Editor: Dana Simian

Proceedings of The International Conference on Applied Informatics

ICDD

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

Romania, Sibiu, May 25 - 27, 2017

Editor DANA SIMIAN

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

Proceedings of the International Conference on Applied Informatics
ICDD

May 25th – 27th, 2017
Sibiu, Romania

"Lucian Blaga" University of Sibiu

"Lucian Blaga" University, Sibiu, 2017

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 the International Conference on Applied Informatics ICDD
May 25th – 27th, 2017, Sibiu, Romania

Preface

This volume contains refereed papers presented within the International Conference on Applied Informatics – ICDD – 2017, which was held between May 25th – 27th, at the Faculty of Sciences, University " Lucian Blaga" of Sibiu, Romania.

The conference is mainly addressed to bachelor and master level students, PhD students and young researchers from all over the world. The conference gives the participants the opportunity to discuss and present their research on informatics and related fields (like computational algebra, numerical calculus, bioinformatics, etc.). The conference welcomes submissions of original papers on all aspects of informatics and related fields ranging from new concepts and theoretical developments to advanced technologies and innovative applications. 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 conference has brought together participants from 5 countries (Bulgaria, Republic of Moldova, Romania, Russia and Serbia).

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

The conference was organized with the support of the following sponsors (in alphabetical order): AUSY, CODEXWORKS, GSD, KEEP CALLING, NTT DATA, PARADINE DEVELOPMENT, OMERON, PROIT, ROPARDO, TOP TECH, VISMA.

May 2017

Dana Simian
Conference chairman

Motto:

“There are no limits, only your imagination”

Proceedings of the International Conference on Applied Informatics
IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT
Sibiu, Romania

Editor *Prof. Dana Simian PhD.*

Scientific committee

Contents:

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

Secured Bootloader: application for initializing and updating microcontroller applications

Barbu Paul - Gheorghe

Abstract

This paper aims to describe a bootloader for embedded devices. The purpose of the bootloader is to be the first to run when the microcontroller starts and to provide a way of controlling what application runs next, similarly to how bootloaders work on the desktop systems. The difference here being that the bootloader's purpose is to allow the embedded device to be much more flexible, since the application running on the microcontroller can be changed without using specialized hardware. This flexibility and the fact that no specialized hardware programmers are needed also leads to a reduction of costs for deploying and maintaining the embedded systems. These two goals, flexibility and cost reduction must be accompanied by a third one: security. The recent growth of embedded systems in the IoT (Internet of Things) domain demands that the device deployed be more secure than in the past.

1 Introduction

Working in an embedded area we have to think about how we can achieve the goals that we've set for the bootloader application.

First of all, any microcontroller that has to be programmed must be connected to a desktop system using a specialized piece of hardware, called most commonly a "programmer device", also known as a JTAG connector [1]. Through this connector the developer of the application can perform the flashing of the application. This is the first point that the application developed in this article will tackle, the fact that an expensive hardware tool is needed for flashing most industrial microcontrollers. This issue will be solved by allowing the bootloader application to communicate through the microcontroller's RS232 port. Hence we will develop a serial protocol for writing applications on the embedded device. According to the goals set, this protocol also has to be secure, when flashing an application on the microcontroller the flashed application's machine code should not be available to the outside world.

The bootloader's role is to allow the developer to flash a new application on an MCU without having to connect a JTAG device to it. This also allows for a simpler flashing procedure that the client can also perform when there is a need for a firmware upgrade. The bootloader loads applications only via RS232 and only in a specific, encrypted file format. Hence, the bootloader makes sure that the flashed application is not tampered with by third parties, in which case it should refuse to boot the application. For security reasons in order to upgrade the bootloader a classical flashing process needs to be used, eg. JTAG. The classical use case for the bootloader is when a client has devices all over the world and a firmware upgrade is necessary, then the client's technicians can perform the upgrade by themselves, using the bootloader and the new firmware file received from the developer. This is done in order to avoid bringing in all the devices and flashing them at headquarters, then shipping them back to their client's locations. In order for the bootloader to work properly it should receive via RS232 a valid firmware file (.fw extension). This has to be created by an application, which should also be responsible for encrypting it, we'll call this application "Bundler", since it creates an encrypted file that will contain the application

that the bootloader will load on a microcontroller. In order for the client to be able to program the microcontroller and to use the bootloader he will need an application that will send the firmware file to the MCU, via the serial port, we will call this tool the "Flasher", since it does the flashing of the application, similar to the classical process of programming a microcontroller using a JTAG connector.

The following chapter of the paper will describe the general way a bootloader works, then in the third section the implementation details will be presented, touching the most important parts of the designed system with its characteristic architecture, while the last section provides the conclusions of the paper and the pitfalls of the application that can be improved in future versions.

2 How does a bootloader work?

A bootloader can be found on any system that needs to load an operating system or, in our case, a third party application. The etimology of this word stems from the word "to bootstrap", which means "to pull oneself up by one's bootstraps" [2].

A bootloader is, without exception, the first application that runs on a system. It is loaded in memory (in RAM - Random Access Memory) from non-volatile storage, such as hard-disks or, in the case of microcontrollers, flash memory. After loading, the bootloader (on x86 systems) generally asks the user what operating systems he wants to load. On embedded devices, this usually cannot happen because a microcontroller, by default has reduced input and output capabilities. Also a key difference on the embedded side of things is the fact that a microcontroller has a smaller non-volatile memory than a personal computer has and this allows only one application to be stored at a time on the microcontroller, so the bootloader, after start-up, will automatically choose just this application. But since one of our goals is to allow the user to change the application, so that the embedded system becomes more flexible, we will have to first erase the old application and then write the new one on the microcontroller. This way we still maintain flexibility, but also overcome the small flash memory space embedded devices have.

With this in mind, the initialization sequence of a Cortex M7 ARM microcontroller starts with reading the first two memory locations of the flash memory, which store the Stack Pointer and the Program Counter. The Stack pointer allows the MCU to know where the stack memory area starts and the Program Counter points to the next memory location of the program that needs to be executed. The following program code shows the first few lines of the the Interrupt Vector Table, the location on flash that holds the Stack Pointer, Program Counter and the Interrupt Service Routines, which are called when an interrupt occurs on the microcontroller.

```
__isr_vector:
  .long __StackTop
  .long Reset_Handler
  .long NMI_Handler
  .long HardFault_Handler
  .long MemManage_Handler
  .long BusFault_Handler
  .long UsageFault_Handler
```

The `__StackTop` entry represents the Stack Pointer and the `Reset_Handler` represents the first Program Counter that needs to be loaded when the microcontroller starts, this entry also represents the Interrupt Service Routine for the "reset" signal of the microcontroller [3].

The `Reset_Handler` routine does nothing else other than call the function `SystemInit`, which is responsible with cache initialization and the Floating Point Unit initialization. Next, the `Reset_Handler` will just initialize the heap memory area and will call the `__libc_init_array` which will then prepare the microcontroller for running the C environment. Lastly, the `main` function of the programmed application will be called. After the `main` function is called, the initialization sequence has ended.

As we can see there is no bootloader involved in the process by default, the microcontroller just initializes the necessary memory areas and structures and then proceeds with calling the loaded application. In order to introduce a bootloader into this initialization sequence we will make the bootloader the

application that will be called when calling the `main` function, and only then, the bootloader may call the main function of another application, possibly after receiving this application form the serial port.

The whole initialization sequence of the microcontroller can be seen in the figure 1.

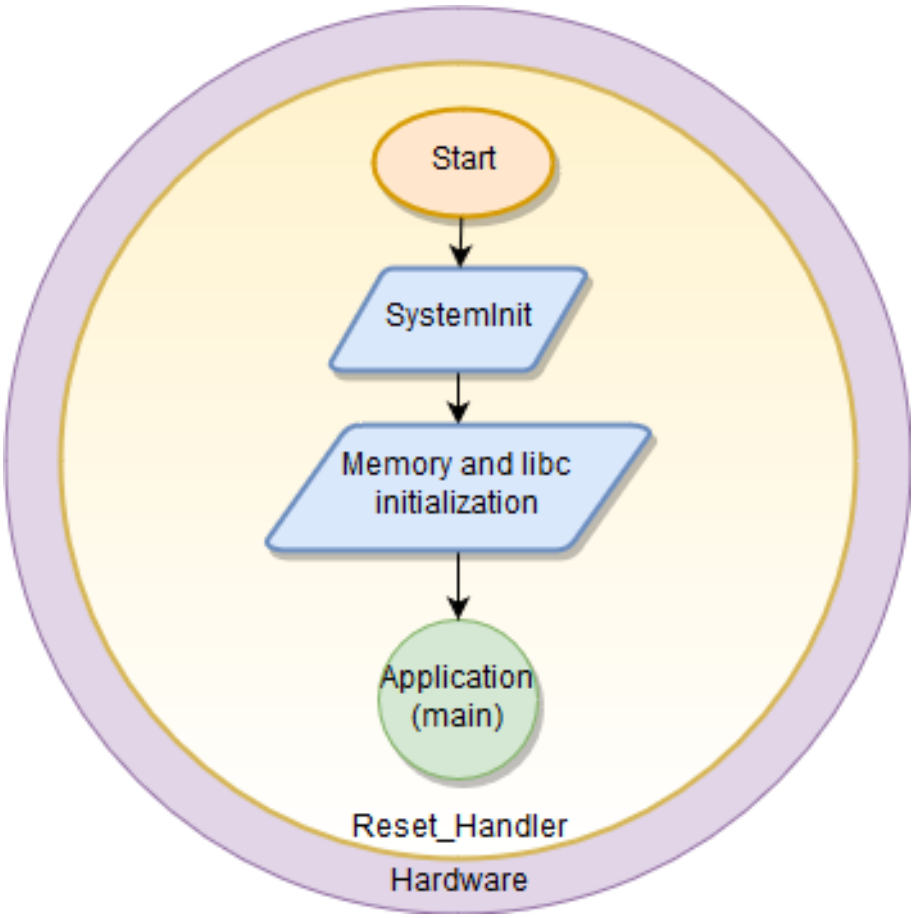


Figure 1: The initialization sequence of an MCU

3 Implementing the bootloader

In the following part we're going to present the technical details of how the bootloader and the helper tools, such as the Bundler and the Flasher are implemented.

As stated briefly in the previous part of this paper, there are three main components:

- the bootloader: an embedded application, which is responsible with managing the microcontroller and receiving the third party firmware file from the serial port. After the firmware is decrypted and loaded, it will be executed.
- the Flasher tool: a PC application, that will be run by the client whenever he wants to change the application on the microcontroller. This tool is responsible with sending the bundled application (the firmware file) to the microcontroller using the serial port.
- the Bundler tool: a PC application (available only to developers) that allows one to create firmware files for the microcontroller. The firmware files are encrypted and only the bootloader may decrypt them. The reason behind this decision was the fact that a potential cracker may intercept a firmware

file and try to reverse engineer its mechanisms and thus, compromising the security of the embedded device.

So in order to maintain security we're going to need to encrypt and decrypt the firmware file. For this job, the symmetric key algorithm Rijndael was chosen, due to its unbroken security [4]. This algorithm is also known as the Advanced Encryption Standard, since it is used by the United States as a standard for encrypting communications [5]. The Rijndael algorithm requires the use of an encryption and decryption key. In order to maintain the security of the firmware file, this key should only be known to the Bundler tool and the bootloader, hence we're going to present how the bootloader and the keys will be laid out in the flash memory of the microcontroller.

3.1 Memory Map

The memory map of the microcontroller represents how the application code and data is laid out in memory. What starting address do they use, and how much space do they occupy.

In order to accomplish all of our goals, the memory will be split in three main sections, the bootloader's code which is the main part of this paper, the encryption keys, which will satisfy the security requirement and finally the user's application which will be decrypted and executed by the bootloader.

For exemplification and testing purposes we chose the MKV58F1M0VLQ24 microcontroller, based on the Cortex M7 ARM core. The microcontroller MKV58F1M0VLQ24 defines its memory map starting with address 0x10000000, this is also the address the bootloader is located at [3].

The memory is divided into the three main sections as the following table shows:

Address	Description
0x10000000	Bootloader's start address
0x10008000	Encryption keys start address
0x1000a000	Application start address

Table 1: The microcontroller's memory map

So the bootloader's size is limited to maximum 32KB (including interrupt vector table and flash config), while the keys may span at most 8KB. The application may span the rest of the space up to the end of the flash memory (1MB – 40KB). The keys are required to span a whole sector on the flash since we need to delete it when the keys are changed. The linker scripts or scatter files of the application have to be modified according to the memory map required by the bootloader, ie. the applications should be placed in memory at 0x1000a000, otherwise they will not be loadable by the bootloader.

Also applications should not write the flash memory between address 0x10000000 and 0x1000a000, otherwise the device is recoverable only via a traditional re-flash, using a JTAG connector.

Please note that if the microcontroller is changed, the memory map and possibly the sector size will change too. That will lead to different starting positions and sizes of the application and the keys.

3.2 The Bundler tool

The Bundler tool is responsible with converting the application executable file (the result of the compilation process) to the firmware file that can be used by the Flasher tool to send to the bootloader. Basically it takes the output of the compilation process (the `.bin` file), encrypts it using an encryption key and writes it back in the `.fw` file format. The resulting `.fw` file is the one that the flasher will use to feed the bootloader. Generally the embedded compilers generate the executable file in the `.hex` format [7]. In order to generate a `.bin` file from a `.hex` file you can use `srec_cat` [6]:

```
srec_cat file.hex -intel -offset - -minimum-addr file.hex -intel -o file.bin -binary
```

If you have the binary file and a key file, an example command for running the Bundler is:

```
bundler.exe keys\xyz.key app\Debug\app.bin
```

After this command runs, we should end up with a `app.fw` file in the `app\Debug\` directory that is suitable for usage with the Flasher tool. Please note that the Bundler only takes two arguments and the order is not relevant. One argument should be the path to the `.key` file and the other one should be the path to the application's `.bin` file.

3.3 The Key files

If you need to generate a new key file, the process is simple (on Linux):

```
head -c 16 /dev/urandom > keys/xyz.key
```

The key files are simple binary files, of size 16 bytes. The 16 bytes represent the key that the Bundler will use to encrypt the `.bin` application with and then store it in the firmware file. It is important that the key used by the Bundler is also the key the bootloader will also use to decrypt the firmware, otherwise the flashing process will fail.

3.4 The Flasher tool

The Flasher tool is used to load a new application on the bootloader enabled MCUs. This is done by sending a firmware bundle to the microcontroller by means of serial communication. The firmware should be received by the client from the developers.

A prerequisite of using the Flasher tool is to first connect the microcontroller and the personal computer using a serial cable and determining the serial port on the PC the cable is connected to. On Windows based machines this can be done by inspecting the `Ports (COM & LPT)` section of the `Device Manager`. The port should look like: COM4.

After the bootloader is started, the client has approximately 20 seconds to properly start the Flasher tool. For the loading to be successful the user needs to use the COM port determined earlier and the path to the firmware file received from the application's developers. In order to run the Flasher tool a command window has to be opened, then the following command has to be run:

```
flasher.exe -p COM4 path\to\app.fw
```

The Flasher takes exactly two arguments, a named one and a positional one:

- `-p COM4`: represents the communication port
- `path\to\app.fw`: this is the path to the firmware file the client received from the developers. This file was created using the Bundler tool.

3.5 The Bootloader

Having to deal with reading a firmware file from the serial port, decrypting it, writing it to the flash memory and loading it, the bootloader is divided into several functional modules that each handle some independent responsibility. The modules will then be combined together in order to achieve the desired functionality: loading a third party application in a secured way.

The architecture of the bootloader may be seen in figure 2.

In this figure we can distinguish the following modules and responsibilities:

- **Flash**: This module is used for writing, reading and erasing the flash memory of the microcontroller.
- **Crypto**: Its role is to manage the cryptograpy library and to provide an abstraction layer over it. The library used here is called `mbedTLS` and it implements the Rijndael algorithm used to encrypt and decrypt the user application.
- **Communication**: Implements the communication channel, it is responsible with opening the serial port, sending and receiving of data and of course is also responsible of properly closing the port.
- **Protocol**: It packages and formats the data in packets in order to facilitate the communication.

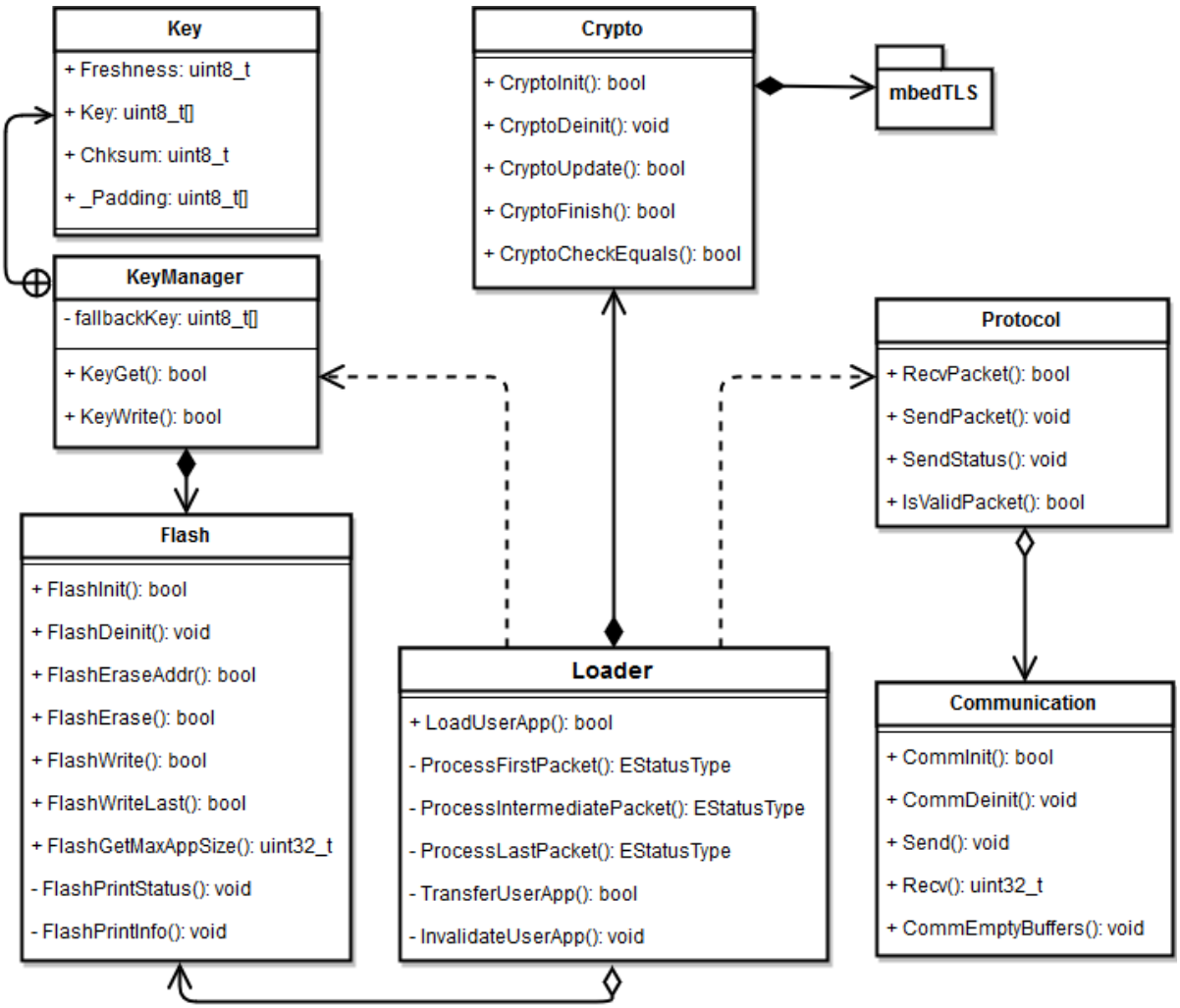


Figure 2: The bootloader's modular architecture

- **KeyManager**: Responsible for writing and reading the encryption keys in their reserved memory areas.

- **Loader**: This is the module that brings all the other functionality together and performs the actual application loading on the microcontroller.

The most complex module of all is the `Loader` module, which combines all the other functionality in order to transfer the application on the serial port, decrypt and execute it. Figure 3 shows how the module works on a high level.

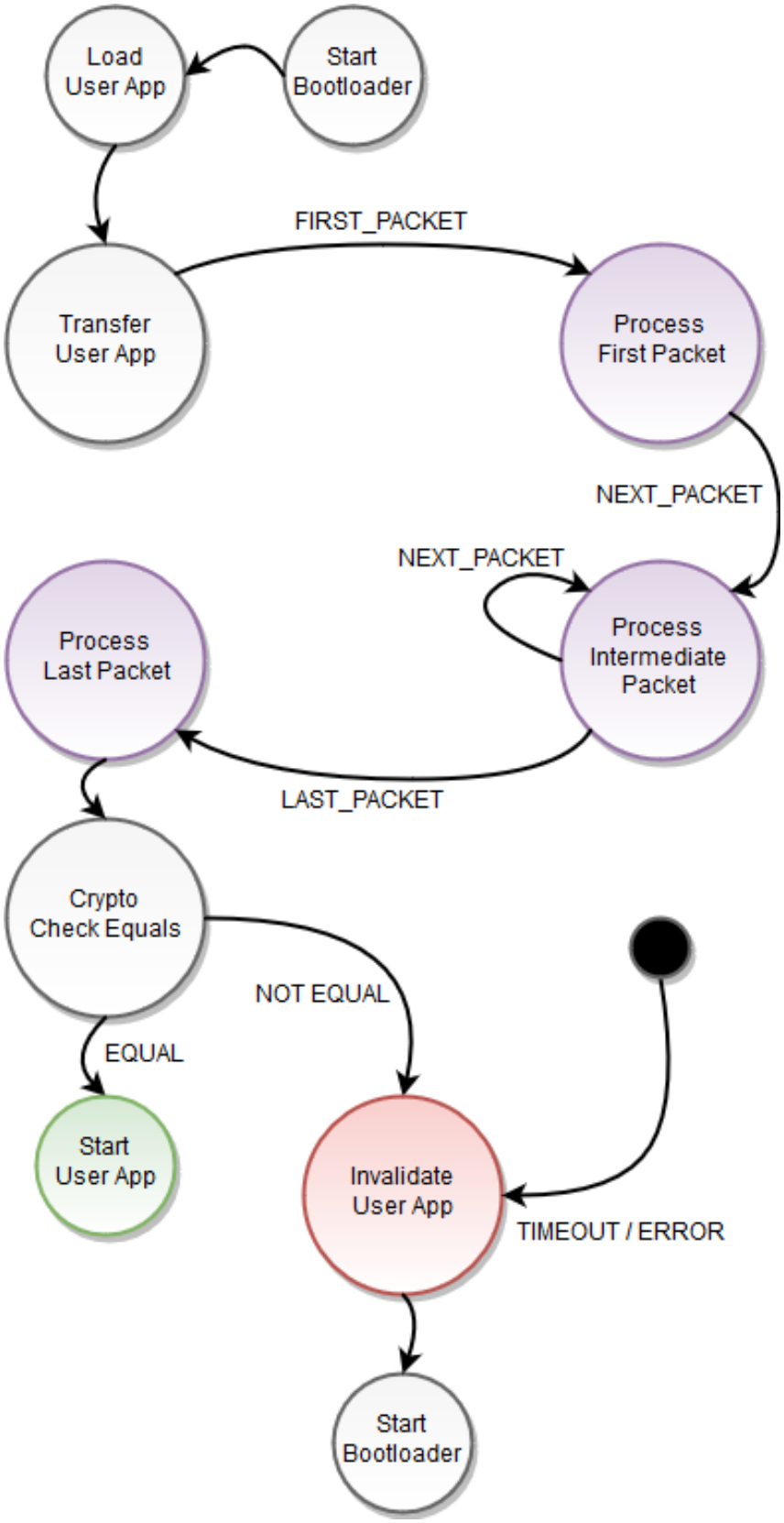


Figure 3: The data flow in the `Loader` module

This data flow is based on the protocol developed specially for the bootloader, the transitions between states are determined by the arrival of different packet types:

- **FIRST_PACKET**: The first packet sent from the Flasher to the bootloader. After this packet only packets of type `NEXT_PACKET` will follow.

- **NEXT_PACKET**: An intermediary packet neither the first, nor the last. The number of these packets sent by the Flasher and received by the bootloader vary with the application's size, the bigger the application, the more packets sent to the bootloader.

- **LAST_PACKET**: This type of packet always follows a `NEXT_PACKET` packet and is surely the last packet that the bootloader will receive.

The states of the `Loader` module represent functions inside the code, these do the processing required after a state transition and hopefully will end up in the `StartUserApp` state, which means that the application was successfully transferred and decrypted and lastly it can be executed.

It should be noted that each of these modules are implemented in a source file, with the `.c` extension and a header file, with `.h` extension.

3.6 The communication protocol

As stated previously, there needs to be a standard way of communicating between the Flasher tool and the bootloader running on the microcontroller. This standard way is implemented by the means of a communication protocol. This protocol is implemented by the `Protocol` module of the bootloader, which is shared by the Flasher tool, too.

The packet structure of the protocol may be seen in the table 2.

Zone	Address	Description	Size (bytes)
Header	0x00	ID (fixed at 0xA5A5)	2
	0x02	Type of packet	1
	0x03	Length	1
Data	0x04	Content of packet	variable
CRC	-	Polynomial code	2

Table 2: Packet structure for the communication protocol

Apart from the fixed ID that starts a packet, the type of packet field may take several values. These values were presented in the last section of this document and in addition to those, there is one more packet type, `STATUS_PACKET` that has the role of communicating the status of the transfer up to this point in time. After the type of the packet, the length field follows, which indicates how many data bytes of the application will be sent from the Flasher to the bootloader in the current packet. After the actual data, a 16 bit CRC is appended to the end of the packet in order to make sure the transmission of the packet is done without errors.

The message sequence diagram in figure 4 illustrates the process of successfully sending a firmware file from the Flasher tool to the bootloader.

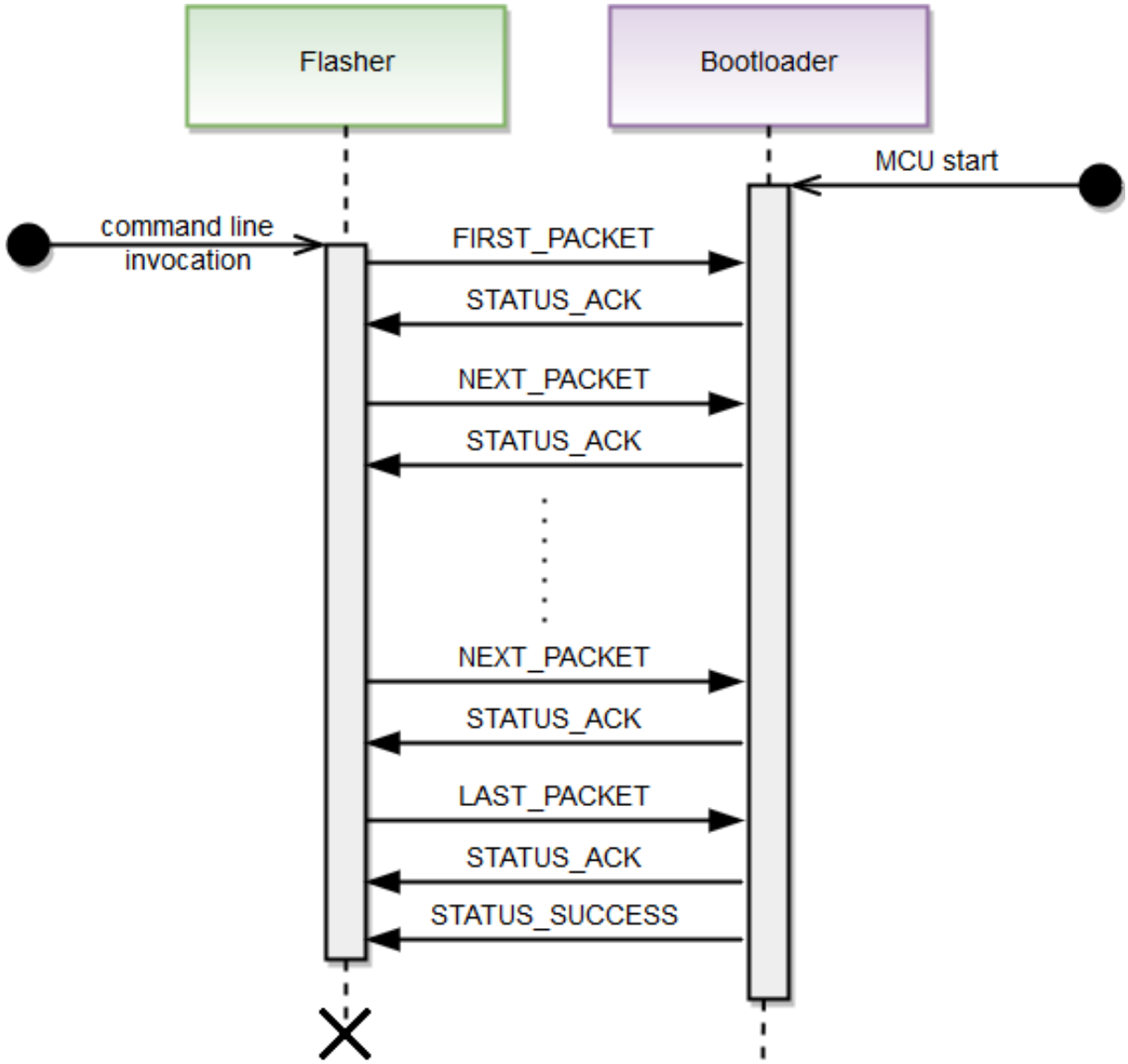


Figure 4: Protocol message sequence diagram

It can be seen in this figure that the transfer stars when both the Flasher and the bootloader are ready to send and receive data and the fact that every packet is followed by a `STATUS_PACKET` indicating that the transfer up to a certain point was successful.

The last two packets are sent by the bootloader, acknowledging the receipt of the last packet and indicating the success of the whole transmission of the firmware file. After the firmware file was successfully transmitted, the bootloader proceeds with executing the third party application.

4 Conclusions and future work

This paper successfully presented the design and implementation of a bootloader for embedded devices. The aim of the bootloader was to allow for cost reductions security and flexibility of embedded applications deployment.

The cost reductions are achieved by allowing the users to program applications on the microcontrollers without the necessity for special programmer hardware. The flashing, when using the bootloader, can

only be done using a serial cable connected to a personal computer. The flexibility comes from the fact that the bootloader allows the client to change the application on the microcontroller whenever there is an update available, without the need to ship the device to the developer so he can program it securely using the special JTAG connectors.

Although the practical implementation on the MKV58F1M0VLQ24 microcontroller was a success, there are still open points that may be improved. One such point would be to allow the bootloader to write multiple applications into the flash memory, given the fact that enough storage space is available. Also the usability of the Bundler and Flasher tools can be improved by implementing a Graphical User Interface, since now they are console based programs. Finally, the portability of the bootloader may be enhanced to other microcontrollers in the future, this task is made simple by the fact that all the modules are written independently. Partly this feature was demonstrated by the fact that the Flasher tool shares the same protocol code with the bootloader, no line of code had to be changed for it to work.

5 References

References

[1] Wikipedia, JTAG *https://en.wikipedia.org/wiki/JTAG*, Accessed March 2017.

[2] Wikipedia, Bootstrapping *https://en.wikipedia.org/wiki/Bootstrapping#Etymology*, Accessed March 2017.

[3] NXP Semiconductors, *KV5x Sub-Family Reference Manual*, June 2016, Revision 4.

[4] Joan Daemen and Vincent Rijmen, *The design of Rijndael: AES — the Advanced Encryption Standard*, 2002, Springer-Verlag.

[5] Wikipedia, Advanced encryption standard *https://en.wikipedia.org/wiki/Advanced_Encryption_Standard*, Accessed February 2017.

[6] Peter Miller, Srecord - Binary Files *http://srecord.sourceforge.net/man/man1/srec_examples.html#BINARY%20FI* Accessed February 2017.

[7] ARM Group, General: Intel Hex File Format *http://www.keil.com/support/docs/1584/*, Accessed February 2017.

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

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

RS.TeamNotifier

István Bialkó

Abstract

This software offers an easy to use, easy to deploy solution to teams formed for university projects, small companies, or even bigger companies with many teams that need to communicate on a regular basis. It is a sever-client based setup, using separate channels and Toast notifications for Windows 10.

1 Introduction

Problems occur when a developer needs access to some resources that are held by a colleague, or want to modify a common environment, use some tools that have an effect on the work of the team. There are platforms that have group chats, but not in every case can you add everybody to it, or not all the colleagues are using that software. Of course you can use group emails, but if the notifications are not set or an application is not open, some people might miss the information. There are a lot of reasons why the notification would be missed.

My design is simple in functions, but its purpose is also just to quickly notify people who need to be informed.

It is also very easy to use for the end-user, install the master service on a server, and to create the database.

The solution is written in C#, using .Net 4.5, and the database is tested in Microsoft SQL Server. I also used some Nuget packages, a few developed at our company, to make it easier to transform some data, and connect the components.

2 Concept

Imagine a scenario: you finished working on a component for the company's software, and it needs to be installed on a common environment (for example a platform which is used by another team). That would mean interrupting their work for the duration, which can take from a few minutes to half an hour or more, in case of errors. They might work on some sensitive data, and those modifications would be lost with the new installation.

You could just email it to everybody, but this update is only for a couple people, why would you bother those who work in sales, or the boss? And this can happen multiple times a day, even every day. It's pointless to overload the email server, or the people's mailbox, this information is specific for that one time, and by the time the installation is over, becomes irrelevant.

3 Design

This software is designed to run on Windows (as a service), and the client uses the new Toast notifications from Windows 10. For other OS-s the code that pops up the notifications has to be modified.

The idea is to have a server, running somewhere accessible to all clients (internal, in the cloud), and this server communicates with the clients and the database. The database contains information about the users (a client service can serve more than one person), last messages and most importantly channels. A basic scheme is shown in "figure 1".

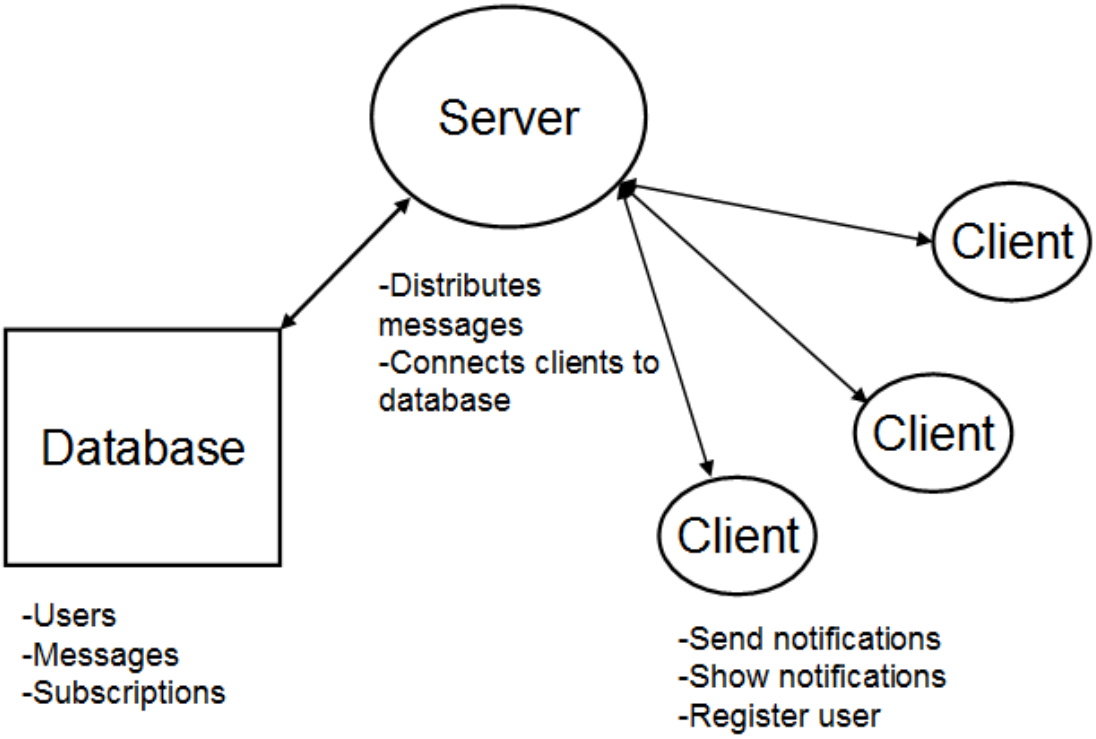


Figure 1: Layout

As a user, you can receive and send messages to any of the channels you signed up for. This means, only those will see your messages who need to. For example, at our firm, we have the Department of Development (Dev), and the Department of Implementation (Imp). This can result in three channels, "Dev", "Imp" and "Dev+Imp". Some people need to subscribe to their own team, and a few for the common channel (like the team leaders). So a message sent to the channel "Dev" won't bother "Imp", but the ones sent to "Dev+Imp" will appear for both teams ("figure 2"). This scenario can be generalized, and applied to any number of departments.

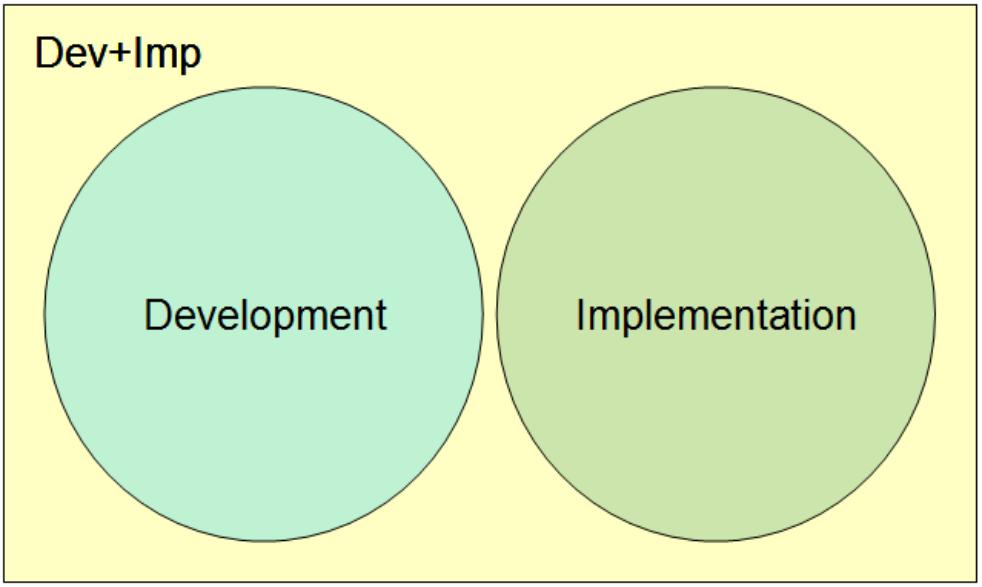


Figure 2: Example of channels

4 Master service

To handle the large number of clients, and to centralize the connection to the database, it was best to implement a master service that can be run on a server accessible to all the clients.

The project is a console application, and on start it creates a Windows service, which contains a few functions to receive messages, register new users, and provide the latest message in the users' respective subscriptions if they are still relevant but were missed. It can be installed as a service only with InstallUtil (a batch file is provided in the solution folder).

The communication between the master service and the clients works on HTTP, as defined in the App.config. The endpoint functions accept one parameter, a wrapper, than can be serialized/deserialized using Newtonsoft Json. To make simpler, these functions define POST method for the parameter.

Functionalities were separated into multiple projects in the solution (a practice I learned at our company). The service doesn't access the database directly. Manager classes are implemented in the business layer (BL) that connect to necessary repository classes in the data access layer (DL or DAL). In DL exists a basic repository that has a public context variable, through which we can access the database. At first this might seem confusing, but it reduces the size and complexity of each project and class.

Since this part of the system is just a central service, it remains relatively simple.

The administrator when installing only has to make sure the database exists (there is an SQL script to create it), and that the connection string is correct.

There is another use of this service: it can be called within any application from the network, notifying users from automated programs, such as installers, deployment managers. This alleviates the need for a user to send messages manually, and on errors alerts a specified group to handle it.

5 Client

This project has more tasks. It starts a service to be able to receive messages from the master service. All the actions performed by the clients are going through the master service, there is no direct connection to the database.

Also the program is made up of two projects: an interface that can be console or windows form application, and a singleton class that manages the service, gets input data from the interface.

Its tasks are:

- On start register the client in the database. It takes the username of the current user logged into Windows, the IP of the machine and sends it to the master service. If the username exists, it will update the IP address, otherwise inserts a new row. By default every user is subscribed to the "default" channel.
- Subscribe to or unsubscribe from a channel. If the channel entered exists, there is no problem, the master service takes care of the action. We have to take into consideration who has the right to create or delete a channel. Can it be anybody, or one of the clients will receive special rights? If anybody can create channels (which could be useful, giving the users freedom to manage small groups between them) one drawback could be mistyped channel names. If that happens, somebody has to manually delete these wrong channels. The master service should not need an interface, and at the moment the client does not support distinguished features.
- Sending and receiving messages. The user can easily send messages to the selected channel. If the interface allows it, selecting the channel can be done using a dropdown menu, or typing it in the console. The master service will select the users subscribed to the selected channel and send them the message. Of course the sender is not in that list. The master will also save the message in the database. The client service has a public function that is accessible by the master service to receive messages.
- Showing notifications. After receiving a message, the client has to show it on the screen. The interface could have a popup window, but Windows has a built in method for handling notifications. On Windows 10, this is called Toast. The user has to enable notifications for the client in Windows Settings.

6 Technical details

The connection between the services is realized by creating service references in both the master and the client service projects. This means that in the project we can create variable to instantiate "clients" which can be consumed by RS.Web. This package allows easy connection and invoking functions on the targeted service (as shown in figure 3 and 4).

Before you can connect to any service, you have to add a service reference to your project. This is necessary to use the service class and its public functions in your code.

Here is an example:

```
//using rs web connect to client service
var client = new ClientServiceReference.RSTeamNotifierClientServiceClient();
var response = string.Empty;
RS.Web.UsingServiceClient.Do(client, c=> response = c.DoWork());
```

Figure 3: Call client from master service

```
1 reference
public string DoWork()
{
    return "Success";
}
```

Figure 4: The function called in the client

To access the database, I used RS.Data package and Entity framework. In the data access layer project after adding the entities of our database, I created a basic repository that has a Datacontext variable, which is basically an interface to the database. Through this variable we can interrogate and update our data. There is a connection string set in the project's App.config, which is used to update the EDMX file (generates classes based on the database), but in the application that references this project we have to specify our database. This basic repository also allows us to create a context on the fly, which is useful if the database name changes over time.

6 Future development

To improve the user experience there will be implemented a graphical interface for the client. It will not have a window, only a taskbar icon. The user is offered a few options: quit the program; send new message to a channel –this will either open a window prompting to select from the subscribed channels, or two text boxes; create a channel; subscribe to a channel; unsubscribe from a channel; change display name.

Because the master service accepts a serialized parameter in a POST method, it can receive instructions from a web page as well. The code that digests the input values from the form has to make sure it is a valid Json string before calling the service function.

There are a few templates available in the Toast notification class, allows greater customization, like showing pictures as well. Beside registering the users' name and display name, it can also save an avatar. Since the description of a notification is an xml string, probably it enables using styles –this can be useful to add extra meaning to a notification, it being urgent or an alert.

Another feature of this system can be multiple devices for a user. Getting notifications on laptop/pc is nice, but there might be a case when the user is not sitting in front of a monitor. Since the components communicate via HTTP, it is very easy to write an IOS, Android or Windows Phone app to be a client. Of course the database and the master service has to be updated to handle multiple devices associated to a user.

