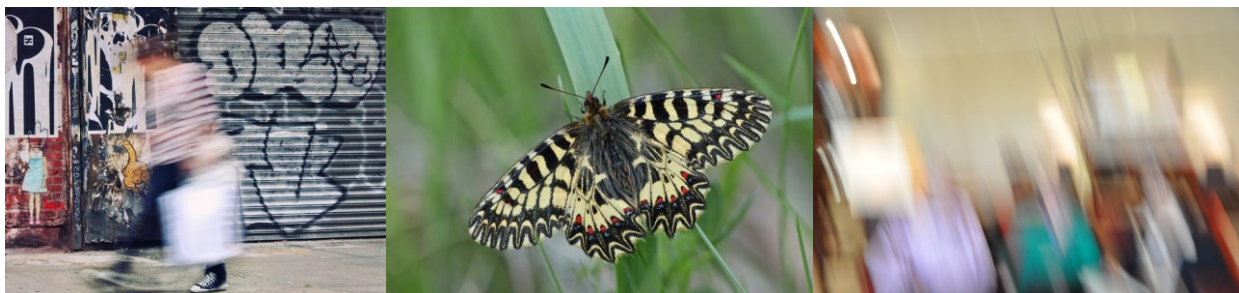


Fig. 7: Original images

## 6 Conclusion

In this paper is suggested algorithm for blur detection and classification. Algorithm first detects blurred images and then blurred regions on the images and class of blur. For detection we used several algorithms for edge detection. Our algorithm has proved to be very effective for blur detection and with planned improvements it could get better. In further research we will include neural networks and try to use them to find out how many edges sharp image have. Also we can experiment with threshold and sizes of the regions. After detection and classification is done, it is possible to refine classification and add new classes like depth of field blur. Blur detection is subject that draws a lot of interest of scientific community and is in constant development. One of major uses for blur detection is exposing fake documents. After blur detection, next logical step is blur removal, subject interesting not only to scientists, but to general public too.

## References

- [1] L. Renting, Z. Li and J. Jia. *Image partial Blur Detection and classification*, Computer Vision and Pattern Recognition, ISSN: 1063-6919, 1-8, 2008.
- [2] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro, A variational framework for simultaneous motion estimation and restoration of motion-blurred video. *Computer vision*, 1-8, 2007.
- [3] Y. Chung, J. Wang, R. Bailey, S. Chen, and S. Chang. A non-parametric blur measure based on edge analysis for image processing applications. *IEEE Conference on Cybernetics and Intelligent Systems 1*, 2004.
- [4] L. Zhou and D. Wang. *Blur detection of digital forgery using mathematical morphology*, Proceedings of the 1st KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, Wroclaw, Poland, 990–998, 2007.
- [5] G. Cao, Y. Zhao and R. Ni. Edge-based Blur Metric for Tamper Detection. *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 1, No. 1, 20-27, 2010.
- [6] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi. *A noreference perceptual blur metric*, ISSN: 1522-4880, 57-60, 2002.
- [7] J.H. Elder and S. W. Zucker. *Local scale control for edge detection and blur estimation*, Proceedings of 1996 European Conference on Computer Vision, Vol. 2, 57-69, 1996.

Viktor TUBA  
Faculty of Organizational Sciences  
University of Belgrade  
Jove Ilica 11, Beograd  
SERBIA  
E-mail: viktortuba@gmail.com

# **S.N.A.P. - Student Network Administration Platform: The Core and Mobile Implementation**

**Kim R. Velker, Adrian A. Bărbos, Cristian Bota, Camelia-L. Ianoş**  
**Teachers Coordinators: Camelia-M. Pinte, Hajdu Măcelaru Mara**

## **Abstract**

The current paper illustrates a new platform for universities, with web and mobile support. Our newly platform is named SNAP. It is inspired by the fast paced evolution of technology and the human urge of acquiring vital information. The way it works will be explained. SNAP brings innovative user-friendly interface. We seek to ease and enhance the relation between students and teachers. The platform looks promising for present generations at achieving big progress on intellectual stimulation and better communication development, that's why it can be adapted to any university in the future.

## **1 Introduction**

A class project was assigned to a team of ten students. What we initially thought to be a simple class project, turned out to be something bigger, something we believe in. We chose a university related theme, because we think it is time to create something new for the benefit of both students and professors.

We all know that frustrating feeling when we can't find the homework a teacher gave us, when we don't know in what classroom the next course is going to be or even when we forget the date of our next exam, right? Well SNAP it! This is going to end because we have the solution! S.N.A.P. (Student Network Administration Platform) is an application that will help solve all these problems and so much more. First of all you don't even have to create an account because you can log in with the information from your usual university login and find all the information your teachers can provide like dates of exams, locations of courses or topics you need to study to pass. The application also includes an interactive map of the university so you won't get lost even if you are a freshman. We hope SNAP is the answer to all your problems and that it will make your student life a little bit easier

## **2 Student Network Administration Platform**

We studied many solutions developed for helping student and teachers do their daily tasks easier. One of the best solutions we found is made by Google and its called Classroom. Classroom is designed to help teachers create and collect assignments paperless, including time-saving features like the ability to automatically make a copy of a Google Document for each student. It also creates Drive folders for each assignment and for each student to help keep everyone organized.



Students can keep track of what's due on the Assignments page and begin working with just a click. Teachers can quickly see who has or hasn't completed the work, and provide direct, real-time feedback and grades right in Classroom.

Google Classroom looks very good so far, but it is only available through Google Apps for Education. The cloud solution is a good solution for most people but some of us like their data with a little more privacy.