The basic client interface uses the Toast system from Windows 10, which is very specific, but the older Windows versions also have a form of notifying the user. The client interface has to recognize the system it is running on and show the notification supported by it.

An installer for the client is also a possible route –it would set the program to start on Windows start up, ask a display name for the user and the address to the master service.

7 Conclusion

The master service centralizes the communication between clients and database. It can be installed anywhere accessible by the clients.

The client program is user friendly, extensible, comes with clear instructions.

I recommend this suite for teams made up of students, developers, where direct discussion is not a solution, or creating a Skype group is difficult. It requires 10 minutes to set up, both the server and client program need little amount of resources.

Acknowledgement: This work was supervised by prof. univ. dr. Grigore Albeanu, from Spiru Haret University, Bucharest.

References

[1] Library of RS.Web: *https://www.nuget.org/packages/RS.Web/* version: 1.51, last updated: April 4, 2017
[2] Library of RS.Data: *https://www.nuget.org/packages/RS.Data/* version: 1.52, last updated: May 4, 2017
[3] Library of RS.Logger: *https://www.nuget.org/packages/RS.Logger/* version: 1.32, last updated: March 24, 2017
[4] Library of RS.Common: *https://www.nuget.org/packages/RS.Common/* version: 1.70, last updated: May 20, 2017
[5] Create a windows service: *https://msdn.microsoft.com/en-us/library/zt39148a%28v=vs.110%29.aspx* last access: May 5, 2017
[6] Create a WCF service: *https://msdn.microsoft.com/en-us/library/bb386386.aspx* last access: May 5, 2017
[7] Entity framework documentation: *https://msdn.microsoft.com/en-us/data/ee712907* last access: May 5, 2017

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

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

Wedoo: Eco-Educational Game

Cristian-Ionuț Cazan, Ionuț Stelian Nicoară, Monica-Andreea Ionescu

Abstract

One of the most burning issues of our current generation is the environmental pollution. During past years, people have been just throwing things away, having no concern whatsoever about where they might end. Studies that are more recent have shown that we produce far more waste than our planet can take. This paper presents an eco-educational game, Wedoo, which aims to increase the people's awareness related to the world pollution issue. The paper also demonstrates how its game engine, especially created for the Wedoo game itself, was continuously refined and still improving so that other new games can use it during their development process.

1 Introduction

Wedoo is a platformer game currently structured on a single level in which the player's mission is to follow Portho's journey through the fictional world. It also acts like an eco-education guide for people that will increase their ecological awareness [1] of pollution in our world.

The character that we control, Portho, is an almighty alien from the Porthonia planet. He had been traveling through different worlds striving to bring beauty and happiness, and of course, to make new friends, when he arrived on Terra. This is when he got stupefied by how dirty our planet looked, so he decided to use his magic on bringing grace back on it. Of course, he can only do that by collecting and recycling the omnipresent garbage and piling waste.

In the end, when he will have had the revelation that our world is a better place, the player can get the magic key, which consists of the knowledge needed for saving the planet. We hope to inspire young people with this key, and awaken them to the reality we are living in, because our planet is "crying" and only we can assuage its pain.

The game is built on top of our CLiX Engine, which has been improved drastically during this process. We have also added some new functionality to the game engine even after the Wedoo project was finished, and we even started developing SonicBat, a new indie game that will hopefully add some networking capabilities to the current engine.

We think that what makes our project special is the fact that the game's educational value can be huge if its message is well applied by players.

Another particular thing of our project is that our CLiX game engine can be used by other

Fig. 1 *Game screenshot when the environment is highly poluated*

developers to create new Microsoft Windows-based games, mostly because it is very simple and intuitive for novice developers.

In the next chapter we will define the concept of eco-education and we will make a short overview some of the other games targeted towards this kind of education. Finally we will present the gameplay of Wedoo.

The third chapter is focused on the development of the engine that is behind Wedoo and the game itself. It presents the technologies used, the initial stages of development and the engine architecture.

Finally the forth chapter contains the conclusion of the paper.

2 Wedoo Game

2.1 Eco-education

Definition 1 *Environmental education is a process that allows individuals to explore environmental issues, engage in problem solving, and take action to improve the environment. As a result, individuals develop a deeper understanding of environmental issues and have the skills to make informed and responsible decisions.[8]*

2.2 Related work

It seems that there are not a lot of videogames focused on environmental education for children and adults, even though the gaming industry is very developed and tends to implement all the new technologies and ideas available. This fact can be seen by simply searching related topics on Google. Nevertheless, there is still hope because even in small numbers they exist.

Most of them are Flash based games that you can download and play for free or run directly in your browser. One such web based game that is dedicated to teaching children about the importance of ecology is Clim`Wai [2]. It tasks the player with upgrading a city and at the same time keeping it eco-friendly. The manual [3] of the game explains not only the gameplay but also some of the scientific aspects like what are Natural Greenhouse Gases or what is Climate Change.

Recycle City [4] is another game that tries to teach people about recycling and eco-friendly practices. The gameplay consists of quizzes designed to test your eco-friendly thinking. If you just want to learn, you can also explore the city map and click on objects in order to get advice about good practices.

Another game that tries to teach people about their impact on the environment is ECO [5]. It is by far the most complex game because it is not as the other ones just based on Flash. It is a multiplayer sandbox experience in which one player can make or break the planet not just for himself but also for the other players on it. This fact makes it the best educational game, in our opinion, because it reflects the real life impact that each and every one of us has on planet Earth.

Even though at the start of this chapter it was mentioned that the game industry does not concern itself with games that educate people on the idea of preserving the natural balance, games like the space themed turn based strategy ones tend to implement the idea of ecologising owned planets in order to maintain the population healthy and productive. This fact may also influence the players of this kind of games to be more caring for the environment in real life.

Now that the stage was set with the existing games we can present what goals Wedoo was built to fulfil.

First on the list is simplicity. In comparison to the games before mentioned, Wedoo is simple to use and understand. All you need to do, as a player, is start the game and choose the “Save the planet” menu option. By doing so you are placed in the middle of the action and you

only need to keep in mind the goal of collecting the objects correctly and the fact that you need to move to the right of the screen with your character.

This makes the gameplay very simple to understand by people of all ages.

Another goal of Wedoo is to be a little challenging in order to have high replayability value. This is achieved through the existence of an energy bar that needs to be maintained above 0 by means of collecting the correct objects before they disappear and through the number of objects generated on screen in a random manner.

The quality of replayability is something that in our opinion only Clim`Wai and ECO possess. The space themed turn based strategy games do not have this property only because of the ecological point of view implemented in their gameplay.

The third and final goal of Wedoo, is to use few computing resources in order to work properly. The browser based flash games are as well low resource consumers but they depend on the browser or a Flash player to work. Wedoo on the other hand uses native Windows infrastructure to function and that is how we achieved low resource usage.

The disadvantage of this is that the game is only playable on Windows platforms.

To sum up by completing those three goals, Wedoo brings a new perspective to eco-educational games. This is achieved by fulfilling the three above mentioned goals.

2.3 Gameplay

The game is currently structured on a single level. The player`s mission is to traverse the world with his avatar Portho and collect the garbage left by humanity. By completing this task he/she will be rewarded with the symbolic key of saving the planet.

The game`s story is that the avatar, Portho, is an alien from the planet Porthonia who travels the universe, intending to make new friends. In his journey he arrives on Terra where he is stupefied by the garbage left by the people of earth, everywhere. During his adventure, he will be confronted with a series of monsters born of people's indifference and in the end he will have the revelation that will save the planet.

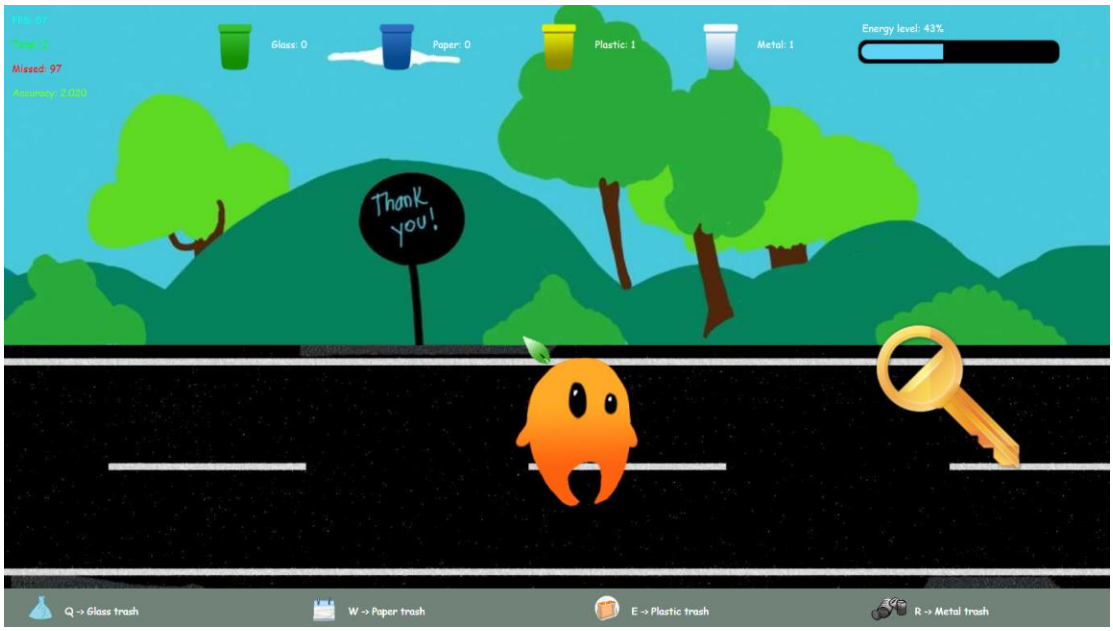


Fig. 2 *Game screenshot when the game is over and the player finds the key to save the planet*

The gameplay is simple and intuitive. Using the left or right arrows the avatar is moved to the left or to the right. The spacebar provides the ability to jump and the collection and sorting of objects is achieved trough the Q, W, E, R keyboard keys.

3 CLiX game engine

3.1 Technologies used

As we designed this engine for game creation, the most important decision that we had to make was about which graphics library to use, so this is where the Microsoft's Direct2D graphics technology was preferred, mainly because it is easier to understand and implement by novice developers, based on the following reasons:

- Microsoft has released a full documentation for its Direct2D graphics library, together with working samples and small projects to demonstrate all of its functionalities [6],
- The OpenGL documentations are mostly outdated, and its code's main goal is to just work rather than be as effective as possible,
- At the time, we wanted to develop something that would run well on the Microsoft Windows operating system.

This also played a decisive role in choosing the Visual Studio 2015 integrated developing environment for our project, mainly because it is very stable and packs many functionalities.

Other software used consists in different small freeware tools required for different graphic manipulations, such as GIMP and Irfan View.

3.2 Initial stages of implementation

Although the CLiX game engine has improved mostly during the development stage of the Wedoo project, it has an even longer history. The engine took its first relevant shape after some research had already been done, and some small projects had been finalized, each of them bringing some new functionalities. During these projects, we tried different technologies, especially graphic libraries such as SDL, OpenGL and Direct2D.

Some relevant early projects are the following:

- BlackTV, a small utility written using Direct2D; its role was to search for a specific display, and once found, cover it with a fullscreen black background, so that it would consume less power; the main functionalities added by this application consist in display operations, such as fullscreening, setting different resolutions, etc.
- GlowingBalls, a colourful entertaining application, similar to a screensaver, which pops different animated circles on the screen, and then slowly fades while moving in different positions; This brought the Particle System functionality in the CLiX engine, heavily used in the Wedoo application too.
- PathFinder, designed initially as a map explorer sketch, implemented some parallax [7] functionality in the engine.

3.3 Engine Architecture

3.3.1 System Layer

All of our CLiX Engine functionality is built over the system layer consisted of a form window which uses a Direct2D graphic surface. Both of these are managed by the Microsoft Windows operating system alone.

3.3.2 Abstraction Layer

Over the system layout we built a custom abstraction layer, which does nothing more than translating and adapting the required, and often more-complex, Windows and Direct2D API calls to simpler and more intuitive custom methods and objects.

One such example is the custom method for drawing an ellipse having its center at the x and y coordinates, with a custom [Red, Green, Blue, Alpha] color, so that the programmer can easily call it without requiring direct access to Direct2D graphic objects:

```
void MainWindow::DrawEll(float x, float y, float size, float r, float g, float b, float a)
{
   pBrush->SetColor(D2D1::ColorF(r, g, b, a));
   d2drender->FillEllipse(D2D1::Ellipse(D2D1::Point2F(x, y), size, size), pBrush);
};
```

Other important methods under this layer are the routines which initialize all the window form and Direct2D environment in only a couple methods which calls automatically the system APIs required.

3.3.3 Custom Functionality Layer

This layer uses all functionality offered by the abstraction layer and creates some interesting and useful tools and graphic effects which can be used in the game.

Wedoo game uses the following functionalities:

- Particle generator: This object offer the functionality of instantiating animated particles using filled circles; every of these will have an individual position, direction, fade time, color and size.

- Flyer generator: An object similar to the particle generator used for generating animated text objects, which will softly fade away while moving upwards/downwards.

- Image Object: Each object will load a specific graphic image, and animate it by doing specific operation with its size, position, rotation angle, etc. For the moment, one limitation is that we cannot colorize it using specific colors.

- Metrics: This is a group of objects and parameters which contains system-obtained data, such as screen resolution, mouse position, keyboard keys status (pressed, released), specific events received from the operating system, etc.

3.3.4 Logic Layer

Being the highest-level layer, the Logic Layer offers the developer all the engine functionality in a structured way.

The developer has the responsibility of overwriting two CLiX Engine methods which will be called automatically by the engine at every single frame:

- onRender - Rendering routine; this is the place where the developer specifies what graphic objects has to be rendered; each object has its own render method which will automatically do its job using their individual logic and parameters.
- onUpdate - Logic update routine, better described as the routine responsible for the actual gameplay; this is where operations like reposition, resize and changes of colors are applied.

Of course, in a usual project the biggest part of this layer is formed by custom application-specific routines written by the developer.

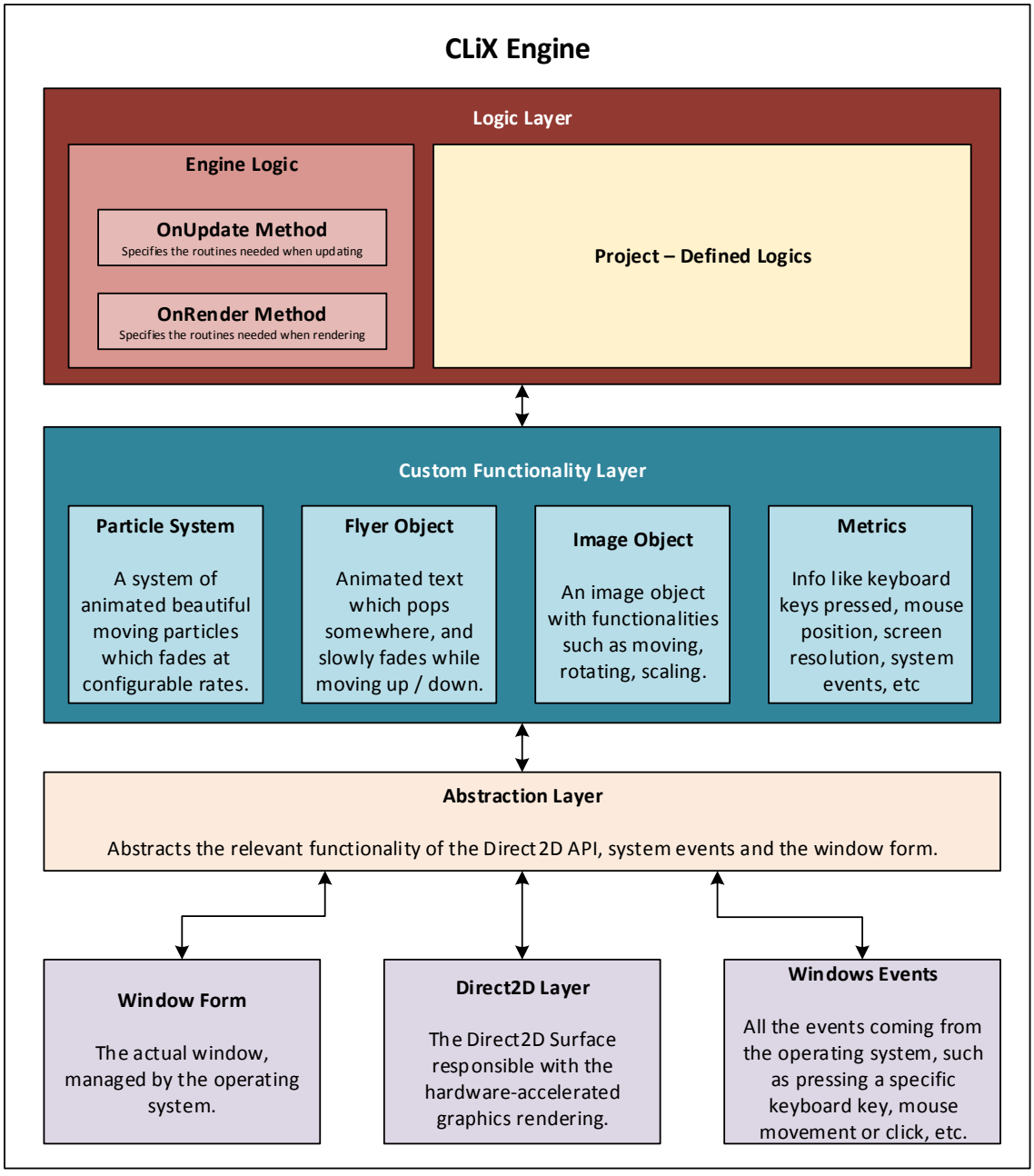


Fig. 3 *– CLiX Engine Architecture*

3.4 Unexpected situations during development

A few unexpected situations raised while developing the Wedoo game. Ordinary problems like pointers mishandling and small design flaws popped very often, but almost all of them were solved in less than thirty minutes. But one which is worth to mention is actually the strangest one: The application was running flawlessly on one computer, but instantly crashing on another computer, although they were running the same operating system with the same software packages installed.

Yet another weird thing about this bug is that it was hard to spot until final stages of development, when it was deployed on other computers for testing purposes. That meant that we had to use one of these computers to track the source of the problem, and it was not quite easy.

A considerable amount of time was spent investigating possible mishandling of pointer allocation / deallocation or misuse of variables values, and of course, a lot of debugging. It seemed that all of these actions were in vain. The solution came when we tried to simplify the whole application by cutting off whole portions of functionality, narrowing down to an application which plotted some textured objects on the screen.

It seemed that one of those textures was instantly crashing the whole application when rendered on some integrated video chipsets, because it was very large in memory, so the guess it was that some video chipsets simply could not handle such big textures. The first saving solution was to reduce the quality of the texture, but it sounded like a compromise for the overall feeling of the game. This led to the second and final solution: to split the texture in multiple parts and rendered separately.

3.5 Future development

Currently we are working at a new indie game named „SonicBat" starting from the same CLiX Engine used at the development process of Wedoo, and there were already a couple of new things implemented in the base engine. The most interesting ones are:

- Sprite Objects, which the ability of using sprites for character animations,
- Network functionality, for multiplayer games,
- More animation effects and properties for Image Objects.

Of course, most of these functionalities are not completely working yet and are not fully tested, but they demonstrate that the future of this game engine looks aspiring.

Meanwhile, it is mandatory that the source code has to be improved and optimized, and a very good inspiration for this is the work of Scott Meyers with his "Effective C++ Digital Collection: 140 Ways to Improve Your Programming"[9], which already helped with some good designs such as template-based containers and some helpful methodologies regarding the C++ code.

4 Conclusions

During the creation of this project we had a lot of good and bad surprises and we learned a lot of important lessons about working as a team, about the design process of a game, and we also got many new technical skills. But maybe the most important thing was the creation of the Wedoo game itself, as we had been working towards creating an educational game, for both children and adults who forgot to take care of our planet. We hope this will inspire and encourage future generations to help the Earth, not only through recycling, but also by engaging in different activities that can stop pollution and help the healing process of our nature.

Also, the most important scope of this publication is to give a start point for any beginner developer which wants to build simple games and maybe offer their contribution with any design concept ideas, functionality concepts or even code related modifications.

Acknowledgement: This work was supervised by Professor *Conf. Dr. Inf. Gabriela Moise,* affiliated to *iTIMF* department of *Litere și Științe* faculty from *Petroleum and Gas University of Ploiești.*

All the graphic artwork related to the Wedoo project was created by *Andreea Giulia Drogoreanu*.

References

[1] S. Jeffery, Ecological Awareness, https://scottjeffrey.com/ecological-awareness/, Last accessed date 05.08.2017.
[2] Clim'City, http://climway.cap-sciences.net/us/index.php, Last accessed date 05.05.2017.
[3] Clim'City, http://climway.cap-sciences.net/us/livret-clim-InEnglish.pdf, Last accessed date 05.05.2017.
[4] Recycle city, https://www3.epa.gov/recyclecity/mainmap.htm, Last accessed date 05.08.2017.
[5] ECO, https://www.strangeloopgames.com/eco, Last accessed date 05.08.2017.
[6] Official Microsoft Direct2D Documentation, https://msdn.microsoft.com/en-us/library/windows/desktop/dd370990(v=vs.85).aspx, 2016.
[7] Dennis Ku, *Parallax Scrolling – To Scroll or not to scroll,* Institutionen för informatik, 2015.
[8] United States Environmental Protection Agency, https://www.epa.gov/education/what-environmental-education, Last accessed date 05.08.2017.
[9] Scott Meyers, *Effective C++ Digital Collection: 140 Ways to Improve Your Programming*, Addison-Wesley, July 10, 2012

CRISTIAN-IONUȚ CAZAN
UPG Ploiești
iTIMF
Bulevardul București 39, Ploiești
ROMÂNIA
E-mail: cristicazan94@gmail.com

MONICA-ANDREEA IONESCU
UPG Ploiești
iTIMF
Bulevardul București 39, Ploiești
ROMÂNIA
E-mail: ionescu.monica.andreea@gmail.com

IONUȚ-ȘTELIAN NICOARĂ
UPG Ploiești
iTIMF
Bulevardul București 39, Ploiești
ROMÂNIA
E-mail: yonutz9337@gmail.com

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

Proserpina - The Godness of the Underworld

Cristiana Constantinescu

Abstract

Starting from the implementations of mathematical and theoretical models of cryptography, the present paper reveals a new and personal concept of steganography, with improved security levels and resistance to detection analysis. This paper is the result of personal commitment, hard work and research, based on most secure cryptographic algorithms known so far and also on advanced and clearly described mathematical theories with applications in cryptography. The originality of the present paper mostly relies on the proposal of a new encryption algorithm meant to provide an enhancement of data security, of cryptanalysis attack resistance and also of the execution time.

1 Introduction

Steganography was firstly introduced thousand years ago and was originally used with no mathematical background [3, 4]. The current concepts of steganography are the same as they were when it was introduced and they rely on the same idea, the one of hiding data in plain sight; past interpretation of the concept referred to small amount of data encoded in different masks that are not too hard to be processed by human abilities, whilst current concepts are limited in the size of used data only by available computer resources. The ideal form of steganography, however, suggest that, having a mask concealing the desired data, one could not detect the presence of foreign data and further more, if this desideratum is broken, it should be impossible to one to extract and recompose the data in such way that the extracted information keeps its original integrity and value. The algorithm proposed in this paper aims to be as close as possible to the ideal steganography state.

Steganography is now heavily related to mathematical models and computer science applications and so it owns today a variety of implementations, based on the needs of certain categories of consumers. Briefly and roughly explained, any type of data can now be encoded in any type of data (e.g.: pictures hidden in pictures, documents hidden in sounds etc). The data type chosen to be encoded and the chosen mask are selected based on the consumers' need and have a huge impact on the information's integrity and security. At the moment, the market holds steganographic implementations mostly based on images hidden in other images, or text hidden in images.

The steganographic concept treated in the present paper is based on hiding, encoding and encrypting text in pictures.

2 State of Steganographic Art

A trusted cryptographic system is described by the complexity of attack resources it requires to potential assailants and by the attack resistance it provides to potential harms that can be brought. Several attack implementations have been studied up to the present. [11, 12] Steganographic systems are not at all different. Studding different concepts and implementations of cryptographic algorithms and data protection techniques, as those described in [1, 2, 5, 8, 13, 14], I have learnt that, in the development of

cryptographic applications, rising the attack costs and offering an as high as possible number of potential leads of data location is crucial for the applications' security.

Besides the chosen technique for raw data manipulation, an important factor of impact in applications' security is given by the chosen algorithm for data encoding. Currently, the most common used encoding algorithm, in steganographic applications, is the LSB (Least Significant Bit) algorithm.

'Computers speak binary' and for any data readable by the human eye, there is a binary translation that computers process, use and manipulate in order to provide 'humanly readable' output. So are the pictures. In the case of pictures, if a single bit value of their binary representation is converted from 0 to 1, or vice versa, two possible scenarios rise for the output result: the difference is either so big and obvious for both the computer and the human eye (if the modified bit is a significant one), or obvious and detectable only for the computer, whilst the human eye can not detect any modifications (if the modified bit is the least significant one).

The LSB algorithm speculates this property of pictures' binary data representation that allows modifications to be brought to it without any perceptible difference of the resulted output, in the interest of steganography. Expressed in a different manner, this property is suitable for the steganographic ideal of being imperceptible and the LSB algorithm took advantage of it.

Let P be a picture of height = h and width = w, having that h,w $\in \mathbb{N}^*$. Knowing that p_{ij}, i = $\overline{0, w-1}$, j =$\overline{0, h-1}$, is the smallest structural unit of the picture P. In binary encoding, each p_{ij} can be written as

$$p_{ij} = (x_0, x_1, ..., x_{n-1}), \tag{1}$$

with n = $\mu \times l$, l $\in \mathbb{N}^*$, μ being the color depth chosen for the binary representation, $x_k \in \{0,1\}$, k = $\overline{0, n-1}$. The x_{n-1} is the binary bit with least impact in data's visual modification. So, given a binary message

$$M = (m_0, m_1, ..., m_{t-1}), \tag{2}$$

with t $\in \mathbb{N}^*$, the bit length of the binary representation,

$$m_q \in \{0,1\}, q = \overline{0, t-1}, \tag{3}$$

if the general LSB algorithm implementation is applied on the tuple (P,M), each value of each least significant bit $x_{n-1} \in p_{ij}$ $\forall i, j \in \mathbb{N}^*$ will be altered with the value of each $m_q \in M$, resulting a new image P'.

Given this general universal type of LSB algorithm implementation, although the binary stream of the two *P* and *P'* is noticeable different for the computer, not a single difference can be detected by the human eye between the original input *P* and the resulting output picture *P'*.

The security of this type of LSB algorithm relies on the presence of any suspicion over the existence of a steganographic algorithm. If this suspicion exists and the attacker proceeds to decompose the picture, than the security level directly depends on the deduction of the channel the least significant bit belongs to and the eventual protection of the hidden information.

Keeping an implementation of the LSB algorithm close to this general one, the encoded data is not protected by numerous, nor strong security layers.

This paper presents a new steganographic algorithm with fundaments in the LSB algorithm and with usage of clearly defined mathematical concepts and cryptographic theories that are meant to enhance the security and data integrity under possible attacks and detection analysis. This proposed algorithm is named *Pixel StH* and will be following described in the present paper.

3 *Pixel StH* Algorithm

Encryption and codification are related as concept but placed apart as provided level of security. While codification protocols are easier to be broken or reproduced by a third party, other than Alice and Bob(fictional characters used in the theory of cryptography, who represent the autentificated entities involved in a communication session, aside Eve who wants to break the protection of their shared data), just by the simple fact of the third party observing the communication between Alice and Bob and theirs

gestures or actions after each communication session, encryption brings in discussion an extra layer of security, allowing Alice and Bob to encode their communication in an unknown and meaningless (for the third party) strings of data. In theory of mathematical models, both codification and encryption are described as bijective functions between a domain and a codomain, as described in [1], the only difference being noticeable in the set of parameters each of them use. Aside the domain value, encryption uses a second parameter that is not present in codification function, called *key*. The *key* provides, in addition to the encoding process, a mathematical meaning and pattern, transforming it in encyption.

Starting with this statement, that defines the difference between encryption and codification as a security manner, I have studied and implemented an algorithm meant to bring steganography closer to its cryptographic meaning, rather than its codification aspects. In other words, the *Pixel StH* algorithm I propose carries an extra-parameter meant to amplify the security of steganographic models that embrace it.

Let Ψ be an input picture of λ height and σ width, λ,$\sigma \in N^*$. Having ψ_{lm}, with $l = \overline{0, \sigma - 1}$, $m = \overline{0, \lambda - 1}$, as being the smallest structural unit of the picture Ψ, the input picture Ψ can be written as

$$\Psi_{\sigma\times\lambda} = \begin{Bmatrix} ((\psi_0^0)_0, (\psi_0^0)_1, ..., (\psi_0^0)_n)) & ((\psi_0^1)_0, (\psi_0^1)_1, ..., (\psi_0^1)_n)) & \dots & ((\psi_0^{\sigma-1})_0, (\psi_0^{\sigma-1})_1, ..., (\psi_0^{\sigma-1})_n)) \\ ((\psi_1^0)_0, (\psi_1^0)_1, ..., (\psi_1^0)_n)) & ((\psi_1^1)_0, (\psi_1^1)_1, ..., (\psi_1^1)_n)) & \dots & ((\psi_1^{\sigma-1})_0, (\psi_1^{\sigma-1})_1, ..., (\psi_1^{\sigma-1})_n)) \\ \vdots & \vdots & \ddots & \vdots \\ (\psi_{\lambda-1}^0)_0, (\psi_{\lambda-1}^0)_1, ..., (\psi_{\lambda-1}^0)_n)) & ((\psi_{\lambda-1}^1)_0, (\psi_{\lambda-1}^1)_1, ..., (\psi_{\lambda-1}^1)_n)) & \dots & ((\psi_{\lambda-1}^{\sigma-1})_0, (\psi_{\lambda-1}^{\sigma-1})_1, ..., (\psi_{\lambda-1}^{\sigma-1})_n)) \end{Bmatrix}$$

Let Φ be an input message of length ξ, with $\xi \in \mathbb{N}^*$, represented by the formula

$$\Phi = (\phi_0, \phi_1, ..., \phi_{\xi-1}). \tag{4}$$

Suppose Alice wants to send Bob a picture containing steganographic content described by *Pixel StH* algorithm. As said before in the present paper, one of the innovative aspects of the algorithm is brought by the key. In order to obtain a secure key, all that Alice has to do is to provide a string (in any combination of characters) that will serve as a seed for data manipulation of Diffie-Hellman type. Mathematical models and fundaments that serve as base for the Diffie-Hellman machine encapsulated in this algorithm are touched in [11, 10].

Let Ω be the key Alice provides. Ω can be represented by any set of characters and is described by the following formula:

$$\Omega = (\omega_0, \omega_1, ..., \omega_{\theta-1}), \tag{5}$$

where $\theta \in \mathbb{N}^*$, representing the key length.

After data manipulation and pseudorandom generation of the number ε (large prime number), the secure key (Υ) of the session is generated from the seed provided by Alice.

At this state, the representation of Υ is the following $\Upsilon = \Omega^{\varepsilon} = (\mu_0, \mu_1,...,\mu_{x-1},\mu_x,...)$, where $x \in \mathbb{N}^*$ is a large number and each $\mu_x \in \mathbb{N}^*$.

Bringing the secure key Υ to context of the picture

$$\Psi_{\sigma\times\lambda} = \begin{Bmatrix} ((\psi_0^0)_0, (\psi_0^0)_1, ..., (\psi_0^0)_n)) & ((\psi_0^1)_0, (\psi_0^1)_1, ..., (\psi_0^1)_n)) & \dots & ((\psi_0^{\sigma-1})_0, (\psi_0^{\sigma-1})_1, ..., (\psi_0^{\sigma-1})_n)) \\ ((\psi_1^0)_0, (\psi_1^0)_1, ..., (\psi_1^0)_n)) & ((\psi_1^1)_0, (\psi_1^1)_1, ..., (\psi_1^1)_n)) & \dots & ((\psi_1^{\sigma-1})_0, (\psi_1^{\sigma-1})_1, ..., (\psi_1^{\sigma-1})_n)) \\ \vdots & \vdots & \ddots & \vdots \\ (\psi_{\lambda-1}^0)_0, (\psi_{\lambda-1}^0)_1, ..., (\psi_{\lambda-1}^0)_n)) & ((\psi_{\lambda-1}^1)_0, (\psi_{\lambda-1}^1)_1, ..., (\psi_{\lambda-1}^1)_n)) & \dots & ((\psi_{\lambda-1}^{\sigma-1})_0, (\psi_{\lambda-1}^{\sigma-1})_1, ..., (\psi_{\lambda-1}^{\sigma-1})_n)) \end{Bmatrix}$$

each pair

$$(\mu_x \mod \sigma, \mu_{x+1} \mod \lambda) \tag{6}$$

serves as a pattern for units that obey the formula $\psi_{\mu_x \bmod \sigma}^{\mu_{x+1} \bmod \lambda}$. Structure units of this shape will be chosen to retain, in their least significant bit, the equivalent of each value of the message Φ encryption's binary representation.

Each value $\psi_{\mu_x \bmod \sigma}^{\mu_{x+1} \bmod \lambda}$ is the equivalent of a n-tuple, n $\in \{3, 4\}$, representing available channels that allow data manipulation. For a 4-tuple, to avoid visual modification, any c channel, c $= \overline{0,3}$ can be used.

Example 1 *Let Ψ be an input picture of height 800 and width 450. Then, Ψ is represented by the formula*

$$\Psi_{800\times450} = \begin{Bmatrix} ((\psi_0^0)_0,(\psi_0^0)_1,....,(\psi_0^0)_n)) & \underline{((\psi_0^1)_0,(\psi_0^1)_1,....,(\psi_0^1)_n))} & ((\psi_0^2)_0,(\psi_0^2)_1,....,(\psi_0^2)_n)) & \dots & ((\psi_0^{449})_0,(\psi_0^{449})_1,....,(\psi_0^{449})_n)) \\ ((\psi_1^0)_0,(\psi_1^0)_1,....,(\psi_1^0)_n)) & ((\psi_1^1)_0,(\psi_1^1)_1,....,(\psi_1^1)_n)) & \underline{((\psi_1^2)_0,(\psi_1^2)_1,....,(\psi_1^2)_n))} & \dots & ((\psi_1^{449})_0,(\psi_1^{449})_1,....,(\psi_1^{449})_n)) \\ ((\psi_2^0)_0,(\psi_2^0)_1,....,(\psi_2^0)_n)) & ((\psi_2^1)_0,(\psi_2^1)_1,....,(\psi_2^1)_n)) & ((\psi_2^2)_0,(\psi_2^2)_1,....,(\psi_2^2)_n)) & \dots & ((\psi_2^{449})_0,(\psi_2^{449})_1,....,(\psi_2^{449})_n)) \\ \vdots & \vdots & \ddots & \vdots & \\ ((\psi_{10}^0)_0,(\psi_{10}^0)_1,....,(\psi_{10}^0)_n)) & ((\psi_{10}^1)_0,(\psi_{10}^1)_1,....,(\psi_{10}^1)_n)) & ((\psi_{10}^2)_0,(\psi_{10}^2)_1,....,(\psi_{10}^2)_n)) & \dots & ((\psi_{10}^{449})_0,(\psi_{10}^{449})_1,....,(\psi_{10}^{449})_n)) \\ \vdots & \vdots & \ddots & \vdots & \\ (\psi_{798}^0)_0,(\psi_{798}^0)_1,....,(\psi_{798}^0)_n)) & ((\psi_{798}^1)_0,(\psi_{798}^1)_1,....,(\psi_{798}^1)_n)) & ((\psi_{798}^2)_0,(\psi_{798}^2)_1,....,(\psi_{798}^2)_n)) & \dots & ((\psi_{798}^{449})_0,(\psi_{798}^{449})_1,....,(\psi_{798}^{449})_n)) \\ (\psi_{799}^0)_0,(\psi_{799}^0)_1,....,(\psi_{799}^0)_n)) & ((\psi_{799}^1)_0,(\psi_{799}^1)_1,....,(\psi_{799}^1)_n)) & ((\psi_{799}^2)_0,(\psi_{799}^2)_1,....,(\psi_{799}^2)_n)) & \dots & ((\psi_{799}^{449})_0,(\psi_{799}^{449})_1,....,(\psi_{799}^{449})_n)) \end{Bmatrix}$$

Suppose having the following key Ω that obeys the formulas described previously, and a pseudorandom large prime number ε that have led to the message $\Upsilon = \Omega^{\varepsilon}$ = (449,7991,451,452,908,809,....,1596,1599).

This leads to a set

$$\chi = (\psi_{449}^{7991 \bmod 799 \bmod 449}, \psi_{451}^{7992 \bmod 7991 \bmod 449}, \psi_{908}^{809 \bmod 799 \bmod 449}, ..., \psi_{764}^{1599 \bmod 799 \bmod 449}) = \\ = (\psi_0^1, \psi_2^2, \psi_{10}^{10}, ..., \psi_{315}^{252}),$$

that represents the string of the picture's smallest structural units ready to be modified. This is the pattern of the units that will accept data alteration on their least significant bit.

If the Pixel StH algorithm is applied, a new picture Ψ' is generated, with the following structure:

$$\Psi_{800\times450} = \begin{Bmatrix} ((\psi_0^0)_0,(\psi_0^0)_1,....,(\psi_0^0)_n)) & \underline{((\psi_0^1)_0,(\psi_0^1)_1,....,(\psi_0^1)_n))} & ((\psi_0^2)_0,(\psi_0^2)_1,....,(\psi_0^2)_n)) & \dots & ((\psi_0^{449})_0,(\psi_0^{449})_1,....,(\psi_0^{449})_n)) \\ ((\psi_1^0)_0,(\psi_1^0)_1,....,(\psi_1^0)_n)) & ((\psi_1^1)_0,(\psi_1^1)_1,....,(\psi_1^1)_n)) & \underline{((\psi_1^2)_0,(\psi_1^2)_1,....,(\psi_1^2)_n))} & \dots & ((\psi_1^{449})_0,(\psi_1^{449})_1,....,(\psi_1^{449})_n)) \\ ((\psi_2^0)_0,(\psi_2^0)_1,....,(\psi_2^0)_n)) & ((\psi_2^1)_0,(\psi_2^1)_1,....,(\psi_2^1)_n)) & ((\psi_2^2)_0,(\psi_2^2)_1,....,(\psi_2^2)_n)) & \dots & ((\psi_2^{449})_0,(\psi_2^{449})_1,....,(\psi_2^{449})_n)) \\ \vdots & \vdots & \ddots & \vdots & \\ ((\psi_{10}^0)_0,(\psi_{10}^0)_1,....,(\psi_{10}^0)_n)) & ((\psi_{10}^1)_0,(\psi_{10}^1)_1,....,(\psi_{10}^1)_n)) & ((\psi_{10}^2)_0,(\psi_{10}^2)_1,....,(\psi_{10}^2)_n)) & \dots & ((\psi_{10}^{449})_0,(\psi_{10}^{449})_1,....,(\psi_{10}^{449})_n)) \\ \vdots & \vdots & \ddots & \vdots & \\ (\psi_{798}^0)_0,(\psi_{798}^0)_1,....,(\psi_{798}^0)_n)) & ((\psi_{798}^1)_0,(\psi_{798}^1)_1,....,(\psi_{798}^1)_n)) & ((\psi_{798}^2)_0,(\psi_{798}^2)_1,....,(\psi_{798}^2)_n)) & \dots & ((\psi_{798}^{449})_0,(\psi_{798}^{449})_1,....,(\psi_{798}^{449})_n)) \\ (\psi_{799}^0)_0,(\psi_{799}^0)_1,....,(\psi_{799}^0)_n)) & ((\psi_{799}^1)_0,(\psi_{799}^1)_1,....,(\psi_{799}^1)_n)) & ((\psi_{799}^2)_0,(\psi_{799}^2)_1,....,(\psi_{799}^2)_n)) & \dots & ((\psi_{799}^{449})_0,(\psi_{799}^{449})_1,....,(\psi_{799}^{449})_n)) \end{Bmatrix}$$

where each underlined element represents one of the units of structure that contain the message Υ data.

4 Algorithm Description

Algorithm 1: *Pixel StH* Steganographic algorithm

```
1  StH (a, b)
   Input  : Original picture P, text message T, key seed S
   Output: Encrypted picture
2  if P is in valid format then
3  |  return binary representation of P;
4  else
5  |  require other P; break;
6  end
7  if T length ≠ 0 then
8  |  return binary representation of T;
9  else
10 |  require T;
11 end
12 if S length ≠ 0 then
13 |  generate Diffie-Hellman-like key;
14 else
15 |  use default S;
16 end
17 while binary representation of T is not entirely looped do
18 |  bit← current bit of T binary rep;
19 |  position← correspondent structure unit in binary rep of P;
20 |  LSB← least significant bit of position's chosen channel P;
21 |  LSB← bit;
22 end
23 form Encrypted picture;
24 return Encrypted picture;
```

5 Algorithm Complexity Analysis and Comparison

As encryption is the improved interpretation of codification, due to security enhancement and mathematical complexity of the models it implements, *Pixel StH* represents an improved interpretation of classic LSB algorithms.

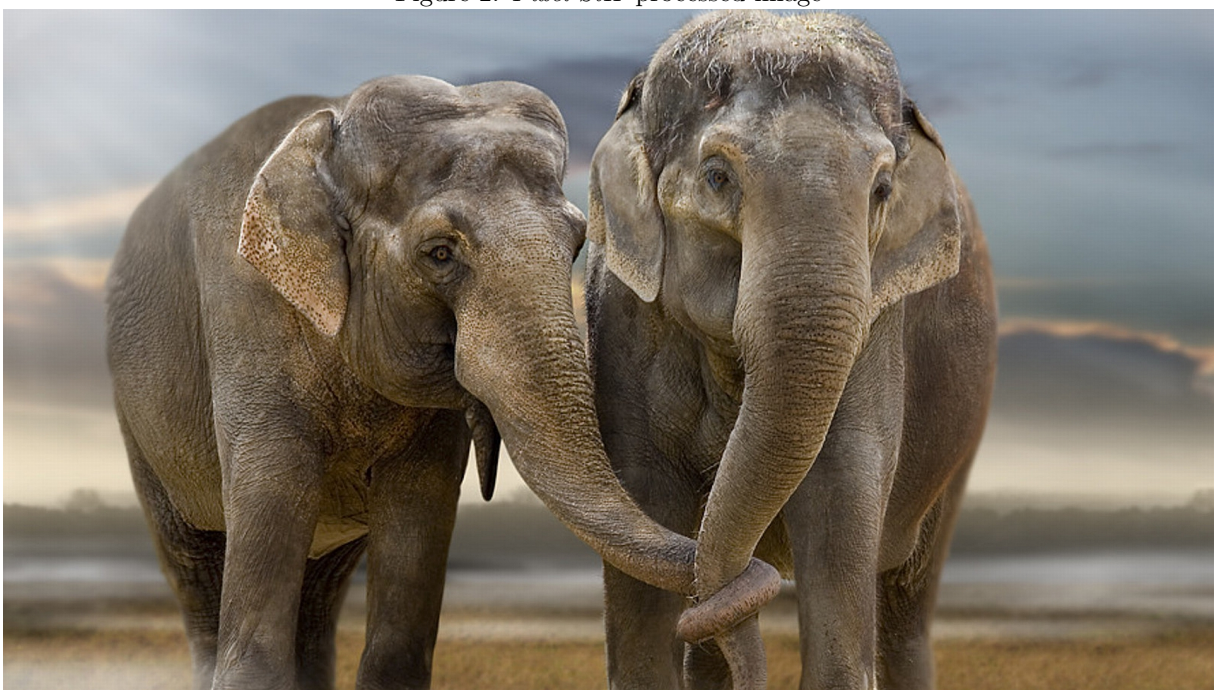
Compared to classic LSB, *Pixel StH* where the security reasons rely on two security layers; the first layer being represented by the suspicion and detection of steganographic algorithm's presence and the second by the correct deduction of the used channel, *Pixel StH* has 4 stronger security layers.