We really liked the solution that Google came up with, but we think we can make it better and these are our improvements:

- **privacy** - Student Network Administration Platform is hosted on an university private server, so privacy isn't an issue
- **integration** - if an university has other platforms with user accounts, like ours does, can import all the users for easier access
- **calendar and schedule** - is divided in two categories, so the calendar will show all the events (exams, paper deadlines) and the schedule to track course activity (grades and attendances)
- **indoor interactive map** - it contains a map of each of the university's floors
- **professors** - a list all the professors so you can interact easy
- **newsfeed** - to be up to date with everything new in the university
- **inbox** - because communication between students and professors is important
- **students** - list of all the students to help professors contact the students
- **push notifications** - alert the user with a message when new data is published

## 3 Core Database and API

### 3.1 phpMyAdmin Database

**phpMyAdmin** is a free and open source tool written in PHP intended to handle the administration of MySQL with the use of a web browser. It can perform various tasks such as creating, modifying, or deleting databases, tables, fields, or rows; executing SQL statements; or managing users and permissions [1,2,3].

Our application database has eleven tables for managing data : '*Students*', '*Groups*', '*Courses*', '*Classrooms*', '*Calendar*', '*Schedule*', '*Newsfeed*', '*StudentActivity*', '*Inbox*', '*Devices*', '*Professors*'. Every table's structure and purpose is detailed.

**Students Table** host all the students' login details and personal data. The table has the columns :

- id\_student - Int(11)
- id\_group - Int(11)
- name - VARCHAR(50)
- surname - VARCHAR(50)
- initial(s) of father's name - VARCHAR(5)
- phone - VARCHAR(15)
- email - VARCHAR(50)
- created - DATETIME

**Groups Table** contain information regarding name of the group, study year and coordinating professor.

- id\_group - Int(11)
- name - VARCHAR(50)
- study\_year - VARCHAR(4)
- id\_professor - Int(11)
- created - DATETIME

**Courses Table** contains the list of all the courses name, description, and coordinating professor.

- id\_course - Int(11)
- name - VARCHAR(50)
- description - TEXT
- id\_professor - Int(11)
- created - DATETIME

**Classroom Table** contains all the information regarding the classrooms available in the University with information about the name and capacity.

- id\_classroom - Int(11)
- name - VARCHAR(50)
- description - TEXT
- capacity - Int(4)
- created - DATETIME

**Calendar Table** has information about all the events like exams and paper deadlines.

- id\_calendar - Int(11)
- name - VARCHAR(50)
- description - TEXT
- priority - TINYINT(1)
- date-time - DATETIME
- created - DATETIME

**Schedule Table** includes information about the courses, seminars, and laboratories.

- id\_schedule - Int(11)
- id\_group - Int(11)
- id\_course - Int(11)
- date-time - DATETIME
- created - DATETIME

**Newsfeed Table** contains the news items with access level, so news can be posted as ‘student only’, ‘professor only’ and ‘both’.

- id\_newsfeed - Int(11)
- title - VARCHAR(255)
- content - TEXT
- small\_image - TEXT
- big\_image - TEXT
- access\_level - TINYINT(1)
- created - DATETIME

**Student Activity Table** stores the grades, activity points, and attendance.

- id\_activity - Int(11)
- type\_activity - Enum(‘grade’, ‘point’, ‘attendance’)
- id\_student - Int(11)
- name - VARCHAR(255)
- value - VARCHAR(10)
- created - DATETIME

**Inbox Table** contains all the messages between students and professors.

- id\_inbox - Int(11)
- id\_professor - Int(11)
- id\_student - Int(11)
- message - TEXT
- created - DATETIME

**Devices Table** is used for storing the device token from every user. The device token will be used to send push notification to the users.

- id\_device - Int(11)
- user\_id - Int(11)
- user\_type - Enum('student', 'professor')
- device\_token - TEXT
- device\_type - Enum('Android', 'iOS')
- created - DATETIME

**Professor Table** contains the login and personal information of each professor. It will be used to allow professors to login and present their public contact information.

- id\_professor - Int(11)
- name - VARCHAR(50)
- surname - VARCHAR(50)
- password - VARCHAR(255)
- phone - VARCHAR(15)
- email - VARCHAR(50)
- created - DATETIME

### 3.2 PHP implementation of the API

**PHP** is a script language and interpreter that is freely available and used primarily on Linux Web servers. PHP, originally derived from Personal Home Page Tools, now stands for PHP: Hypertext Preprocessor, which the PHP FAQ describes as a "recursive acronym." [4]

We used PHP in developing the Web **API** (short for Application programming interface) that we use to connect our application to our database and manipulate data more easily. All the *php* methods we used to create our API will be explained with code snippets and interpretations.

One of the most important methods is the one that connect to the database and for this we made 2 PHP files: 'Config.php' and 'Connect.php'. The first one defines our connection variables:

- **DB\_HOST** - defines the host of the database
- **DB\_USER** - defines the database user
- **DB\_PASS** - defines the database password
- **DB\_NAME** - defines the database name

The second file uses these variables in the following code snippet to attempt a connection to the database and if it succeeds, it will select the database.

```
$db_connect = mysql_connect(DB_HOST, DB_USER, DB_PASS) or die;  
$db_select = mysql_selectdb(DB_NAME, $db_connect) or die;
```

Once connected to the database we can start using the next API methods that let us interact with the applications data. Every API JSON formatted response respects the same format and it will send an array of two keys:

- **data** - containing an array/array/key/keys of the requested information
- **status** - contains an INT value of 1 for success and 0 for failure. We use this status value to check if the request to the database succeeded

#### 3.2.1 List of the APIs and explanation

All the APIs use PHP MySQL queries to manipulate and get data. The basic queries we used are for

- **selecting data:** example - `mysql_query("SELECT * FROM 'table_name' ")`
- **inserting data:** example - `mysql_query("INSERT INTO 'table_name' ('key') VALUE('value')`

- **updating data:** example - `mysql_query("UPDATE 'table_name' SET 'key' = 'value' WHERE 'key' = 'value'")`
- **deleting data:** example - `mysql_query("DELETE FROM 'table_name' WHERE 'key' = 'value' ")`

Requesting an API from the mobile application or the web application is done with http requests by sending 'method' = 'method\_name' and other needed parameters through **POST Method**. For each request if it succeeds a **JSON** formatted response will be sent. If the API method did the requested data selection, update or delete the data will be send along with a status flag of '1' . If not, the only thing sent will be the status flag of '0' value.

The list of the main API methods will be enumerated.

- **Login** (*api method = 'login', database tables: 'Students', 'Professors'*) - allows students to log in the application with a 'name' and a 'cnp' and professors with a 'name' and a 'password'
- **Schedule** (*api method = 'get\_schedule', database table: 'Schedule'*) - by sending the user's id, it will list all the courses, laboratories and seminars that the user is attending
- **Subjects** (*api method = 'get\_subjects', database table: 'Courses'*) - retrieves all the subjects that are available in the University starting with the subjects the current user is taking
- **Calendar** (*api method = 'get\_calendar', database table: 'Calendar'*) - shows all the events like exams and paper deadlines with a priority tag and which will be listed in a calendar layout
- **Professors** (*api method = 'get\_professors', database table: 'Professor'*) - requests the list of all the Universities professors with their contact details with a tag to show the professors that are teaching the current logged in student
- **Students** (*api method = 'get\_students', database table: 'Students'*) - method designed for professors to request a list of all the students attending their classes with contact details
- **News** (*api method = 'get\_news', database table: 'News'*) - method for requesting the announcements available for the logged in account type by sending as parameter 'type' = 'account\_type' (professor or student)
- **Inbox** (*api method = 'get\_inbox', database table: 'Inbox'*) - retrieves all the messages received by the logged user

Along the main APIs are secondary ones for adding, updating or deleting data from almost all the modules ( example: *add\_news, add\_subject, edit\_subject, send\_message*, etc). A very important API for the mobile application is the one that lets you know when there is something new or when you received a message and doesn't let you forget about an exam and a paper deadline, the *PUSH Notification API*.

*PUSH Notification* is a message received almost instantly on your smart-phone. Android or iPhone notifications for the end user are the same; a message is displayed in the notification bar. The API selects all the entries from the 'Devices' table and takes the 'device\_token' and 'device\_type'. An if statement divides the received request in two separate array, one of Android devices and the other of iOS devices.

*Android* uses the Google Cloud Messaging service from Google's Api Console to send notifications based only on the *device\_token* retrieved from the database.

The *iOS* method uses Apple's Push Notification Service and requires more than the device token to send notifications, it needs a push notification certificate (one for development and one for production) configured from Apple's Developer Certificate Center.

## 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. It has the same modules that the web application has to ease access on the go. The mobile application and its modules will be presented further on, while the web application implementation and development will only be discussed. Both Android and iOS applications contain the following main modules that will be explained.

- **Login** is the first module and the first screen a user will see when the application is started. 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.
- **User profile** is a common module for both professor and student accounts. Basic information like the name, surname, email, phone, and profile picture will be shown and the possibility of changing the password for professor accounts.
- **Schedule** presented in a day calendar format has the information about the class schedule.
- **Calendar** module allows examining the events, exam dates, and paper deadlines.
- **Professors'** module lists all the professors associated to the courses the current logged in user takes; users can access a detail page of professors, see their contact information, and send messages.
- **Students'** module is just like the one above, where professors can view student information and send them a message.
- **Newsfeed** specifics news based on the logged in account type. The news items have three access levels: professor (news can be viewed only by professor accounts), student (news only viewed by student accounts) and universal.
- **Inbox** is standard email-like inbox.

### 4.1 Android & iOS data request method development

**iOS** application development is done using the official Apple SDK Development program, named XCode 6. The programming is done by combining the two programming languages available for iOS development. The first one is Objective C, older but still the best method for creating dynamic frameworks. **Objective-C** is the primary programming language you use when writing software for OS X and iOS. It's a superset of the C programming language and provides object-oriented capabilities and a dynamic runtime. Objective-C inherits the syntax, primitive types, and flow control statements of C and adds syntax for defining classes and methods. It also adds language-level support for object graph management and object literals while providing dynamic typing and binding, deferring many responsibilities until runtime. The second language is Apple's **Swift** programming language, very new but powerful enough to convince you to start developing new applications with it. Writing code is interactive and fun, the syntax is concise yet expressive and **Swift** code works side-by-side with Objective-C, which is very useful [4].