The first level of security is ensured, as in the case of LSB algorithms, by the lack of suspicion of steganographic algorithm being used. Besides bits differences that are present in the computer's processes and perceptible only by the computer, no visual differences are perceptible to the human eye. If data modification perceptible only by computer is observed by human, after raw data decomposing and processing, no confirmation of steganography is offered, since the modification could be the result of data compression.

Figure 1 and 2 represent a visual comparison between the original image, containing no *Pixel StH* implementation, and the same image after being processed by *Pixel StH* algorithm. As observed in the last picture, human eye can not interpret any difference at all, nor one that can lead to the forming of a suspicion of any steganographic tool usage.

Figure 1: Original Image

Figure 2: *Pixel StH* processed image



The suspicion of steganography is harder to be risen in the context of *Pixel StH* (due to lack of consecutiveness of the altered structural unites) than it is in the context of classic LSB algorithms where bits are consecutively modified and characters string can easily be observed and extracted.

Suppose, anyway, that Eve knows Alice and Bob use *Pixel StH* steganography in a certain picture and an identical copy of it has somehow ended up to Eve. Knowing for sure that the picture must conceal steganographic content, Eve has broken the first layer of *Pixel StH* algorithm. At this moment, Eve faces the second layer of the algorithm's security.

This second level of security is represented by the must of knowing the pattern used in data encoding. For the picture to be useful to Eve, it is a must for Eve to deduct the pattern. As mentioned above in the present paper, the pattern is given by a secure key obtained by Diffie-Hellman type of data manipulation of a seed provided by Alice. Starting from mathematical models described in [8, 6, 7], I have studied particular fields over non-supersingular elliptic curves suitable for being generators of seeds. The complexity of cracking a Diffie-Hellman type of system is described in [9, 11].

Suppose Eve has managed to compute the secure key used in the picture. Knowing that the picture contains *Pixel StH* algorithm, Eve faces now the third level of security of the algorithm.

This third level relies on the correct deduction of the channel used in encoding. For the case of a 4-tuple channel representation, Eve must apply a decoding algorithm for up to 4 times, in order to deduce the correct channel.

Suppose, however, that Eve has managed to break the third level of security, has found the correct channel. Now Eve is facing the fourth layer of security of the *Pixel StH* algorithm.

The fourth protection layer is granted by possibility of reconstruction of the extracted message. Suppose Eve has managed to reconstruct the message and now has a string of characters without any meaning. The second sub-layer of this security level Eve is now facing, relies on the validity and intelligibly of the once reconstructed message. There are two situations possible now: either Eve manages to obtain a clear text representation of the message, but cracking the *Pixel StH* algorithm has took Eve huge time resources that the information is no longer valid, nor valuable, either Eve has no possibility of decrypting the recomposed message. Either faced situation, *Pixel StH* has managed to take the protection of the encoded data to huge security levels.

For second, third and fourth layer to be broken, large amount of resources have to be allocated and

multiplied by three to manage to crack all the layers. Raw data manipulation, decoding, decrypting are huge consumers that add extra protection to the *Pixel StH* algorithm.

6 Algorithm Implementation

I have developed the mathematical model of this algorithm and later implemented the algorithm in the frame of Research Laboratory of Models of Nonlinear Analysis and Applications. I am a student member of this research laboratory since my first year of University. Members list can be found *here*.

All the hard work and commitment in developing the algorithm was held under the apprenticeship of Univ. Conf. dr. Nicolae Constantinescu to whom I owe everything I know so far and I will ever learn from now on cryptography topics.

The algorithm is currently implemented in a macOS steganographic application, named Proserpina.

Proserpina - The Godness of the Underworld represents a personal complete steganographic solution proposed for macOS users. Having as encoding engine the *Pixel StH* algorithm described in the present paper, Proserpina is the strongest and most trusted application available on macOS dedicated market.

The completeness of the solution is based on ensuring the user that all the actions necessary in a complete steganographic process (hiding, encoding, encryption, saving or sharing the resulting image) are handled in the most secure manner, suitable for each separate process. The following elemets compose the completeness of the application:

- Encoding security - *Pixel StH* algorithm described above in the present paper represents the steganographic engine.
- Encryption - Proserpina encrypts the message users give as input before encoding it
- Time efficiency - The implementation's of the *Pixel StH* algorithm execution time is the lowest among any other steganographic applications available for macOS platforms.
- Image security - The proposed solution allows users to take new, fresh images using camera devices, increasing the security of the application. New captured images are carefully and securely manipulated and are not retained in memory in their original state, providing any Eve no comparison material between a raw mask and a data holding one.
- On-line sharing - Not only that Proserpina allows on-line sharing within the app, but it also provides access to the sharing options that are the most secure for steganographic communications and manages the data manipulation in order to preserve binary information, without any data loss.

  The proposed solution oferres three possibilities of third party distribution, as presented below:

  - Messages - Proserpina prepares the image and securely handles it to Messages application, informing the users of the available at the end saving methods
  - eMail - Proserpina prepares the image and securely attaches it to Mail app, in order to obtain an untraceable transmission between the two apps and to preserve all the vital steganographic information
  - AirDrop - Proserpina prepares the image and securely handles it to AirDrop application
- Supported set of Characters - Proserpina allows users to write in any set of characters, emoji or special characters being also supported.
- Innovative, adaptive G.U.I. - The proposed solution is addressed, by its functionality and security to all types of users, whilst offering an interface available to the large public, that comes in four different interpretations.

7 Conclusions

The *Pixel StH* algorithm was proven higher above in the current paper to be worth of being trusted due to the security layers it brings in a new interpretation of the classic LSB algorithm. Also, the mathematical models and concepts that represent the theoretical base of the *Pixel StH* algorithm prove the applicability of the proposed implementation.

Pixel StH algorithm is encapsulated in and represents the steganographic engine of my personal developed complete macOS solution, Proserpina.

In the aiming of obtaining the most secure and trusted steganographic complete solution, Proserpina is currently finding itself in a testing session of cryptographic attack endurance, before it will be released on App Store. The security of Proserpina is heavily related to the *Pixel StH* algorithm complexity described in this paper.

After Proserpina will come out of security testing of cryptographic attack sessions, I intend to release it on App Store in order to obtain users' feedback and suggestions, that will be used for the development of future versions of Proserpina.

A Pro version of this application, with user settings for depth, channel, encryption algorithm etc. is desired to be developed, along with my dedication to learning cryptography and to research for new solutions in this domain.

Any future version of Proserpina will encapsulate either the *Pixel StH* described in this paper, either future implementations of this algorithm.

Acknowledgement

Proserpina was designed and developed (from the mathematical models, to features and design) by me, student of Faculty of Science, Dpt. of Informatics, University of Craiova, with commitment to cryptography and programming, in the heart of Research Laboratory of Models of Nonlinear Analysis and Applications and under the help, suggestions and teaching of Univ. Conf. dr. Nicolae Constantinescu.

Due to the development of the mathematical background that sustains the implementation of the *Pixel StH* algorithm, I am honored to be invited to present my concept and to accumulate fresh knowledge in mathematics and cryptography, to University of Thessaloniki and University of Paris-Orsay.

References

[1] Nicolae Constantinescu, Criptografie, *Ed. Academiei Romane, Bucharest* **2009**

[2] Frank Y. Shih, ıDigital Watermarking and Steganography—, CRC Press, **2008**

[3] Eric Cole, itHiding in Plain Sight: Steganography and the Art of Covert Communication, Wiley, **2003**

[4] Gregory Kipper, itInvestigator's Guide to Steganography, CRC Press, **2003**

[5] O.A. Ticleanu, N. Constantinescu, and D. Ebanca. *Intelligent data retrieval with hierarchically structured information*. In Intelligent Interactive Multimedia Systems and Services - Proceedings of the 6th International Conference on Intelligent Interactive Multimedia Systems and Services, IIMSS 2013, Sesimbra, Portugal, 26-28 June 2013, pp. 345–351, 2013.

[6] O.A. Ticleanu, *Differential operators over particular elliptic curves spaces with cryptographic applications*. E. Journal of Differential Equations, **2015**(303), pp. 1–9, 2015.

[7] O.A. Ticleanu, *Endomorphisms on elliptic curves for optimal subspaces and applications to differential equations and nonlinear cryptography*. Electronic Journal of Differential Equations, **2015**(214), pp. 1–9, 2015.

[8] R. Alsaedi, N. Constantinescu, V. Radulescu, *Nonlinearities in Elliptic Curve Authentication*, Entropy, Vol. **16**(9), pp. 5144–5158, 2014.

[9] N. Constantinescu, *Security system vulnerabilities*, Proceedings of the Romanian Academy Series A-Mathematics Physics Technical Sciences Information Science, Vol.13 (2), pp. 175–179, 2012

[10] N. Constantinescu, G. Stephanides, *The GN-authenticated key agreement*, Journal of Applied Mathematics and Computation, Elsevier, London, Vol. **170**, pp. 531–544, 2006.

[11] E. Simion, N. Constantinescu, *Complexity Computations in Code Cracking Problems*, Concurrent Engineering in Electronic Packaging, IEEE Communication, May 05-09, pp. 225–232, ISSE 2001.

[12] N. Constantinescu, G. Stephanides, M. Cosulschi, M. Gabroveanu, *RSA-Padding Signatures with Attack Studies*, International Conference on Web Information Systems and Technologies: Internet Technology/Web Interface and Applications, Portugal, ISBN 978-972-8865-46-7, pp. 97–100, 2006.

[13] I. Iancu and N. Constantinescu, *Intuitionistic fuzzy system for fingerprints authentication*, Applied Soft Computing, Vol. 13(4), pp. 2136–2142, 2013.

[14] A.I. Golumbeanu, *Application of differential cryptography to a GN authentication hierarchy scheme*, Electronic Journal of Differential Equations, **2017**(20), pp. 1–8, 2017.

Cristiana CONSTANTINESCU
University of Craiova
Dpt. of Informatics, Models of Nonlinear Analysis and Applications Research Laboratory (M.A.N.A)
A.I Cuza Street, No. 13, Postcode 200585, Room 221, Craiova city, Dolj county
Romania
E-mail:
constantinescu.cmaria@gmail.com

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

The Weather Stylist App

Gheorghe – Cătălin Crișan

Abstract

The article present a web application that aims to solve everyday problems related to clothing to be warn depending on what the weather is like outside. Thus through the application use, one can see the current temperature, and other information about humidity, speed, wind direction and weather forecast for the next three days specifying the minimum and maximum temperature. Another objective is to suggest clothing items considering the external factors of the time. Because today increased emphasis is laid on image, each person who wants to make a position in modern society must adopt appropriate fashion styles also by considering the weather conditions. To use the application at any time of the day and wherever you are, it is important to have a cross-platform app, so that the user can access it, for example: in the morning after he wakes up and checks his emails on the mobile or even on a desktop device. Moreover, to help our users, they can also subscribe by setting a time interval, daily or weekly, at which to receive notifications even if the user doesn't use the application. Thus, the user will be constantly updated with the latest news about the weather, but also in terms of clothing recommendations.

1 Introduction

Today, most people have access to internet and own a mobile device or desktop. The web application is designed for users who want to add style to their lives. The technology for web application uses a service worker - scripts that communicate with the browser application to facilitate the user's interaction.

The motivation for building this application is given by the desire to help people remain active in the modern world by suggesting clothing styles without wasting their daily time to find an outfit appropriate to weather conditions. Using recommendations offered by the application, the user saves not only time to be regularly invested in the latest trends in fashion, but also money, if we consider the idea of hiring a personal stylist to advise clothing. The application can be extended by increasing the range of clothing styles so that the user can choose from several styles the ones that fit him. In addition, it can try out new looks, if tired of the current style, and also make himself noteworthy among his friends. Why should you be constrained to find out news in fashion when you could get professional advice from someone who is aware of all this? Using the application, the user is offered a wide range of recommendations and streetwear styles that are not only casual and smart but bussiness-like so that he can be ready at any time of the day to visit the city or to go out for a business meeting.

Currently there are applications partially similar to a very low level of interaction with the user, and they offer only some basic information about how you should dress according to the weather conditions. They also contain repetitive images, without providing advice and recommendations of fashion style - this application is called *DailyDressMe*. Another alternative is the application *What to Wear* provided by *Runner's World* in which the user must manually complete weather conditions after receiving a recommendation plain text. This application also

uses a low number of recommendations - one for each garment (e.g. winter cap, running shoes, long-sleeve shirt).

For the practical purpose of the work, the application uses the server for user interaction through an extension called *Web Server for Chrome*. With it you can easily configure a server to communicate with the user's browser to send notification application. In addition, to make the application responsive, so that the user may access app on mobile and desktop, the Bootstrap open-source framework is used for front-end.

The article is organized as follows: in the second section it presents the application and related information about the user interface and application compatibility; the third section contains information about configuring the application to communicate with the user and information about sources used on the project; the last section contains the conclusions.

2 Presentation of the application

2.1 User friendly concept

Application design is created based on the concept of user friendly application. Font color range used is minimal as black and white are the main colors used. For the background darker modern colors are used to serve as the anchor for the user's eye. Delineating sections are also very intuitive for the user as on first contact with the application can distinguish between the app sections, like forecast section from styling section. Without too many colors on which to focus, the users can easily navigate. Making reference to the weather part of the application, it uses a series of icons based on familiar visual language of the user, so that it can determine weather from a simple and fast glance.

2.2 Weather effects

Application interface wants to be closer to the user, and this is done with the help of features like the changing of the background of the time of the day based on the time in the location where the user has been for a more real approach. It also implies the existence of special effects depending on current weather and rain (next version will contain more complex effects such as falling snow, sunshine glow effect and lightening effect).

Fig. 1:*iPhone 6 Plus UserInterface*

Fig. 2:*Desktop User Interface*

2.3 Device compatibility

Using a new technology for Web applications, based on service workers, at the moment the application is compatible with most of the existing browsers. The most popular browsers like Google Chrome, Mozilla Firefox, Opera and Internet Samsung (for mobile devices with Android) have already implemented this feature. However, service workers are fully in progress and browsers like Edge and Safari work on it (Safari has considered that - there is the possibility of implementing it in the coming years). Given these facts, we can say that the application is cross-platform and can be used on both mobile on Android and on iOS and desktop operating system Windows, Linux and Mac [2].

Chrome: Shipped.

Firefox: Shipped.

Samsung Internet: Shipped. Based on Chromium 44.2403 with some additions and changes. (See "Service Worker" section.)

Safari: Under consideration, Brief positive signals in five year plan.

Edge: In development.

Support does not include iOS versions of third-party browsers on that platform (see Safari support).

Fig. 3:*Service worker-browser compatibility[2]*

3 Service Worker at work

For the implementation there will be presented notions related to service worker, how it works, how it can notify the user using a specified time and how it can permanently link you to the server with the user's browser for he to receive the latest news.

The major benefit that this application brings can be compared to other partially similar applications and it consists in opportunity to notify the user of the desired information from time to time. Thus, it requires sending notification to the user even if he is not active to see the notification at that moment. The only thing to be done is the user's registration to receive notifications. This will generate the sending of a user ID registration to the server so that the server can then send to the user the information based on this IDs.

A service worker is a script that runs in the background of the user's browser separated from the web application. The operating mechanism is based on the following events: checking the server at the time when a certain condition is fulfilled, forwarding the request on a service such as Google Cloud Messaging notification instructions or Mozilla Push Service. The necessary information from the server and browser ID used by the client is successfully redirected by the respective service worker.

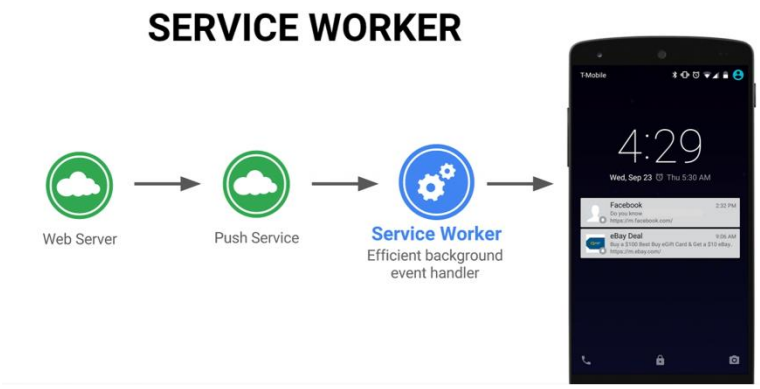


Fig. 4: *Service worker-push notification flow[3]*

The first step is registering a service worker script and checking the possibility to send notification to user [1]:

```
if ('serviceWorker' in navigator && 'PushManager' in window) {
        console.log('Service Worker and Push is supported');

        navigator.serviceWorker.register('sw.js')
        .then(function(swReg) {
        console.log('Service Worker is registered', swReg);
        swRegistration = swReg;
        initialiseUI();
})
.catch(function(error) {
        console.error('Service Worker Error', error);
 });
} else {
        console.warn('Push messaging is not supported');
        pushButton.textContent = 'Push Not Supported';
}}
```

So, after being successfully registered by the service worker, we can provide the user the ability to subscribe to receiving notifications from the application. This is done using a form that sends to the server information that you want to receive notification. Along with information about the range of notification is sent also a unique ID by which the push notification service can redirect notification to the user.

```
Function subscribeUser() {
        constapplicationServerKey = urlB64ToUint8Array(applicationServerPublicKey);

        swRegistration.pushManager.subscribe({
```

```
                    userVisibleOnly: true,

                    applicationServerKey: applicationServerKey
 })
.then(function(subscription) {
        console.log('User is subscribed.');

        //Subscription send to server with unique ID
        console.log(JSON.stringify(subscription));

        updateSubscriptionOnServer(subscription);

        isSubscribed = true;

        updateBtn();
})
 .catch(function(err) {

console.log('Failed to subscribe the user: ', err);

updateBtn();
 });}
```

Considering that the user wants to receive notifications from applications, we must configure the service worker so that it knows what to display when the server receives an event such as receiving a notification.

```
self.addEventListener('push', function(event) {
        console.log('[Service Worker] Push Received.');
        console.log(`[Service Worker] Push had this data: "${event.data.text()}"`);

        const title = 'Check this before you go out !';
        const options = {
                body: 'Ready for today style advice?',
                icon: 'images/logo.png',
                badge: 'images/badge.png'
        };

        event.waitUntil(self.registration.showNotification(title, options));
});
```

At the opening of the notification the user must be forwarded to the web application to check the latest updates, and service worker must have the capability to manage such an event as can be seen further down:

```
self.addEventListener('notificationclick', function(event) {
        console.log('[Service Worker] Notification click Received.');

        event.notification.close();

        event.waitUntil(
                clients.openWindow('https://www.theweatherstylist.ro')
        );
});
```

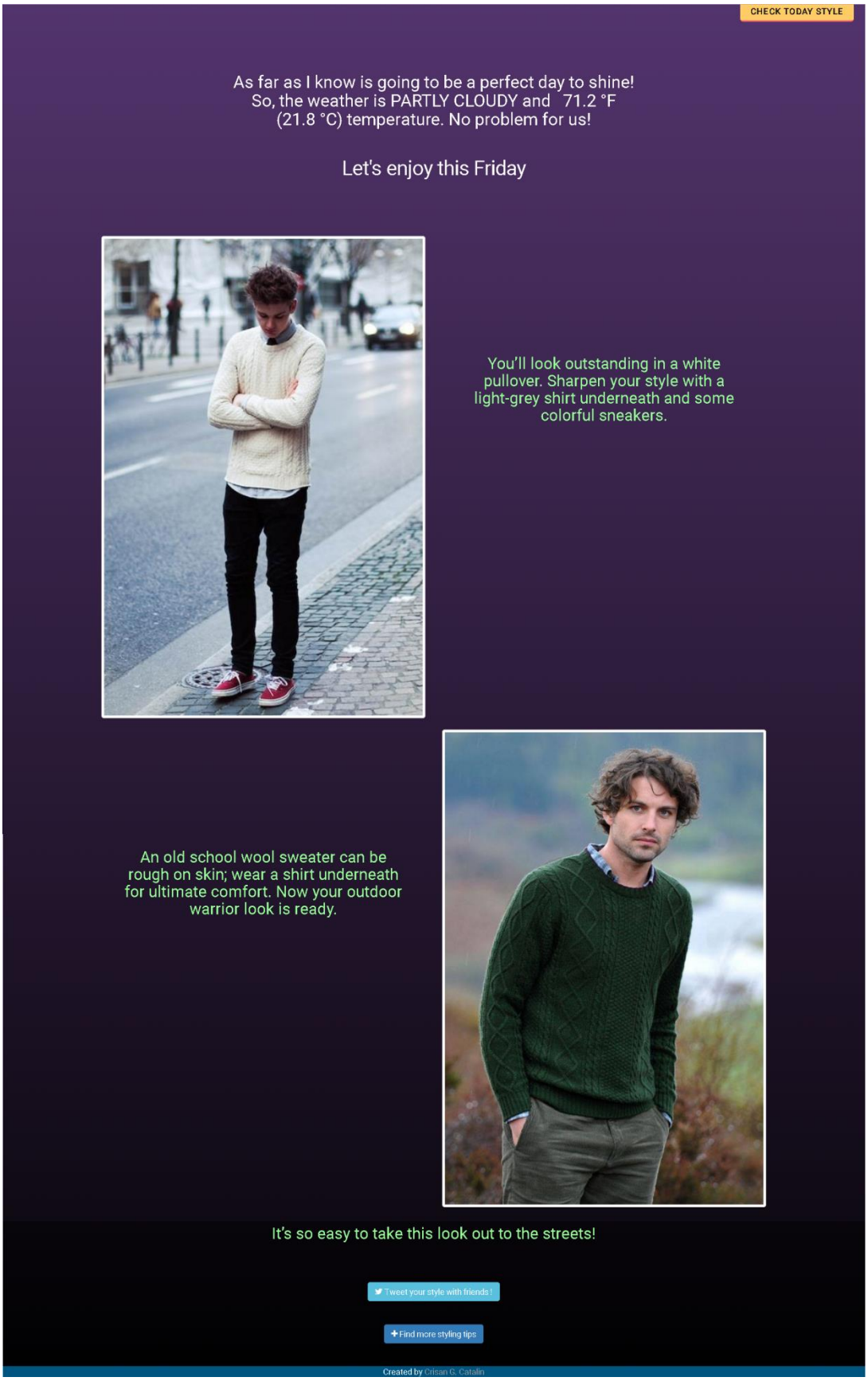


Fig. 5: *Style recommendation section based on local weather*

4 Conclusions

Compared to other similar existing applications, the one presented in this article brings some additional benefits. At present there are many applications - mostly mobile - that suggest the fact that if it is hot you should wear a shirt and trousers without further details. However, for people who need to be seen in the communicative relational environment, this application helps the user to move to a higher level.

Due to the limitations provided by mobile applications, a web application is the best solution today. Web applications take a considerable advance getting closer to replacing the native mobile application. Given the ease with which web traffic and web applications can be found, such an application that does not require installation, offering to the users a greater comfort in use. In addition to other applications, this app uses a new type of script in modern browsers designed to facilitate communication with the web application through periodic synchronizations running in the browser's background. So, the browser will be updated with the latest news provided by the responsive application, simulating real experience. All this turns the application into a web progressive application - reliable, fast and exciting. So, it manages to establish a permanent contact with the user's application on both desktop and mobile.

For developing the next version of the application I want to increase the complexity of its recommendations of plain text automatically generated and also to configure a web crawler in order to give the user a large number and diverse suggestions; moreover, to add an option to choose which sites the user should trust to receive the preferred clothing recommendations.

In addition, I wish to add the possibility that the user can select clothing styles so that he can receive useful recommendations to be used by the user.

Currently the application is configured to give styling tips only for men because of the high number of existing styles and clothing items. Certainly the next version will be accessible for women planning to have a larger variety.

Acknowledgement: This work was supervised by Lecturer PhD. *Laura Stoica*, from Faculty of Science, *"Lucian Blaga" University of Sibiu*.

5 References

[1] Matt Gaunt, *Adding Push Notification to a Web App*, https://developers.google.com/web/fundamentals/getting-started/codelabs/push-notifications/
[2] Jake Archibald. *Is service worker ready?*, https://jakearchibald.github.io/isserviceworkerready/
[3] Ido Green, *Chrome 48 Updates And News*, https://greenido.wordpress.com/2015/11/27/chrome-48-updates-and-news/

GHEORGHE – CĂTĂLIN CRIȘAN
Lucian Blaga University of Sibiu
Faculty of Science - Informatics
Str. Dr. I. Ratiu, No.5-7, Sibiu, 550012
ROMANIA
E-mail: crisan.gheorghecatalin@gmail.com

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

Devising an improved University Management System

Elian Doran, Alex Negru

Abstract

This paper proposes creating a University Management System (UMS) aimed at improving the life of both the university staff and students. A different approach towards unifying actual concepts of UMS with student-oriented software is the core philosophy of the project. The platform being created aims at implementing more unique features when compared to the existing market like Android & iOS apps, cloud integration and a campus navigation system. The final goal is devising an intuitive and elegant solution.

1 Introduction

The life of a student is riddled with information, but more than often there is also the need of managing daily academic chores: keeping track of the timetable (odd and even weeks, laboratories, seminaries, etc.), noting assignments and most importantly, checking back on exam grades.

In order to facilitate students, applications have been developed for tending to some of these activities. Overlooking the application market for such types of software, two distinct categories can be conceived:

1. **University-oriented (centralized) applications** – usually university-appointed secretaries manage the student database, which includes authentication data, academic year structure, timetable, exam grades and the financial data of students. The system can usually be accessed by means of a web application.

2. **Student-oriented (decentralized) applications** – these are administered per each student, which includes manually entering the subjects, timetables and other information. One of the core advantages is the ability of receiving notifications of today's schedule and impeding exams. These can be accessed in a variety of mediums: web-based applications, mobile applications or a combination of both.

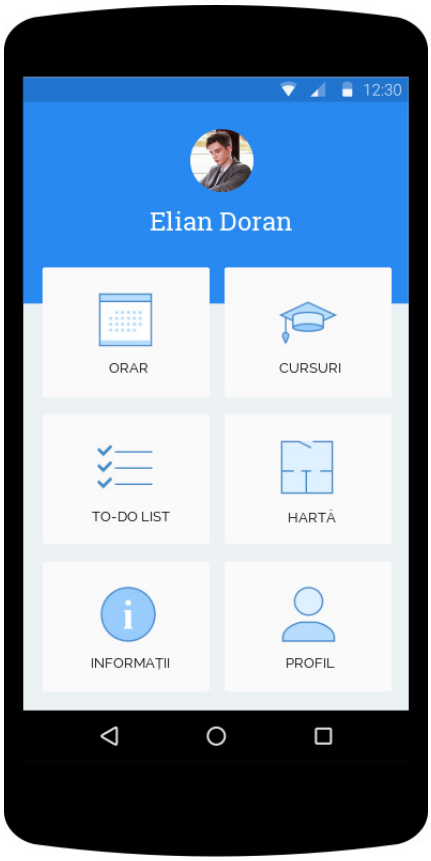


Figure 1: Home screen concept of our mobile application

Centralized applications are mandatory for defining common information like the academic year structure, time table and exam grades, whereas student applications add the ability of adding individual information like keeping track of tasks, notes. One of the

downsides of student-oriented applications is the fact that every
student must personally define common information that should probably have been handled by a centralized application. Combining both of these concepts results in a system that provides benefits of both worlds, while reducing downsides like data duplication.

This paper proposes creating a unified software platform which acts as a centralized University Management System whilst adding functionality from student-oriented applications. The application being created is designed to run directly from the web, as a web application, with correspondent mobile applications for Android and iOS.

Apart from the different approaches, these types of software are usually the victim of not tending the entire needs of a student, thus the user resorting to use multiple applications. Using existing features from different applications as a baseline in the development of a platform would aid both the student and the university in making the entire process more efficient.

1.1 Structure of the article

1. Introduction	Comparison of existing software
2. Core Features	A detailed overview of some of the defining characteristics (timetable, grades, task list and maps) we intend on implementing.
3. Implementation Details	A description of the processes we undertook in turning our concept into reality (client-server architecture, roles, the database and its components).
4. Conclusions	Advantages/disadvantages analysis of our end product, planned features and summarization.

1.2 Existing software

1.2.1 Student-oriented applications

As mentioned earlier, there are already multiple software platforms that apply the concept of digital University Management Systems.

Regarding student-oriented applications, three different applications have been studied for the purpose of competition analysis:

Name Author	**School Assistant** Gil Castro	**Pocket Schedule Free** Appxy	**My Study Life** My Study Life, Ltd.
Android Support	✓		✓
iOS Support		✓	✓
PC/desktop Support			✓
Has overview screen	✓		✓
Personal to-do list		✓	
Assignments per subject	✓	✓	✓
Calendar view	✓	✓	✓
Agenda view	✓	✓	✓
Keeping track of exams	✓		✓
Password protection	✓(pro)	✓	
Cloud-based			✓

Most of the features compared in the table above will be detailed further on.

1.2.2 University-oriented applications

Due to the fact that university-wide applications are harder to obtain access to, only a single platform has been thoroughly analyzed for the purpose of this paper.

The platform in question is titled *UMS*, standing for this specific type of applications (University Management System). It is developed by the Romanian software company Red Point Software Solutions. According to self-reported statistics, as of May 2017, it is being used by 24 Romanian universities [1].

The software features multiple facilities for university administration, as well as most features of a university-oriented application. A comparison of Redpoint UMS and the software proposed by this paper will be further discussed.

2 Core Features

2.1 Timetable

Figure 2: Home screen of the mobile application

Timetables represent one of the most popular features implemented by student-oriented university software, fact which is reflected by analyzing the market for such applications.

The user must define the following information:

1. The *academic year structure*, mostly the division in terms (i.e. semesters, trimesters, etc.) and their corresponding start and end dates.
2. The *list of subjects* (e.g. Mathematics, Computer Science, etc.) which are part of the schedule.
3. The *list of locations* where courses are held and the list of teachers, these can be manually defined or automatically inferred from what the user has previously entered.
4. Manually creating each distinct *time entry* (e.g. Math class each Monday at 08:00 am). These are characterized by:
   (a) The *subject*, taken from the list defined above.
   (b) The *date of the first instance* of this entry, usually placed in the first week at the beginning of a semester/module (e.g. first Monday since the start of the first semester).
   (c) The *date of the last instance* of this entry, it can span until the end of a semester/module or at the end of the entire year (e.g. last Monday of the end of the second semester).
   (d) The *start time* (e.g. 08:00) and *duration* (e.g. 2 hours) of the entry.
   (e) The *repetition mode* – how often the entry repeats itself: daily, weekly, bi-weekly (e.g. only on odd or even weeks, or in this case, weekly – every Monday).
   (f) The *location* where it is held and the *teacher* holding it – selecting from a list or defining it through user input.

This is a time-consuming activity, which in the case of student-oriented software, every student must personally enter dozens of entries for each semester with care. Most universities have already gone through the process of creating a digital timetable (usually in the form of a web page or PDF document), but the ability to transfer the timetable into student-oriented applications is usually lacking.

[1] The statistics can be found at: `http://rpss.ro/ro_RO/products/university-management-system/`.

2.2 Grades

Whereas the timetable is one of the main features of student-oriented software, grading information is more common in university-oriented software. Universities sometimes use their on-line platform to list exam grades, to which the student has to constantly check for updates.

Grades can have the following structure:

1. The *subject* being graded.
2. The *date* the grade was given.
3. A *title* and/or *description* of the grade, indicating the purpose (e.g. *final exam*).
4. The *grade* itself, in a globally-specified grading system/format (e.g. from 1 to 10, from A to F, etc.).

Student-oriented software like *School Assistant* do allow keeping track of different grades per subject. But the downside to this is the fact that the user has to manually enter each of their grade which is tedious and error-prone. Final grade calculation is also possible, although this does require entering all the details about the grading system and the percentage/weight of the specific grade.

University management systems usually have a dedicated section for listing exam grades. When examining Redpoint UMS, it presented no notification method of new exam grades (no push notifications or e-mail notification support), which represents a downside. Furthermore, the fact that Redpoint UMS occasionally resets the user's authentication session, refreshing in longer periods of time results in the user having to authenticate over and over.

2.3 Task List

Keeping track of daily school assignments is a feature unique to student-based implementations. Implementing this ability into our University Management System would theoretically be unique.

There are two distinct types of task lists:

1. **Assignments List** – items have deadlines which are assigned to *timetable entries*, these are usually used to represent homeworks or projects (e.g. *Computer Science presentation on June 05th.*).
2. **To-Do List** – unlike, *Assignments Lists*, these have a variable or no deadline and have no attachment to the timetable, can be used to represent other task a student might need to be reminded of (e.g. *Get certificate of student status from secretary.*).

Both of these types of task lists have the following characteristics:

1. A *description* of the task at hand.
2. Whether the task was *completed* or not.
3. The *deadline* date, if any.

2.4 Maps

University campuses can be of significant size, and for students, navigating through might turn out to be hard at first. This is especially true when a student has to present themselves to a location for the first time, with the pressure of not arriving late.

The platform seeks to solve this issue by allowing universities to blend their campus and building maps into the platform. When faced with a new location, the user can simply tap an option to show the map indicating it.

This feature is intended to be the 'ice breaker' feature of the platform. This is due to the fact that it doesn't seem to be implemented, at least not on a grand scale, in other types of University Management Systems.

Most modern smartphones and some tablets have geolocation features built-in. These devices could be used by the student, with the aid of the University Management System, to easily navigate to the desired location.

There are two different types of maps:

1. **Campus maps** – these are open maps, spanning over larger areas, can easily be tracked via GPS and the maps are very easy to obtain via Google Maps or other similar services.
2. **Building maps** – some universities offer maps of buildings, indicating the name of rooms. These are harder to implement due to the need of creating custom map software.

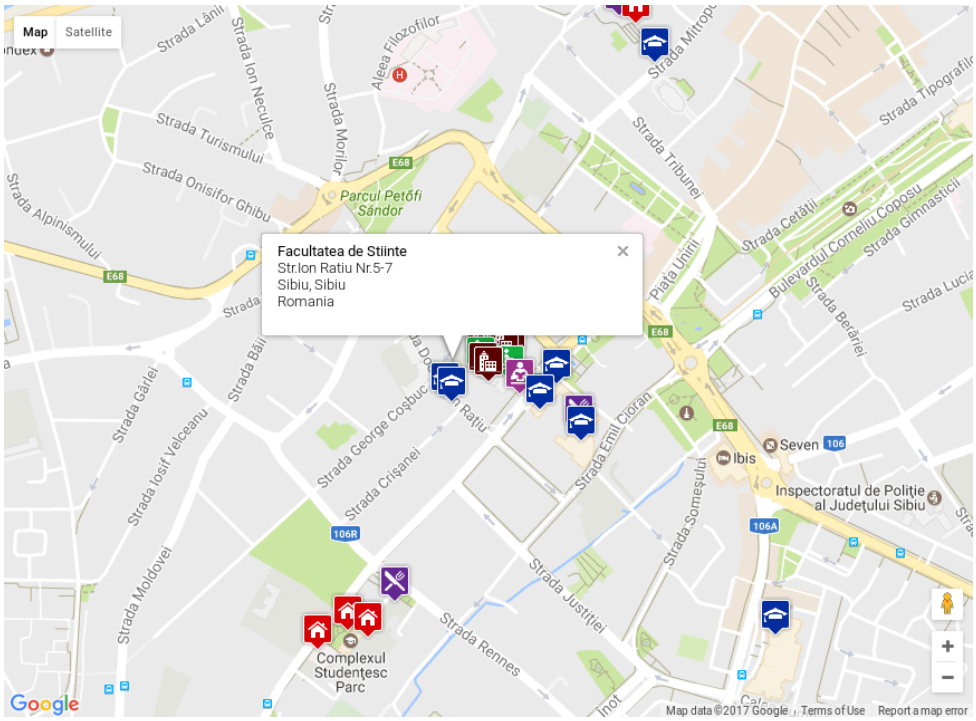


Figure 3: Lucian Blaga University of Sibiu's campus map[10]

3 Implementation Details

A University Management System requires intricate systems design to create a modular architecture capable of handling the features mentioned in this paper, while also allowing room for improvement and new features.

This section is intended to present a fraction of the development process while gaining general insight of the mechanism required for operating this kind of software.

3.1 Client-server architecture

In the introduction section, two distinct types of applications were identified: centralized (university-oriented applications) and decentralized (student-oriented applications). The goal of this paper is to describe a hybrid of the two concepts: creating a software platform containing both a centralized and decentralized part.

The implementation consists of a server platform which undertakes the tasks of a university-oriented application whereas the client communicates with the server, while also providing individual features like task lists and customization.

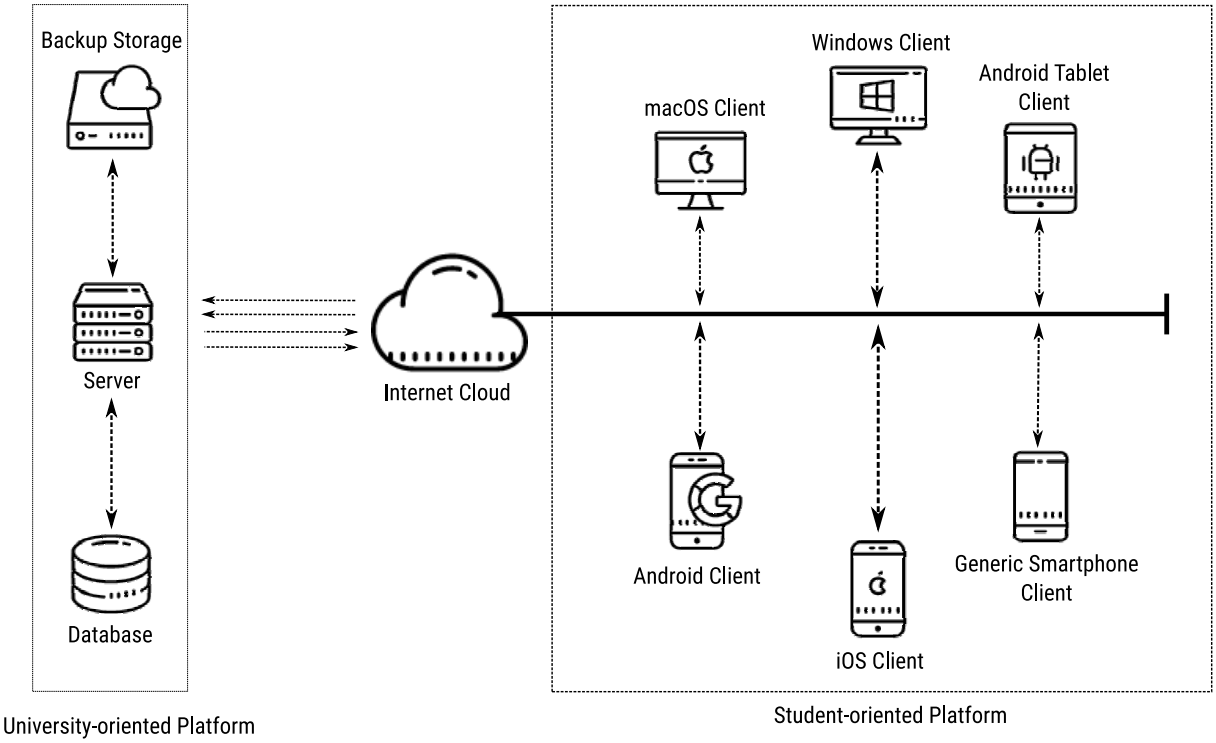


Figure 4: An illustration of the bilateral communication between the server and the client.

Apart from avoiding data duplication, the advantage of using a cloud-based application instead of manually defined by students, there is also the possibility of a single user using multiple devices to access their data.

3.2 The roles between client and server

The **server** must handle following tasks:

- Store important information such as the academic year structure, timetable, exam grades, etc.
- Provide an Application Program Interface (API) to be able to manage the said information.
- Provide authentication support, making distinction between administrators, teachers, students and guests.
- Maintain multiple connections between the server and the client, while allowing for a large number of clients with minimal lag.
- Regularly do tasks for optimizing the database and removing redundant data.
- Send out notifications when needed.
- Provide a Web interface for administering the data.

Whereas the **client** must:

- Have a friendly user interface for displaying and manipulating data.
- Be capable of receiving and dealing with notifications sent by the server.
- Store a local cache of user input until an Internet connection can be established.
- Synchronize remote data with local data whenever possible, with least power consumption and CPU usage.
- Be customizable by the end user.

3.3 Server-side technology

A wide variety of technologies can be used to create the server-side of the University Management System. When devising the development plan of our prototype, it has been taken into consideration the fact that the server would have a Web interface that is going to be interpret data sent out by the server. This translates into the need of using the three de-facto technologies of the Web (HTML, CSS and JavaScript).

Thus, a programming language well integrated with these three technologies would be best suited. Considering that and the fact we had prior experience with the said programming language, it was decided to use PHP. Some of the alternatives considered were Python and ASP.NET (via C#).

PHP is one of the core programming languages of the Web, being used by 244 million sites by January 2013.[6] One of the downsides for not using PHP as our server programming language was its workflow: it runs as a script, rather than a service. The code is being invoked only when it is requested via a URL and data can be difficult to persist among sessions.

Even though the API was foreseen to be vast in order to accommodate all the features of a University Management System, the commands themselves were going to be relatively simple: *fetch today's schedule*, *add a new user*, etc. Thus, for the sake of simplicity, we stood by PHP's side.

One of the problems with a PHP implementation mentioned earlier was the difficulty in data persistence, but taking into consideration the possibly large amount of data that needs to be stored for a University Management System, a DBMS (DataBase Management System) would be more than suitable.

In order to communicate with the client, an approach had to be selected – there is quite a significant amount of different technologies and patterns that can be used. The main contenders were using SOAP[11]

communication, or a RESTful Web Service Client[9]. We chose the former as it seemed to be the simpler approach.

3.4 Client technology

As mentioned earlier, the platform is aimed at both desktop and mobile devices. There are two approaches to achieving this: develop native solutions to the same problem to be compatible with only one platform or find a common denominator.

We chose the latter, and that is creating a *hybrid web application*[2] – a native application that uses an integrated web browser to display a web application. This simplifies the process of development on multiple platforms because standard web development techniques can be applied in order to create a single solution for both mobile, tablet and desktop environments. The source code of the web application consists of HTML, CSS and JavaScript. Once the decision has been made, a technology that applies this concept must be sought for – in our case, Cordova was chosen.

In order to improve user experience, web applications are in fact Single Page Applications (SPAs). The concept proposes using a single HTML page loaded with JavaScript that further loads needed data asynchronously[4]. These can be implemented from scratch or an existing framework can be used to improve the quality of the end product and speed up the development. A personal favorite of ours is Framework7, apart from being a SPA framework, it features the ability of designing native-looking applications that mimic the user interface of both Android and iOS.

The abstract of this paper mentioned the fact that creating an intuitive design is a priority, thus the user interface must be carefully planned. Presently, there are a number of design philosophies that are commonly adopted – Google, Microsoft and Apple have standardized design principles that are widely recognized. Because Framework7 that we intended to use already has built-in support for Google's Material Design[5] and Apple's flat design, we had to choose between the two of them. Our decision was unanimously in favor of Google's Material Design due to their detailed specifications when compared to Apple's design.

3.5 Database structure

The database will store all information pertaining to the University Management System, from users to schedules. As for the technology involved, we decided to use a SQL-compatible DBMS that follows the relational model.

The relational model[3] is a way of storing data into tables where each table presents a different set of information, with columns representing fields and rows representing records. Records are usually identified a key column. Tables can be linked with each other via keys to add relationship information between tables and their fields.

Once we decided to use the relational model and an SQL DBMS, the next step was choosing an implementation. We went with the second most popular database engine on the market[7], MySQL which is an open-source project owned by *Oracle Corporation*.

There are many alternatives to both relational DBMS (Oracle, Microsoft SQL Server, PostgreSQL, DB2), but also to other approaches towards storing data (Document store – Mongo DB, Wide column store – Cassandra, Key-value store – Redis), but the relational model was deemed suitable for our purposes and MySQL offers all the needed features.

Due to the complexity of the project, this section only defines and attempts at explaining a fraction of the real database, but it should help create a general overview of the underlying database structure.

3.5.1 User Management

Due to it being a cloud-based application, a good authentication system is a must. The system must be able to provide an authentication API by means of an user name and password. A common alternative to simple user-password authentication is two-factor authentication[1], which requires an additional method of confirming one's identity – usually through another password generated by a disconnected token. This has not been taken into consideration for early implementation due to the amount of work and the lack of need for such security measures (at the moment).

Every user, that is every teacher, student or administrator, is stored in the `Users` table. The only information stored so far are the user's name and its password. For security reasons[8], passwords are being hashed with a SHA-256 algorithm in order to prevent password leakage in case of a database intrusion.

User information, currently comprising of 31 columns of information, is stored separately because of a design choice to keep general information out of the main table as to not clutter the structure and create unnecessary complexity. Between `Users` and `UserInformation` is a one-on-one relationship, meaning that each distinct user can only have zero or one corresponding user information entries.

`Students` and `Teachers` tables store information pertaining to `Users` that belong to this kind of category. This is a forward-looking move as to allow custom information added to either these categories, for example the faculty assigned to the user.

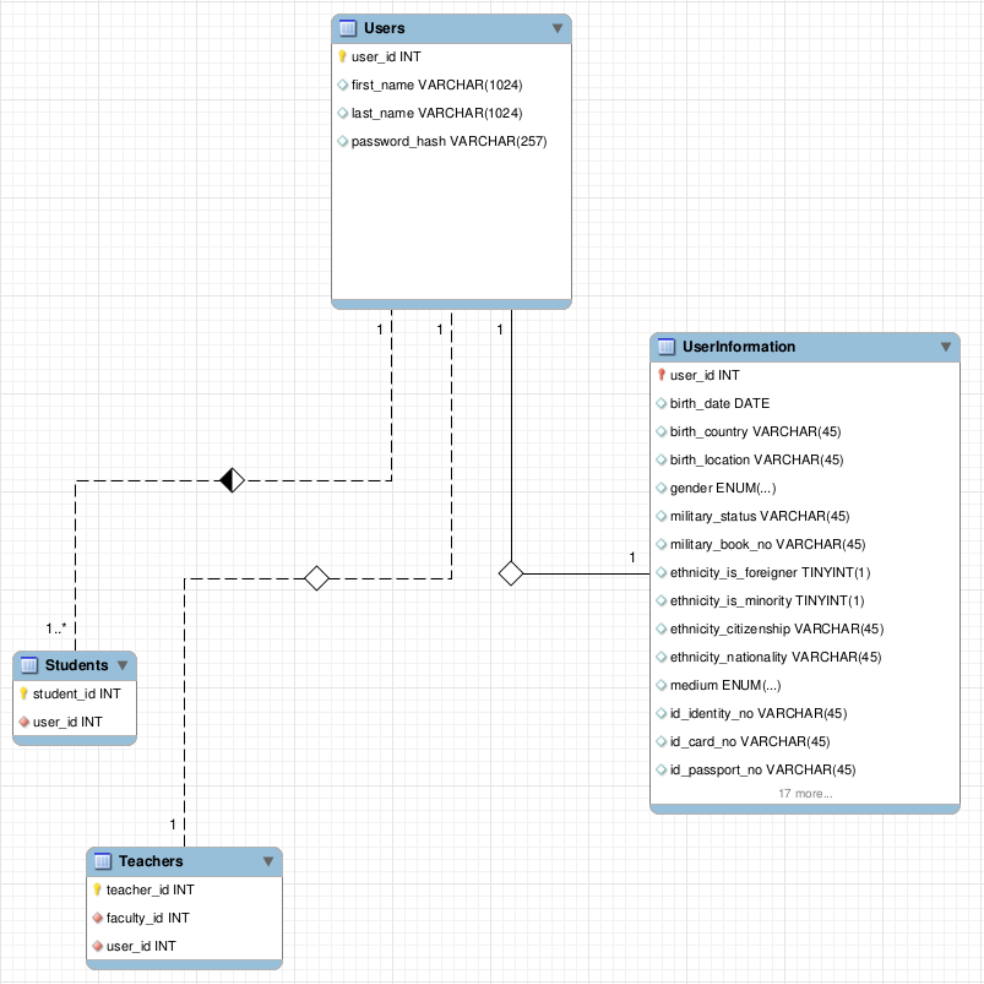


Figure 5: The EER database model for user-related information.

3.5.2 Academic year structure

The entirety of data handled by a University Management System, like the timetable and grading information, revolves around the concept of academic year structure.

The application has to be aware of the academic structure in order to correctly present and categorize information.

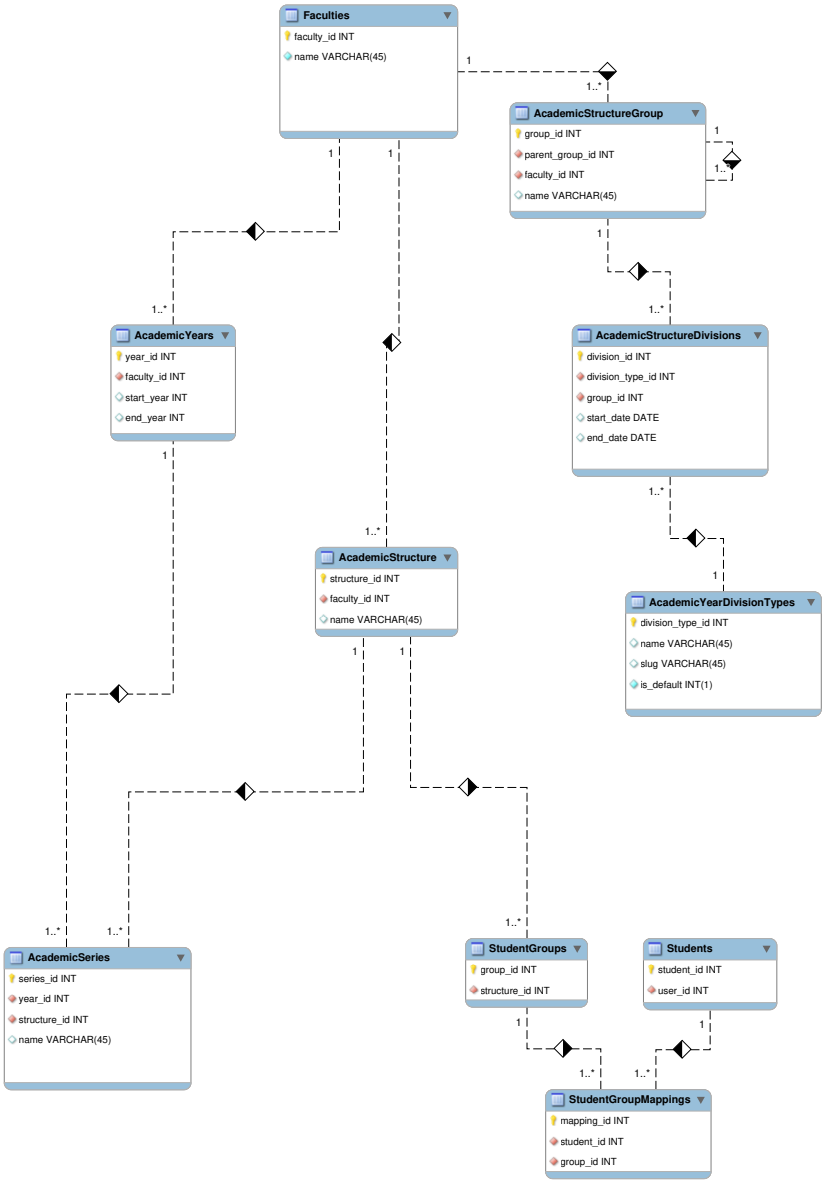


Figure 6: The EER database model for academic year structure information.

The academic year structure, for practical reasons, is divided into multiple time periods with different activities. From now on, these periods will be referred to as *academic year structure divisions*.

The year structure can differ between a university's faculties, thus this information must be separately stored. In order to create that distinction, the `Faculties` table was created which is used to specify which faculty the academic structure belongs to. The `faculty_id` field of subsequent tables indicate which faculty it applies to.

In an effort to standardize the representation of the year structure, the following taxonomy has been conceived:

1. **Structure groups** (stored in `AcademicStructureGroup`) are the largest components of the year structure. They can be:
   (a) *Semesters* – the year is usually divided into two semesters, with different subjects and different final exams. Defined as a `AcademicStructureGroup` having the `parent_group_id` field NULL.
   (b) *Subdivisions of semesters* – some faculties further divide a semester into 'modules' (or similar), which similar to semesters, feature their own subjects and final exams. Defined as an `AcademicStructureGroup` with the `parent_group_id` set to the `group_id` of its parent *semester*.
   (c) *Stand-alone periods* – periods of time alloted to practice, recuperation or re-examination that usually aren't attributed to any semester or subdivision of it. Database-wise, these behave exactly like *semesters*.
2. **Structure divisions** (`AcademicStructureDivisions`) are further divisions of a *structure group* (`AcademicStructureGroup`). These are characterized by the *start date*, *end date* of the period and a *type* describing it. The type of division can vary wildly depending on faculties, which means having them stored in the table `AcademicYearDivisionTypes` for extensibility. Common divisions include:
   (a) Didactic activities;
   (b) Final exams;
   (c) Holidays.

Each `Faculties` record can have multiple attached `AcademicYears`, which contains distinct academic years like 2016-2017, 2017-2018. In turn, `AcademicYears` can have multiple `AcademicSeries` (which is defined by name and year). That was created in order to accomodate the fact that there are multiple categories of students in a year: first year undergraduate, senior year, etc.

The information contained within (1) and (2) is part of the parent table `AcademicStructure`, which contains the entire structure for an academic year. To one or more `AcademicSeries`, a single `AcademicStructure` can be assigned. This allows for adding a single structure for multiple series, in order to avoid data duplication.

3.5.3 Timetable database storage

Once the academic year structure has been defined, the next step would be defining the timetable database table structure. Just as with the previous database schema, its complexity is rather high, in order to make the administration part and the API easier.

The `TimeTable` table contains all the time entries. An entry has the following table relationships:

1. The `AcademicStructure` it belongs to.
2. The `Subjects` record it belongs to (e.g. Mathematics, Computer Science), cannot be `NULL`.
3. The `Locations` record it is hold at, can be `NULL`.
4. The `Teachers` record which holds the class, can be `NULL`.
5. The `CourseTypes` record which indicates its type.

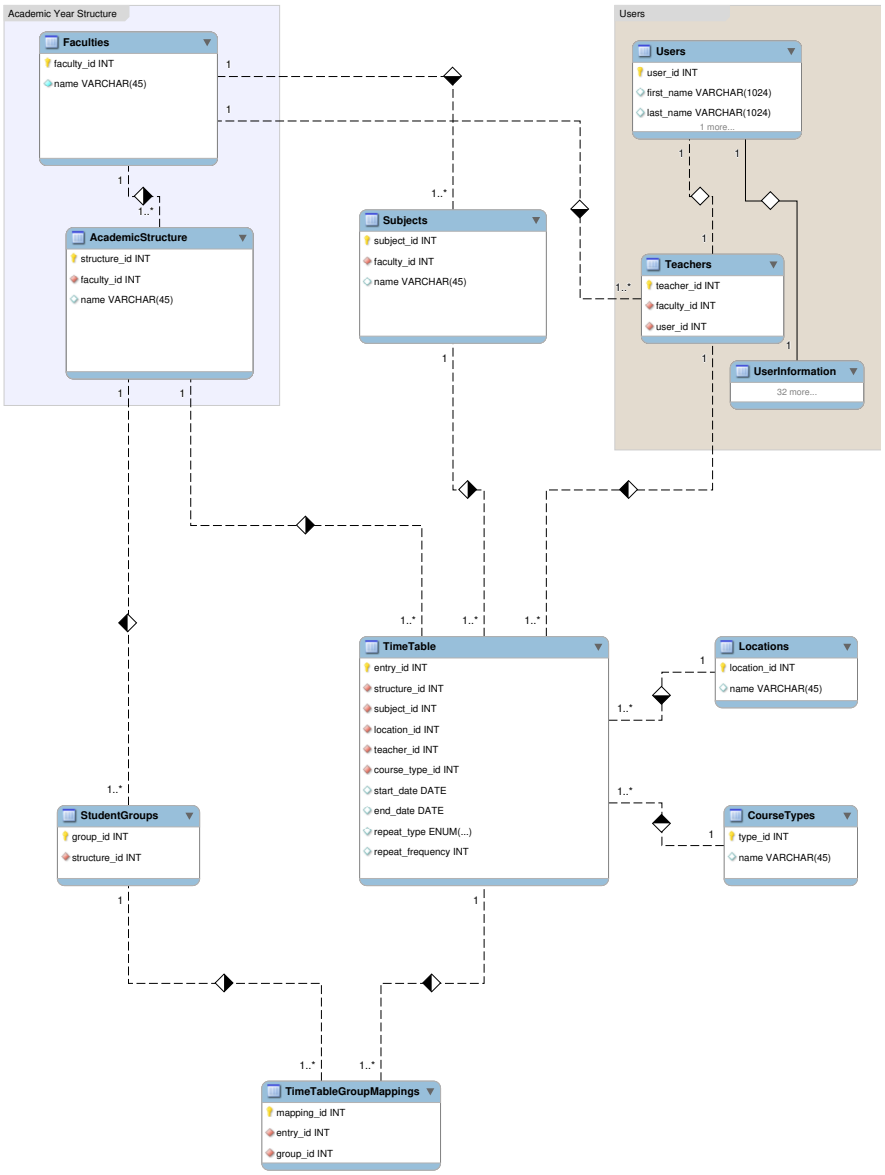


Figure 7: The EER database model for timetable information.

The topic of the timetable feature was already discussed in the *Core Features* section of this article.

An `AcademicSeries` can be further divided into what was named `StudentGroups` because sometimes group of students belonging to the same year (i.e. undergraduate) have different time entries for seminaries and laboratories. The `TimeTableGroupMappings` is what allows grouping multiple `TimeTable` records to multiple `StudentGroups`.

4 Conclusions

4.1 Existing market comparison

The purpose of this platform is to create a software that is better than what the existing market provides.

Advantages

This is a non-exhaustive list of features that uplifts our project:

- ✓ Combining the concept of university-oriented software with the features of student-oriented software.
- ✓ Avoiding duplication of data, requiring minimal intervention for the student.
- ✓ Using cloud technology to synchronize the user's data to all of their devices.
- ✓ The creation of native-looking mobile applications for Google Android and Apple iOS.
- ✓ Merging useful features found throughout different applications into a singular one.

Disadvantages

These are some possible disadvantages regarding our platform:

- ✗ Because of limited human and financial resources, development of all the proposed features is going to take a long while.
- ✗ The existing competition already has a loyal user base, making it more difficult to convince people and universities to transition.
- ✗ Due to the software being a hybrid web application, not native, performance might be suboptimal on low-end devices.

4.2 Future directions

The main goal after preliminary development of the University Management System would be to apply it in real life situations. This translates into having convinced a university to adapt a UMS system should they not already have one in place, or convince to switch from a pre-existing UMS to ours.

Planned features

- ♦ Implement notification system.
- ♦ Display the year structure for students.
- ♦ Improve the map service to allow for in-building maps.
- ♦ Add a target-able news service for academic notifications;
- ♦ Add some form of inter-user communication (personal message or forums/message board);
- ♦ Add the ability to download courses which are made available by the teachers;
- ♦ Add file submission system for assignments/homework.
- ♦ Add students' feedback regarding teacher's performance.

4.3 Summarization

The idea of creating our own University Management System materialized from the frustration due to:

(a) the lack of intuitiveness and good functionality of a certain UMS platform

(b) the need for more features than existing ones provide

(c) having to use multiple software to do what a single one can.

We are responsible for designing the concept of a 'hybrid' UMS (both university- and student-centered), design a working model for the database system, and implementing a rough working demo of our client and server applications. Hopefully, our software will be able to become a full-fledged UMS in the Romanian market or even world-wide.

Acknowledgment

(i) This work was supervised by Professor Ph.D. *Dana Simian*, from *Dept. of Mathematics and Informatics, Faculty of Sciences of the Lucian Blaga University of Sibiu*.

(ii) Icons found throughout this document and the accompanying application are part of the free, attribution-required Icons8 platform (`https://icons8.com`). The icons pack used in this document were *Wired* and *1em*.

References

[1] Duncan de Borde. *Two-factor authentication*. Siemens Insight Consulting. 2012. URL: `https://web.archive.org/web/20120112172841/http://www.insight.co.uk/files/whitepapers/Two-factor%20authentication%20(White%20paper).pdf`.

[2] Raluca Budiu. *Mobile: Native Apps, Web Apps, and Hybrid Apps*. Nielsen Norman Group. URL: `https://www.nngroup.com/articles/mobile-native-apps/`.

[3] E. F Codd. *The Relational Model for Database Management*. Addison-Wesley. ISBN: 0-201-14192-2.

[4] David Flanagan. *JavaScript - The Definitive Guide*. 5th ed. O'Reilly, 2006, p. 497.

[5] Google. *Material Design*. URL: `https://material.io/`.

[6] The PHP Group. *PHP Usage Stats*. 2013. URL: `http://php.net/usage.php` (visited on 05/12/2017).

[7] solid IT GmbH. *DB-Engines Ranking. Popularity ranking of database management systems*. 2017. URL: `https://db-engines.com/en/ranking` (visited on 05/13/2017).

[8] Thomas Pornin. *Why passwords should be hashed*. Stack Exchange Security Blog. URL: `http://security.blogoverflow.com/2011/11/why-passwords-should-be-hashed/`.

[9] PHP Pot. *PHP RESTful Web Service*. URL: `http://phppot.com/php/php-restful-web-service/`.

[10] Lucian Blaga University of Sibiu. *Hartă locații*. Romanian. URL: `http://dga.ulbsibiu.ro/harta/`.

[11] Stephen Thorpe. *Creating Web Services with PHP and SOAP, Part 1*. URL: `https://www.sitepoint.com/web-services-with-php-and-soap-1/`.

Doran Adoris Elian
Lucian Blaga University of Sibiu
Faculty of Sciences, Dept. of Informatics
Str. Dr. I. Rațiu, No. 5-7, Sibiu 550012
România
E-mail: *elian.doran@ulbsibiu.ro*

Negru Alex Damian
Lucian Blaga University of Sibiu
Faculty of Sciences, Dept. of Informatics
Str. Dr. I. Rațiu, No. 5-7, Sibiu 550012
România
E-mail: *alex.negru@ulbsibiu.ro*

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

SmartViewReward

Ioana Teodora Duma

Abstract

SmartViewReward is a project for mobile devices in Android Studio, created to help children spend less time with their gadgets and learn something every time they use them. The application will lock the smartphone screen and the child can play games or watch videos only if he answers correctly a few questions. The questions for now are about mathematical equations, but in a future version they will be from multiple domains. The application is proposed as a solution to one of the problems of parents in the digital age, namely that children spend too much time on the smartphone and can no longer concentrate with school. In this paper, I will present the motivation, the analysis and the implementation of the project.

1 Introduction

Children are motivated by rewards they receive on the spot and they do not realize that what they learn will be helpful in life. To influence them in a positive way and give them the motivation they need, SmartViewReward was created. Shortly, after children respond correctly to some answers, they gain access to their mobile phones. This way, they will motivate themselves to enter the phone, and will start learning to be able to answer the questions, or they will get bored and search for other fun activities besides the phone itself.
SmartViewReward is written in Android Studio [5] which is an IDE where learning java is made easier. I chose java because it is a native language for phones with Android, an operating system being used by 76.6% of the smartphones. In this way, the application will be able to run on a wide range of phones and many children can benefit from it.
The application's target users are children between 4 and 16 years old, who are too focused on their smartphones. A couple of students said that they would like to use it because of the social media applications which consume their time and distracts them from learning. Using the application, they can focus on the courses and get better grades.

2 The MinionMath game

A first version of SmartViewReward is MinionMath game which was inspired by the children's need for motivation to do something productive. Written in python [2], using the Eclipse IDE, in MinionMath the user had to solve some mathematical equations to gain access to the game itself.
Characters from the Minion movie [1] are used as a theme of the game, with funny images to get the child's attention. It is used Tkinter as a GUI package [4] for the python application.

In the main layer (fig. 4) the player can see a cute minion and insert the user name.

2.1 The UI

The menu contains a label with information about game, a picture with minions, a label with points and four buttons:
“Adunare”, “Scadere”, “Joc”, “Iesire”
Most of the children will press "Joc" button which means "Game" and will wait for the game to begin, but this wouldn't happen because they need 10 points. Points are accumulated from good answers at addition or substitution operations. They need to press "Adunare" (Addition) or "Scadere" (Substitution) buttons.

2.1.1 “Adunare” & “Scadere”

Sends the user to a layer which shows a random addition/substitution operation, a background image with a movie character and two buttons (fig. 3), the terms of the operations are random generated using random module. After the answer is entered in the empty box and the “Verifica” button is clicked, the game will show a message with the result and you get a point if the answer is correct. (fig.1)

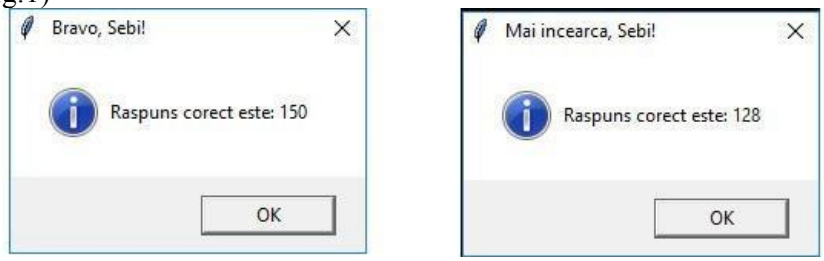


Fig. 1 - Good and wrong answer

By button “Inapoi” you can go back to the menu.

2.1.2 “Joc”

In the game, the user can enter only if he has accumulated at least 10 points and he can stay as many seconds as points. At that time, the user had to catch many random minions, which appeared on the screen every second. This game develops the attention and reaction speed of the children, while the Minions are making it funny and attractive.

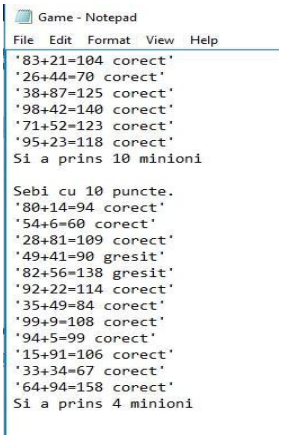


Fig. 2 - Player information

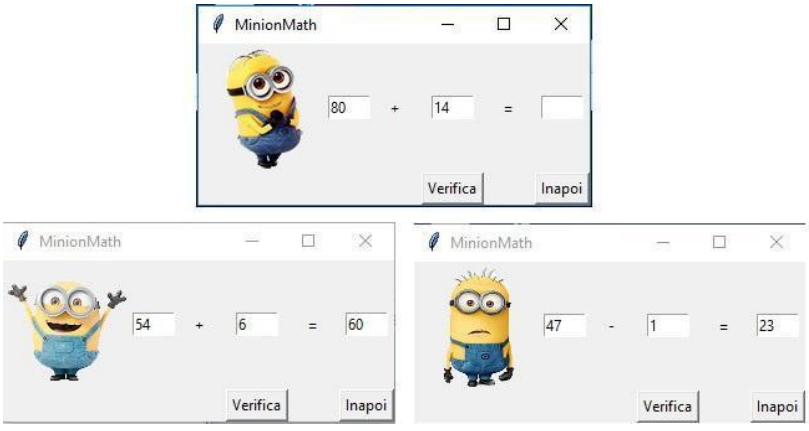


Fig. 3 - Layers

2.1.3 "Iesire"

This button closes the game and writes in a file the information about the player. The parents can use the data to see the progress of their child.

Fig. 4 - The MinionMath game

2.2 The code

2.2.1 Player class

```
class jucator(object):
    def __init__(self, prenume, pct):
        self.prenume=prenume
        self.pct=0
        self.pct_joc=[]
        self.opList=[]
    def addOp(self, op):
        self.opList.append(op)
    def addToFile(self):
        f=open("Game","a")
        f.write(self.prenume+" cu "+str(self.pct)+"
puncte."+'\n')
        for s in self.opList:
            f.write(repr(s)+'\n')
        sum=0
        for i in self.pct_joc:
            sum+=i
        f.write("Si a prins "+str(sum)+"
minioni"+"\n")
        f.write("\n")
        f.close()
```

2.2.2 Operation class

```
from random import *

class operatie(object):

    def __init__(self):
        self.TermGenerate()

    def TermGenerate(self):
        c=Random()
        n=100
        self.t1=c._randbelow(n)
        self.t2=c._randbelow(n)
        self.op=""
    def Adunare(self):
        return self.t1+self.t2
    def Scadere(self):
        if self.t2>self.t1:
            self.t2=self.t2+self.t1
            self.t1=self.t2-self.t1
            self.t2=self.t2-self.t1
        return self.t1-self.t2
```

3 SmartViewReward

MinionMath is a computer game that is harder to be used by children, they prefer smartphones because they can play it anywhere and anytime. That's how SmartViewReward was born. Created specifically for children's smartphones, on the same operating principle as MinionMath.

Fig. 5 - SmartViewReward icon

Fig. 6 - Default view on the phone

There are two kinds of reactions from the child regarding this application:

- he gets bored and gives up his phone
- learns to answer the questions

3.1 Rules and functionalities:

3.1.1 The child:

When the function "Lock screen" is activated, at the opening of the phone the child's menu is shown (fig.6) at which the child can either enter the parent preset password, or answer the questions to unlock the phone. Also, the time is shown, and there is an emergency call button, so they can call one of the parents.
To unlock the screen, the child must respond to some questions; if more than half of the answers are wrong, the child cannot unlock the phone, he needs to study more; if more than half are correct, the child unlocks the screen and can play a time set by the parent. They cannot play after the sleep time and there is a dedicated button for emergency calls
In a future version, the child will be able to choose from more domains the one which he enjoys the most, and depending on his skill, the generated questions will have a higher or lower difficulty.

3.1.2 The parent:

With the help of a switch button the parent can activate or deactivate the "lock screen" function of the application. He can also change answers, set the time, set the password and stop the application.
In the future version, the parent will be able to see statistics with the child's progress and see what domain the child likes.

3.2 The code for switch button

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_parent_menu);
    Switch switchOnLockScreen = (Switch) findViewById(R.id.switch1);
      switchOnLockScreen.setOnCheckedChangeListener(new
      CompoundButton.OnCheckedChangeListener() {
      @Override
      public void onCheckedChanged (CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
          startActivity(new Intent(ParentMenu.this, MainActivity.class));
        } else {
          stopService(new Intent(ParentMenu.this, LockScreenService.class));
        }
      }
    });
}
```

3.2 The code for locking the screen

```
public void onCreate() {
  super.onCreate();
  IntentFilter intentFilter = new IntentFilter(Intent.ACTION_SCREEN_OFF);
  registerReceiver(screenReceiver, intentFilter);
  windowManager = ((WindowManager) getSystemService(WINDOW_SERVICE));
  layoutParams = new WindowManager.LayoutParams(
      WindowManager.LayoutParams.MATCH_PARENT,
      WindowManager.LayoutParams.MATCH_PARENT,
      WindowManager.LayoutParams.TYPE_SYSTEM_ERROR,
      WindowManager.LayoutParams.FLAG_LAYOUT_IN_SCREEN
          | WindowManager.LayoutParams.FLAG_DISMISS_KEYGUARD
          | View.SYSTEM_UI_FLAG_HIDE_NAVIGATION,
      PixelFormat.TRANSLUCENT);
}
```

4 Discussion

The children were happy about this idea, but not for a long time. When they realized that this is not a joke and they really need to study to gain access on the smartphone, they got very upset, but in time they got used to it and progressed in the ability to calculate mathematical equations. Also, they don't use the phone so much time as before and spend more time with other games (Lego, football etc.). The parents are happy because they don't need to worry about the child's waste of time and his loss of interest for school lessons.
Psychologist opinion: "I think that it is a very good idea.", "Children who start to play for the first time on the smartphone will adapt easier to the application.". Features proposed by the psychologist: emergency button, limited time, shut down after 10 p.m., test the application with groups of parents and children

5 Future development

A future development of the application is a platform for teachers where they can introduce questions from other domains. In this way there will be more types of questions so children can choose from the desired field, introduce statistics for the parent menu and include more themes and colours for parents and children to enjoy the application even more.

The application will be tested with some children from a school and their parents [6], to see what the reactions are, what type of questions are chosen by children and what proposals will have the parents.
A new age is in front of us, an age of smart people who will solve more and more difficult problems. Every generation is smarter than the last one and is our duty to develop our children and make them smarter, to guarantee them a good and happy life.

Acknowledgements: The project was coordinated by drd. Bota Florentin, Babes-Bolyai University
Several features were suggested by Psychology Lecturer Dr. Dan Octavian Rusu, Babes-Bolyai University
"Minions" are Illumination Entertainment Productions. The author is in no way affiliated with Illumination Entertainment, or the Minions official website. MinionMath game is an application made by a fan and was not distributed to the public.

References

[1] Minions [cinema] produced by Illumination Entertainment, [1 June 2015], https://en.wikipedia.org/wiki/Minions_(film) .
[2] Object-oriented programming laboratory, python language by drd. Bota Florentin. Available: http://www.cs.ubbcluj.ro/~botaflorentin/. [2016]
[3] Zed A. Shaw. *Learn Python the hard way*. A very simple introduction to the terrifyingly beautiful world of computers and code. Addison-Wesley, 2013.
[4] TkInter module, [Online]. Available: https://docs.python.org/2/library/tkinter.html [2016]
[5] Android Studio(IDE), [Online]. Available: https://developer.android.com/training/index.html [2017]
[6] Macgill, Alexandra Rankin. *Parents, teens and technology.* Pew Internet and American Life Project (2007).

IOANA TEODORA DUMA
Babes-Bolyai University
Faculty of Mathematics and Computer Science
Mihail Kogalniceanu nr.1, 400084, Cluj-Napoca,
ROMANIA
E-mail: ioanad1220@gmail.com

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

TestIT

Florea Adrian-Cristian

Abstract

TestIT is an application designed for phones with Android operating system, which aims to record the phone's performance, both at the hardware level by recording CPU temperature and battery charging time and at the software level by detecting the copying and creating speed of a file, respectively the download and upload speed of a network. Following these tests, the app creates various statistics that not only determine the performance of the phone, but can also suggest improvements that can be made to the device's thus preventing a possible replacement with a newer and therefore more expensive model. The application was developed in Android Studio v2.2.3, with tests carried out both through Genymotion emulator and on my mobile phone.

1 Introduction

Today, almost everyone has a smartphone and children and students use them more than computers, they are a part of our daily life. Software application development has become a desirable skill not only in the IT field, but also in other technical and less technical fields. [1] [2] Nevertheless, Steve Jobs also insisted on this idea saying that *"Everybody in this country should learn to program a computer, because it teaches you how to think"*. [3]

TestIT is an application designed for all smartphone users with Android operating system, which aims to record the performance level of the phone through different tests at both hardware and software level. The idea of developing this application derived from a passion, which evolved during the 11th grade, while I was attending one of my Android classes, that is taught in my school. The era we live in is an age of speed, progress and continuing development and the desire to bring something innovative and useful our world, led me to create this application. The purpose of this application is to check some of the phone functionalities and to inform the owner, if there are improvements that can be made to his device and also to establish the performance of his smartphone.

Google Play application package provides several free and paid applications dedicated to identifying available sensors in your phone. Some of these application are "Sensor Box for Android", "AndroSensor" and "Sensors Sense". However, not only as the saying "Do It Yourself" goes, which states that the best learning method is by developing your own applications, but also under the supervision of a teacher passionate about everything new in programming intelligent systems I considered helpful and also liked to implement an application in Android capable of achieving the Human – Computer interaction acquiring some useful data about the performance of

the smartphone. In the near future I would like to implement the Human – Computer – Environment application interface. My choice to program in Android has been motivated by the fact that, after a market survey conducted in 2012, 49.2% of mobile devices sold worldwide were based on Android System Architecture. In addition to the previous mentioned information, in November 2014 Ericsson experts forecast that by 2020 more than 90% of the population which is older than 6, will have a mobile phone. [4], [5]

2 User interface and application functionality

The first activity of the application is exhibited in Fig. 1 and shows the copy and create file menu that allows the user to create a file on his device, in order to calculate the average speed used for creating or copying a file. The file used for this action is created in the default location of the application.

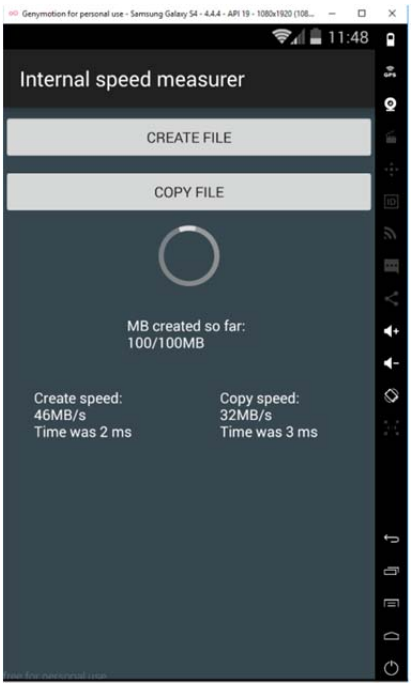


Fig 1. Measurements of the creating and copying speed

Passing from one page to another page of the application is done using the Android Gestures facility. [6] Through a Swipe, the user can move to another functionality of the app, the CPU temperature measurement, in order to establish the effect it has at the smartphone's performance level. In order for the temperature to be measured, the mobile phone needs to have a temperature sensor or it can be read by accessing a reserved folder. If the phone does not have a temperature sensor, a list with the available sensors is shown, as you can see in Fig. 2.

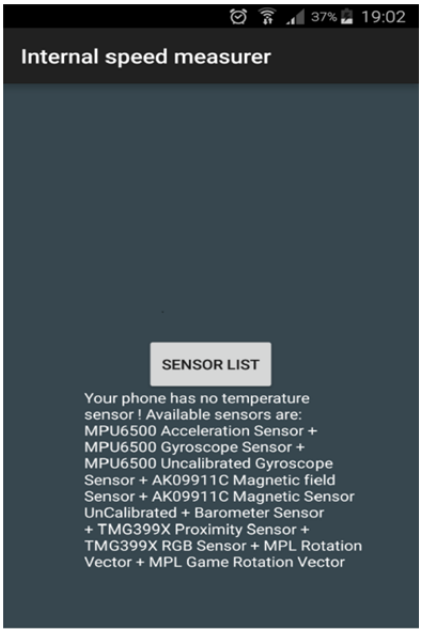


Fig 2. List of the available sensors

Sliding further, the third functionality of the application is being reached. By pressing the Test It button, a 5MB .zip file is being downloaded and the download time and speed are being measured.

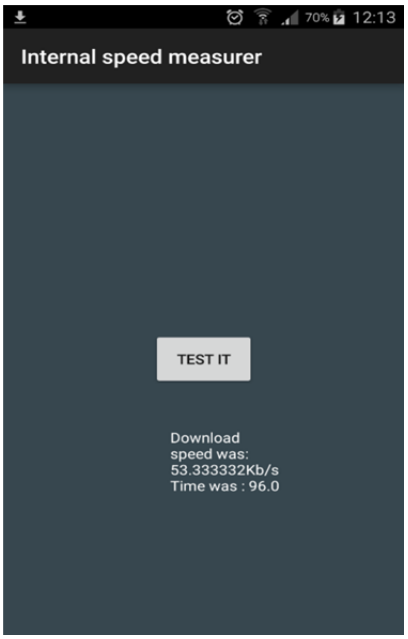


Fig 3. Download time and speed

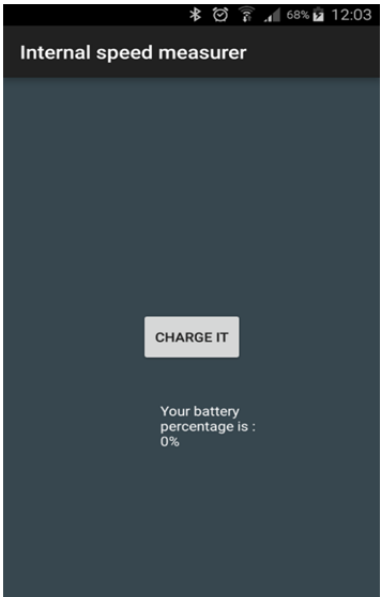


Fig 4. Battery percentage

In Fig. 4, the user reaches another option of the application, which measures the battery charging speed and the time needed for a full charge.

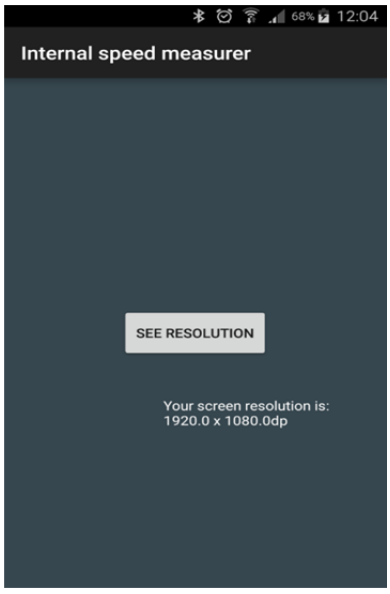


Fig 5. Screen resolution

The last part of the application, as shown in Fig. 5, measures the screen resolution and displays it in density-independent pixels (dp).

3 Application development software

3.1 Programming environment – Android Studio

TestIT was first developed in Android Studio v2.2.3 and v2.3, using Java language skills. Throughout its development I used Android Software Development Kits – Android SDKs, that contained the API interfaces which I had to use in order to run some application tests. Application tests were carried both on the Genymotion Android Emulator and on Samsung Galaxy S4, S5 and S6 devices. [7]

3.2 Measuring create and copy speed

After launching the application, two buttons appear on the user's screen that can allow him to create a 100 MB file, which can be copied in the same folder, where the creating process took place. To copy the file, Create File button must act first. After building the file called 100BigFile1.dat, the creating speed can be set using the formula creating speed = (file size) / (time needed to create the file). If the file was created and File Copy button has been activated, the original file is copied and renamed 100BigFileCopy.dat in the same parent folder through the CopyFile method.

The CreateFile method receives as a parameter the file size and creates a 100 MB file through an array of bytes and sets it the name 100BigFile1.dat. In order to record the time needed for the two operations I saved the starting and ending time in two int variables start1 and end1. Create speed = (file size) / (time needed for creating the file). To provide a dynamic interface for the application, I chose to display at each 5 MB how many MB have been copied until that moment.

```
private void CreateFile(final int fileSizeInMB) throws IOException, InterruptedException {

    start1 = SystemClock.elapsedRealtime();
    int bufferSize = 1024*1024;
    byte[] buf = new byte[bufferSize];
    OutputStream out = openFileOutput(fileSizeInMB + "BigFile1.dat", MODE_PRIVATE);
    for(int i = 0;i<=fileSizeInMB;i++){
        final int p = i;
        if(i%5==0)
        {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    textview.setText("MB created so far: " + p + "/" + fileSizeInMB + "MB");
                }
            });
        }
        out.write(buf, 0, bufferSize);
    }

    size1 = fileSizeInMB;
    out.close();
    end1 = SystemClock.elapsedRealtime() - start1;
    x1 = (size1*1000)/end1;

    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            spinner.setVisibility(View.GONE);
            textview2.setText("Create speed: " + x1 + "MB/s\nTime was " + (end1/1000) + " ms");
        }
    });
}
```

Fig 6. Code sequence for measuring the creating speed of a file

Inside the method CopyFile, the file is being copied under the name of 100BigFileCopy.dat in the same folder as the original file 100BigFile1.dat. Before copying the file, the application checks if the file was created before and erases it. The copying process is being done using a byte array. To calculate the ellpased time, I used two int variables to save the start and end time of the process. Using the formula (file size) / (time needed copying the file) we get the copying speed.

```
private void CopyFile(String source,String destination) throws IOException {

    start2 = SystemClock.elapsedRealtime();
    file = new File(getApplicationInfo().dataDir + "/files/" + source);
    fileSizeInMB=100;
    File destFile = new File(getApplicationInfo().dataDir + "/files/" + destination);

    if(!destFile.exists()){
        destFile.delete();
    }

    InputStream in = openFileInput(source);
    OutputStream out = openFileOutput(destination, MODE_PRIVATE);

    byte[] buf = new byte[1024*1024];
    int len;
    for(int i = 0; (len = in.read(buf)) > 0; i += len) {
        final int totalSize = i;
        if(i%5==0)
        {
            runOnUiThread(() -> {
                    spinner.setVisibility(View.VISIBLE);
                    textview.setText("MB copied so far: " + (totalSize/1024)/1024 + "/" + fileSizeInMB + "MB");
            });
        }
        out.write(buf, 0, len);
    }
    in.close();
    out.close();

    end2 = SystemClock.elapsedRealtime() - start2;
    x2 = (size1*1000)/end2;

    runOnUiThread(() -> {
            spinner.setVisibility(View.GONE);
            textview1.setText("Copy speed: " + x2 + "MB/s\nTime was " + (end2/1000) + " sec");
    });

    destFile.delete();
    file.delete();
}
```

Fig 7. Code sequence for measuring the copying speed of a file

3.3 Measuring temperature

The first possibility to measure the temperature involves the existence of a temperature sensor. The application measures the ambient temperature and it is desired to determine its effect on the phone's CPU. Android Studio allows the activation of a sensor service including the temperature sensor (Sensor.TYPE_AMBIENT_TEMPERATURE) and every slight change in temperature (onSensorChanged) is displayed on the screen. Unfortunatelly, tests carried out on Samsung Galaxy S4, S5 and S6 devices revealed that they do not have such a sensor. As a result, the following image displays the values obtained when using the accelerometer, which is an available sensor on the previously mentioned devices. [8]

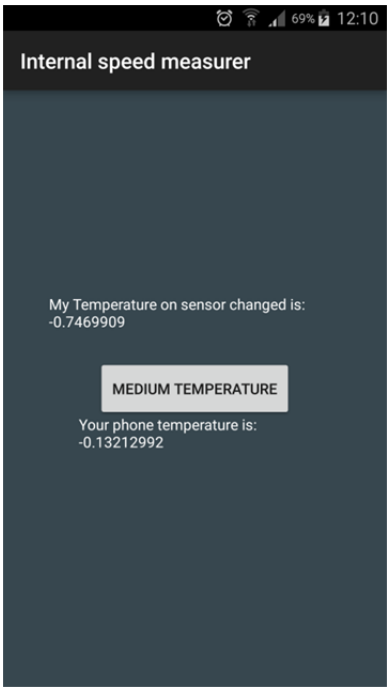


Fig 8. Results offered when using the accelerometer

```
mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
mTempSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
mSensorManager.registerListener(this, mTempSensor, SensorManager.SENSOR_DELAY_NORMAL);

if(mTempSensor!=null)
    button2.setText("Medium Temperature");

if(mTempSensor!=null)
    mySensorList.setText("Your phone temperature is: "+(alltemp/ct));
```

Fig 9. Code sequence in order to be able to use the accelerometer

Another possibility to measure the temperature involves the use of the adb shell tool in command line that allows communication via Linux commands to the device and, this way, the thermal_zone files are being listed. This files contain data about the temperature of the processor. There are only 3 folders containing details that can suggest the proper temperature. [9]

I connected to a shell and typed the commands below. The adb executable is located in the platform-tools folder that comes with the SDK.