The '**GetData**' method works by making a http request to the server and interpreting the response as **JSON Array** or **JSON Object**. Code snippet below.

```
let swiftObject:AnyObject = NSJSONSerialization.JSONObjectWithData(reply!, options:
NSJSONReadingOptions.AllowFragments, error:&jsonerror)!
// JSONObjectWithData returns AnyObject so the first thing to do is to downcast this to a known type
if let nsDictionaryObject = swiftObject as? NSDictionary {
```

```
    if let swiftDictionary = nsDictionaryObject as Dictionary? {  
        let temp:Dictionary = swiftDictionary;  
        tempData = temp  
    }  
}
```

GetData method is called using parameters like: *URL* ( the host address of the API ), *Parameters* and *Values Array* ( a collection of keys and values, example: ‘method’ : ‘get\_news’ ) and a debugging variable of true or false. In the debugging is on, the application console will show if a non *JSON* response is received. Example method call: “Functions.getData(“method\_name”, ParamsArray: [“a”, “b”], ValuesArray: [“a”, “b”], MainURL: “http://host.domain/api/”) as Dictionary”.

**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 byte code that can run on any Java virtual machine (JVM) regardless of computer architecture [1,3,4].

**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 [1,3,4].

## 5 Conclusion and Further Work

The Student Network Administration Platform (SNAP) application can be further enlarged with more modules and maybe some modifications to the current modules for better usage. The application could be developed on other mobile platforms like Windows Phone 8.1 or even on desktop operating systems like Mac OS X or Windows 8.

## References

- [1] S. Tanasa, C. Olaru, *Java de la 0 la expert*, Ed.Polirom 2011
- [2] R.C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*, Prentice Hall 2008
- [3] J. Friesen, *Learn Java for Android Development*, Apress 2010
- [4] Wikipedia

Kim VELKER  
Technical University  
Cluj-Napoca  
Faculty of Science  
76 Victoriei  
ROMÂNIA  
kim.velker@trecadis.ro

Adrian BĂRBOS  
Technical University  
Cluj-Napoca  
Faculty of Science  
76 Victoriei  
ROMÂNIA  
adrian.barbos@trecadis.ro

Cristian Bota  
Technical University  
Cluj-Napoca  
Faculty of Science  
76 Victoriei  
ROMÂNIA

Camelia-L. Ianoş  
Technical University  
Cluj-Napoca  
Faculty of Science  
76 Victoriei  
ROMÂNIA

## **S.N.A.P. - Student Network Administration Platform: Web Design and Implementation**

**Vlăduț-F. Mureșan, Alin G. Bonaț, Ramona Dragomir,  
Sergiu A. Molnar, Anamaria Palfi, Diana M. Codrea**  
**Teachers Coordinators: Camelia-M. Pinteă, Hajdu Măcelaru Mara**

### **Abstract**

*Student Network Administration Platform* is a platform for universities offering easy access to information for both students and professors. Its purpose is to enhance the way vital information is exchanged through all the modern communication channels, adapting to all the existing platforms, web, mobile and also desktop application and widgets. A follow-up of the article, *S.N.A.P. - Student Network Administration Platform: the Core and Mobile Implementation*, the current work illustrates the web design, the implementation of the application, a detailed description of the technologies used and also the targets regarding user experience.

## **1 Introduction**

The Student Network Administration Platform, SNAP, application develops a new concept, the idea that every user should be able to access easily the information. What does the application do for the user? It offers detailed and important information about the university and any other news that refers to the student life. The application is dedicated to both students and professors, providing features according to their individual needs. The design is simple and easy to understand that's why it is inspired by the accelerating change and fast evolution of technologies.

## **2 Technologies**

### **2.1 Adobe Photoshop & Minimalistic design**

Adobe Photoshop is the predominant photo editing and manipulation software on the market. Its uses range from full featured editing of large batches of photos to creating intricate digital paintings and designs for websites or applications. Many web designers use Photoshop for creating full mockups of their pages. Those mockups are later sliced by the web developers. Photoshop has been the most secure option for designers for years. Most front-end developers know how to use Photoshop and how to easily extract graphic elements from the PSD files. There is a variety of tools and plug-ins that make that job even easier [1,2].



CSS3 can create almost everything, the only impediment being the support for older browsers or actual photos/artwork. Often, CSS can be regressed to a working stage to offer support for older browsers. Minimalism that is about breaking things down to the least number of elements required for a design to work. It's about removing things until the only thing that remains is the sole purpose of the design. "Less is more" is probably the most well-known catch phrase of the minimalist movement. In Web design, less is more is achieved by using only elements that are necessary to a given design. A good example of minimalistic design is Google's design language "Material Design" introduced at Google's I/O Developer Conference in June 2014 [1,4,5]. Material Design guidelines state that every single element must have a purpose in the design, giving the user meaning about what they can use.

## 2.2 HTML & CSS

HyperText Markup Language (HTML) is the publishing language of the World Wide Web (WWW). This is a markup language used to make web pages, being one of the main components of Open Web Platform. The first version of HTML was made by Tim Berners-Lee at the end of 1991. HTML is made up of a series of labels and markup tags. They are between the < and > symbols and they describe the contents of a HTML document, which can have the *.html* or *.htm* extension. The web browser reads the HTML documents by interpreting the tags to compose the content of the web page. The web browsers can define the content and the design of some elements by using CSS.

Cascading Style Sheets (CSS) is a standard used for formatting elements in a HTML document. The styles can be attached to HTML through external files with the *.css* extension or in the HTML document itself with the <style> tag or the style attribute. CSS was made to separate the content of the document from its aspect. It has a simple syntax; it uses keywords from the English language to define the naming of style properties [3,4,5].