```
$ cat sys/class/thermal/thermal_zone1/type
mtktscpu

$ cat sys/class/thermal/thermal_zone1/temp
30600
```

The temperature is listed above, because the first command returns "mtktscpu". MTK is the processor of the device I have run the tests on, while 30600 means 30.6 degrees Celsius.

```
try {
    String commandLine = "cat sys/class/thermal/thermal_zone11/temp";
    Process process = Runtime.getRuntime().exec(commandLine);
    InputStreamReader streamReader = new InputStreamReader(process.getInputStream());
    BufferedReader reader = new BufferedReader(streamReader);
    String temp = reader.readLine();
    myTempValue.setText(temp);
} catch (Exception e)
{
    e.printStackTrace();
}
```

Fig 10. Code sequence for reading the temperature from the thermal_zone11/temp folder

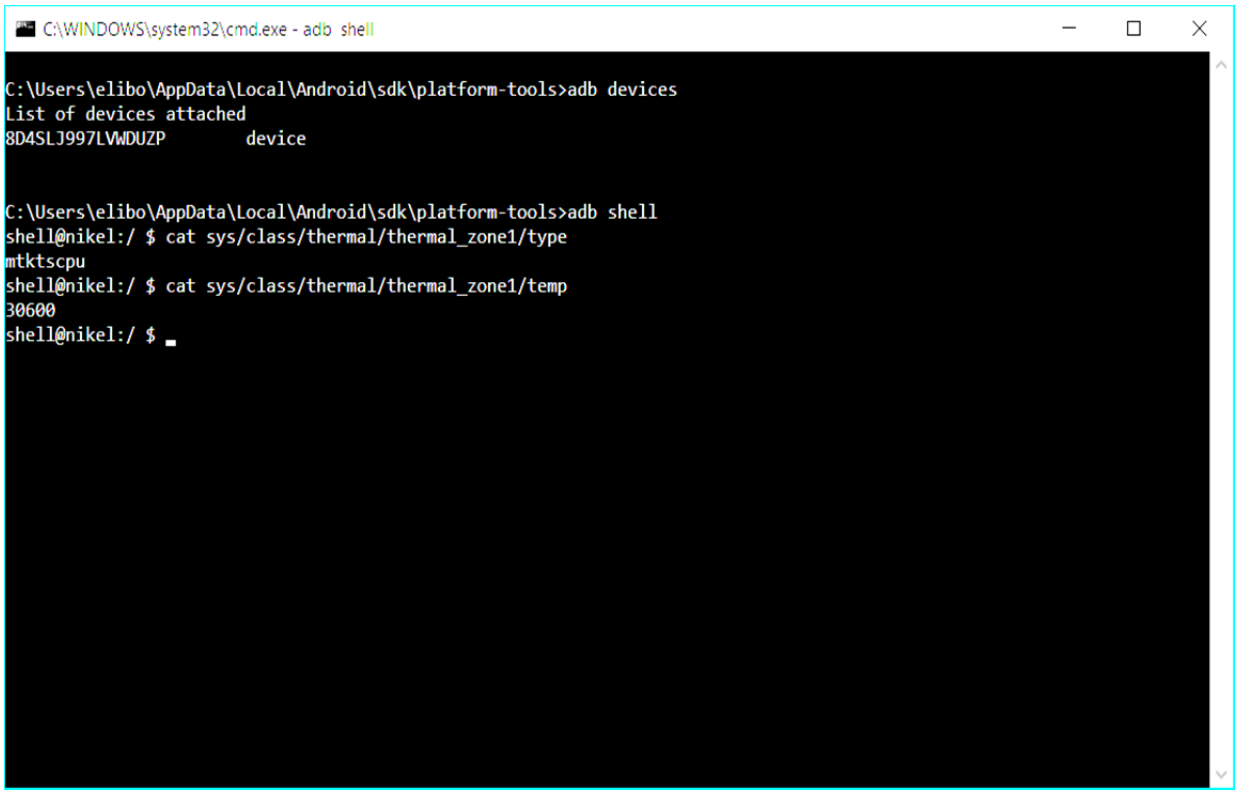


Fig 11. Terminal sequence using the adb shell command in order to read the temperature from the thermal_zone11/temp folder

3.4 Listing available sensors

If the device does not have a temperature sensor, the application lists all the available sensors, as a list, in order for the user to see all the available sensors. [10]

```
public void MyTemp(View v) {
    /*Toast.makeText(getApplicationContext(), myTempValue.setText("My Temperature value is: "+temp);
            "sensor temp = " + temp, Toast.LENGTH_LONG).show();*/

    if(mTempSensor!=null)
       mySensorList.setText("Your phone temperature is: "+(alltemp/ct));
     else{
     mySensorList.setText("Your phone has no temperature sensor ! Available sensors are: ");

    List<Sensor> sensors = mSensorManager.getSensorList(Sensor.TYPE_ALL);
    for (Sensor sensor : sensors)
        mySensorList.append(sensor.getName()+" + ");}
}
```

Fig 12. Code sequence listing phone sensors

3.5 Measuring download speed

In order to be able to access the internet, the application needs to have some permissions set. TestIT application connects to the internet, downloads a file from a predefined location and also plans to measure internet download speed. At the moment, download time measurement has not been fully implemented. [11]

```
public void DownloadStuff(View v) {

    float start = SystemClock.elapsedRealtime();
    String url = "http://ipv4.download.thinkbroadband.com:8080/5MB.zip";
    DownloadManager.Request request = new DownloadManager.Request(Uri.parse(url));
    request.setDescription("Downloaded file");
    request.setTitle("TestFile");
    // in order for this if to run, you must use the android 3.2 to compile your app
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
        request.allowScanningByMediaScanner();
        request.setNotificationVisibility(DownloadManager.Request.VISIBILITY_VISIBLE_NOTIFY_COMPLETED);
    }
    String s = "TestFile.zip";
    request.setDestinationInExternalPublicDir(Environment.DIRECTORY_DOWNLOADS, s);

    // get download service and enqueue file
    DownloadManager manager = (DownloadManager) getSystemService(Context.DOWNLOAD_SERVICE);
    manager.enqueue(request);

    Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS+s).delete();
    float end = SystemClock.elapsedRealtime() - start;
    textview.setText("Download speed was: " + (5/end)*1024 + "Kb/s\nTime was : " + end);
```

Fig. 13 Code sequence for measuring download time and speed

3.6 Measuring battery charging speed

The application wants to measure the current battery percentage, repeating this process after a few seconds, calculating the difference between percentages in order to estimate not only the time needed for a full charge, but also the battery charging speed. At the moment, battery charging time has not been fully implemented.

```
public void ChargeBattery(View v)
{
    IntentFilter ifilter = new IntentFilter(Intent.ACTION_BATTERY_CHANGED);
    Intent batteryStatus = context.registerReceiver(null, ifilter);

    int level = batteryStatus.getIntExtra(BatteryManager.EXTRA_LEVEL, -1);
    int scale = batteryStatus.getIntExtra(BatteryManager.EXTRA_SCALE, -1);


    float batteryPct =  level / (float)scale;
    textview.setText("Your battery percentage is : " + batteryPct + " %");
}
```

Fig. 14 Secvență de cod pentru măsurarea vitezei de încărcare a bateriei

3.7 Measuring screen resolution

Measurements of the screen resolution return the number of pixels of the device's screen. All the facilities listed so far are designed to establish the performance level of the device on which TestIT is tested on.

```
public void MeasureResolution(View v)
{
    Display display = getWindowManager().getDefaultDisplay();
    DisplayMetrics outMetrics = new DisplayMetrics ();
    display.getMetrics(outMetrics);

    float dpHeight = outMetrics.heightPixels;
    float dpWidth  = outMetrics.widthPixels;

    textview.setText("Your screen resolution is: " + dpHeight + " x " + dpWidth + "dp");
}
```

Fig. 15 Code sequence measuring screen resolution

4 Conclusions and further work

Android operating system sensors are listed in three categories: motion, location and environmental sensors. In this article, I used motion sensors – the accelerometer, as a substitute for the temperature sensor, because the devices I tested the application on, did not possess such a functionality. Furthermore, this project targeted the performance of intelligent phones, measuring the screen resolution, the data download speed when connected to the Internet, the create and copy speed of a file and the CPU temperature, especially when multiple games or apps are being accesed. The application is far from being complete. I will continue to add new features when using the existing motion sensors, like the accelerometer, in order to be able to track the older persons that live by themselves and if an abnormal change is being detected by the accelerometer, a fall may be referred and the family members will be notified about this incident via SMS. I will use the environmental sensors to communicate with other external sensors in order to determine some climatic parameters like humidity, atmospheric pressure, pollen, dust, or even the pollution degree in the air we breathe. This will affect not only the world we work in, but also the one we live in. I also intend to realize an effective battery management by enabling or disabling certain applications depending on certain conditions. I am working now also on changing the user interface of the app to make it more easier to use. Nevertheless, I would like to port the application on other mobile platforms such as Windows Phone, iPhone or iPad.

Acknowledgements: This work was supervised by Professor *Delilah Florea*, from "Samuel von Brukenthal" National College of Sibiu.

References:

[1] Riley, D., 2012, *Using Mobile Phone Programming to Teach Java and Advanced Programming to Computer Scientists*, Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, pp. 541-546, North Carolina, USA.

[2] Tigrek, S., Obadat, M., *Teaching smartphones programming using (Android Java): Pedagogy and innovation*, International Conference on Information Technology Based Higher Education and Training (ITHET), pp. 1 – 7, 2012, Istanbul, Turkey.

[3] Rosoff, M, 2011, *The Most Interesting Things Steve Jobs Said In A "Lost" Interview Showing Next Week*, Business Insider, Tech Branch.

[4] Pereira, O.R.E., Rodrigues, J.J.P.C. (2013) *Survey and analysis of current mobile learning applications and technologies*, ACM Computing Surveys, 46(2), pp.1–35.
[5] Ericsson Energy and Carbon Report Including Results from the First ever National Assessment of the Environmental Impact of ICT, November, 2014, https://www.ericsson.com/assets/local/about-ericsson/sustainability-and-corporate-responsibility/documents/ericsson-energy-and-carbon-report.pdf, accessed 14 January 2017.
[6] Udemy Online Courses, https://www.udemy.com/courses/search/?q=the%20complete%20android%20developer%20course&src=ukw, accessed 03 February 2017.
[7] Android Developer Tools, 2014, https://developer.android.com/studio/intro/index.html, accessed 05 March 2017.
[8] Stack Overflow, http://stackoverflow.com/questions/11931681/how-to-get-device-s-temperature-in-android, accesed 07 March 2017
[9] Stack Overflow, http://stackoverflow.com/questions/20771070/how-do-i-get-the-cpu-temperature, accessed 02 March 2017.
[10] Aram, S., Troiano, A., & Pasero, E. (2012, February). *Environment sensing using smartphone. In Sensors Applications Symposium (SAS)*, 2012 IEEE (pp. 1-4). IEEE.
[11] http://stackoverflow.com/questions/19258254/how-to-correctly-measure-download-speed-with-java-android, accesed 06 March 2017

ADRIAN-CRISTIAN FLOREA
”Samuel von Brukenthal” National College
Informatics and Computer Science
Sibiu, Romania
E-mail: acflorea99@gmail.com

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

Art School "Art Popovo" – Web Application

Svetoslav Hadziivanov

Abstract

Websites are somehow mandatory nowadays. With the time passing Internet and the technologies are growing very fast. Building a website is not that hard but it takes a while to learn and understand the basics. The paper provides information on why each of us should have a personal website and how is the art school's website build. The realization of some basic features is described. The main tools and techniques for optimization and website security are discussed.

1 Introduction

Everybody should have their own website. That's because it's easier to inform people, share opinions, publish news, achievements, etc. It is a big bonus which may work even as an advertisement. Many people rely on the information posted there. However, the things aren't like that in the small settlements, at least in Bulgaria. The most of the people aren't used to Internet and they still prefer to ask someone rather than start their PCs and look in the network. Many people might have been witnesses how hard it is to find an address, specific information or something else. The author of the website came up with that idea and he decided to create the website.

2 How the app works

2.1 Main functionality

The design of the website is based on Bootstrap v3.3.7 [1]. Bootstrap is an open-source Javascript framework developed by the team of Twitter. It's a combination of HTML, CSS, Javascript code designed to help users build interface components. Nowadays, the most of the websites are using it. Art School's website is valid for HTML5 and CSS3. It has some Javascript codes like jQuery UI, jQuery Easing, jQuery Form Validator, ScrollToTop, Livestamp "Time ago" [2] and let's not forget the jQuery library which is mandatory for some Bootstrap components and the most of the scripts.

The website contains both types raster and vector graphics images. The raster graphics images consist of pixels unlike the vector ones which are based on vectors, which lead through locations called control points or nodes. The vector graphics images have their advantage over the raster that their quality remains the same after a zoom and their loading speed is also better (Fig. 1).

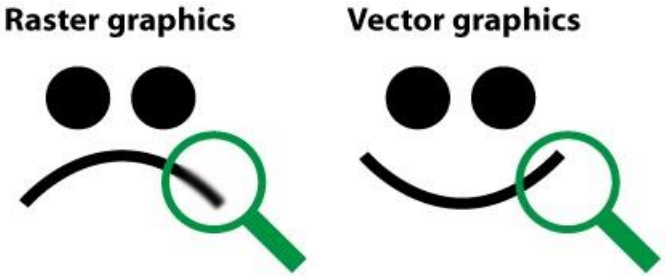


Fig. 1: Raster versus vector graphics

2.1.1 Optimization

Each serious website should be optimized at both loading speed and Search Engine Optimization (SEO). To make the loading speed faster, the author minimized the amount of the redirects, minified the CSS and JS files, set up gzip compression, checked for broken links, reduced the most of the images' size by changing their formats (for example, images that haven't transparent areas, can be JPEG instead of PNG), by making their actual size same as in the CSS file and by pushing their quality a bit so there wouldn't be any difference for the human's eye. A few website analysts [5] are used to check the loading speed of the website and then he fixed the major problems. In Fig. 2 the loading speed of the website is presented.

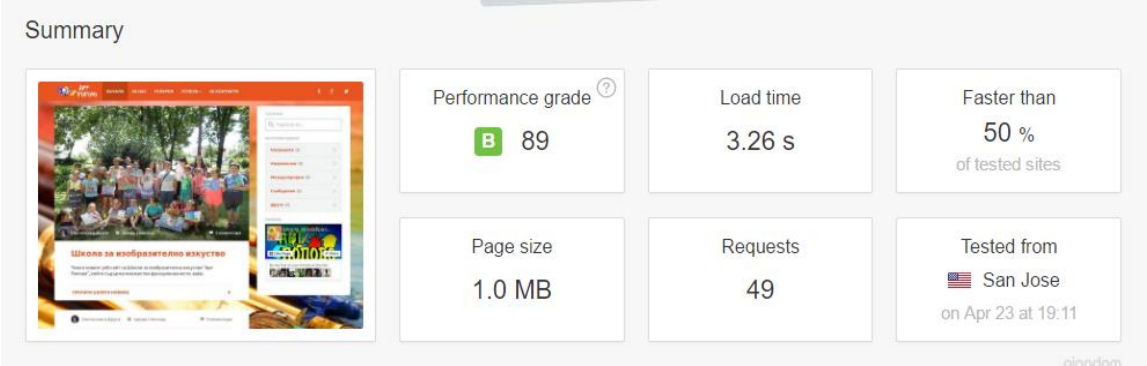


Fig. 2: Loading Speed Test performed by Pingdom Tool

The SEO is the second part of the site's optimization because it really matters if the website is going to be on the first page or somewhere else in the search engines. There are a few SEO checkers [6] which are telling us what's wrong and what has to be fixed to achieve the desired effect (Fig. 3). The missing rates are coming from the URL Canonicalization Test which means http://site.com can be accessed via http://www.site.com, turned on server signature, Libwww-perl Access Test and some more that are going to disappear once the website is uploaded at the real host.

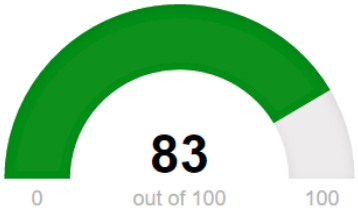


Fig. 3: SEO checkup performed by seositecheckup.com

Another part of the optimization is optimization for various browsers. Some commands in CSS are having different syntax which suits the browser's compatibility. Mozilla Developer Network [7] is preferred site for command's browser compatibility checking. In the figure below browser compatibility for "border-radius" is presented.

Browser compatibility

Desktop | Mobile

Feature	Firefox (Gecko)	Chrome	Edge	Internet Explorer	Opera	Safari
Basic support	1.0 (1.7 or earlier) -moz [1] 4.0 (2.0)[10]	1.0 -webkit 4.0[9]	(Yes) -webkit (Yes)	9.0	10.5[8]	3.0 -webkit 5.0[9]
Elliptical borders	3.5 (1.9.1)	(Yes)[7]	(Yes)	(Yes)	(Yes)	(Yes)[7]
4 values for 4 corners	(Yes)	4.0[6]	(Yes)	(Yes)	(Yes)	5.0[6]
Percentages	(Yes)[5] 4.0 (2.0)	(Yes)[2]	(Yes)	(Yes)	11.5[3]	5.1[2]

Fig. 4: border-radius browser compatibility

Another also important part is the mobile device optimization. Nowadays almost everyone has a smartphone or a tablet. It is essential the website to have a mobile version and to be optimized for different resolutions. Bootstrap applies some initial settings for low resolution devices. In Fig. 5 the mobile version of the website is presented.

Fig. 5: The website's mobile version

2.1.2 Security

Everyone who is writing PHP code should try to secure it as much as possible in the process of writing. What does that mean? Let's say http://domain.com/news.php?id=2. It will show us news with id: 2. The problem is that we haven't any type restriction and we can write anything else there. It means that our page is vulnerable to SQL Injection. SQLi is the most famous hacking method today which gives a hacker the opportunity to dump any information from the database and the possibility to upload a shell. Fig. 6 shows us where the vulnerability is coming from.

```php
<?php
$id = $_GET['id'];
$sql = "SELECT id, news, author FROM news WHERE id = ".$id."";
```

Fig. 6: SQL Injection vulnerability

To secure that we should cast the type we want, use prepare statements or htmlspecialchars. Nowadays, there are plenty of tools which help us to find vulnerabilities in a website. One of them is a program called Acunetix Web Vulnerability Scanner [8]. It finds even the smallest and the most harmless vulnerability. In Fig. 7 results from Acunetix scan are presented.

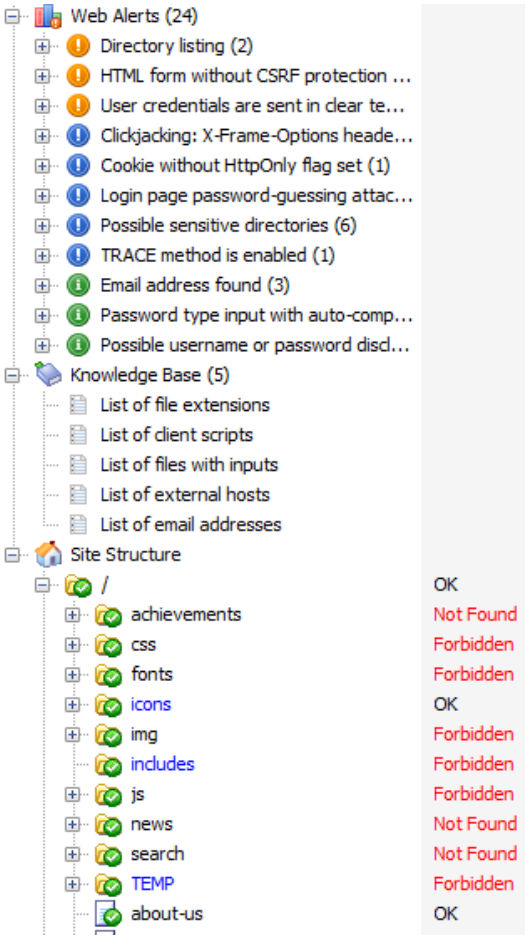


Fig. 7: Art School's website scanned with Acunetix

Furthermore, there are a few also famous vulnerabilities which are called Local File Inclusion (LFI), Remote File Inclusion (RFI) and Cross Site Scripting (XSS). An example for LFI and RFI could be a developer who includes a GET method. It gives the opportunity to a hacker to change the headers and upload a shell. The shell is a PHP file which works like FTP (File Transfer Protocol) and it gives us permissions to add, edit, delete or download files. We can set certain rules for our files in order to restrict hacker's permissions. If our host isn't secured enough and if we have an upload form somewhere (usually one which is used for user avatars) then we might upload a shell by sending changed headers and their content type to image/jpeg with Tamper Data. If that doesn't work, we can try spoofing the extension by making it looks like an image (for example, shell.php;.jpg), as long as the content type remains image/jpeg. However, to secure the variables while selecting data from the database the creator of the website used PHP Data Objects (PDO) connection and its prepared statements. For example, piece of code which is used in the achievements page to call out the selected category and its sort. The website uses PDO's prepared statements. As we see on the picture below, the variable cat has been bonded to an integer parameter with the same name. That makes sure the input for the category will be integer. In Fig. 8 how does the PDO's prepared statements look like is presented.

```
$stmt = $dbh->prepare("SELECT * FROM `achievements`
WHERE category = :cat ORDER by `sort` ASC");
$stmt->bindParam(':cat', $cat, PDO::PARAM_INT);
$stmt->execute();

$result = $stmt->fetchAll();
```

Fig. 8: PDO's prepared statements

The author also made a restriction in the htaccess file to prevent some folders from being scanned (Fig. 7). Furthermore, If you're hosting your website on your home PC make sure to remove the phpmyadmin folder and whenever you need it, put it back because it can be bypassed.

2.1.3 Website Preview

The main page consists of a news system, categories related to them, search and a Facebook page. The news system includes comments system by Disqus [9]. Many famous websites are using Disqus as news commenting system, even sites which are using WordPress (even though they have an integrated one). WordPress is a very famous open-source Content management system (CMS) which is preferred for blogs. There are a few more pages: view news, about the art school, gallery, achievements, contacts. View news' page has included options to share the news in the social mediaanas. About the art school page, it is pretty much only text but the gallery and the achievements pages consist of a PHP code which is selecting a data from the database and it can be edited through the administration panel. The contacts page gives us information for the developer and the leader of the art school. There is also a contact form which sends the messages to a chosen e-mail address by the developer. The administration panel is able to control almost everything in the website. It consists of a simple account system for authorization (possible features with it: edit profile, upload avatar, change password), view/add/edit/delete news, their categories, achievements and achievements categories. In the figures below are presented the most of the pages that the website contains.

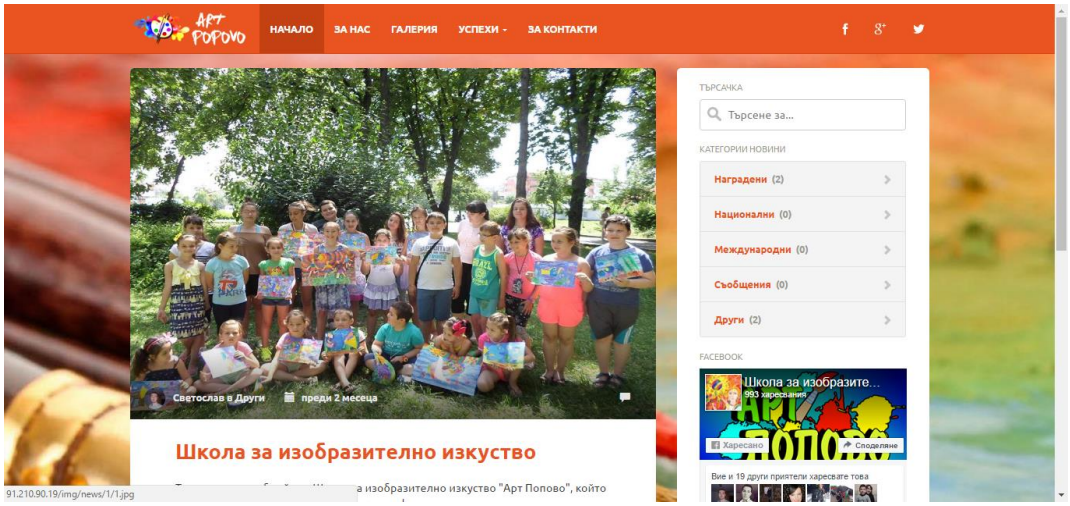


Fig. 9: Homepage

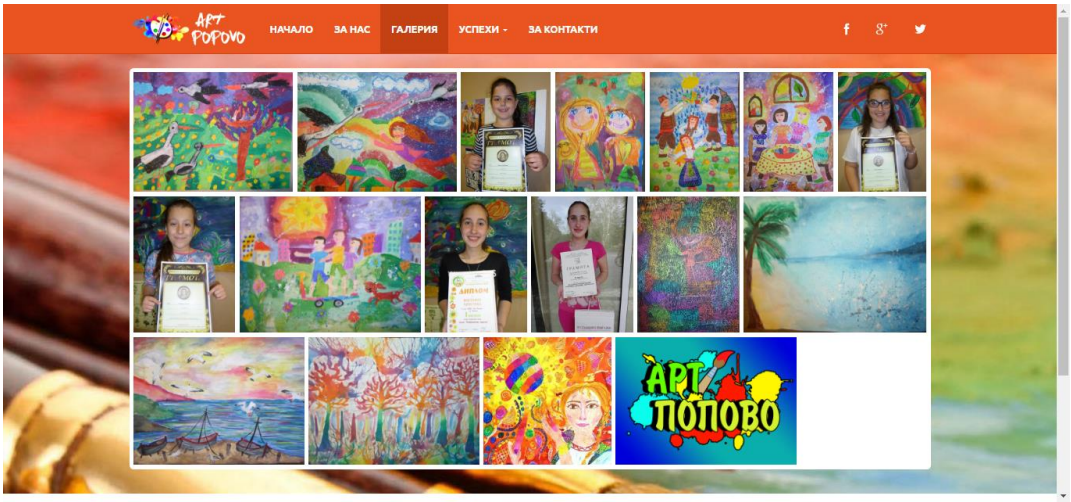


Fig. 10: Gallery page

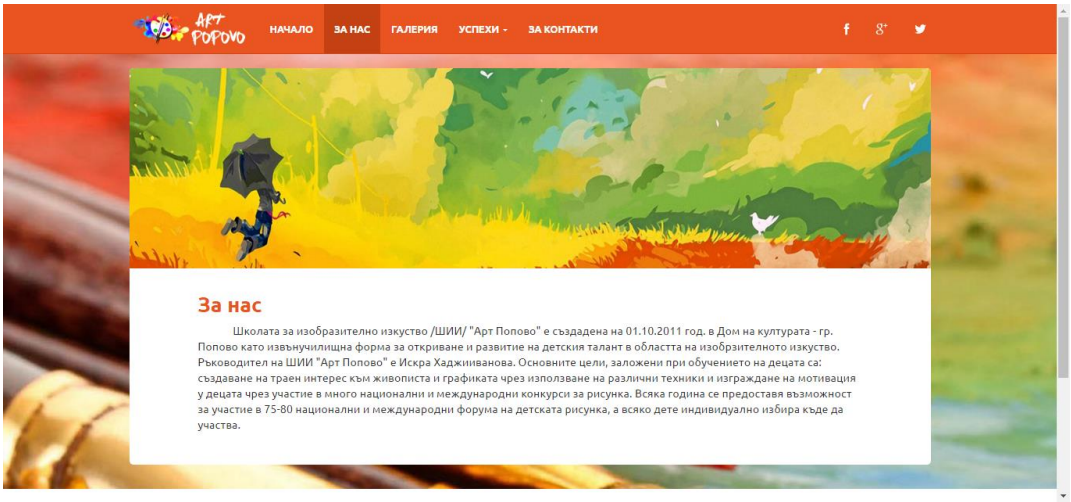


Fig. 11: About the Art School "Art Popovo"

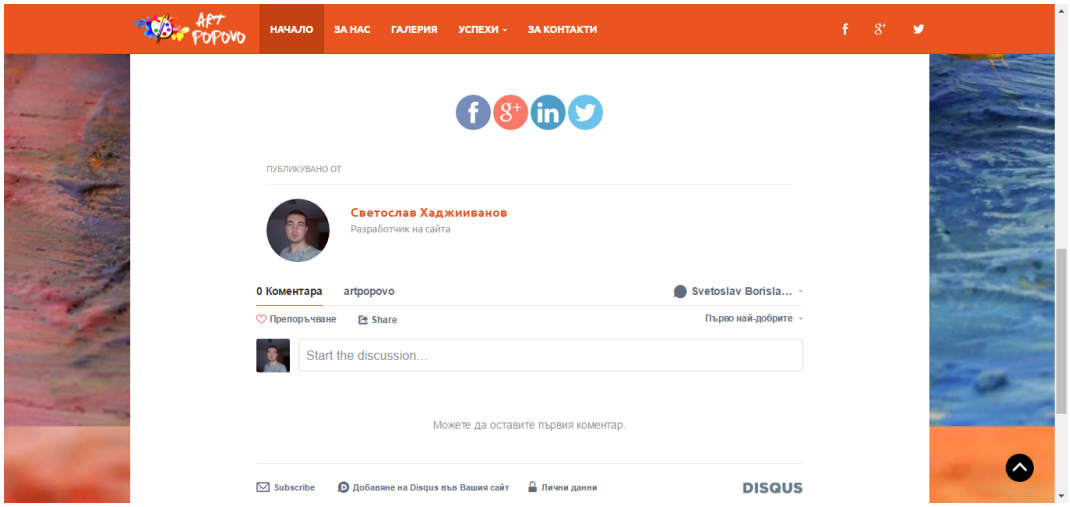


Fig. 12: Page view news

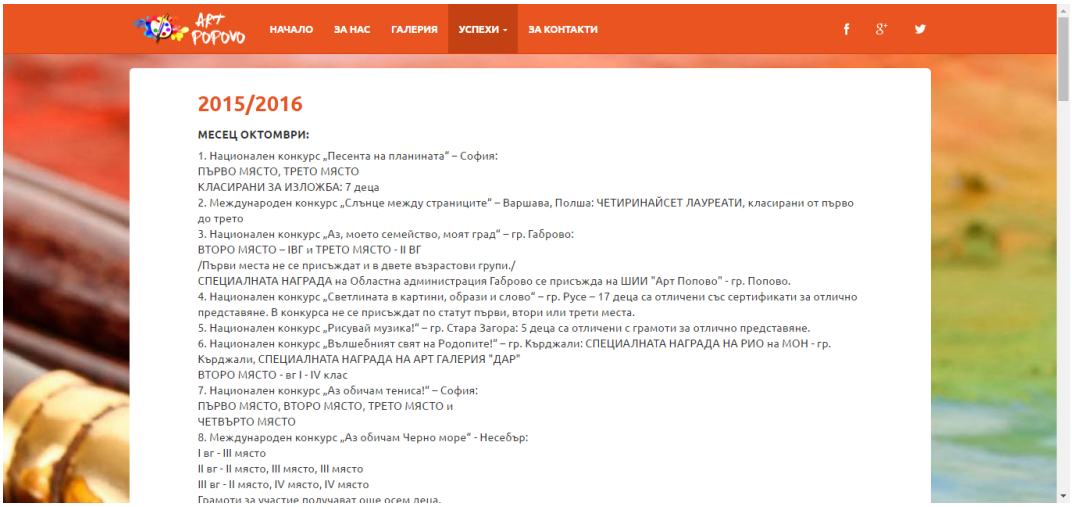


Fig. 13: Achievements preview page

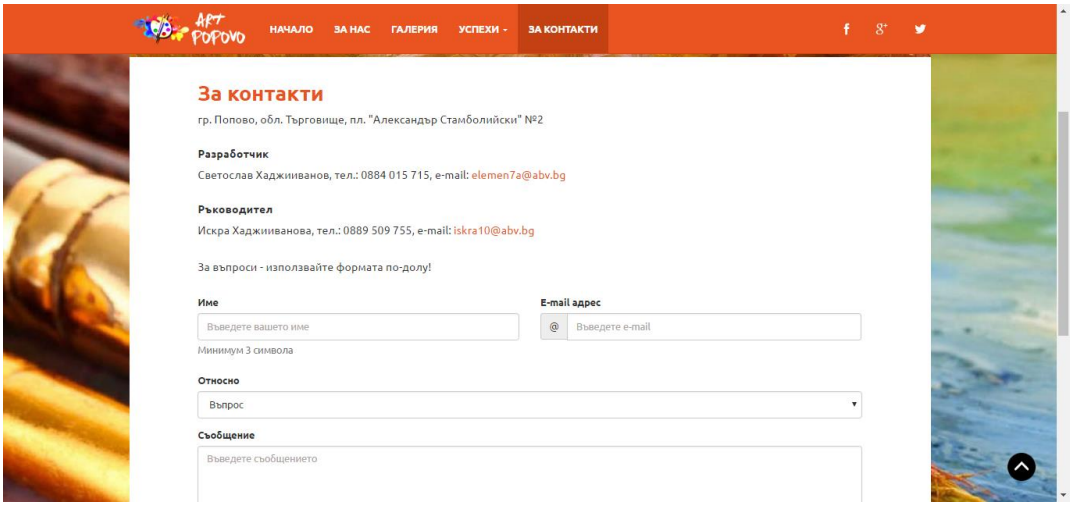


Fig. 14: Contacts page

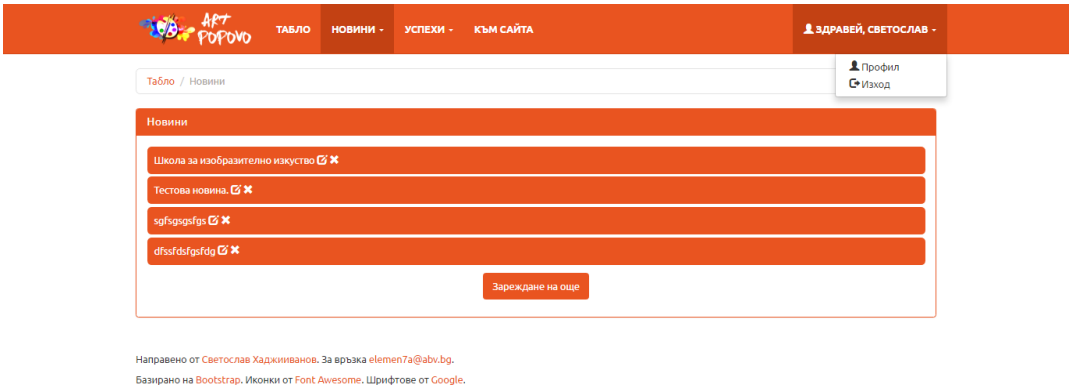


Fig. 15: Administration panel

3 Conclusion

The website is not finished yet. Therefore, there are many modules left to be implemented or optimized.

The author plans to expand the capabilities in the following areas:

- language system which will make the website readable for people from other countries;
- ability to upload a picture via drag & drop technology;
- website visitor statistics in the administration panel;

Acknowledgement: This work was supervised by Prof., PhD Katalina Grigorova, from University of Ruse, Bulgaria.

References

[1] Bootstrap, http://getbootstrap.com/
[2] Livestamp, https://mattbradley.github.io/livestampjs/
[3] Justified Gallery, http://miromannino.github.io/Justified-Gallery/
[4] Swipebox, http://brutaldesign.github.io/swipebox/
[5] Loading speed checker, https://tools.pingdom.com/
[6] SEO checker, http://neilpatel.com/seo-analyzer/
[7] Mozilla Developer Network, https://developer.mozilla.org/en-US/
[8] Acunetix – Web Vulnerability Scanner, http://acunetix.com/
[9] Disqus, http://disqus.com

SVETOSLAV HADZIIVANOV
University of Ruse
Department of Informatics and
Information Technologies
8, Studentska str., Ruse 7017
BULGARIA
warrolen@gmail.com

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

Student Share - A MEAN stack based application

Dragoș Hodină, Ionuț Hodină

Abstract

Web applications have developed substantially in last years due to easy scalability, easy access and better business model. Comparing with desktop applications, web applications are less expensive to develop. On a hand, users across the world can access a webpage fast and easily, without installing anything, having possibility to continue their work on a wide range of devices from smartphones to large enterprise screens. On the other hand, companies or web applications owners can easily push updates, target and manage various client needs, have access to data analytics which translates to better products, growing business and happier clients.

The evolving trend of web applications helped web technologies to gain popularity and visibility across developers. As web applications continue to grow in complexity, web programming languages are also growing in terms of features. Here jumps in JavaScript, the "web programming language", which is most used programming language accordingly to largest online developers community (StackOverflow) 2016 survey and is language with most active repositories in GitHub, largest code host service in the world.

1 Introduction

Many developers ask themselves how JavaScript evolved so much from a programming language used to animate content of web pages to existence of hundreds of dozens that can offer full technology stack for most business needs. To better understand this fact, there are a couple of reasons that must be mentioned.

Almost any application needs an user interface and for web applications there are HTML and CSS technologies that can handle this need. These technologies are widely known and used, but their purpose is to create static design. User needs interaction with application: validation of input content, handling of content changes, making pages more dynamic. As web applications grew up as complexity various client-side frameworks appeared to ease developing process. Angular JS is one of this solutions. It incorporates bidirectional user interface binding, organizing through reusable components and facilitation of Model-View-Controller pattern.

Having user interface also needs strong backend to communicate with. One solution is NodeJs, a packaged compilation of Google's V8 JavaScript engine. It uses non-blocking, event-driven I/O to remain lightweight and efficient in the face of data-intensive real-time applications that run across distributed devices. Communicating with server is accomplished through rest API calls. Using

RESTful architecture client and server are very loosely coupled which means client can be easily be represented by various devices and technologies.

So we have solutions for client and server side. Most applications needs a way to store and load data. MongoDB is an open-source database engine that stores data in JSON-like documents that can vary in structure. Of course that there are other database engines, but working with JSON format, which are native JavaScript objects, facilitates developing flow having unique data format from client to server, database and reverse flow.

This is how appeared MEAN concept, acronym that stands for MongoDB, ExpressJs, AngularJs and NodeJs, a full stack JavaScript solution for developing web applications. It's worth also to mention ExpressJs as a NodeJs framework designed for building web applications and APIs. All this technologies common thread is JavaScript. It's cost effective because of open source, easy to transfer and manipulate data. A lot of developers claim that MEAN or parts of it will lead the future of web applications[1].

The application we developed is a platform meant to standardize and organize content of teachers and students and ease the way they communicate as part of an university. Some of application features are: authentication, password reset via email, friendly URL routing, initial provisioning of data via shell scripts, support for various university schemes (multiple faculties, multiple specializations, bachelor, master and so on), files management using integrated Google Drive API, management of posts and user profile, real time data updates on dedicated smartphone application.

Solutions available on the market are more oriented to sharing of resources needed to learn. We want an application that is focused on easy sharing news and posts from teachers to students and reverse. Other solutions (mostly web applications) offer features related to user groups, but they don't provide a hierarchy of groups as specific to university.

2 Technologies

2.1 HTML, CSS and Bootstrap

HTML stands for Hyper Text Markup Language and describes the structure of web pages using HTML elements. Those elements are like building blocks and are represented by tags. Browsers are using tags for rendering content defined inside those.

CSS is used to describe how HTML elements are to be displayed on screen and stands from Cascading Style Sheets. Using CSS, web pages may look different having exactly same HTML elements, but using different arrangements and colors. CSS can be stored in external files and can be applied on multiple different pages.

Bootstrap is a free and open source framework aiming to help front-end developers to design their web sites. It contains many predefined interface components, like forms, buttons or navigation and supports also JavaScript extensions. It's first goal is to add responsiveness to web pages across many different screen sizes starting with smart phones and ending with larger desktop computer screens.

2.2 JavaScript

JavaScript is a programming language used to create interactive web pages. Instructions are executed by the user's browser and interpreted at runtime using a Just In Time compiler. JavaScript files are only once downloaded on the user machine and can be modified, but none of modified files

will be saved after closing browser or reloading the page. No special developing environment is required. Development can be done even with notepad writing plain text.

Although many times JavaScript is confused with java, another programming language, they are two completely different languages. At first, JavaScript was only interpreted by the browsers and was used only within web pages creation. Nowadays, with JavaScript, programmers can create complex mobile games, integrated development environments, manipulate databases or implement server side logic.

HTML defines what a web page contains, like images, paragraphs or links and those elements are rendered by the browser interpreting some predefined tags. Visual aspect is dictated by CSS and JavaScript is like an animator. This powerful programming language can insert new HTML tags, delete others, hide or even change the way they look in the browser by modifying their respective CSS attributes.

It is popular among developers due to it's flexibility. It is an object oriented programming language, like C# or Java, and weakly typed as opposed to the two mentioned before. In JavaScript, any variable can hold at first an integer and later a string can be assigned to it without any compilation error. Some or all of the parameters may be missing also without any error. This freedom given to the programmer is very powerful and can be quite productive, because complex syntax of strongly typed languages is avoided, but it is also error prone leading to hard to debug applications.

All features of modern object oriented programming languages like polymorphism and inheritance are also available in JavaScript using special keyword "function". This keyword is used for both declaring functions and creating classes writing nested functions and setting inner properties using "this" keyword.

ECMAScript is a scripting language specification standardized by the European computer manufacturers association. Because JavaScript is interpreted by many different browser rendering engines, the above association appeared in order to standardize the way JavaScript is interpreted by different browsers. Even today, not all the features of JavaScript are interpreted the same way, thus leading to frustration among web developers.

2.3 Angular 2

Angular 2 is based entirely on JavaScript. It is free and open source designed for web applications and maintained by Google and many community contributors. Angular is used to create single page applications. Unlike usual web sites, a single page applications is changing it's content dynamically through JavaScript, so no navigation to other web pages is required. Components are at the core of angular. A page consists of one or many angular components. Components can be added, deleted or modified and they live in a angular view.

Angular is platform independent and requires no installation because everything angular produces is pure CSS and HTML content manipulated by JavaScript instructions. Single page applications are fast and intuitive due to decreased network traffic involved. It is available for the best known server side frameworks, like .NET, PHP and NodeJs. Components based system provides a lot of benefits and one of them is code splitting. Every components has it's own behavior and only code for used components is loaded, further components being loaded when user or other components request them, unlike usual web pages where all scripts were loaded when requiring a web page.

A component is nothing more than a typescript class. Angular offers a series of data annotations which are letting the developer to add extra metadata. This metadata specifies, for example, whether

the class should be used as component or directive. As stated earlier, components are at the core of angular and are the most used during developments. Components consists of a class decorated with some data annotations and a template. Every components must define a “selector” which acts just like HTML tags, but in the end, it will be rendered using standard HTML tags. A template is just a plain HTML file with or without respective CSS file. Components allow event binding, property binding or both. Event binding is for taking action when user triggers an event, like clicking on a button or typing in a text box input. Event handling is done in classes mentioned earlier. Property binding refers to modifying HTML elements properties using instructions defined in typescript classes. Angular know which elements to edit using a routing module which pumps out the specific element.

Angular 2 has a powerful dependency injection system. The developer must only add the service provider name in the “providers” array at global or local level. Angular will automatically inject an instance of this service in the constructor parameter and developer has nothing more to do.

It is also possible to customize the way a component is looking using directives. A directive is a typescript class which is decorated with “directive” data annotation. Directives are used inside HTML tags just like attributes and provide additional information about an element, like “href” from an “a” HTML tag.

2.4 NodeJs

NodeJs is a JavaScript runtime built on Chrome's V8 JavaScript engine. NodeJs uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. NodeJs package ecosystem, npm, is the largest ecosystem of open source libraries in the world.[2]

It is free, open source and cross platform running on windows, mac, linux, webOS and others. It is used by some of the most important companies over the world, such as IBM, Microsoft, PayPal, GoDaddy or Netflix.

NodeJs is used for server side applications and it’s flexibility is provided by event driven mechanism. This kind of application has a main loop which listens for events and triggers a callback function when one of those events is detected[3]. In normal applications, commands are executed one after another and every command starts executing after previous were done. In NodeJs, every command executes asynchronously. Event driven is much more simpler for developers, because threading and concurrency at the same time are hard to implement and can easily lead to errors and sometimes to poor performance. Non blocking input/output operations and parallel command executions are on top of NodeJs advantages.

Unlike .NET which requires IIS or PHP requires Apache HTTP, NodeJs has a built-in library that allow application to act like a server. Once installed, developers can start writing code and testing. One key advantage among it’s competitors is represented by community which is permanently improving NodeJs based on their needs.

2.5 MongoDB

MonogoDB is a NoSQL database. It is free, open source and cross platform available on windows Vista or later, linux, mac and others. MongoDB is a document oriented database. It stores JSON-like documents with schemas, unlike tables used by sql databases, like MySQL or SQL server. This type of data saving provides flexibility over time, because every document has it’s own structure that can

be modified without affecting others entries from database like in SQL databases where adding a field for specific entry was not possible without any error.

Schemas are used for mapping objects from application to documents from database. Schemas are like table description from SQL databases. MongoDB allows referencing between different documents, data aggregation, indexing and ad-hoc queries.

This database is a distributed one providing high availability, scaling horizontally using sharding and load balancing based on geographic locations.

Replication is also easy in MongoDB using replica sets. A replica sets consists of two or more copies of the data. When primary copy is unavailable or corrupted, secondary replicas comes in actions and replaces first copy in order to provide high access to data. If primary copy fails, it is replaced by a secondary copy and a third copy is made in order to maintain at least two copies of the same data.

2.6 Google drive REST API and OAuth 2.0

Google drive is a file storage and synchronization service powered by Google company. It provides 15 GB of free space offering multiple paid plans up to 30 TB. All files are saved forever unlike other free cloud storage server. Google drive can be accessed by normal users using official web pages, using some client application which is synchronizing with Google drive account or using REST API. REST comes from Representational State Transfer and is the widely spread all over the world, mostly because it is platform independent and many applications can “consume” the same API endpoint regardless of their operating system.

In order to access Google drive account, an applications must provide an OAuth 2.0 authorization token generated from Google developer console. At first, a json file is generated from Google developer console and consists of an *client_secret, client_id* and *redirect_uri* keys. This file is one time generated and used to obtain an OAuth 2.0 token which consists of an *access_token* and an *refresh_token* along with an *expiry_date*[4]. Every request must provide an access_token at every submitting to Google drive using REST API. If the access_token is expired, a new access_token can be retrieved from API using refresh_token.

In order to upload a file, a post request is submitted on API endpoint using OAuth 2.0 client authorization using *multipart/form-data* bodies. As response, Google drive API returns a collection of unique id’s which can be used further for downloading using an authorized get request.

2.7 Firebase cloud messaging

Firebase is used to notify a client application when something changes: receiving an email or some data is available on the server. Firebase is ideal for sending small instant messages. It is cross-platform and currently supports iOS, Android, Web, C++ or Unity clients. Firebase can be used by different server side clients such as NodeJs, Java or Python. Messages can contain text up to 4 KB which is enough for every context, because large data is kept on the server side. Messages can target a single device, groups of devices or devices subscribed to topics. Instant messages are sent using web sockets which are based on opening and maintaining a two way battery-efficient TCP connection between target device and cloud.

In order to receive a message in your application, you must first setup server side application to authenticate and initialize with firebase cloud messaging. After authenticating to service, server

application can start sending messages to the cloud, however, at this time nobody is going to receive any message just because none is subscribed to any topic or user group. Client application must handle user authentication, subscribe user to topics, handle incoming messages and display them and also save current application state.

3 Student Share

3.1 Data provisioning

These scripts are files written in JavaScript and run using NodeJs execution shell. Mainly they are storing and configuring MongoDB documents(similar to tables) before using application.

3.1.1 Groups

Every university has a clear organization scheme, which can be broke down to several groups of interests regarding news sharing. In its simplest structure, an university has multiple faculties and each faculty has several departments. Each department has multiple users assigned to, grouped by year of study. Additionally, students can be split up to bachelor and master study program.

Data that has to be provided consists of a file containing single JSON object that represents groups of interest from university as a tree object. To give an example, *University* group can be the root group of this tree which means that every user has access on information(news, posts) in this group because is relevant across entire university. Above group can have subgroups like *Science* and *Engineering*. An user assigned to *Science* group can access information only from this group or from *University*. This tree of groups can have any organization and theoretically any number for level depth.

3.1.2 Users

Users data must be also provided through a provisioning script. For every user it must be specified name, email, image profile, and groups that is assigned to (usually just one because a regular student learn at only one faculty/department at a time, but this is not a restriction due to abstraction of groups). It must be specified only leaf group(s) ids from groups tree explained above. If a user is assigned to group *Second year Informatics* then has access to all groups from parents path through three root: *Department of Informatics, Faculty of Science, University ULBS*.

3.2 Authentication and password reset

Every user must initially set a new password by receiving an email with a reset token. Same flow must be followed every time user forgets his password. Reset token is a random generated string which uniquely identifies a reset password request. A token can be used only once.

Authentication is made using *PassportJs* framework. This generates a JsonWebToken when user logs in and expects this token for every request of which API route is protected from unauthorized access.

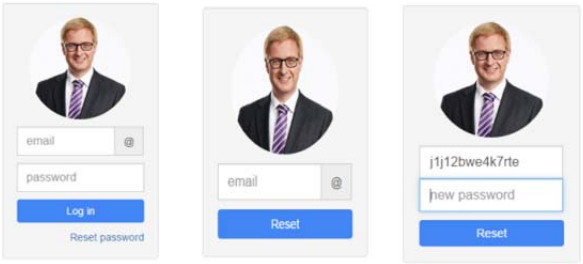


Figure 1: Log in & reset password

3.3 Routing and main layout

Layout is divided into 4 main parts. Head of application is first one. It contains logo, information about current logged user and several messages for user input validation or other user relevant information depending on active view. Left column contains current user groups and a button that opens up a popup for adding a new post. On right column are displayed contact information of users. In the middle are loaded several views depending on user actions flow.

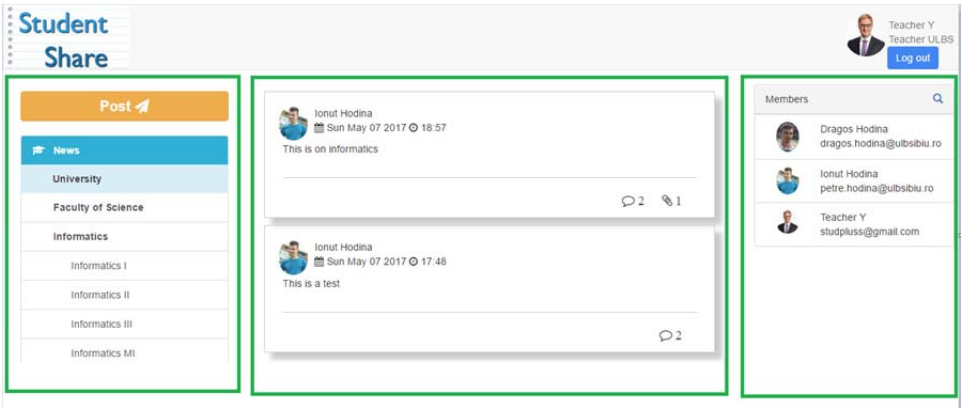


Figure 2: Main view

All views are loaded with a specific friendly url that uniquely identifies each one. If a user is not logged in and tries to access a protected route then is redirected on login view. Routing is managed using Angular Router module as described below.

```
const appRoutes: Routes = [
    { path: '', component: HomeViewComponent, canActivate: [AuthGuard] },
    { path: 'groups/:groupId', component: HomeViewComponent, canActivate: [AuthGuard] },
    { path: 'login', component: LoginViewComponent },
    { path: 'reset-password', component: ResetPasswordViewComponent },
    { path: 'reset-password/:resetToken', component: ResetPasswordViewComponent },
    { path: 'profile', component: ProfileViewComponent, canActivate: [AuthGuard] },
    { path: 'post', component: PostViewComponent, canActivate: [AuthGuard] },
    ];
```

3.4 Posts and groups view

On the left of page are retrieved groups that are related to current logged user. By selecting a group, user can see posts added to that group ordered descending by publish date. To see posts across

all groups then *News* group must be selected so posts are loaded ordered descending by last update date.

A post view contains information about the one who posted (image, name) and about post itself like content, date time of published and other useful information.

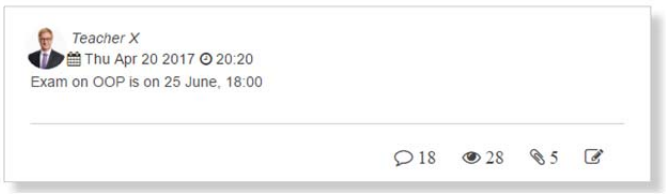


Figure 3: Post view

3.5 Add post view

User can add new post from an opening popup having default group selected. Other groups can be selected also using autocomplete dropdown control. Selected groups are displayed as labels. A selected group can be removed using button from right top corner of label. Multiple files can be attached to new post. For an optimized and scalable solution these files are uploaded to Google Drive using application account credentials. All operations are made using asynchronous HTTP requests.

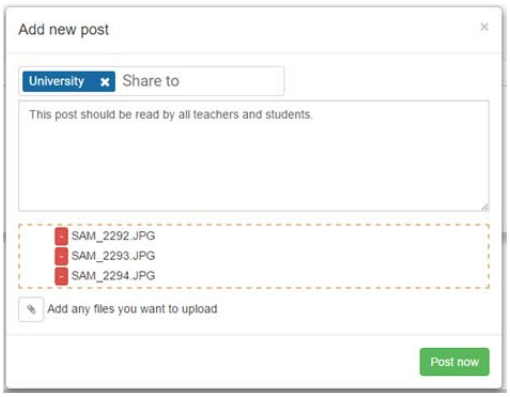


Figure 4: Add new post view

3.6 Profile view

This view display information about current logged user or any other user selected like: name, faculty, department and email. On the bottom of view are loaded posts created by user.

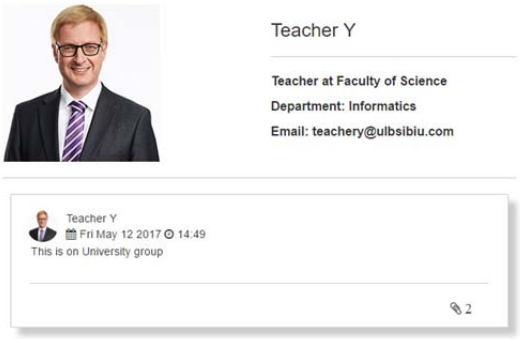


Figure 5: Profile view

3.7 Add homework

Users having role of teacher can add new homework for every leaf group. Leaf groups are those groups that don't contain any other groups, like Informatics III or Informatics MII. One homework contains title, content, one leaf group assigned, list of attachments and a deadline.

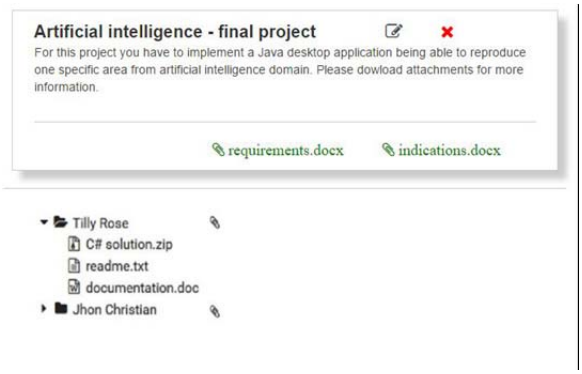


Figure 6: Homework view

Every teacher has a list with all created homework in the left side view. Once he adds a homework, it will be displayed in the left side view. Selecting one homework from this list allows teacher to view all details about a specific homework, being able to modify it and also permanently delete it, resulting in list updates for all students who have been assigned to this homework. In the same view, teacher can see all students with their respective homework files and can download them on his own computer.

A student can see his assigned homework in the left side view, right next to posts view. All assigned homework are displayed in a list like view, which is updated every time new homework are assigned to student leaf group. Students can respond to a homework by uploading attachments. They can upload again new files that will override old files in case they discover some mistakes in their initially uploaded files. Uploading is possible only if current day is not passing deadline.

3.8 Firebase messaging using Android client application

As Android is the most popular operating system for mobile devices and tablets, Student Share comes with an application which is able to receive instant messages sent from server using Firebase cloud messaging.

User must sign-in for being able to receive any message. Application will automatically subscribe user to one or many topics, which are denoted from user groups. At this moment, application is able to receive push notifications.

Figure 7: Push notification when phone is locked

Push notifications are handled within two background services providing high availability for receiving messages in foreground, background and even when application is closed. Tapping on a notification opens main view where user can see more details like sending date-time, title, group and notification content about each notification.

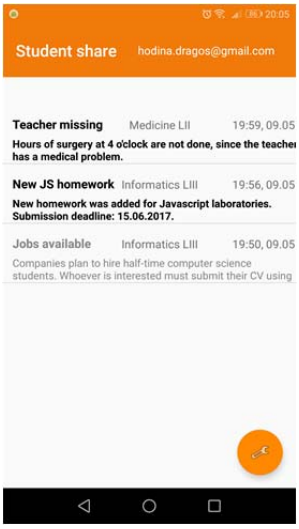


Figure 8: Notifications view for logged user

Every new notification is visually emphasized and appears in the top area of notifications list for higher visibility when user is opening main view. Tapping on a notification opens another view where user can see notification body on entire screen. Settings button brings up settings view where users can log in or out.

4 Conclusions and further development

This application uses a fresh new stack of technologies having in common programming in JavaScript. While these technologies are young comparing with well-known other solutions, they are growing up fast on market mainly because they are backed by big companies in development and services providers fields: Angular - developed by Google, support for MongoDB was added on cloud providers, NodeJs is already used to fulfill big companies needs.

This application is organized around users groups, having this feature as main advantage. Application offers clean and intuitive user interface for sharing news and posts across university.

Acknowledgement: This work was supervised by Professor dr. Dana Simian, from "University Lucian Blaga from Sibiu".

References:

[1] Advantages of MEAN, https://www.quora.com/What-are-the-advantages-of-developing-with-the-MEAN-stack-mongoDB-Express-js-Angular-js-Node-js
[2] NodeJs, https://nodejs.org/en/
[3] Event driven, https://dzone.com/articles/what-are-benefits-nodejs
[4] OAuth 2 https://developers.google.com/identity/protocols/OAuth2

HODINĂ DRAGOȘ
Faculty of Science "Lucian Blaga"
Department of Informatics
St. Dr. I. Ratiu, No. 5-7, Sibiu, 550012
ROMÂNIA
E-mail: hodina.dragos@gmail.com

HODINĂ IONUȚ
Faculty of Science "Lucian Blaga"
Department of Informatics
St. Dr. I. Rațiu, No. 5-7, Sibiu, 550012
ROMÂNIA
E-mail: ionut.hodina@gmail.com

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

S.O.S. Cardio

Holerga Flavius Adrian

Abstract

S.O.S. Cardio is a bracelet designed to detect the pulse of a person with a cardiac condition, who doesn't benefit of permanent medical assistance, but is monitorized by at least another person, be it from the family or from a social program. This device is connected via bluetooth to the user's mobile phone and communicates with it through an app made using „Eclipse" which was created with the aid of the programming language „Java". The application receives the data captured by the proximity sensor from within the bracelet and if the pulse is either too high or too low, a warning message is sent to a phone number which is initially set by the user when the application is first installed on the phone.

1 Introduction

The paper presents a device and an application, made with the purpose to help the people with cardiac type health issues. S.O.S. Cardio was designed to decrease the intervention time in case of a cardiac attack or a similar situation. The project consists of two main components: the electronic component and the software one. The idea of the project is that the mobile phone of the pacient can help decrease the intervention time in case of a cardiac attack by always checking the pulse of the pacient using a proximity sensor placed on the ear lobe of the user. If a very high or very low pulse is detected by the electronic component, the application installed on the phone of the pacient kicks in and sends a message using the "Short Message Service" (SMS) to a phone number specified by the user when the application is initially started.

In the following sections, we will cover the applicability of the device, the advantages and disadvantages of using it, similar devices and the differences between them. After that, we will take a look at the main components of the project and at the existing features of both the application and the bracelet, as then to present upcoming features and future goals.

2 Applicability

In the span of the last 20 years, the life expectancy has grown significantly, especially in the more developed countries due to the increase in technology and in medicine. With all of these being said, the cardiovascular diseases remain some of the most common causes of death on a global scale. In these cases, it is especially important to monitor the cardiovascular activity of a person with such problems, as the intervention time, in case of an attack, has to be as low as possible for the pacient to have a better chance to survive.

For this purpose, we designed S.O.S. Cardio as an alternative mean to monitor the activity of a pacient's heart who doesn't need or doesn't benefit of hospitalization. The hospitals in our country are often overpopulated, therefore many pacients with cardiovascular diseases or who are not in need of immediate interventions are, more often than we would like to admit, left behind. To their help, we built the S.O.S.

Cardio device, which is cheap and accesible to anyone, made primarily out of recyclable materials and built using the Arduino technology. S.O.S. Cardio provides supervision continuously in an easy to use way, giving the pacient and his supervisor an alternative way of detecting possible heart problems and making sure that it is acted fast enough in case of a health situation to stop the possible consequences which can follow a heart attack. Also, in contrast to other seemingly similar devices, the S.O.S. Cardio device communicates with the user's phone using the bluetooth service, which makes the whole device available everywhere and does not require any sort of internet connection in order to work. And, besides that, also in contrast to other seemingly similar technologies, the device was designed to be mobile and easy to use on a day to day basis without creating any discomfort to the pacient or to the superviser.

S.O.S. Cardio is especially designed for people within medical-social care systems, people in the care of their family or in the care of a personal assistant. (Fig. 1)

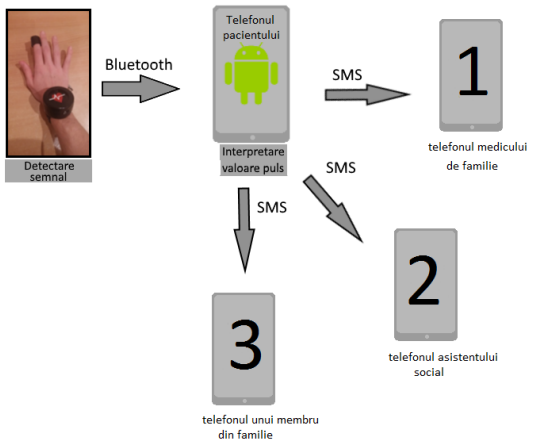


Fig.1: Communication system diagram

Seemingly similar devices are used:

- in hospitals or private medical cabinets:

- the Holter monitor: is a device used to interpret the readings of a pulse sensor with the purpose to identify possible cardiovascular diseases. This device, in contrast to S.O.S. Cardio, is not mobile, requires an activity journal and a schedule. Even more, the Holter monitor requires the pacient to use it for at least 24 hours before it can precisely read the data provided by the ECG sensors.

- the Oximeter: is a device used to determine the level of oxygen in the pacient's blood. Like S.O.S. Cardio, the oximeter uses a proximity sensor to detect the number of heartbeats per minute, similarly to our device. In contrast to S.O.S. Cardio, the oximeter, even though is mobile, it is not recommended for every day usage, can not provide live data or communicate with an external device and is placed on the finger, which can easily lead to some sort of discomfort to the pacient.

- in the sport industry

- the fitness bracelets: - are devices used especially for physical activities, are not recommended to be used with a medical purpose and are not designed to give a precise value of someone's BPM (heart beats per minute), therefore are unreliable in the case of a cardiac attack.

3 Functionality

S.O.S. Cardio is made up of 2 main components:

- The electronic component
- The android application

3.1 The android application

The application was made using the developement enviroment Eclipse and initially was designed using MIT App Inventor 2. It has the main purpose of receiving the data sent by the device through the bluetooth module and sending a short warning message to the person in charge of the pacient if values too high, or too low were detected by the electronic component.

From an algorithmic point of view, the SMS is sent using a basic function if the value received from the electronic component is higher or lower than the values initially inserted by the user in the fields mentioned previously. If the value detected is not between the "safe" parameters, a warning message is written by the application. The application reads, if available, the location of the user through the GPS service of the phone, then concatenates that information with the pulse value detected by the sensor.

```
bluetoothIn = new Handler() {
        public void handleMessage(android.os.Message msg) {
             readMessage = (String) msg.obj;
             new CountDownTimer(1000, 1000) {
                public void onFinish() {
                   sensorView0.setText(readMessage);
                   i=Integer.parseInt(readMessage.replaceAll(„[\\D]”, „”));
                   if((BPM> maximum || BPM<minimum) && ok==0)
 {
                        LocationManager lm =
(LocationManager)getSystemService(Context.LOCATION_SERVICE);
                        Location location =
lm.getLastKnownLocation(LocationManager.GPS_PROVIDER);
                        double longitude = location.getLongitude();
                                          //gets the longitude value
double latitude = location.getLatitude();
                                          // gets the latitude value
                        body = body+ „ x=” + longitude + „ , y=” + latitude
+ „ ; Valoarea pulsului detectata: „ + i; // making the message which will
                                         // be sent in case of an emergency
                        SmsManager smsManager = SmsManager.getDefault();
                        smsManager.sendTextMessage(numar, null, body, null,
null); // sending the message
                        Toast.makeText(getBaseContext(), „Mesaj de
avertizare trimis la numarul: \n” + numar, Toast.LENGTH_SHORT).show();
             /*showing a notification on the pacient's phone to notify him
that a warning message was sent in case of a false alarm*/
                        ok = false; /* debounce variable */
                   }
             }

             public void onTick(long millisUntilFinished) {
             }
              }.start();
        }
    };
```

Fig 2: The method used to send a SMS

The interface (fig. 3) is simple, and features 3 main interactive components. In the first text box, situated at the bottom of the screen, the user is required to write one or more telephone numbers, which can be contacted in case of an emergency. The second and the third fields represent the parameters between which the pulse values are considered to be normal and are situated on the top half of the screen, right under the title of the application. After inputting the preferred values, in order for the settings to be saved, the user must press the save button situated in the lower half of the screen.

Fig. 3: The interface of the application

3.2 The electronic component

The device was made using Arduino technology and has in its composition an Arduino Nano microcontroller, a HC-05 bluetooth module, a proximity sensor and a couple of resistances. The software used in order to program the board is a software made by the company Arduino [1]. The programming language used to do so was C++.

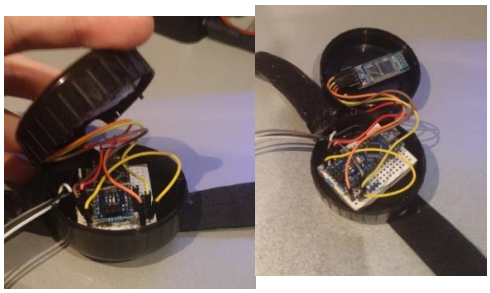


Fig.4: The interior of the bracelet

The images above show the circuit from the inside of the case (Fig. 4), and the images below (Fig.5) show the circuit in an unfolded form, on a breadboard. The circuit is made of 2 resistances of 2.2kΩ and 4.7kΩ, a bluetooth module HC-05 ,an Arduino Nano microcontroller and a proximity sensor.

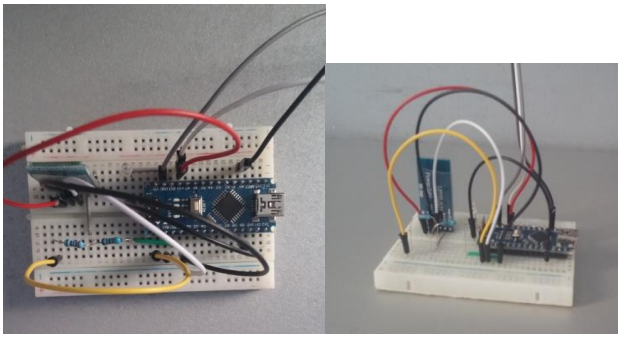


Fig.5: The circuit unfolded on a breadboard

The bluetooth module HC-05 - has the purpose of sending and receiving data, essentially making up a connection between the application and the device.

The Arduino Nano microcontroller - has the purpose of memorating and executing the code, therefore controlling the other components in the bracelet. The microcontroller receives the data sent by the proximity sensor, reads it, and processes it, as then to send it to the bluetooth module.

The pulse sensor - has the purpose of detecting the values of the pacient's pulse using a proximity sensor. This sensor works the best when placed on a finger tip (Fig. 6), or on a ear lobe.

The bracelet is powered using 2 lithium batteries or a USB Cable and also works just fine with a power bank.

From an algorithmic point of view, the microcontroller uses the "A0" pin to read the signal given by the proximity and calculates the number of heart beats per minute, using the time between the beats.

In order to send the data to the phone, we simply sent the data to the serial from which the bluetooth module handles it.

```
void serialOutputWhenBeatHappens(){
  switch(outputType){
    case PROCESSING_VISUALIZER:
      sendDataToSerial('B',BPM);
      sendDataToSerial('Q',IBI);
      break;

    default:
      break;
  }
}
```

4 Future improvements

In the future, we plan to change the whole design completely into a more compact one and at the same time, removing the bracelet in it's entirety by transforming the whole device into something more like an earphone.

To do that, we started working on a circuit board to contain all the circuitry in a relative small space.

To sketch out the circuit, we used an online circuit designing application called EasyEDA [2] and the software Fritzing for mapping out the components. (Fig. 7)

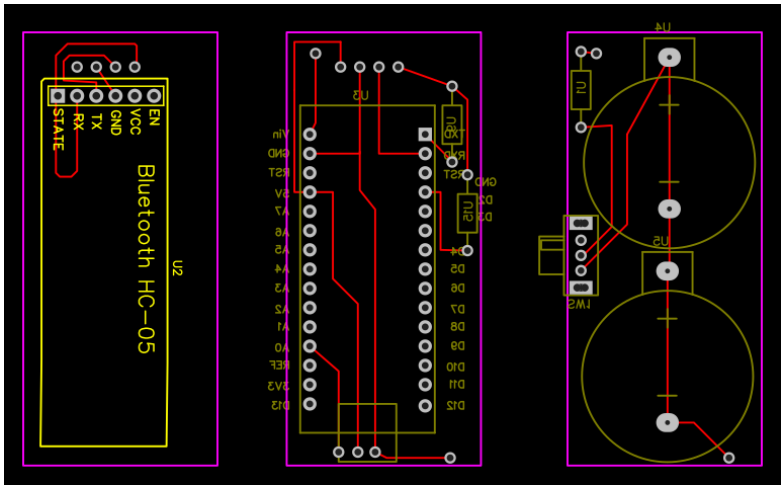


Fig. 7: The main boards sketched

After that, we used the designs to make 2 circuit boards. To corrode the copper on the boards, we used Ferric Chloride, which left us with 2 boards which then we completed with the right arduino components. (Fig. 8)

Fig. 8: The 2 main boards

In the future, we plan to finish the product, to make a custom case and improve the design of the Android application and possibly add more features to it.

5 Conclusions

In conclusion, because the mobile phone is an indispensable device for almost anyone, S.O.S. Cardio (Fig. 9) is a viable solution to decrease the intervention time in case of a cardiac attack and it is, in our opinion, a fast and efficient way to send a warning a message and to inform a person in charge of supervising a pacient in case of an emergency.

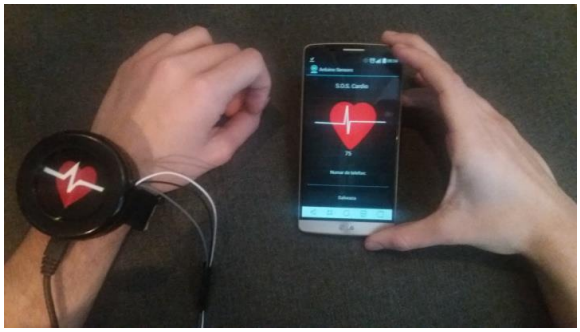


Fig. 9: The final form of the bracelet and the application

Acknowledgement: This work was supervised by Professor Gabriela Florea, from "Axente Sever" Theoretical High School of Mediaş.

References:

[1] ARDUINO's official website, https://www.arduino.cc/en/Main/Software, march 2017
[2] The electronic circuit design tool used in sketching out the main board of the device featured in the paper, https://easyeda.com/, may 2017

FLAVIUS ADRIAN HOLERGA
"Axente Sever" Theoretical High School of Mediaş
Mediaş, ROMANIA
E-mail: flavius.holerga@gmail.com

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

Game Boy emulator

Borislav Kosharov

Abstract

Game Boy emulator is a project which purpose is to mimic real hardware with software implementation. It was created to give insight on low-level programming, processor architecture and memory management in a simplified manner. Each hardware module's functions will be described and followed by explanation of it's implementation. Some of the systems are yet to be implemented and are skipped, but are planned to be done in the future.

1 Introduction

Game Boy is a 8-bit handheld game console manufactured by Nintendo. It was first released on April 21, 1989 and has been an object of interest of many programmers, because at the time it solved complex problems with very limited hardware resources. The main reason this emulator was created is to learn more about computer architecture and how all hardware sub-systems interact with each other. Because this emulator was made for learning purposes, it is not meant for end user use and there are far better and more complete emulators that do the job. The D [1] programming language and a general purpose multimedia library called SFML [2] were used for implementing the emulator. The project is open-source and publicly hosted on Github at https://github.com/nikibobi/gameboy-d. The code is licensed under the MIT license. Each hardware module is implemented as a separate D language module. In app.d is where classes are instantiated, that is where the window is created and the main game loop.

2 Emulator Implementation

2.1 Processor and registers

Game Boy's CPU is 8-bit and is based on Intel's 8080 and Zilog's Z80 processors. It has 7 general purpose registers - A, B, C, D, E, H, L; flags register F with 4 flags - Z (Zero), N (Negative), H (Half carry), C (Carry); SP (Stack Pointer) and PC (Program Counter). The instruction set [3] has 256 (not all are used) different instructions and there is also a secondary instruction set that has bit manipulation instructions. When an instruction is fetched the first byte defines its type and address in the instruction set. If the instruction accepts an argument then up to two more bytes are red. After that the PC register is incremented by that many bytes and now points at the next instruction to be red.

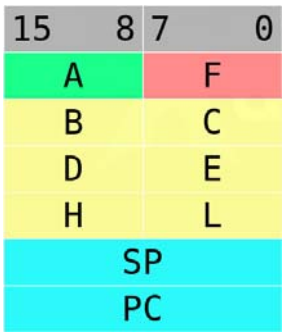


Figure 1: CPU Registers

2.2 Memory space and cartridges

It has a memory space of 64KB which includes ROM, RAM, Video memory and memory-mapped IO registers. Games for the Game Boy are distributed trough cartridges. When a cartridge is inserted into the slot it's data is memory-mapped as ROM (Read-only memory) to the first 32KB of the memory space (addresses from 0x0000 to 0x8000). But not all games are exactly 32KB and are most often larger. Nintendo solved this problem by splitting the data into 16KB chunks called memory banks. For those games there exists an external chip located on the game cartridge called MBC (Memory Bank Controller) [4] switching the memory banks that are memory-mapped. The first bank - 0 is always mapped in address range 0x0000-0x4000 and never changes. The current active bank is mapped at address range 0x4000-0x8000. Game Boy supports up to 125 (from 0x1 to 0x7F) different memory banks. The control registers of the MBC are also mapped to 0x4000-0x8000, but they are write-only and don't actually write over the ROM, but instead they switch the active bank. Some cartridges have external RAM part that is used to save the progress of the game. It is mapped to address range from 0xA000 to 0xC000.

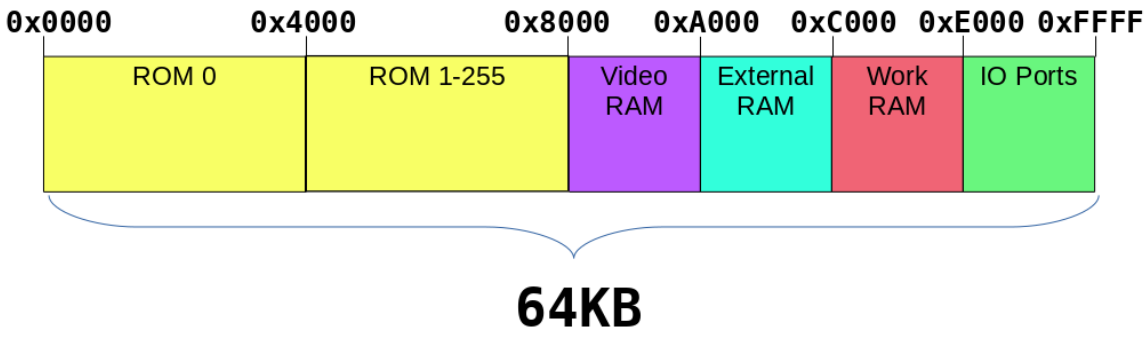


Figure 2: Memory map

2.3 Display and video

The Game Boy has and LCD video display that is 160×144 pixels. It has 2-bit (4 total) colors that are shades of green. Memory addresses from 0x8000 to 0xA000 are reserved for Video RAM. It contains sprite and tile information. Part of it is reserved for background tiles and foreground tiles. Sprites are the moving objects like the player. They can be 8×8 or 8×16 pixels. There is enough space in the VRAM for 40 sprites and that part of the memory is called OAM (Object Attribute Memory) [5]. Each sprite is exactly 4 bytes long. The first two bytes contain the X and

Y coordinates of the sprite on the screen. The third byte contains the pattern number (the index of the tile that represents the sprite). And the last fourth byte contains 4 flags. One of them is priority that represents if the sprite should be drawn in the background (when 1) or foreground (when 0). Two flags contain information if the sprite's pattern should be flipped by x-axis, y-axis or both. And the last flag is the palette index (0 or 1). A palette is a table that maps indexes to actual colors. Game Boy supports two palettes at a time. It is fascinating how a single tile pattern can be reused to generate 96 (4 flips × 4! palettes) different unique sprites.

2.4 IO Ports and Input

The memory area from 0xFF00 to 0xFFFF are reserved for control registers of different hardware devices. This makes working with hardware devices as easy as writing to or reading from a memory address. Every subsystem uses the memory to do different things with the same instruction. There is no special instruction for working with hardware it is all just a LD instruction.
Game Boy has a total of 8 physical buttons. Four buttons for each direction, button A, button B, Start and Select. Working with hardware is done by writing/reading a specific memory location. A single byte is used at memory address 0xFF00. Writing to that location specifies which set of button is about to be red. Writing 1 to the 4th bit reads the 4 special buttons and writing 1 to the 5th bit reads the 4 direction buttons. Reading that address returns the state of the buttons in the set in the lower 4 bits with 0 meaning pressed and 1 released.

2.5 Memory module

The memory module is important since all other modules depend on its interface. The Memory class holds two hash tables one for reading and one for writing operations. Because every module owns only a part of the address space, a mount() function has been made. It accepts start and end address and a callback function to be called when memory is accessed at that range. The range itself serves as a key in the corresponding hash table. The value is a function that returns a byte (for reading) or a void function that has a single byte argument (when writing a byte to an address). The implementation of the callback functions can be as trivial as reading a value from an array. There are also indexer operators so that an instance of the memory class can be intuitively used as an array. Writing a value 127 to an address 0xFF00 is intuitively: mem[0xFF00] = 127;

2.6 CPU module

2.6.1 Instruction structure

This structure is holding the mnemonic, arguments count and action of a single assembler instruction. It has 3 different constructors accepting callbacks with 0, 1 and 2 arguments for instructions taking 0, 1 and 2 operands. In it there is an union with 3 different pointers to functions. The arguments count is used to differentiate which pointer to use. Unions contain a set of variables that all share the same memory location. So only one of its variables can be used at a time.

2.6.2 Processor class

The Processor class implements the two instructions sets as hash tables with the byte code as a key and Instruction structure as a value. CPU registers are contained in a hash table with a char as a key (the name of the register) and byte as it's values. A function that accepts a variable length string and combines the listed registers into one larger value. There are instructions that operate on 16-bit values and use the combinations BC, DE, HL. So that the first register is the high part and the second is low. The HL register is often used to access an arbitrary memory location at address the value of the register. The CPU flags are implemented in a similar fashion. The F

register can be read as a whole or each bit can be accessed through its char code.
A lot of the instructions are exactly the same and just use a different register. For example the ADD instruction has 8 different variants for each register – A, B, C, D, E, H, L, (HL). The last one is the value at memory address HL. The ADD instruction adds a value to A (often called accumulator) register. It takes byte-codes from 0x80 to 0x87 including. This reduces the number of unique instructions from 256 to around 32 or even less. For each instruction there is a single generic function that does the logic. Later lambda expressions are made that use that function and the only difference is the register used.
When the byte code 0xCB is read the second instruction set activates which again has 256 different byte-codes, but only 11 actual instructions. 64 of them are a single SET instruction that sets a bit with index 0-7 of one of the 8 registers to 1. The same is for the RES instruction that does the same but instead sets the bit to 0.

2.6.3 Interrupts

There are times when the CPU has to stop executing the current instruction in line and do something more important. Game Boy has 5 interrupt types – VBlank, LCDStat, Timer, Serial, Joypad. VBlank happens when the screen refreshes the graphics. LCDStat is when some control register of the display changes. Timer is when one of the internal timers reaches 0 and something has to happen. Serial is when serial port communication begins and Joypad is when a button is pressed or released. When an interrupt happens the current value of PC (Program Counter) is pushed to the top of the memory stack. Then a jump instruction is performed to a fixed location depending on the type of the interrupt. For example for VBlank it goes to 0x40 address of the ROM. After the interrupt has been handled the value of PC is popped from the stack and its value is restored and continues where it left of. In some cartridges interrupts can even happen during other interrupts. Memory location 0xFF0F is used to flag what type of interrupt is triggered, but interrupts can't be triggered if they are not enabled in location 0xFFFF (the last address in the memory). Also there are two instructions that enable/disable all interrupts EI (from Enable Interrupts) and DI (from Disable Interrupts).

2.7 ROM module

This module is responsible for reading the raw binary files of the cartridges and extracting metadata information about the loaded game. The cartridge header [6] is located in range 0x100 to 0x14F and contains the game title, version, manufacturer, MBC type, memory banks, external ram size and checksum. If the checksum of the header is incorrect the game doesn't load. The cartridge class is mounting the raw bytes read from the file to memory location 0x0000 to 0x8000. Currently MBC controllers aren't supported, but are planned to be added.

2.8 Video module

The video module handles Video RAM, sprites (OAM), tiles and a state machine that simulates the GPU states. A Sprite structure has been added that has the exact format as described in the introduction and has friendly properties for accessing the raw bytes.
The GPU has four states: HBlank, VBlank, OAM and VRAM. Game Boy renders the screen by running a scan line that goes from the top and goes down through each pixel row. When the line moves to the next row a HBlank is triggered. After the scan line goes to the bottom of the screen a VBlank state is triggered, the screen clears and the line goes back up. The sprite table can be modified only when the GPU is in OAM state. And the same goes for the VRAM state. There are a few other functions that work with palettes, tiles and sprites. This module also renders the game screen on a off-screen texture that later is rendered on the program window.

2.9 Input module

This module handles keyboard and gamepad input and translates the states to the format described in the introduction and mounts it to address 0xFF00 using the memory module interface.

3 Conclusion

A lot has been learned while doing this emulator and it gave me insight on low level programming, hardware, computer architecture and design. MBC controllers and the sound sub-system are still to be implemented. I hope my work helps other aspiring programmers to learn more about this interesting 8-bit computer. This project was inspired by several other open source emulators [7] [8].

Acknowledgement: This work was supervised by Prof., PhD Katalina Grigorova, from University of Ruse, Bulgaria.

References

[1] *D programming language*, http://dlang.org/
[2] *SFML library*, https://www.sfml-dev.org/
[3] *Gameboy CPU (LR35902) instruction set*, http://www.pastraiser.com/cpu/gameboy/gameboy_opcodes.html/
[4] *Memory Bank Controllers*, *http://gbdev.gg8.se/wiki/articles/Memory_Bank_Controllers*,
[5] *VRAM Sprite Attribute Table (OAM),* *http://gbdev.gg8.se/wiki/articles/Video_Display#VRAM_Sprite_Attribute_Table_.28OAM.29*
[6] *The Cartridge Header, http://gbdev.gg8.se/wiki/articles/The_Cartridge_Header*
[7] *Cinoop emulator in C, https://github.com/CTurt/Cinoop*
[8] *LuaGB emulator in Lua, https://github.com/zeta0134/LuaGB*

BORISLAV KOSHAROV
University of Ruse
Department of Informatics and Information Technologies
8, Studentska str., Ruse 7017
BULGARIA
bosakmaw@gmail.com

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

Automatization of the process of bibliography making for article-level metadata

Dorina Luca

Abstract

Nowadays, for greater indexing recognition, scientific journals tend to be present in the most prestigious international databases such as SCOPUS, Thomson Reuters, INSPEC, zbMATH, DOAJ, etc. The goal of this paper is to present the problem of article-level metadata preparation for international databases and the necessity of automatized approach to the process of bibliography making in this framework. A tool that was developed and used already by the Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova is described in the paper. This system helps us to solve the problem of bibliography automatizing when preparing article-level metadata – the process which is necessary if the journal tends to obtain a higher impact factor. The goals of this project are to reduce the preparation time for metadata and to obtain more accurate and complete information about an article.

Keywords: indexing, metadata, impact factor.

1 Introduction

At the moment, the scientific researches develop very fast because we have more and more scientists and so many fields of research. From the other hand time of publications preparation and deliver to readers essentially decreased due to information technologies development. The result is an exponential rise of the number of journals, conferences, books, articles and other publications bringing research results. These publications are referred to as primary publications or primary sources of information, pointing to the original character of the information presented. Convenient search of such a huge number of publications issued in different countries, in different languages and stored in different media would be impossible without special aids, these are secondary publications or secondary information sources, which process, analyze and summarize primary publications and help in their target search. As such a great amount of publications and their authors require proper evaluation, these secondary sources of information are also used for scientific validation, based on the high criteria they employ in the selection of primary publications to be systematically followed and processed.

Since each article is followed and processed (popularly termed indexing) by specific databases, for greater indexing recognition, scientific journals tend to be present in the most prestigious international databases. However, its true usefulness for scientists will be determined by the number of citations received by its articles. Each article is represented by bibliographic information that contains

data such as author name, title of the article, name of the publication where the article has appeared, volume, issue, year of publication, abstract, author affiliation, original language of the article, type of the article, etc. [1]

The databases are now updating their lists of journals and ask respective editorial boards to provide updated information about the journal, at the same time they are asking for metadata about journals and articles published by them. In different databases there are different requirements and different format of metadata presentation.

Our goal is to develop a tool that would automate the process of obtaining the metadata needed the publication to be indexed in international databases. This paper is organized as follows: A short introduction of this topic, in Section 2 there is specified the problem, the requirements needed to solve it and how should be written the bibliographic resources in order the process of metadata preparation to be automated. Section 3 describes the practical application, the algorithm and also some examples of how the automate process works. In the end we will conclude with the benefits of this project, future directions of the study and how it can be improved.

2 Statement of the problem and solutions

2.1. What is article metadata

According to the definition given by Merriam-Webster dictionary [2], metadata is "data that provides information about other data".

Types of metadata are specified in different ways, depending on the field where metadata are used. As to National Information Standards Organization (NISO), there are three main types of metadata [3]: descriptive, structural, and administrative.

- *Descriptive metadata* describes a resource for purposes such as discovery and identification. For example, such elements as title, abstract, author, and keywords.
- *Administrative metadata* is information for a resource management. For example, when and how the file was created, file type and other technical information, including who can access it.
- *Structural metadata* indicates how compound objects are organized. For example, how pages are ordered to form chapters.

As to the field of journals and books publishing, we need to consider such metadata as *articles metadata*. This data can be referred to as Descriptive metadata. Journal Publishing Tag Library NISO JATS defines article metadata as "Container element for information concerning the article that identifies or describes the article" [4]. Such type of metadata may contain bibliographic data (authorship, article title, copyright year, and publication date), descriptive material (keywords and abstracts), any numbers identifying an article. Some journals distinguish three types of article metadata: about the specific article, about the journal, and about the issue of the journal containing the article.

The American Mathematical Society states that "By creating metadata for journal articles, authors and publishers document key elements relating to their works. This metadata facilitates data exchange between authors, publishers, the scientific community and other online service providers" [5].

When creating metadata for a journal article, the user (author or publisher) identifies its basic elements (author, title, journal, volume, issue, etc.) and creates a Document Type Definition (DTD) consisting of those elements. A DTD is a structured, tagged representation of an article. This DTD usually is created according to the requirements of the respective database or community. Using a DTD and data files, information regarding your articles can easily and accurately be communicated to other members of scientific community, publishers, or online service providers.

2.2. Bibliography in article metadata

Initially, the list of references in scientific articles was intended for the reason that authors could be based on already published results without their repetitions. Using this list, if necessary, readers can find and read more information about these publications. To facilitate the search for the publications cited, the information must be accurate and complete. There were some rules of how to write a list of references. The accumulation of knowledge, the increasing number of publications, the development and use of informational technologies led to some changes of these rules. The bibliographic references began being used to calculate the index of citations of scientific publications. Citation index – is one of the most important quantitative indicators to measure the importance of a scientific publication. The process of indexing the bibliographic resources is an important factor that determines how useful the articles which are published in scientific journals are.

Nowadays, for greater indexing recognition, scientific journals tend to be present in the most prestigious international databases such as SCOPUS, Thomson Reuters, INSPEC, zbMATH, DOAJ, etc. Each data base elaborates its own format for article-level and journal-level metadata.

For example, in DBLP (the on-line database with bibliographic information on major computer science publications) previously the input "language" was in HTML style. The idea is to enter tables of contents (TOCs) in a format, which is very close to their appearance on the DBLP TOC web pages. At the given moment in this database new XML submission format is accepted.

DOAJ (The Directory of Open Access Journals) provides for publishers three options [6]:

1) to prepare the metadata in XML format and then to upload the XML article (DOAJ Native XML);
2) to enter the metadata about the articles online - using the online form on their website;
3) to use the DOAJ API to upload content.

In zbMATH (the database containing abstracts and reviews of publications in pure and applied mathematics) they have been developing a new indexing procedure. New requirements for journals indexed by them are elaborated. Among these requirements, metadata are to provide in .xml format the information about published articles and bibliographies contained there. It is a specifically developed zbJATS input format: an XML format based on the Journal Article Tag Suite by NBCI.

The preparation of such information in XML-formal is a long routine. In this context it is necessary to develop a system that would allow automatic processing of references according to the international standards, in order to achieve the desired results in the shortest time. The Tag Library is used as a reference to look up XML tags and how to use them, to browse around and familiarize you with this Tag Set, to see examples of correct or recommended usage, and even to find guidance for how to implement software to handle documents that are marked up using this Tag Set. [7].

Writing a bibliography is the hardest and time consuming part of the preparation of this information, that's why, we used this Tag Library in order to develop a tool that will obtain faster the information we need. Data processing becomes more difficult if the authors provide incomplete information in their list. Therefore, we need strict discipline when you are writing a bibliography, because the source described has to be informative, complete and accurate.

Thus, to solve this problem, we work on an application whose purpose is to process bibliographic data in XML format which is required by zbMATH. As the input data, we take an article bibliography written in LaTeX, in accordance with the requirements of the journal Computer Science Journal of Moldova (CSJM) based on IEEE2015 standard.

LaTeX is a document preparation system for high-quality typesetting. It is most often used for medium-to-large technical or scientific documents but it can be used for almost any form of publishing [8]. The writer uses markup tagging conventions to define the general structure of a document (such as article, book, and letter), to stylize text throughout a document (such as bold and italics), and to add citations and cross-references.

A paper can contain many types of references: books, journals, conferences, online sources, chapters, thesis, dissertations, reports etc. Every category has its specific elements consisting of: author names, title of the article, subtitle, source, publisher name, publisher location, date and year of publishing, editor/s, digital object identifier (DOI) number, ISSN, ISBN, page numbers etc. If some reference is originally published in language other than English, additionally respective titles (of articles, journals and books) in original language are provided and the language is written in round brackets.

3 Practical application

One method to satisfy the necessity of automation the process of bibliography making is to elaborate a system that would simplify the redactor's work. The advantages of using such system are enormous because the time for metadata preparation will be reduced, time is money so we will save money at the same time, we will send the information on time to be indexed and the last but not the least important is that the information will be accurate and complete.

We started to develop this system last year, and until now it has processed already 4 issues of the "Computer Science Journal of Moldova" – the journal published by the Institute of Mathematics and Computer Science of the Academy of Sciences of Moldova. This system was developed using PHP programming language, CSS styles and HTML markup language, also it has incorporated a MySQL database where all the results are saved.

Let's consider an example for book reference and an example for periodicals reference, the template of these references written in LaTeX should look as follows:

```
%{book}                                                                    (1)

\bibitem{21} J. K. Author, \textit{Title of His Published Book} (Title of Series) x-th ed., vol. x, J. S. Editor, Ed. City of Publisher, (only U.S. State), Country: Abbrev. of Publisher, year, xxx p., ISBN: xxx-xx-xxx-x-xxxx.