## 2.3 JavaScript, jQuery & PHP

JavaScript is a scripting language developed by Netscape to enable Web authors to design interactive sites. Although it is similar to Java, it was developed independently. JavaScript can interact with HTML source code, enabling Web developers to enrich their pages with dynamic content.

jQuery is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to simplify the usability of JavaScript on every website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

PHP is an HTML embedded Web scripting language. This means PHP code can be inserted into the HTML of a Web page. When a PHP page is accessed, the PHP code is read or "parsed" by the server the page resides on. The output from the PHP functions on the page is typically returned as HTML code, which can be read by the browser. Because the PHP code is transformed into HTML before the page is loaded, users cannot view the PHP code on a page. This makes PHP pages secure enough to access databases and other secure information [3,4,5].

### 3 The user experience

The user experience must be seamless, that's why we opted for a simple and effective user-friendly design that works for everyone. The interface is split into three main sections: the navigation menu, notification bar, and the content section. The first section (left side) is the navigation menu with the schedule, subjects, calendar, professors, and news.

The header of the application (top side) contains the event and inbox notifications, as well as a link to the user profile and a *logout* button. Every user will be notified of any new information that is added or updated throughout the platform. Another interactive feature is that users have the possibility of sending a private message to a professor or even to a student. The “*My Profile*” page shows the user account information and allows the user to view and edit some of his personal information such as the phone number and the e-mail address. (Fig. 3.1.)

The core of the homepage (last section) includes schedule, calendar, latest announcements, and future events; each one includes only the most important information and for the full details the user can get what he desires simply by selecting: “*Full schedule*”, “*All news*” or “*All events*”. (Fig. 3.2.) The schedule section updates daily with data: name, type of the subject and the class number.

The “next events” container shows the details of the events that will take part in the next week. For a specific event the student can participate by selecting a given date from the calendar and confirming his attendance or by clicking “join”. Important news posted by the professors can be read in the “*Latest News*” section. The navigation menu is positioned on the left side of the viewport and it contains a list with each item aligned vertically, that links to a different page. Above this menu there is a small section that displays the user's avatar, his full name and account type.

The schedule page contains a detailed timetable for the current semester that is available and easy to access at any time. The timetable updates very fast if any unforeseen changes occur. It has eight columns and seven rows. The first column represents the intervals of time and each of the remaining columns represent one day of a week. This schedule timetable is linked to a 2D interactive map. If any subject is clicked then a map with the university is displayed and the class of the selected subject is highlighted. (Fig. 3.3.)

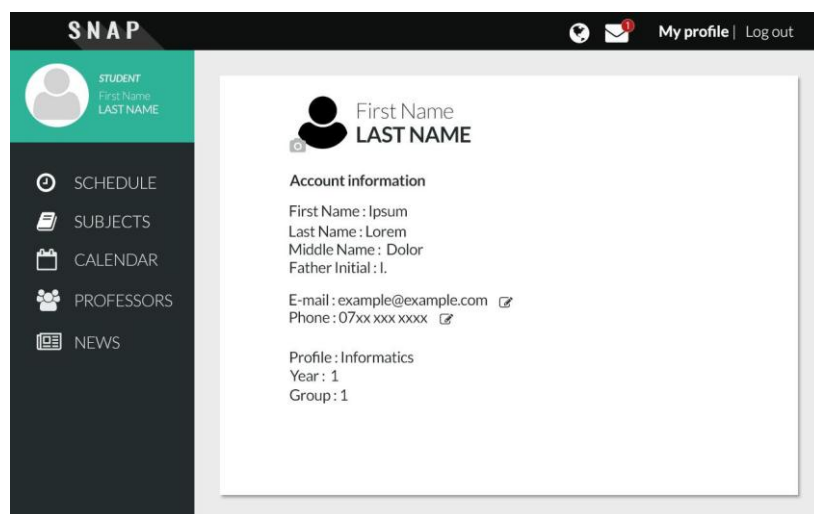


Fig. 3.1. My profile page

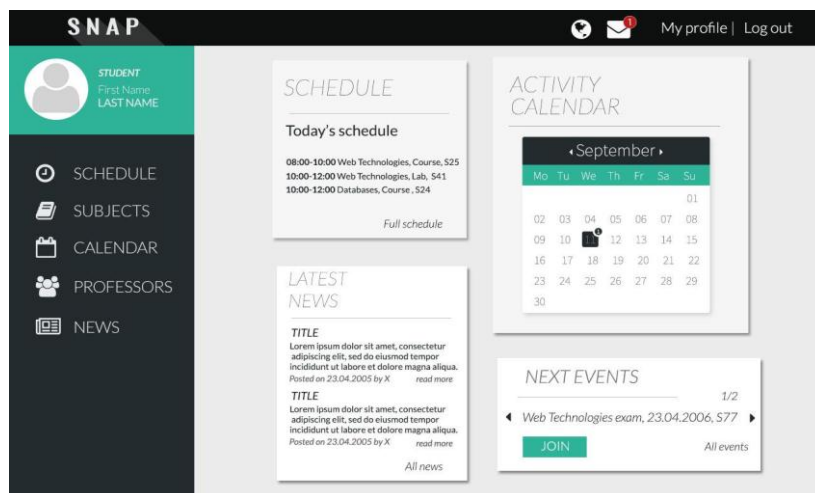


Fig. 3.2. Homepage

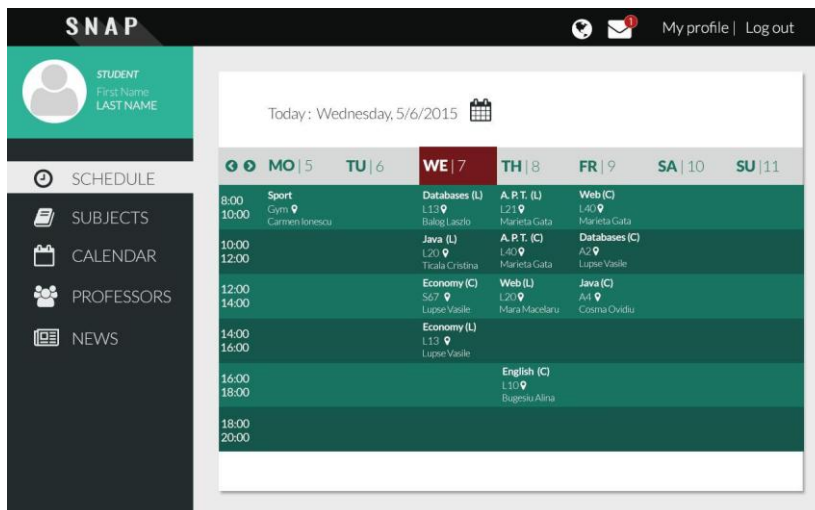


Fig. 3.3. Schedule page

A list with all the subjects for the current semester is presented in the “subjects” page. (Fig. 3.4.) Here, every student can see the professors responsible for the course and seminar, the type of examination, his own attendances, and his grades during exams or over the semester. Every selected subject will be underlined with a red color, meaning that its information can be found in the right section. Each subject displays the professor name for the course and seminar. Also, there are two containers, called messages and files. The message container is an archive of all the important posts from the professor. The second container has a list with all uploaded files. They can be downloaded by all the students for better understanding of the subject.

The next page contains an interactive calendar which is based on events such as final exams, competitions, and deadlines for assignments. Navigation through the calendar is easily done by selecting the left or right arrow near the month name. A date from the calendar that has a particular event is highlighted, so that all users are informed in a friendly way. By clicking a date with an event, the user can view the details and even join. (Fig. 3.5.)

All the professors from the university are listed in the “*professor*” page so that anyone can easily find their profile and contact them. Only professors of the current semester are displayed in the “*My professors*” list. The user can click the “*show all professors*” button to update the list with the remaining professors in the university. Every selected professor will be underlined with a red color, meaning that his information is shown in the right side. Every subject taught by the professor is listed under his name. Also, some particular listed items display a small book icon, which is an anchor to the respective subject details accessible in the subject’s page. The book icon only appears only if the subject is studied in the current semester. Using the contact information provided, the user can get in touch with the professor by phone, e-mail, or private message.

In the news page, the user can view all the important announcements related to the university; the categories are: modifications in the schedule, scholarship related topics, tax payments or other useful information. The user can select the category of news he wants to view by choosing an option from the dropdown submenu on the right side of the page. (Fig. 3.6.) Only the professors have the privilege of adding news, the student has only the ability to read them. The news has title, a thumbnail, and a short preview of the content, a date, and an author.

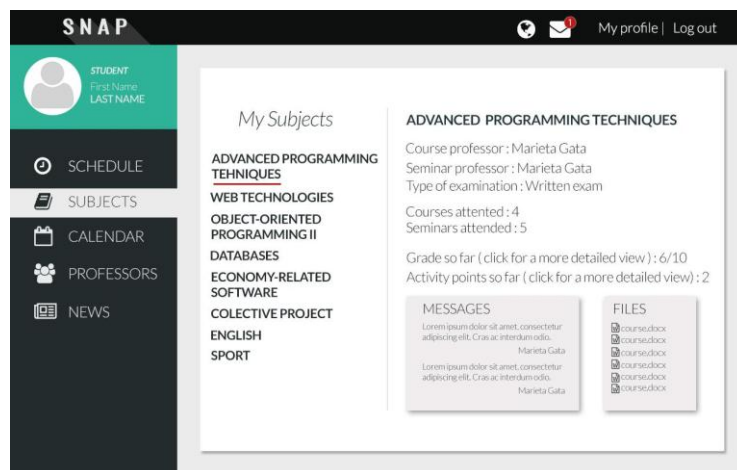


Fig. 3.4. Subjects page

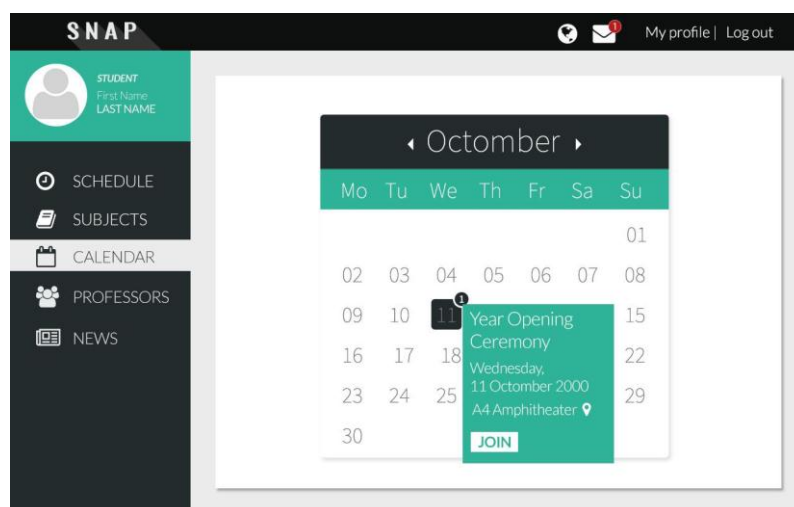


Fig. 3.5. Calendar page

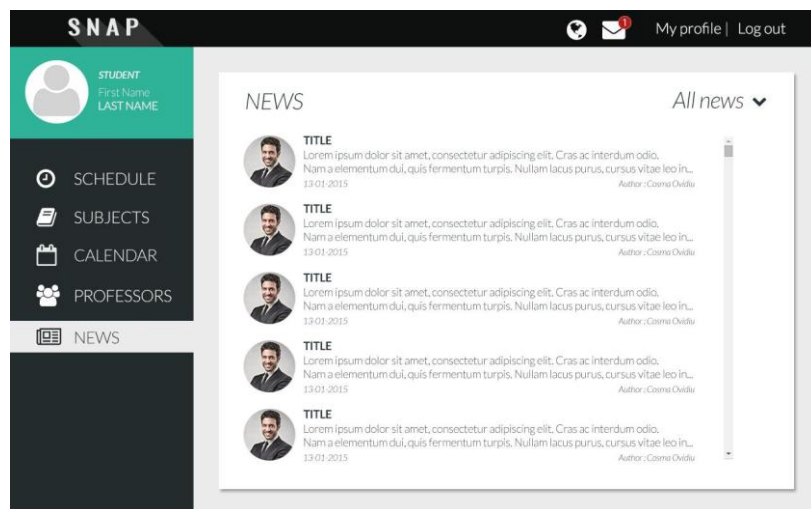


Fig. 3.6. News page

## 4 Conclusion

The application *Student Network Administration Platform, SNAP* is meant to encourage students and professors to easily interact. We chose to make a change and improve the way we access information on SNAP. The platform is promising showing a good communication platform that can be adapted to any university in the future.

## References

- [1] <http://graphicdesign.stackexchange.com/>
- [2] <http://rafaltomal.com/should-we-still-use-photoshop-to-design-websites/>
- [3] <http://stackoverflow.com/>
- [4] <http://www.w3schools.com/>
- [5] <https://en.wikipedia.org>

Vlăduț-F. Mureșan  
Technical University  
Cluj-Napoca  
Faculty of Science  
76 Victoriei  
ROMÂNIA  
vladmuresan02@gmail.com

Alin G. Bonaț  
Technical University  
Cluj-Napoca  
Faculty of Science  
76 Victoriei  
ROMÂNIA  
alin.bonat@gmail.com

Ramona Dragomir  
Technical University  
Cluj-Napoca  
Faculty of Science  
76 Victoriei  
ROMÂNIA

Sergiu A. Molnar  
Technical University  
Cluj-Napoca  
Faculty of Science  
76 Victoriei  
ROMÂNIA

Anamaria Palfi  
Technical University  
Cluj-Napoca  
Faculty of Science  
76 Victoriei  
ROMÂNIA

Diana M. Codrea  
Technical University  
Cluj-Napoca  
Faculty of Science  
76 Victoriei  
ROMÂNIA

**List of authors:**

- Cristian ALDULEA  
Transylvania University of Braşov  
Department of Mathematics and Informatics  
Str. Iuliu Maniu Nr. 50, 500091  
Braşov, ROMANIA  
E-mail: alduleacristi31@gmail.com
- Ladislau ANDRASI  
"National School of Gas" Technology High School, Mediaş  
Mathematics - Informatics  
Mediaş, ROMANIA  
E-mail: ladiladi85@gmail.com
- Vlad BĂDOIU  
"Mircea cel Bătrân" National College  
Educational Software  
Rm. Vâlcea, Vâlcea, ROMANIA  
E-mail: vlad\_andrew@openmailbox.org
- Adrian BĂRBOS  
Technical University of Cluj-Napoca  
Faculty of Science  
76 Victoriei  
Cluj-Napoca, ROMANIA  
E-mail: adrian.barbos@trecadis.ro
- Paul – Gheorghe BARBU  
"Lucian Blaga" University of Sibiu  
Department of Computer Science and Electrical Engineering  
10, Victoriei Bd., 550024, ROMANIA  
Sibiu, ROMANIA  
E-mail: barbu.paul.gheorghe@gmail.com
- Cristian BECKERT  
„Samuel von Brukenthal” National College  
Piaţa Huet nr.5  
Sibiu, ROMANIA  
E-mail: beckertrysty@yahoo.com
- István BIALKÓ  
Spiru Haret University, Bucharest  
Faculty of Mathematics and Informatics  
Str. Ion Ghica, nr. 13, CP 030045, Sector 3  
Bucharest, ROMANIA  
Email: bialko.istvan@gmail.com
- Paul BOLDIJAR  
The National College of Computer Science "Grigore Moisil"  
Computer Science  
Braşov, ROMANIA  
E-mail: paul.bv96@yahoo.com
- Alin G. BONAT  
Technical University of Cluj-Napoca  
Faculty of Science  
76 Victoriei  
Cluj-Napoca, ROMANIA  
E-mail: alin.bonat@gmail.com

Cristian BOTA	Technical University of Cluj-Napoca Faculty of Science 76 Victoriei Cluj-Napoca, ROMANIA
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
Andrei CĂLBĂJOS	Technical University of Cluj-Napoca Faculty of Science 76 Victoriei Cluj-Napoca, ROMÂNIA E-mail: andrei.calbajos@trecadis.ro
Georgi CHAVDAROV	University of Ruse Computer Sciences Studentska St. 8 Ruse, BULGARIA E-mail: powerbsd@abv.bg
Diana M. CODREA	Technical University of Cluj-Napoca Faculty of Science 76 Victoriei Cluj-Napoca, ROMANIA
Ramona DRAGOMIR	Technical University of Cluj-Napoca Faculty of Science 76 Victoriei Cluj-Napoca, ROMANIA
Bogdan-Mihai FIERARU	Transylvania University of Brașov Department of Mathematics and Informatics Str.Iuliu Maniu Nr. 50 Brașov, 500091 Brașov, ROMANIA E-mail: fieraru_bogdan2000@yahoo.com
Bogdan GRECU	„Lucian Blaga” University of Sibiu Faculty of Sciences Str. Dr. I. Rațiu, Nr.5-7 Sibiu, ROMANIA Email: bogdy_grecu@yahoo.com
Darius HAȚEGAN	„Lucian Blaga” University of Sibiu Faculty of Sciences Str. Dr. I. Rațiu, Nr.5-7 Sibiu, ROMANIA E-mail: hategandarius71@gmail.com

Ionut HODINA	<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: ionut.hodina@gmail.com</p>
Camelia-L. IANOS	<p>Technical University of Cluj-Napoca  Faculty of Science  76 Victoriei  Cluj-Napoca, ROMANIA</p>
Mihaela ITU	<p>Ovidius University of Constanța  Faculty of Mathematics and Computer Science  124, Mamaia Blvd.  Constanța, ROMANIA  E-mail: itu_mihaela_93@yahoo.com</p>
Arpad KERESTELY	<p>Transilvania University of Braşov  St. Iuliu Maniu 50, 500091  Braşov, ROMANIA  E-mail: k_arpi2004@yahoo.co.uk</p>
Raičević LJUBOMIR	<p>Megatrend University  Faculty Of Computer Science  Bulevar Umetnosti 29,  Beograd, SERBIA  E-mail: rljubo90@gmail.com</p>
Ioana MARTIN	<p>“B. P. Hasdeu” National College  Mathematics and Computer Science Area  No.1, Bd. Gării  Buzău, ROMANIA  E-mail: ioana.martin98@yahoo.com</p>
Alexandru MITRICA	<p>University of Craiova  Department of computers and Information Technology  Blvd. Decebal nr. 107  Craiova, ROMANIA  E-mail: alexmitrica@yahoo.com</p>
Sergiu A. MOLNAR	<p>Technical University of Cluj-Napoca  Faculty of Science  76 Victoriei  Cluj-Napoca, ROMANIA</p>
Vlăduț-F. MUREȘAN	<p>Technical University of Cluj-Napoca  Faculty of Science  76 Victoriei  Cluj-Napoca, ROMANIA  E-mail: vladmuresan02@gmail.com</p>



Radović NIKOLA	Megatrend University Faculty Of Computer Science Bvd Umetnosti 29 Beograd, SERBIA E-mail: nikolaradovic85@gmail.com
Cristian OANCEA	Transylvania University of Braşov Department of Mathematics and Informatics Str.Iuliu Maniu Nr. 50, 500091 Braşov, ROMANIA E-mail: oancea.c.cristian@gmail.com
Anamaria PALFI	Technical University of Cluj-Napoca Faculty of Science 76 Victoriei Cluj-Napoca, ROMANIA
Petru ROGOJANU	Children's Palace of Sibiu 7th grade Sibiu, ROMANIA E-mail: herrpetru01@gmail.com
Robert SĂNDICĂ	„Lucian Blaga” University of Sibiu Faculty of Sciences Str. Dr. I. Raţiu, Nr.5-7 Sibiu, ROMANIA E-mail: sandica.robert@gmail.com
Georgiana SECAREA	Transylvania University of Braşov Department of Mathematics and Informatics Str.Iuliu Maniu Nr. 50, 500091 Braşov, ROMANIA E-mail: georgiana.secarea@gmail.com
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
Alecsandru SPERILA	University of Craiova Department of computers and Information Technology Blvd. Decebal no. 107 Craiova, ROMANIA E-mail: sperila.alecsandru@gmail.com

Cosmin STOIAN	The "B. P. Hasdeu" National College Mathematics and Computer Science Area No.1, Bd. Gării Buzău, ROMANIA E-mail: cosmyn9708@gmail.com
Vlad TEODORESCU	„Mircea cel Bătrân” National College Educational Software Rm. Vâlcea, Vâlcea ROMANIA E-mail: vladjohn2013@gmail.com
Marius-Nicolae ȚOC	”Horea, Cloșca și Crișan” National College of Alba Iulia Mathematics-Informatics intensively Informatics No. 11, 1 Decembrie 1918 Avenue Alba Iulia, ROMANIA E-mail: toc.marius@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
Andreea Monica TOPORAS	Ovidius University of Constanța Faculty of Mathematics and Computer Science 124, Mamaia Blvd. Constanța, ROMANIA E-mail: andreemonica88@yahoo.com
Eva TUBA	University of Belgrade Faculty of Mathematics Studentski trg 16 Belgrade, SERBIA E-mail: eva.tuba@gmail.com
Ira TUBA	Megatrend University Faculty of Computer Science Boulevard Umetnosti 29 Belgrade, SERBIA E-mail: ira.tuba@gmail.com
Viktor TUBA	Faculty of Organizational Sciences University of Belgrade Jove Ilica 11 Beograd, SERBIA E-mail: viktortuba@gmail.com
Kim VELKER	Technical University of Cluj-Napoca Faculty of Science 76 Victoriei Cluj-Napoca, ROMANIA E-mail: kim.velker@trecadis.ro

**Sponsors:**



**CoSSoftware – Solutions**



**EBS Romania – Expectations Beyond Software**



**Gemini – Solutions**



**Global Solutions for Development**



**iQuest Romania**



**Omeron Technologies**



**PAN FOOD**



**ROPARDO**



**Top Tech**