%{periodicals}                                                             (2)

\bibitem{1} J. K. Author, ``Name of paper,'' \textit{Title of Periodical}, vol. x, no. x, pp. xxx--xxx, Abbrev. Month, year.
```

Below we have proposed two examples for each type of bibliographic references. Of course it is desirable for a bibliographic reference to be complete and to contain as many specific elements as possible, but references may not include all information, sometimes, some details of the source are missing and the author offers few information about it.

```
%{book}

\bibitem{Nielson1992} H. R. Nielson and F. Nielson, \textit{Semantics with applications: a formal introduction}, Wiley Professional Computing, John Wiley \& Sons, Inc., New York, 1992, 252 p., ISBN: 0-471-92980-8.
```

Using the new requirements of zbJATS input format which is an XML format based on the Journal Article Tag Suite by NBCI we will obtain the following information about one bibliographic reference:

```
<ref>
<element-citation publication-type="book" publication-format="print">
   <name>
            <surname>Nielson</surname>
            <given-names>H. R.</given-names>
   </name>
   <name>
            <surname>Nielson</surname>
            <given-names>F.</given-names>
   </name>
<source>Semantics with applications: a formal introduction</source>
<publisher-loc>New York</publisher-loc>
<publisher-name>Wiley Professional Computing, John Wiley \& Sons, Inc.</publisher-name>
<year iso-8601-date="1992">1992</year>
<size units="pages">252 p.</size>
<isbn>0-471-92980-8</isbn>
</element-citation>
</ref>
```

And the same criteria are used for periodical references:

```
%{periodicals}

\bibitem{14} P. Mahonen, M. Zorzi, "Cognitive Wireless Networks," \textit{IEEE Wireless Commun.}, vol. 14, no. 4, pp. 4--5, Aug., 2007.
```

```
<ref>
    <element-citation publication-type="journal" publication-format="print">
            <name>
                    <surname>Mahonen</surname>
                    <given-names>P.</given-names>
            </name>
            <name>
                    <surname>Zorzi</surname>
                    <given-names>M.</given-names>
            </name>
    <article-title>Cognitive Wireless Networks</article-title>
    <source>IEEE Wireless Commun</source>
    <volume>14</volume>
```

```
        <issue>4</issue>
        <fpage>4</fpage>
        <lpage>5</lpage>
        <year iso-8601-date="2007">Aug. 2007</year>
        </element-citation>
    </ref>
```

This is how a book and a periodical reference should look like after being processed. Writing all this information by hand requires a lot of time and resources, and these are only two bibliographic references. But an article can have tens of these references, and every journal has five, ten or even twenty articles, it's a lot of work here. We intend to automate every process of articles preparation for publishing in Computer Science in the future and this is what we're trying to do now.

3.1 The algorithm

How does the system recognize the elements of a bibliographic reference? We use the regular expressions. A regular expression or regex (sometimes called a rational expression) is, in theoretical computer science and formal language theory, a sequence of characters that define a *search pattern*. Using regular expression you can search a particular string inside another string, you can replace one string by another string and you can split a string into many chunks.

PHP offers functions specific to two sets of regular expression functions, each corresponding to a certain type of regular expression. You can use any of them based on your comfort.

- POSIX Regular Expressions
- PERL Style Regular Expressions

The structure of a *POSIX* regular expression is not dissimilar to that of a typical arithmetic expression: various elements (operators) are combined to form more complex expressions.
The simplest regular expression is one that matches a single character, such as g, inside strings such as g, haggle, or bag.
Perl-style regular expressions are similar to their POSIX counterparts. The POSIX syntax can be used almost interchangeably with the Perl-style regular expression functions. In fact, you can use any of the quantifiers introduced in the previous POSIX section. A meta-character is simply an alphabetical character preceded by a backslash that acts to give the combination a special meaning.
For instance, you can search for large money sums using the '\d' meta character: **/([\d]+)000/**, Here **\d** will search for any string of numerical character. Following is the list of meta-characters which can be used in PERL Style Regular Expressions [9].

. - Matches any single character. Within POSIX bracket expressions, the dot character matches a literal dot. For example, `a.c` matches "abc", etc., but `[a.c]` matches only "a", ".", or "c".

[ ] - A bracket expression. Matches a single character that is contained within the brackets. For example, `[abc]` matches "a", "b", or "c". `[a-z]` specifies a range which matches any lowercase letter from "a" to "z".

| - The choice (also known as alternation or set union) operator matches either the expression before or the expression after the operator. For example, `abc|def` matches "abc" or "def".

$ - Matches the ending position of the string or the position just before a string-ending newline. In line-based tools, it matches the ending position of any line

***** - Matches the preceding element zero or more times. For example, ab*c matches "ac", "abc", "abbbc", etc. [xyz]* matches "", "x", "y", "z", "zx", "zyx", "xyzzy", and so on.

? - Matches the preceding element zero or one time. For example, ab?c matches only "ac" or "abc".

+ - Matches the preceding element one or more times. For example, ab+c matches "abc", "abbc", "abbbc", and so on, but not "ac".

For every specific element of an article, we construct search patterns depending on what we are searching for in the references. Using the next search pattern our system searches and finds in the bibliographic reference written in LaTeX only the name of the authors:

```
"/\\\\bibitem\{[0-9a-zA-Z, _-]+\}(.*?)\\\\textit/i"
```
(3)

For each element like publication year, publisher location, publisher name etc. we use a specific search pattern. In this example, the system looks for any letter that comes after "\bibitem[]" – is a command used to enter a reference in a thebibliography environment.

PHP offers following functions for searching strings using Perl-compatible regular expressions − ***preg_match_all(), preg_replace(), preg_match()*** and others.

- The *preg_match_all()* function matches all occurrences of pattern in string.
- The *preg_replace()* function operates just like ereg_replace(), except that regular expressions can be used in the pattern and replacement input parameters.
- The *preg_match()* function searches string for pattern, returning true if pattern exists, and false otherwise.

The complexity of this system is based on the correctitude of writing an article, each comma, each punctuation sign has an important role for obtaining the right information. Thus we permanently work:

- with article authors, insisting on their accuracy and maximally possible completeness;

- on this application to improve its function and to facilitate the work of redactors and researchers.

4 Conclusions and future work

Our experience has shown that the system elaborated to automate the process of obtaining metadata provides good results in a record time. In order to obtain better results, more accurate and complete it is necessary to respect the requirements for journal indexing, to write the bibliographic references in LaTeX according to IEEE style manual standards and to provide as much possible information as one can.

We continue to work on this project and in the future we want to develop and improve it:

- to use BibTeX format for bibliography description as input too;
- to introduce new requirements from different databases because until now it is build based on zbJATS input format and the articles can be indexed only in ZbMATH database using the generated XML-files.

It will take time and a combined effort of many people but in the end we want to obtain a finite product that will allow us to compete with big publishing houses.

Acknowledgement: This work was supervised by Mrs. Tatiana Verlan, researcher at the Institute of Mathematics and Computer Science. I would like to take this opportunity to express my profound gratitude and deep regard to her and also to Prof. Svetlana Cojocaru for their exemplary guidance, valuable feedback and constant encouragement throughout the duration of the project. Their valuable suggestions were of immense help throughout this project work. Their perceptive criticism kept me working to make this project in a much better way.

I would also like to show my gratitude to all the members of the Institute of Mathematics and Computer Science for sharing their pearls of wisdom with me during the course of this research.

References

[1] "The role and relevance of bibliographic citation databases," [Online]. Available: http://www.biochemia-medica.com/content/role-and-relevance-bibliographic-citation-databases

[2] "Definition of metadata", [Online]. Available: http://www.merriam-webster.com/dictionary/metadata

[3] Zeng, Marcia (2004). *"Metadata Types and Functions"*. NISO. Retrieved 5 October 2016.

[4] "Article Metadata", [Online]. Available: https://jats.nlm.nih.gov/publishing/tag-library/1.1d1/n-y6c0.html .

[5] Creating Metadata, "How to create metadata for articles", [Online]. Available: http://www.ams.org/publications/journals/create-metadata .

[6] Directory of Open Access Journals, "How to get your journal content / article metadata indexed in DOAJ. Uploading content" [Online]. Available: https://doaj.org/publishers

[7] "ZbJATS input format", [Online]. Available: https://zbmath.org/zbjats/

[8] "The LaTex Project: An Introduction to LaTeX", [Online]. Available: https://www.latex-project.org/about/

[9] "PHP Regular expressions" [Online]. Available: https://www.tutorialspoint.com/php/php_regular_expression.htm

DORINA LUCA
Institute of Mathematics and Computer Science of The Academy of Sciences of Moldova
5 Academiei, street, Chisinau MD-2028
MOLDOVA
E-mail: dorina.luca@math.md

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

Life of a Kutit

Andrei Muresan

Abstract

This project is about the development of a game created to study human preferences, their emotions and decisions, where we tried to make people laugh in order to improve player retention. The game titled "Life of a Kutit" was created using Python scripts, and the Pygame module, all written using the Eclipse program.
The game has a simple knife as the main character and presents some unexpected and difficult choices for the player. It first started as a learning tool for programming and developed into a stand-alone game with unique assets and game experience. In this paper we will present the software architecture, the unique parts of our application, the user feedback and our developing process.

1 Introduction

Many games created by students usually start as a project or homework, and this is how "Life of a Kutit" came to be, as a programming learning project. The game was born based on some imaginary ideas, a knife that can jump and move like the known Mario [1] character. The theme of the game proves that game design is a creative discipline where anything is possible.
The application was developed in Python [8], using Eclipse [11] as IDE. Eclipse is a well known Java development environment and we used the PyDev [10] plugin for our project. This combination made it easy to use Pygame [9], a python module, specifically made for video game developing.
"Life of a Kutit" is a hybrid 2D platform game, which can be deployed and played on most desktop systems. The original part of this project is the way we combined puzzles and decision games with arcade elements which keep the players stimulated during the game. The targeted users for this application are 16 to 25 years old students who enjoy platform games and want to relax by playing unique indie games.
Indie games [5] are video games that are usually created without the financial support of a publisher or a big company, and are developed by individuals or small groups.
In section 2 of this paper we will present the state-of-the-art of game design and the structural development of the game will be explained in section 3. We also gathered user preferences and feedback which will be discussed in section 4, followed by our conclusions.

2 Related work

In our development process we studied other frameworks and ideas regarding the development of a game.
The MDA framework[3] consists in the focus on the development approach on the game. It explicitly describes the design and its impact to the developing process. By understanding the dynamics of the game, it allows us to improve the user's experience and overcome undesired behaviours.
The advantages and disadvantages of Python game programming were studied by Dawson B. in [6]. Like any programming language, Python has it's particular run-time problems, but with a low level complexity. Although the GUI elements in Python allows you to faster implement a friendly interface for the user and to control graphical resources better than C++.
In [4] the authors created a definition for the abstract field of "game design". Depending on the team that worked at creating a game, specific feedback was received from surveys, testers, QA teams, etc. Each game is different than others in unique ways which implies that creating a definition for "game design" needs a lot of attention and documentation on different types of games. Because of that, the term "game design" can be defined as an art at which each developer adds something new, depending on personal preferences and user feedback alike. Using this knowledge we began working on our own platformer game.
After we received the user feedback, the game was changed to increase the player satisfaction, while still keeping the original combination of aesthetics and game story intact. We then combined puzzle parts with arcade gameplay, adding random elements to keep the humor at a higher level.

3 The Game

The game is divided in three main loops. First one is the starting screen, second one is the story, and last one is the game loop itself. The graphics were created so that each backgrounds theme matches the moment of the game, while the music was chosen to sound more like a platform game [2], to correspond with this specific type of game and to add something new and funny to the genre.

3.1 Start screen

The menu is simplistic as it only has 2 buttons that have to be clicked, and it also displays the top 10 scores without any further complication (fig. 1). During the game, hints are popped on the screen, keeping the players attention focused and making the game experience smoother.

The "Exit." button is present on almost all the backgrounds, and it can be clicked to exit the game at any time. On the left side of the start screen, there is a top 10 of the high scores and the players names.

Clicking on the "Start game." button will send you to the story loop function, and continue further.

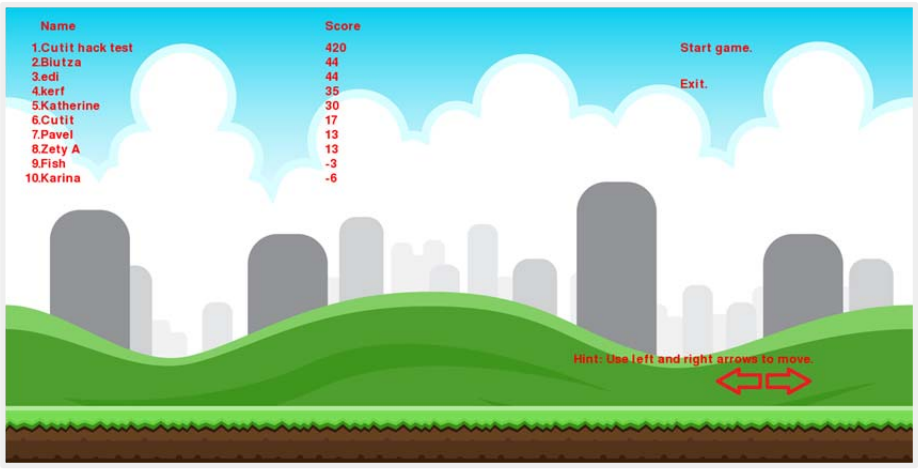


Fig. 1: Start screen exemplification

3.2 Story loop

The story is present to give the players the option to have more points at the end, at the price of playing for a longer period of time, but they can also choose to skip it, and potentially not earn a few extra points. As shown in fig. 2, there are two buttons that you can click, “Yes.” or “No.”.

Clicking the “No.” button will send you directly to the game loop, while clicking the “Yes.” button will start the story, at which the player has to pay attention, because the question asked after the story will be related to it.

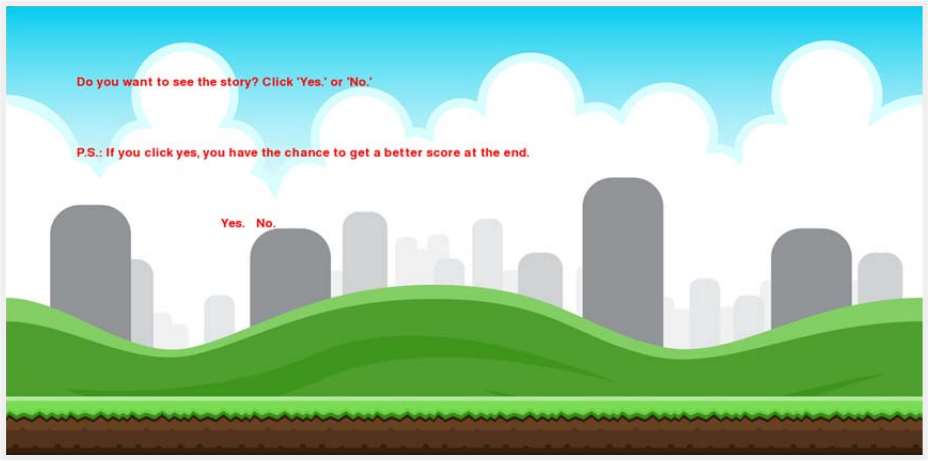


Fig. 2: Story screen exemplification

3.3 Game loop

Using the left and right arrows, the knife image will move across the screen (Figure 3). To move from one screen to another, you have to get with the knife at the end of the background, if the requirements of that specific part of the game are completed, otherwise, a hint will pop-up telling the player what is left to be done.

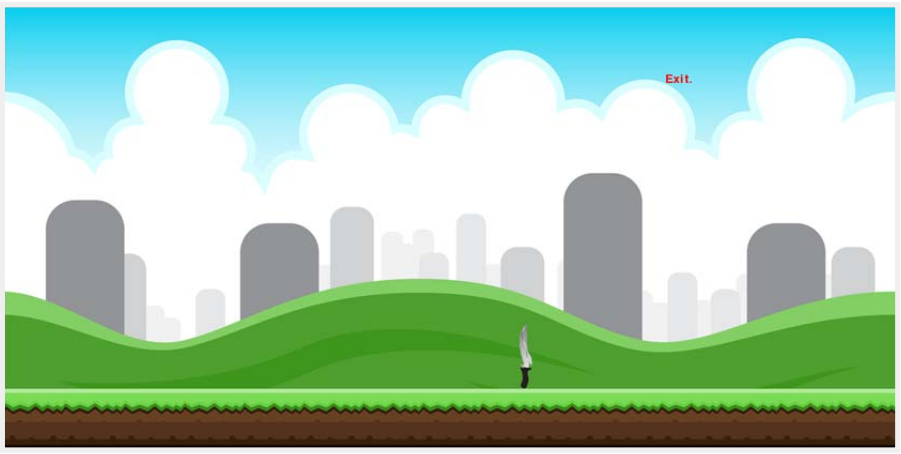


Fig. 3: Game start exemplification

As before, using the left and right arrows, the player has to dodge the ice-bolts coming from above (Figure 4).

The more the player dodges, the better the score.

The requirement for this background is to dodge as many ice-bolts as possible.

The maximum speed has been set to a difficult one, but not impossible to dodge. The ice-bolts start at +- 100 pixel in the left and right of the knife, but being at the edges of the background will not make them disappear outside the screen.

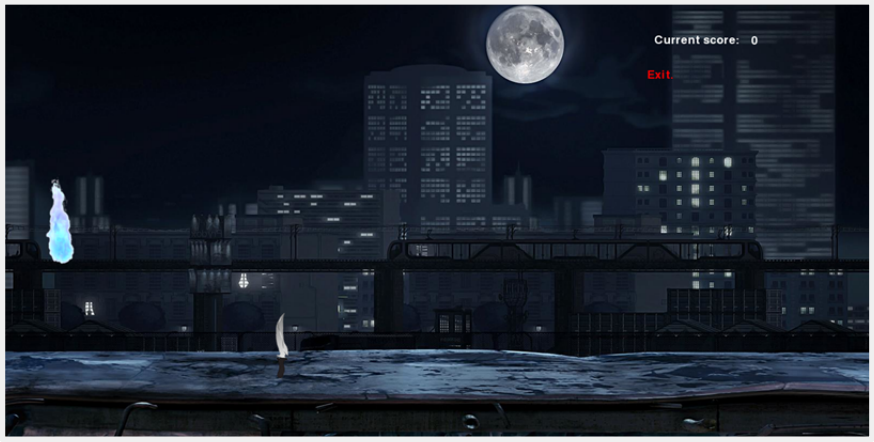


Fig. 4: Gameplay exemplification

After the ice-bolts background, comes the blacksmith one (Figure 5).

Here, the player has to choose one of 3 options, by clicking on which it thinks is the better one. Depending on its choice, the player can remain at the same score, get a better one, or lose some of it.

Same as before, the player can not jump from one background to the other without clicking one of the options, or the “Exit.” button.

Fig. 5: Blacksmith exemplification

The final background (Figure 6) is the shop one. Here the player has the chance to win or lose the most points, so choosing the right option is the best thing. After clicking an option, there is a small delay between automatically jumping from this background to the credits one.

Fig. 6: Shop exemplification

3.4 Program Code

Python code of how the ice-bolt is modifying its position and speed:

```
li#edfnjurxqgbqxpehu#@@#4=#
########li#ghdwk@@3=#
############li#lfherowbfxuhqwd|#@@#3=#
################li#hdgb{?83=#
####################lfherowb{@udqgudqjh+hdgb{0483/#hdgb{.6<,#
################hdi#hdgb{?:3=#
####################lfherowb{@udqgudqjh+hdgb{0453/#hdgb{.8<,#
################hovh=#
####################lfherowb{@udqgudqjh+hdgb{0563/#hdgb{.83,#
################li#lfherowb{A4433=#
####################lfherowb{@udqgudqjh+<33/4433,#
################lfherowb|#@0483#
################lfherowbfxuhqwd|#@#4#
############hovh=#
################jdphGlvsod|1eow+lfherowbpdjh/#+lfherowb{/lfherowb|,,#
################lfherowb|@lfherowb|.lfherowbvshhg#
################li#lfherowb|A@974=#
####################lfherowbfxuhqwd|@3#
####################fxuhqwbvfruh@fxuhqwbvfruh.4#
################li#lfherowbvshhg?@<=#
####################lfherowbvshhg@lfherowbvshhg.3134
#
#
```

This is a simple overlap of images, first being the background, second the knife, and third but not least, the ice-bolt itself (Fig. 4).

Because the game is made to work at 60 ticks per second, there is also an ice-bolt speed limit included. The reason behind the speed limit is that the player could actually play the game for a long time, if attention is being given. Without a speed limit, the ice-bolt would move faster and faster, and that made the game impossible to survive for longer than 40 seconds.

5 User preferences

A survey was created with the purpose of finding the preferences of the targeted users and use them to further develop the game. The survey was completed by 32 students from Babes-Bolyai University of Cluj, Romania, the rest up to 51 being filled by twitch [7] streamers and some professional gamers.

75 people then played the game and about 27 agreed to send us their choices made inside the game and other statistical data. We used this data to analyze if the real choices were consistent with the declared ones.

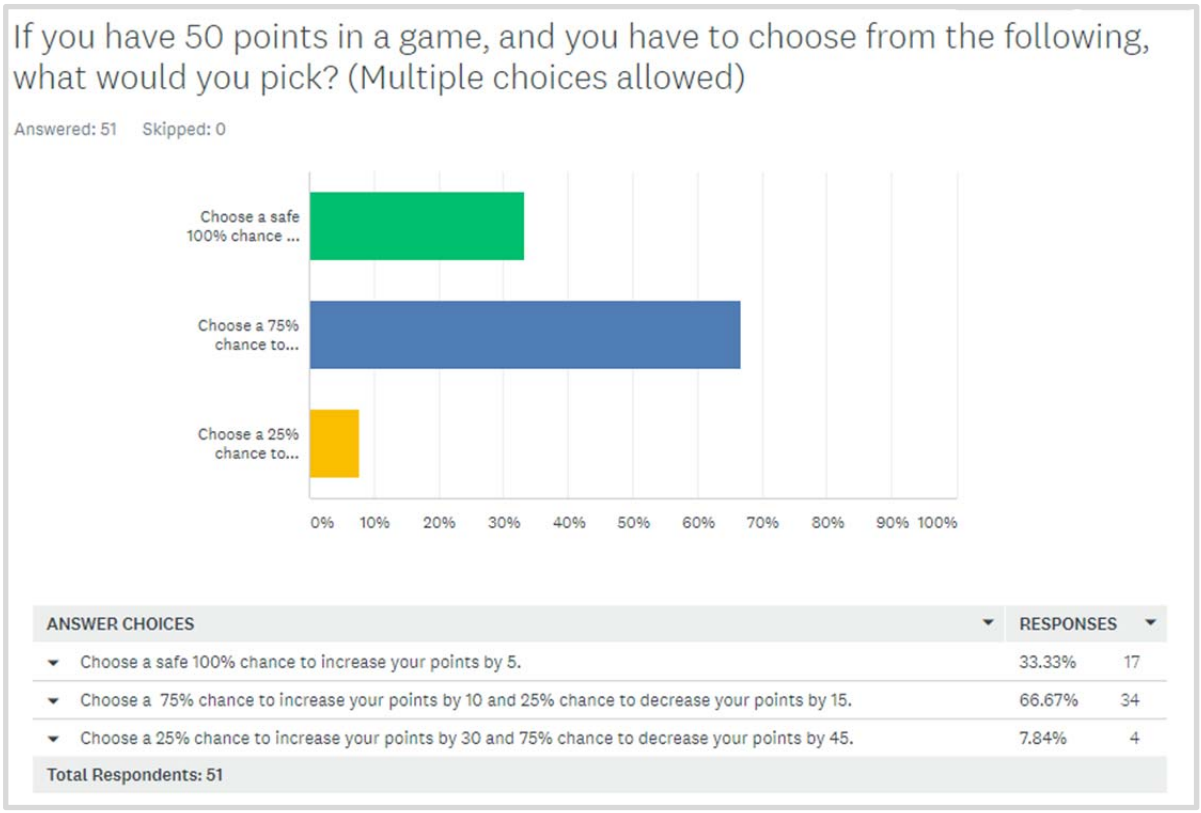


Fig. 7: Risk aversion question from the survey

We asked the players how and if they are willing to risk to gain points, at the cost of losing some (Figure 7). First option was a safe one to gain points with no risk involved, and according to the real result, 40% of the players chose that option, compared to the 33% from the survey. The second option was a small risk to lose some points at the cost of winning more than the first option. Compared to the 66% from the survey, 50% of the testers chose this option. The last option was a high-risk with high-reward one, and, it was the least chosen, by 8% in the survey and 10% in the actual game.

As shown in Fig. 2, players have the option to choose whether they want to see the story or not. The people who completed the survey were asked if they want to play a game for the story or to gain more points (Figure 8). The real result and the survey result were similar: 49% of the surveyors chose both and 39% chose the story, a close result to the 75% of the testers who chose the story during the game.

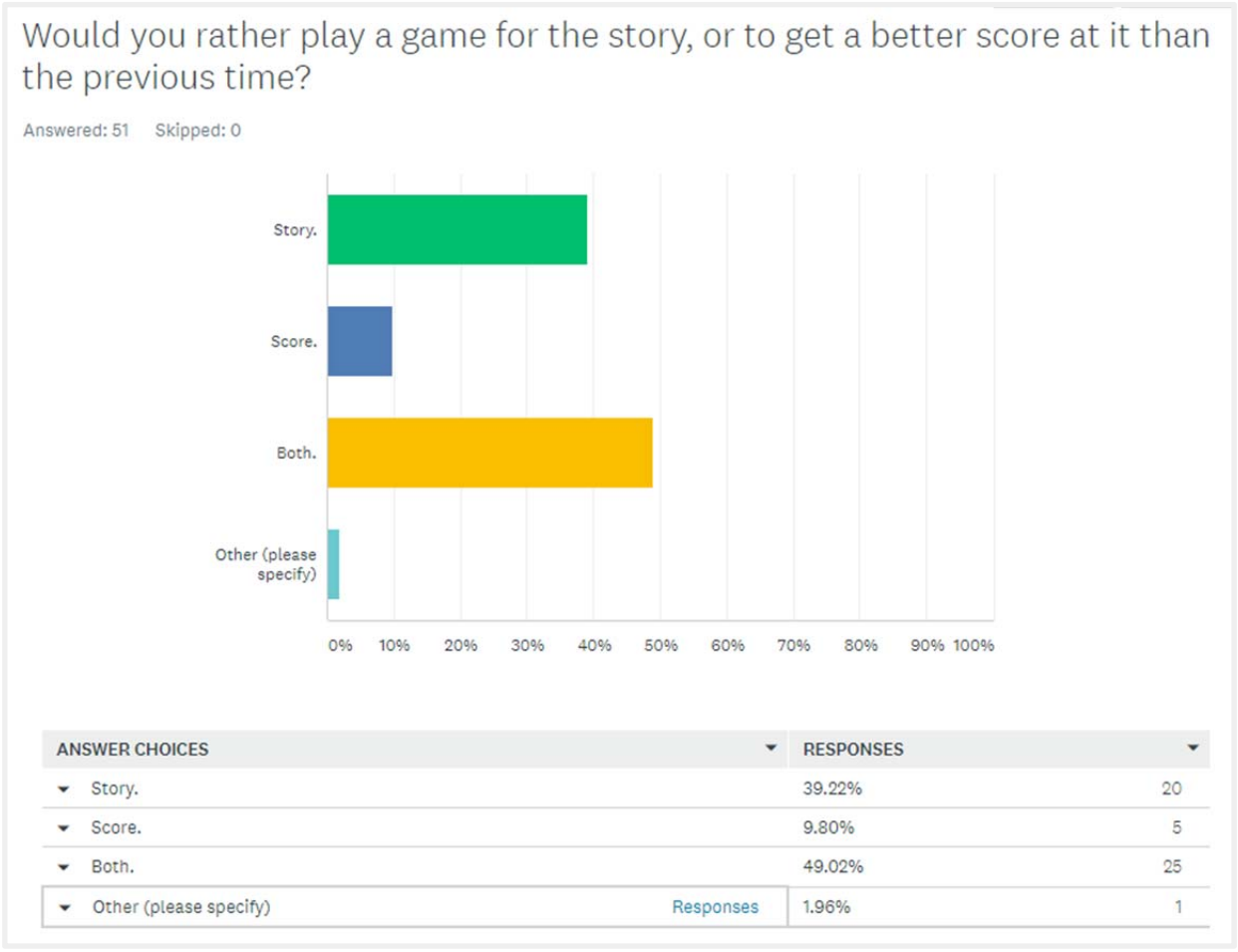


Fig. 8: Story versus points question in the survey

6 Conclusions

In conclusion, most of our objectives have been met, and we successfully implemented and tested our game. We managed to capture the player's attention, constantly being asked if the game is being continued, even asked for a copy of the finished product.
In the first weeks, the main focus of this project was learning game design and that aspiration progressed into a complex game, where we took user feedback and preferences into account.
In the future, we would like to add a custom starting background for the highscores, add animations to the knife movement and a sprint option. Boss fights and other puzzle levels will be added, including a *jumper* level.
Further development of the game will be done in Unreal Engine, for an easier implementation of the image sprites, image overlay and game mechanics. The latest version of *Life of a Kutit* can be accessed at https://goo.gl/qHZue3.

Acknowledgement: Project was coordinated by Drd. Bota Florentin, Babes-Bolyai University.

References

[1] Ryan, J.,*Super Mario: How Nintendo Conquered America,* Portofolio/Penguin, 2011
[2] Adams, E., *Fundamentals of Game Design*, New Riders, 2013
[3] Hunicke, R., LeBlanc, M., & Zubek, R., *MDA: A formal approach to game design and game research.* In Proceedings of the AAAI Workshop on Challenges in Game AI (Vol. 4, No. 1, p. 1722), 2004
[4] Salen, K., Zimmerman, E., *Rules of Play: Game Design Fundamentals,* MIT Press, 2004
[5] Diver, M., *Indie Games: The Complete Introduction to Indie Gaming*, Michael O'Mara Books, 2016
[6] Dawson, B., *Game scripting in Python*,. Proc. GDC, 2002
[7] Hardin, C., *Twitchin': The Basics of Twitch Streaming for Beginners*, Pronoun, 2016
[8] thenewboston, Pygame (Python Game Development) , [Online], Available: https://www.youtube.com/channel/UCJbPGzawDH1njbqV-D5HqKw [15 January 2017]
[9] Pygame Documentation, Available: https://www.pygame.org/docs/ [12 November 2016]
[10]Russell, J., Cohn, R., *Pydev*, Book on Demand, 2012
[11]Holzner, S., *Eclipse: Programming Java Applications*, "O'Reilly Media, Inc.", 2004

Andrei Muresan
Babes-Bolyai University
Faculty of Mathematics and Computer Science
Mihail Kogalniceanu nr.1, 400084, Cluj-Napoca
Romania
E-mail: muresanandrew@gmail.com

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

Making Animations in Blender

Alexandru Pintea

Abstract

Computer-generated imagery (CGI) is used nowadays in many fields as medicine, industry, entertainment and education. It is encouraged the development of animation. The current work presents the main steps to create *2D* and *3D* animations using *Blender*. Several original projects for educational purposes are included to illustrate the animation processes.

1 Introduction

From ancient drawings in Egypt about 4000 years ago to modern day digital animations the development is considerable. Character design has expanded from hand drawing on paper or other surfaces, to using special graphics pads (so as to transfer the images) onto a computer, or directly modeled and finally animated on a computer. [2].

Blender is a multifunctional *3D* graphics platform. It can be used for multiple tasks from drawing a character to animation. The program is free and can be installed on most operating systems Apart from its capabilities and popularity, *Blender* is easy to learn and use [1]. *2D* animation was the first type of animation. It was basically a fast sequence of manually drawn pictures placed on a spinning system. Today most cartoons are based on *Computer Generated Imagery (CGI).*

As in any field, computer graphics have many tools that can be used, but are often not free of charge. When talking about *2D* most notable animation is *Adobe After Effects*, which requires license, second is *Go! Animated*, the next is *Xtranormal*, and the list can continue. Blender is by far the best free software because it can produce *CGI* almost professionally and can be used for a wide variety of projects [4]. Another free *2D* cartoon maker is *Animaker* that can be used similarly to PowerPoint presentations and offers a professional look, but is limited by a number of stored images and characters [4].

In the last years, the best visual effects for movies have used *AutoDesk's Maya* animation software. This software allows the inclusion of animation scenes, *2D* and *3D*, in real-life movies. *Blender* is mainly a *3D* modeling, animating and rendering program; meaning that an entire animation could be done with *Blender*.

A selection of programs used in animation both in *2D* and *3D* animation will be further presented.

Section 2 illustrates the technology for *3D* animation on a case study, drawing and animating a bird to fly; the small project is called "*Flying bird*" [6]. Section 3 presents the stages of creating a *2D* cartoon on a case study: creating an original cartoon sequence, a *2D* animation using *Blender*. The sequence is called "*Holiday*" [5]. Further work and main challenges concludes the article.

2 Creating 3D Animation in Blender

The section includes a presentation of the main steps of making a *3D* animation in Blender.

At first images are inserted in *Blender* by adding it as a background image from the menu on the right in the *3D* module to start *3D* modeling around it (Fig. 1). On can set the view only from the *x*, *y* or *z* axis or from multiple axis at the same time. This is how could also be inserted a movie clip when using *VFX*. To see the image or movie clip should be used the orthographic projection.

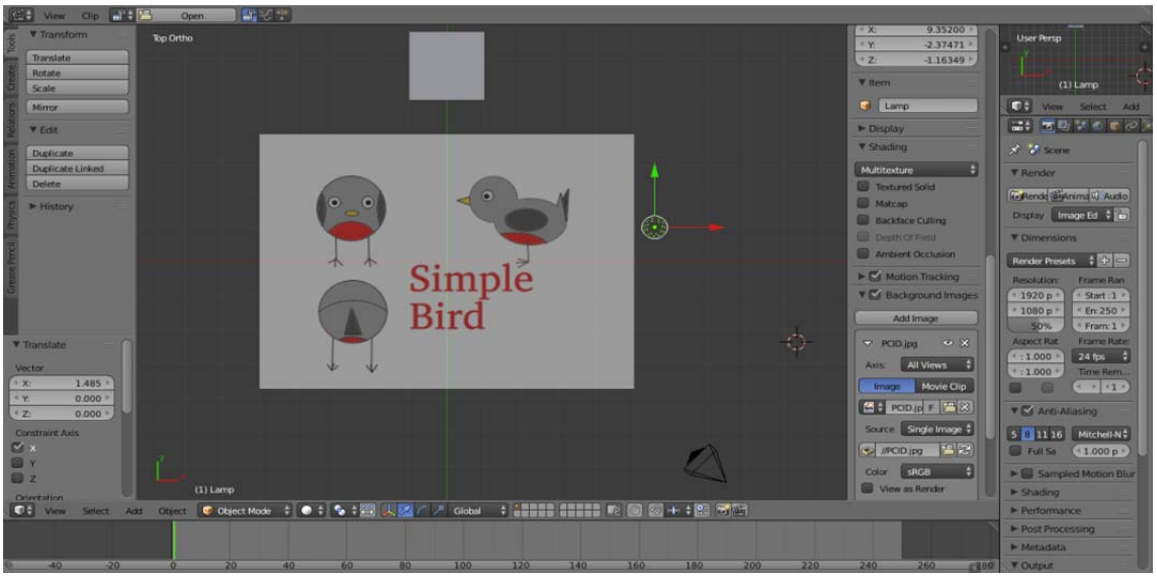


Fig. 1. How are made in Blender the views of bird in the "*Flying bird*" project version [6].

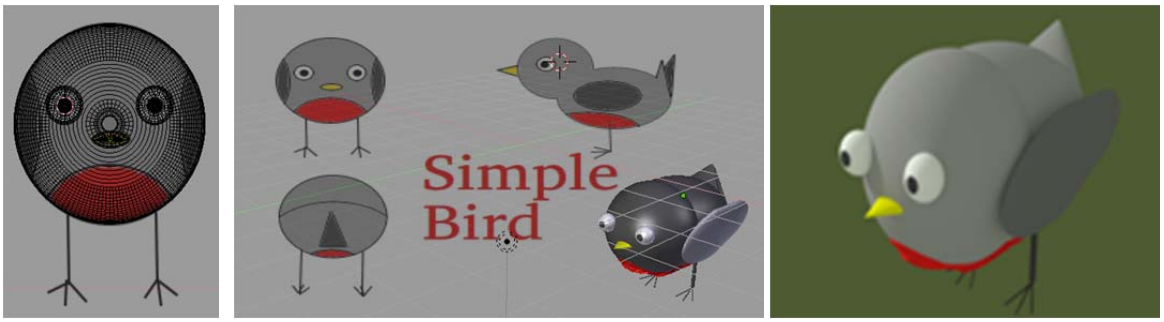


Fig. 2. Example: Views of the birds from initial to final images: "*Flying bird*" project version [6].

Fig. 2 illustrates also how is made a *3D* model of an image. If the image is disproportional from one angle to another it is not a problem because it is possible to move the *3D* object in depth. To view the image through the *3D* object press the "z" key to switch to grid view in the *3D* module.

Multiple meshed and primitive objects to the *3D* model could be added to enhance the image. After modeling set the *Ambient Occlusion* of the world to value *0.3* and the *Sampling* for example to *200* both in preview and in render; the resolution used is *100%.*

An animation with the created objects could be now created. For a simple background change the world's color in the properties module and the intensity of the chosen color. Another thing to do is to add a panel with another color to complete the background. As in *2D* models it is easy to insert *key frames* and make an animation. *3D* animation takes more time than *2D animation* to render.

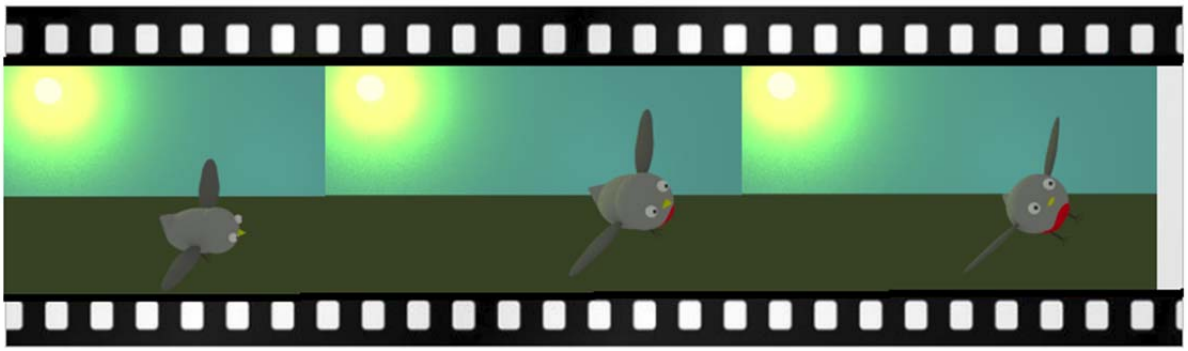


Fig. 3. The "*Flying bird*" version project is now rendered: some sequences are illustrated [6].

After the images are rendered (Fig. 3), *JPEG* format as they are more efficient, they can be post processed as *AVI JPEG, AVI Raw, Frame Server, H.264, MPEG, OGG Theora* or *Xvid* files. After the frames are done, sound should be added using *Blender*'s movie clip editor. If the frames are rendered as a movie file, insert the movie in *Blender* and add sound. Some freeware programs further to be used are *Kdenlive* to add sound to a movie file and respectively Audacity to modify the sound.

3 Creating 2D Animation in Blender

At first the animator has to make the storyboard for each scene on paper or using a graphic pad. The storyboard of the animation is essential to know exactly what *CGI* is going to implement.

3.1. Creating characters

The characters can be drawn in any graphic editor, for example in *Paint* under *Windows, Pinta* or *Gimp* under *Ubuntu.* The difficult part is the level of detail added to the character. In a complex cartoon, the character has many gestures and facial expressions. For this, the background of each element is transparent. It takes long but it is necessary to make personalized and well-organized libraries of *2D* drawings.

3.2. Include the drawings into Blender

Even though it seems easy to make a surface in *Blender* and to attach a picture, in practice the things are more complicated. The image should be repeated until it covers the object.

This is why is enabled *Blender* add-on called Import images as planes so that the images are properly attached to the plan. This will work with the installed *Blender* render, *Blender* and *Cycles* render engines. Furthermore, set in the *Cycles* render and *Blender* game: *"Use Alpha: Premultiplied*." To view the drawings, use the "*Texture*" mode in "*3D View*".

3.3. Rigging a character

The rigging stage is rather difficult. The more facial gestures and expressions the character will have, the more difficult it will be.

Rigging is the addition of *2D* shapes, generally lines, which control the character as a doll, similar to the strings of the puppet theater characters. This is necessary to move the character elements easily without the need for rotation or translation problems. The complexity of the doll (in number of components) will increase the number of elements to be animated, but this will improve the quality of the final image. For example, Fig. 4 includes the action of tying characters to the "*Holiday*" sequence.

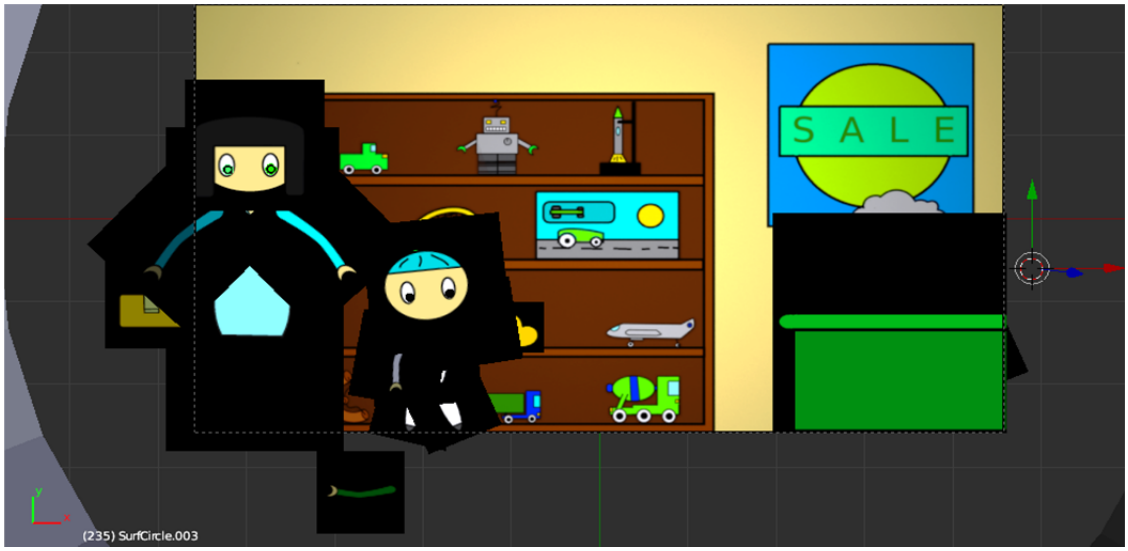


Fig. 4. Example: Rigging characters in the "*Holiday*" project [5].

3.4. Setting the scene

Once are arranged all the characters it is necessary to use separate workspaces, opened in *Blend*er to see the final image after render. It should be checked that there are no two visible elements that may overlap. This is hard to do when there are a large number of elements in a scene, but verification is essential to have no problems with the final drawing of the animated drawing.

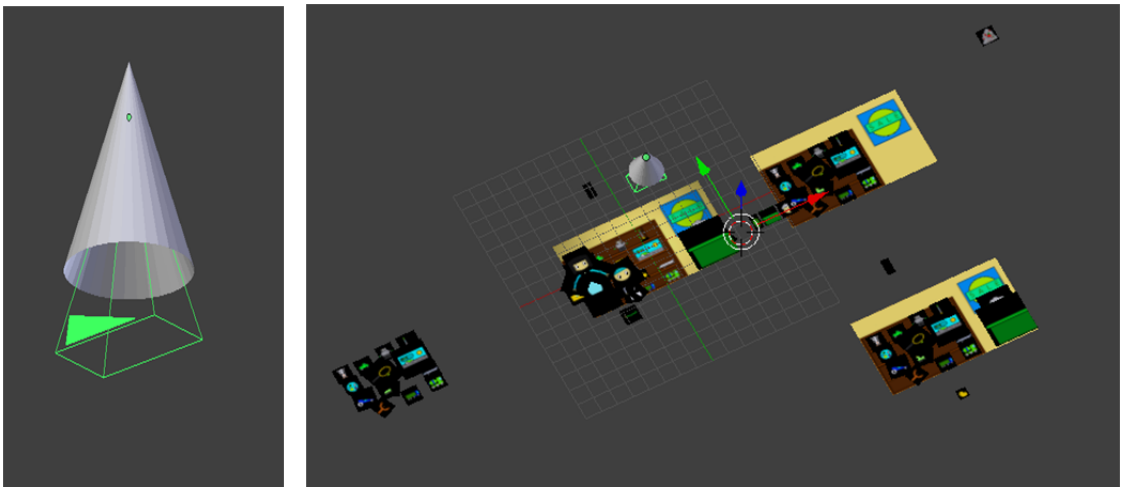


Fig. 5. Light source (left) and several sequences of "*Holiday*" project with light sources (right)[5].

Moving an item away from the background can be detrimental, as they could shade the other elements. At this point it is useful to make a light source, a lamp, so that the characters and the rest of the scene are visible to the camera. The lamp should be placed on the same coordinates with the camera above the scene, and with a slightly larger coordinate than the camera. This will shorten the rendering process because there are fewer rays of light coming into the camera from the single light source and shorten the computing time. Fig. 5 includes light sources on "*Holiday*" project version sequences.

3.5. Animating the scene

Blender easily makes frame-frame animation by adding a timeline and many functions to set a key frame. For example, consider the last action in the "*Holiday*" sequence where the child loses the toy bought in the first sequence. The toy will fly through the air and reach the sea.

The toy must move on *x* and the *y*-axis (not on the *z*-axis so it does not exceed the background), it will rotate and it will appear smaller and smaller. To do this, we will set up a start-up key frame outside the filmed area and one in a shot taken by the camera. Key frame 1: at first, the object has a *1* scale, a *0* rotation on the *z* axis, and the *x* and *y* coordinates, in this case, outside the camera view. The second frame key is for the camera at the specified coordinates *x* and *y* with a *0.23* scale and *z -342* rotation axis. This key frame should be placed as far away from the other key frame to increase quality. These key frames appear as "*LocRotScale*" in the Blender menu, but there are other options, such as "*LocRot", "Scale"* and *"Visual Location".* In the end on check everything to be correct before the rendering action. The rendering stage will take quite some time.

3.6. Rendering

At this stage most of the animation is ready. At the rendering stage, the computer calculates all the textures, light and frame motion. The result of the rendering action is to get successive multiple images that later will allow the animated drawing as a video file. One can work sequentially on specified images. There are many online rendering sites [3,4]. Fig. 6 includes the example for the cartoon sequence "Holiday" after the rendering action.

3.7. Post-processing

The final step of creating a *2D Cartoon* is obtaining a video file. After the image strip is done rendering, use: "*Post Processing*" item in the "*Render*" menu of the *"Properties"* module and check "*Compositing*" and "*Sequencer*". In the "*Timeline*" module add image strip, choose the video format from the "*Render*" menu and the folder where to save and press the render animation button. Blender's official website [1] shows free of charge tutorials, examples and includes projects to involve users.

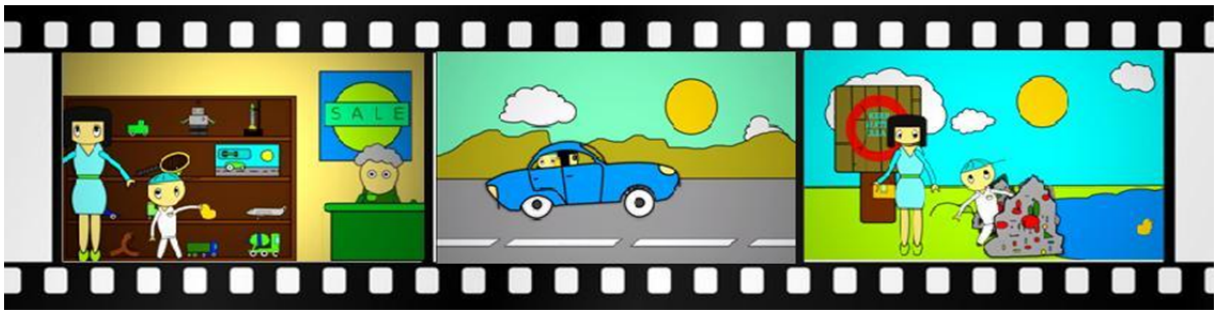


Fig. 6. Original cartoon sequences from "Holiday" project after the rendering action [5].

4 Conclusion

Creating animation with Blender is both useful and interesting. The saved scenes from the animations can be reused in similar projects. The differences between *2D* and *3D* animation are the time and methods used to make cartoons look good. The rendering time also differs from one to the other; the longest cartoon renders for about several months and small projects finished rendering in several hours on a single computing unit. The main challenge was animating the scene while dealing with the malfunctions *Blender*, as the parts of the characters were not always moving with the characters and therefore had to be animated separately. In the future more animations projects will be made with a highly improved quality.

References

[1] Blender official website. Online at: https://www.blender.org.
[2] History of animation Wiki website. Online at: https://en.wikipedia.org/wiki/History_of_animation.
[3] *SheepIt. A free distributed renderfarm for Blender.* Online at: http://www.sheepit-renderfarm.com.
[4] E. Larson: *5 Easy-to-Use Animation Tools for Beginners*, 21 May 2013. Online at: http://mashable.com.
[5] A. Pintea, *Models 2D in Blender*. Proceedings of the Conference of Programming, Communication, Imagination, Design (PCID) 2017 University of Sibiu. 2017, in press.
[6] A.Pintea, *Flying Bird* project version in Blender. Online at: https://github.com/AlexPintea/flying-bird.

ALEXANDRU PINTEA
National College E.Racovita
Mathematics & informatics
9-11 Kogalniceanu, Cluj-Napoca
ROMANIA
E-mail: al_pintea@yahoo.com

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

ADiagnostic

Rînea Iulia Georgiana

Abstract

The widely dependence on mobility has became nowadays a priority for people. The automotive industry has lately grown impressively a lot. Most of the people give cars an important place in ranking their priorities, and this is the reason of developing an Android application for mobile phone, to help resolve certain car problems, which often occur. This application is divided into three categories that facilitate the elimination of car damages. The first part is composed of symptoms, which users visual or auditory notice (observing a particular engine noise, unusual smoke detection). This part helps the car owner to repair the vehicle with only basic information in this domain. The second part is composed of a Google Map, which shows all the car services in a specific area. This part of the application is dedicated to serious problems of the vehicle, which can not be detected and which can damage the car. The third part consists of connecting the phone to the onboard computer through a device OBD II ELM-327 Bluetooth that allows certain functions to appear on the mobile screen display (speed, air intake, fuel level). In conclusion, I believe that this application is useful, because it helps to clarify car key issues, contributing in this way also to solving the environmental problems regarding polution.

1 Introduction

The "Diagnostic" application is developed to simplify drivers problems that may occur in traffic. This application is developed for the Android OS operating system, and this choice has been made following the frequent use of mobile phones to solve various problems. I chose this language because my interest in operating the mobile phone in the form of a useful device has always increased and I attended a course in the school and this is why, i created a passion for developing programms in this language.

Analyzing the information around me, I tried to establish the priorities of people nowadays and came to the conclusion that two of them are the mobile phone and the cars.

The application is therefore designed to detect vehicle malfunctions. Often these problems are encountered while the car holder is driving, which causes great discomfort. In these cases, the application will prove useful, because it helps the user to make a brief diagnosis after the symptoms they notice at the car, discovering the reason of their problem.

For example, he notices that the engine sounds or knocks in an unusual way, and looking for the problem in the application, he realizes that it has occurred due to poor quality or impurity in fuel, and investigating the last fuel supply of the vehicle, he realizes that this is the problem he has and he can solve it by himself, with only basic information. He can choose the fuel model and car brand, and then a search bar appears, where he can enter the detected symptoms. He must choose the two specified criterias, because every car brand has some specified malfunctiones, which differ from car to car. Premature detection of malfunctions can prevent advanced vehicle damage and can also help to keep the environment cleaner through less pollution. In Chapter 4 we have presented some types of malfunctions that may occur and can cause increases in emissions that the proposed application might identify in the future.

Problems can also be very complicated, so the only option is bringing the car to a car service. If he is located in an unknown area and does not know where to call and if the machine is having serious failures, then the only option is to search for car services in that area and select those which are most convenient for him. Again, the app will make life easier by offering all the services within a radius of 10km. With the help of Google Maps, we were able to create the map, and with the help of specific features available to application developers, we were able to display the Marker service, which indicated the car services. Each of the services is accessed as an HTTP request, and returns either an JSON or XML response. All requests to a Places service must use the https:// protocol, and include an API key.

The Google Places API Web Service uses a place ID to uniquely identify a place. After the marker has been set, the user can click on it and some option will appear on the display.

The third part of the app tries to offer something unique to car owners, namely the possibility of communicating with the car. The only way of communicating with the machine they knew was through the mechanic, who through some devices managed to connect the PC to the car's computer. Now, with this application, they have the chance to do this on their own with the help of their mobile phone. With an OBD II (on-board diagnostics) device, the ELM-327, it is possible to connect this device to the mobile phone via Bluetooth. There are many sensors throughout the car: oxygen sensors, engine knock sensors, manifold pressure sensors and so on and so on. Each one of these sensors sends a signal to the car's computer the Engine Control Unit (ECU). The ECU uses that information to adjust different elements of the engine operation, the fuel injection or the spark timing for example. If the information that the ECU gets from one of its sensors is too far out of whack, it saves a code called a Diagnostic Trouble Code (DTC), which are shown by the application. Through this OBD interface, it is permissible to charge certain mobile computer parameters to the mobile phone, such as speed, oil level, air intake, fuel type, etc. The general workflow of the application functionality should go like this:

1) connect to the OBDII adapter through Bluetooth;

2) initialize OBDII adapter with AT commands;

3) continuously get data from the vehicle through issuing the corresponding PID(Parameter ID) codes.

2 Interface

The first access to the functions of the application is realized by a menu, containing 3 buttons, representing the three parts mentioned in the introduction, which can be seen in Fig. 1.

Fig 1. Main Interface

Fig 2. Diagnosis Button

This button allows identification of a malfunction after the symptoms reported by the user.

Fig 3. Loading map button

This button allows the display of car services according to the user's location.

Fig 4. OBD-II connection button

This button allows the phone to connect to the OBD II interface.

3 Application software design

3.1 The programming environment

This app was created with the Java-based, Android object-oriented language, created in the Android Studio 2.2.2 programming environment. For compatibility, we used the Android SDK (Software Development Kit), which is needed to develop and test the application. It has been tested both on the phone and on the virtual device, Android Emulator, which can be created in this programming environment [1], [2]. For testing the third part of the app, the one with the OBD II, it was necessary to use a real car, to have acces to the on-board computer. The application requires two API Keys, one for map implementation and one for Google Place API implementation. We got the two APIs through Google Console Developer through creating the project and adding the key obtained by registering it.

3.2 Code sections

3.2.1 Diagnoza Activity

In this section, you can choose the fuel type of your machine and its brand, which are passed as parameters for later diagnosis.

```
spinner1.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
   @Override
   public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

      if (position == 1 || position==2) {
        spinner2.setAdapter(adapter2);
            iv.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
            new Handler().postDelayed(new Runnable() {
       @Override
       public void run() {
       String mStr=spinner2.getSelectedItem().toString();
       Intent mIntent = new Intent(DiagnozaActivity.this, Search.class);
            mIntent.putExtra("Value", mStr);
                startActivity(mIntent);
                }, 1000)}}); }
```

The following figure shows a diagnosis for an Alfa Romeo type Diesel engine:

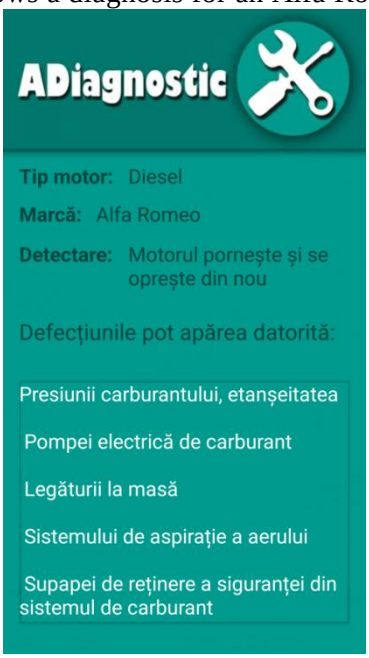


Fig 5. ECU malfunctions

To set the search function, it was created a method to call up existing features in the Android Search libraries.

```
private void setupSearchView() {
   mSearchView.setIconifiedByDefault(false);
   mSearchView.setOnQueryTextListener(this);
```

```
mSearchView.setSubmitButtonEnabled(false);
mSearchView.setQueryHint(getString(R.string.cheese_hunt_hint));}
```

3.2.2 Maps Activity

To set up the map, the functions below are called, creating a variable in where the map is saved (googlemap), and then will be set up the style (json_style) into resources. All functions will be called using the GoogleMap variable (**mMap**) [3], [4].

```
mMap = googleMap;
MapStyleOptions style=MapStyleOptions.loadRawResourceStyle(this,R.raw.json_style);
mMap.setMapStyle(style);
```

In the next section of the code, I will present the usage of GPA over a certain radius (prox_rad =10000)

```
googlePlacesUrl.append("&radius=" + prox_rad);
googlePlacesUrl.append("&key=" + "AIzaSyATuUiZUkEc_UgHuqsBJa1oqaODI-      3mLs0");
return (googlePlacesUrl.toString());
```

The picture below shows a location map relative to the driver's position that has a malfunction.

Fig 6. Services on map

3.2.3. OBD Activity

The Bluetooth initialization for connecting the onboard computer to the OBD II device [5] is done as follows:

```
//initializare Bluetooth
final BluetoothAdapter btAdapter = BluetoothAdapter.getDefaultAdapter();
if (btAdapter != null)
    bluetoothDefaultIsEnable = btAdapter.isEnabled();
```

Function to start GPS and Bluetooth:

```
private void startLiveData() {
    tl.removeAllViews();
    doBindService();
   // pornirea comenzii
    new Handler().post(queueCommands);
     if (prefs.getBoolean(ConfigActivity.ENABLE_GPS_KEY, false))
        gpsStart();
    else
  gpsStatusTextView.setText(getString(R.string.status_gps_not_used));
   wakeLock.acquire();
if (prefs.getBoolean(ConfigActivity.ENABLE_FULL_LOGGING_KEY, false)) {
   prefs.getString(ConfigActivity.dfulllog_key,
      getString(R.string.defaultdirname))
          ); }}
private void stopLiveData() {
gpsStop();
doUnbindService();
releaseWakeLockIfHeld();
  if (writec != null) {
      writec.closeLogCSVWriter();}}
```

In the illustration below you can see the parameters provided by the OBD II, which is connected to the machine

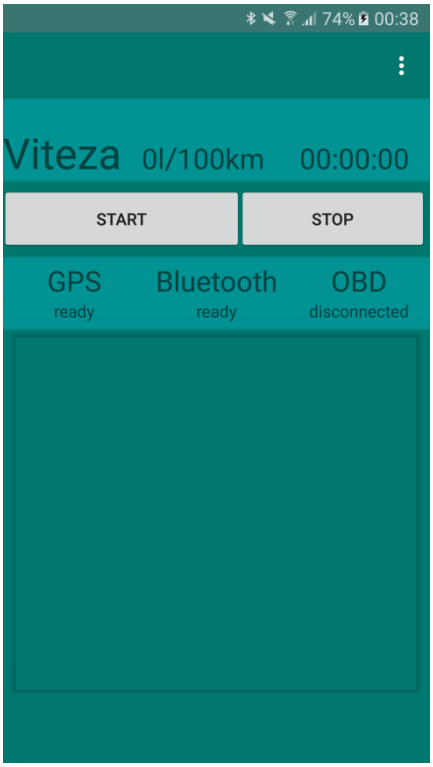


Fig 6. OBD II result

3.2.4 Permissions

To access the different bluetooth, internet, location, and other options, a series of permissions were required that are set in the manifest project file, namely:

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="myapplication.myapplication.MAPS_RECEIVE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="com.android.vending.BILLING" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.BLUETOOTH_PRIVILEGED"/>
<uses-permission android:name="android.permission.READ_LOGS" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

4 Future development

By getting the application up to this level, I'm still thinking about designing more and more future improvements. I will try to develop this application in favor of environmental protection. Cars are one of the negative factor, which cause it, and such an application can not only develop the user's knowledge in this domain, but can also reduce the emission of many inert exhaust gases that can cause environmental disaster. By doing this, I will try to create a server through which the user can connect, always providing feedback when it solves a malfunction with one's own application, making the user aware of his contribution to the quality of the environment. With this information I could make some statistics to see how much it can protect the environment. On the other hand, the big change to the application, which I want to do is to make a diagnosis: the phone analyzes the on-board computer, giving accurately the defects of the car, the user being exempt from paying the diagnosis of the car services. I think that the application can help both, the environment and the economy.

5 Thanks

We would like to thank Mr. Rinea David from SC Lativicom SRL and Dr. Lucian Roman from SC Autohaus Huber SRL Sibiu for the technical advice.

Acknowledgements: This work was supervised by Professor *Delilah Florea*, from ”Samuel von Brukenthal” National College and Professor PhD Adrian Florea, from Hermann Oberth Faculty of Engineering, *“Lucian Blaga” University of Sibiu*.

References

[1] Riley, D., 2012, *Using Mobile Phone Programming to Teach Java and Advanced Programming to Computer Scientists*, Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, pp. 541-546, North Carolina, USA.
[2] Android Developer Tools, 2014, https://developer.android.com/studio/intro/index.html, accessed 02 March 2017.

[3] Google Developers (GPA services), https://developers.google.com/places/, accessed 02 March 2017.
[4] Stack Overflow, http://stackoverflow.com/questions/26562848/how-to-integrate-google-maps-in-android-app, accessed 02 March 2017.
[5] Informații OBD, despre interfață și funcționare, http://blog.lemberg.co.uk/how-guide-obdii-reader-app-development, accessed 02 March 2017.
[6] A. Florea, L. Berntzen, M.R. Johannessen, D. Stoica, I.S. Naicu, V. Cazan, *Low Cost Mobile Embedded System for Air Quality Monitoring: Air quality real-time monitoring in order to preserve citizens' health*, submitted for The Sixth International Conference on Smart Cities, Systems, Devices and Technologies, SMART 2017, June 25 - 29, 2017- Venice, Italy.

IULIA GEORGIANA RÎNEA
"Samuel von Brukenthal" National College
Mathematics-Informatics Specialization,
intensive Informatics
Piața Huet no. 5
550182 Sibiu, ROMANIA
E-mail: iulia_rinea@yahoo.com

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

Virtual Pathfinder

Săndică Robert, Bota Petrică

Abstract

The purpose of this article is to demonstrate that augmented reality (AR) can be used in everyday activities to enhance the user's life. Even though there are multiple apps that help and guide the user to reach his desired destination we consider that we can improve the experience using AR. For this purpose we created an AR app, Virtual Pathfinder, that helps the user to reach his destination faster and easier by overlapping virtual cues (that indicate the path to follow) on top of the real world. In order to run the app the user must use a Smartphone with a good Internet connection. One of the most difficult problems that we have encountered was given by the impossibility of using the maps from Google. Due to the license agreement, it is not allowed to use Google products for creating another app base on Google services. Therefore, we created a special map in order to demonstrate the principle of our application.
For mapping the real world coordinates to geographical ones we introduced a formula to determine the right approximation of three different points in plane. We used this formula in Unity engine.
We have observed that by using our app principle, users are able to arrive at their destination without making any bad turns during their travel.

Key words: virtual reality, pathfinder, smartphone

1 Introduction

Nowadays Augmented Reality (AR) has become a technology increasingly used to enhance the experience and interest of tourists in visiting different objectives, of readers form libraries or book stores, of virtual game players, etc. AR has become more accessible with the evolutions of cameras and improvements of displays.

AR systems created overlap 3D virtual objects over the real world through the use of a quality camera and screen.

Augmented Reality uses the smartphones camera input and overlaps the virtual world over the feed received from the camera. There are two types of AR: one that uses a marker to identify the position where to overlap the camera feed and the virtual space, and another type of AR based on the phones gyro called marker less. The main purpose of AR is to enhance the real world image through the device; more information about AR principles can be found in Augmented Reality: Principles and Practice [2].

Originally, in order to run an AR app it was necessary to have real life markers. These markers were recognized by the software and 3D virtual objects would be placed over said marker (Fig. 1). With the advance of technology we are able to create and place 3D objects with the software alone making the real life markers unnecessary [5].

Fig.1 presents an augmented reality app based on markers that represented a virtual book that allows the user to interact with the virtual elements in an augmented way by pressing virtual buttons and observing the environment at free will by moving the smartphone. The marker that is used in the app is the image representing dried soil, over it, all the rest of the objects found in the figure are overlaid. The most important part in this app are the virtual buttons, that can be pressed as seen in the picture by hovering the hand over the marker image in the real world, the result of pressing the virtual buttons is the change of the picture and the details showed in the upper part of the figure.

Fig. 1 AR with physical marker
Source: Screenshot of smartphone

The most popular AR app to date (May 2017) has to be Pokémon Go. In this game users walk around the real world in order to catch virtual creatures represents the first big success of an AR app.

Pokémon Go got more than 650 million downloads and over 40 million daily users at its peak [4]. Though it does not have the same popularity and it has lost many users Pokémon Go proved that people have a big interest in discovering and using augmented reality apps.

In this article we are using Augmented Reality to improve the way of navigation by foot or by vehicle.

The rest of the article is organized as follows, in Section 2 we present the existing technology for the navigation. Section 3 contains our main result: virtual path finder app. In the last section of the article conclusions and the future directions of work are summarized.

2 Existing technologies

In the past decade the way in which we navigate in order to reach our destination has changed drastically. With the rising popularity and accessibility of the Internet and of the Smartphone we are able to use these tools to our advantage when it comes to arriving at our destination. Long past are the days when it was necessary to buy or print a physical map (that was probably outdated), or when getting lost was part of the journey. At the moment, with a few clicks or taps we can receive the correct, up-to-date

information (in real time) and how to use it.

When this article was written, the navigation application that had the most users was Google Maps with over 1 billion people using it monthly [3]. Observing the trend of fast and optimized way of traveling, it is natural to assume that getting from point A to point B fast and safe is a necessity. A lot of people are interested in how to use the existing technology to improve their results. Thus Google maps or other GPS navigation system become more and more used.

Google Maps holds many tools designed to help the user reach his destination faster. Such as 2D or 3D maps (Google Street View) to audio indications, real-time GPS tracking or even an electronic compass that works on your device. Despite all of these tools the system is far from perfect and we consider that with the help of AR we can improve on the current tools and help the user navigate faster and more effectively.

In the area of augmented reality, Hololens the device created by Microsoft is the most advanced in terms of projection and manipulation of the virtual environment projected over the real world. It uses both marker and marker less principles of augmented realities.

3 Main results Virtual Pathfinder app

3.1. App Description

Virtual Pathfinder is an Android app that uses the Smartphone's camera and GSP location in order to assist the user with navigation. At the moment, the app is intended to be used by a person travelling on foot. Once the user inserts the point of destination into the app, Pathfinder will draw a route for the user to follow and update his position to said route accordingly. So far everything is similar with what most navigation apps provide. Pathfinder differences itself by providing the user with graphical cues by the form of virtual signs. These virtual signs appear either under the form of a virtual line that the user must follow or as an arrow that points the user in what direction they should go.

The principle of our app is similar to that of a compass. A compass indicates the direction by drawing a straight line from your location to your destination. Pathfinder also draws a line from your location to your destination but it indicates the route by mapping it accordingly to the roads available. Therefore, it is like to a compass that guides you step by step from one intersection to another.

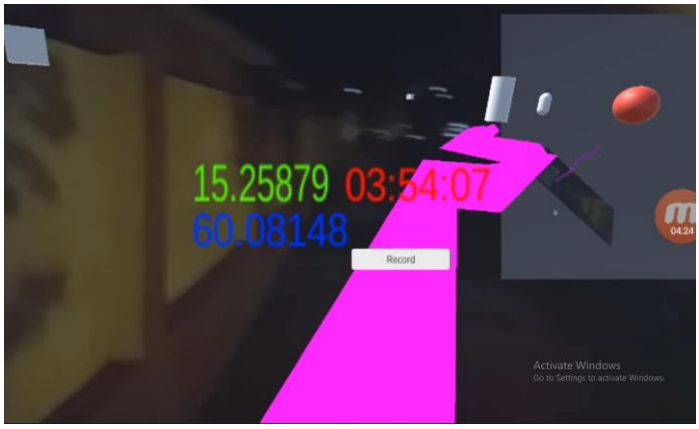


Fig. 2 Guiding line from Virtual Pathfinder

Virtual Pathfinder works in the following way: first it records a route that you have previously been on then it is able to assist you if you are required to take the same or similar route as well. The

reason why you need to first enter in the GPS data is that as of now, we were unable to access Google Maps' GPS data due to technical and legislative reasons.

The main advantage of Virtual Pathfinder is the AR part where the user needs to follow the line by orienting the device in the correct direction of traveling.

3.2. Technologies Used

Unity Game Engine

Unity is a powerful game engine [3] that allows the developers to code a single app for multiple platforms. It allows you to program the app in either Java or C# then you are able to export it to a multitude of platforms such as Android, IOS, Windows, Linux, etc.

Unity contains many libraries and tools that allow the programmer to create quality applications at an incredible speed. For a long period of time programming in-game physics (especially 3D) was a hassle to say the least. With the advent of game engines this is no longer the case.

Furthermore, Unity provides us with the frameworks necessary to access the device's camera, GPS chip and most importantly it allows us to overlap virtual objects on top of the real world camera feed. Therefore, it provides us with all the tools necessary in order to create an AR app that incorporates GPS data into its lifecycle.

Due to all of the tools that this game engine provides and the fact that it can export to multiple platforms, we considered Unity to be the most appropriate development environment for creating Virtual Pathfinder.

3.3. App implementation

As mentioned above Virtual Pathfinder is created in Unity with the help of C# scripts. First it is necessary to create a script that combines the augmented reality with the camera feed and the device's gyroscope. This script receives a 2D plane and prints the camera input into the plane. By accessing the device's gyroscope in the 3D environment we can measure if the device is aligned with the virtual objects or if it is not.

The gyroscope has three degrees of movement. We use the gyroscope to measures the device's tilt and orientation in the virtual plane. As a result the user is able to rotate his camera until the virtual objects/cues (virtual 3D plane) aligns with the camera's feed . Therefore, it is extremely easy and intuitive for the user to to navigate with the help of the Pathfinder app.

Next we use the device's geolocation in order to move the camera forward -backward and left-right into the 3D space by using the script presented in Fig. 3.

```
void UpdateLocation()
    {
    positionOfCameraInVirtualWorldRelativeToLongitudeFromGps =
100000*(Input.location.lastData.longitude - longit);
    positionOfCameraInVirtualWorldRelativeToLatitudeFromGps = 100000*
(Input.location.lastData.latitude - latitud);
    System.DateTime dtDateTime = new System.DateTime(1970, 1, 1, 0, 0, 0, System.DateTimeKind.Utc);
    dtDateTime = dtDateTime.AddSeconds(Input.location.lastData.timestamp).ToLocalTime();
    timeStamp = dtDateTime.ToLongTimeString();
  }
```

Fig. 3 Geolocation update function. Source: C# script

After the GPS connection is initialized and the start position is stored, we continuously receive the coordinates from the GPS location and get the difference from the original point of initialization to the destination point. Using this functionality we are able to store the current path for future usage as well as to follow predetermined paths.

We have tried to access existing maps such as Google Maps in order to have the route data backlogged. This process proved to be difficult due to the fact that Google does not make this information public, or if it does it is very limited. The solution of this problem was to create our own maps. That a bigger problem than our possibilities so we created a way of recording a path. To prove our app principle the recorded path was used to guide the user in his journey. Another problem was the mapping of the real word in geographical coordinates and the in to unity virtual world coordinates. Our proposed solution was to represent everything in a 2d plane instead of a 3d space. This has allowed us to get a better approximation of the location from the real world of the user and then projecting that location in unity virtual environment.

4 Conclusion and Future Developments

Virtual Pathfinder is an application that in still in development. Our goal is to improve Pathfinder so that it can be used easily and seamlessly by people around the world. We strive to make the application as reliable and as user-friendly as we can.

The biggest issue with Pathfinder is the fact that does not have access to a well-defined route database, such as Google Maps. To solve this issue we must either find a way to access Google Maps legally or to find another service that could provide us with the data required.

At the moment, the app updates the user's geolocation every 2 meters. While this is not necessarily a bad time frame we consider that this distance could be improved upon. Also, the geo-location is sensible to weather phenomena.

We desire to export the app on multiple platforms. As mentioned previously Unity simplifies this endeavor by providing us with all the tools necessary for multi - platform exporting.

We consider that people are willing to use AR apps if they integrate into our current society, and if they add a layer of value to the existing technologies those app are more valued than the rest.

Acknowledgement: This work was supervised by Professor PhD. Dana Simian, from "University Lucian Blaga from Sibiu"

References:

[1] Parisi Tony. (2015) *Learning Virtual Reality : Developing Immersive Experiences and Applications for Desktop, Web, and Mobile*, O'Reilly Media.
[2] Dieter Schmalstieg, Tobias Hollerer. (2016) *Augmented Reality: Principles and Practice*, Addison –Wesley.
[3] Google Maps user count (accessed 15.05.2017)- http://www.businessinsider.com/google-services-with-1-billion-users- 2015-10
[4] Pokemon Go user count (accessed 15.05.2017)- https://www.bloomberg.com/news/articles/2016 -08-22/these-charts-show- that-pokemon-go-is-already-in-decline
[5] Unity homepage (accessed 13.05.2017)- https://unity3d.com/

SĂNDICĂ ROBERT
"Lucian Blaga" University, Faculty of Science
Sibiu, Ion Rațiu streer, nr 5-7
ROMANIA
E-mail: sandica.robert@gmail.com

BOTA PETRICĂ
"Lucian Blaga" University, Faculty of Science
Sibiu, Ion Rațiu streer, nr 5-7
ROMANIA
E-mail: botapetre1@gmail.com

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

Interregional logistic management of transport system on the basis of the fuzzy relations of preference

Maria Sokolova

Abstract

In this article a variant of a method of a multicriteria alternative choice ELECTRE, which allows to account for uncertainties of expert estimates, was developed and studied. Modified method ELECTRE was tested on a problem solution of logistics management transportation system "point of departure – point of destination". The suggested approach can raise the effectiveness of logistic management of the goods delivery transport system and reduce delivery time.

1 Introduction

Annually losses from inefficient spatial structure of the Russian economy make 2,25-3,0% of GDP per year. Essential heterogeneity and irregularity of the Russian economic space have defined as one of the major priorities in the strategy of development for the country ensuring its unity, integrity and indivisibility of its space. In this regard special relevance is acquired by the analysis of development of Russia from positions of economic space as whole with the subsequent formation on this basis from groups of the united territories of its rational structure.

One of the most important components of interregional integration is the logistics - movement of goods between regions of the country. Activity in the field of logistics includes several directions, one of which is the transport logistics. Transport logistics problems include the creation of transport corridors, the choice of a transport type, the drawing up schedules and the route choice of freights transportation.

The analysis of various works on transport logistics has shown that mathematical models at the choice of a transport type are not learned well enough. Any numerical estimates of compliance of some type of transport to requirements of these criteria are in most cases limited to some set of criteria. Such approach doesn't consider first of all uncertainty of expert estimates, various physical and logical sense of the used criteria.

2 The choice of the method

The problem of the choice of a transport type for logistic system creation can be presented as a problem of the multicriteria alternative choice where alternatives can be: automobile, railway, air, water (river or sea) means of transport. As criteria usually we can use [1]: speed of cargo delivery, expenses on cargo delivery, frequency of departures, safety of freights, dependence on climatic conditions, reliability of observance of the schedule, ability to transport different freights, territorial availability.

Considering this enrollment, it is easy to see that only by the first three criteria can be exposed, though approximate, but numerical estimates for a concrete type of transport, on the others we can do either mark in some chosen scale, or verbal estimates, like "low, average, high".

Thus, there is an uncertainty in system of estimates of criteria compliance which has an obviously expressed not statistical character. Therefore the standard problem of the multicriteria alternative choice is transformed to a problem of the choice of the best alternative with not statistical uncertainty of criteria estimates.

For decision-making in the conditions of uncertainty usually we can use Wald, Gurvits, Bayes-Laplace's or Sevidzh criteria [10]. The theory of fuzzy sets will be the most adequate in this situation.

As it was already noted, criteria for evaluation of a transport type have various physical and logical nature therefore the correctness of integrated convolution demands additional proofs. In these conditions use of methods in which the choice of the best alternative is based on establishment of the relations of preference between alternatives, in particular ELECTRE[2] method is more preferable.

3 The main stages of the ELECTRE method

The main stages of the ELECTRE method can be tracked on the chart of the top level made in the notation of IDEF0 (Fig. 1):

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 (Fig.2);

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 (Fig. 3);

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 (Fig. 4);

5) the last base includes the best alternatives. The sequence of bases determines the orderliness of alternatives by quality (Fig. 5) [3].

As we understand the main stages of the ELECTRE method on the chart IDEF0, it is possible to find the solution of a problem of the choice of a type of transport. For the solution of a task it is necessary to give each type of transport a set of marks of criteria compliance, to define the weight of criteria and length of the corresponding scales. Considering that the specified parameters of a task are estimated in the expert way, they will be represented in an fuzzy form [4].

For this purpose it is necessary to create a term set of variables and the corresponding verbal and numerical scales (Table 1).

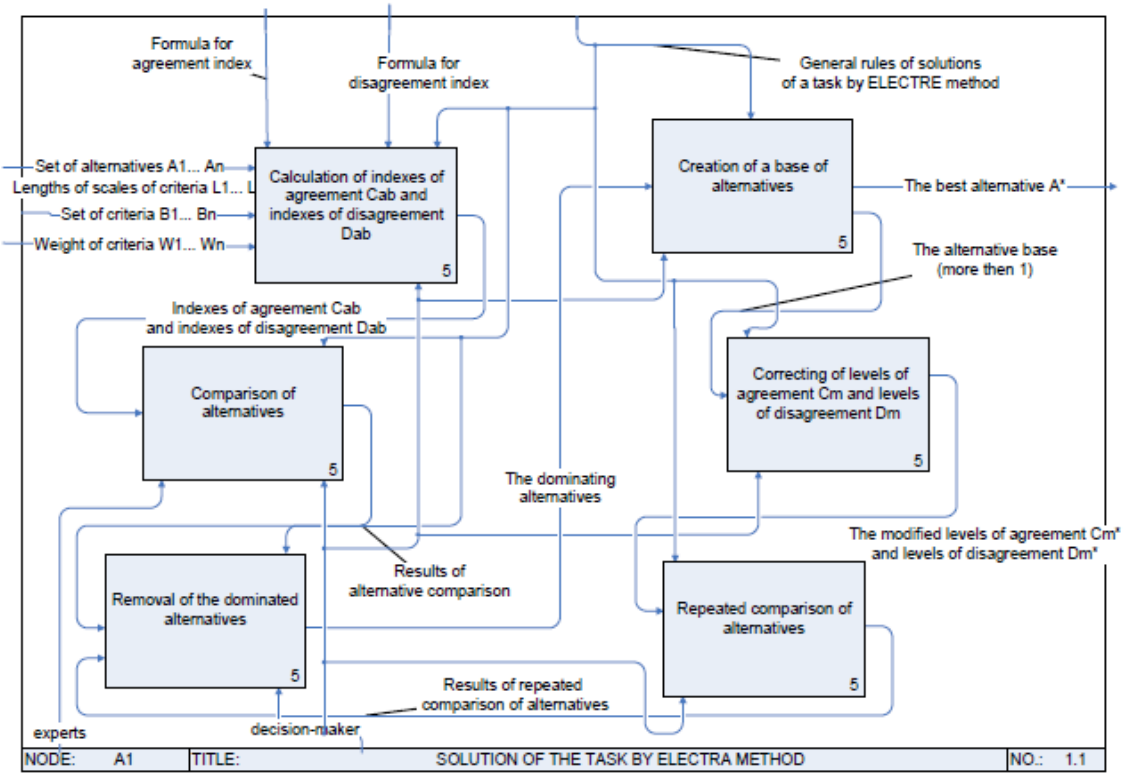


Figure 1 – Solution of the task

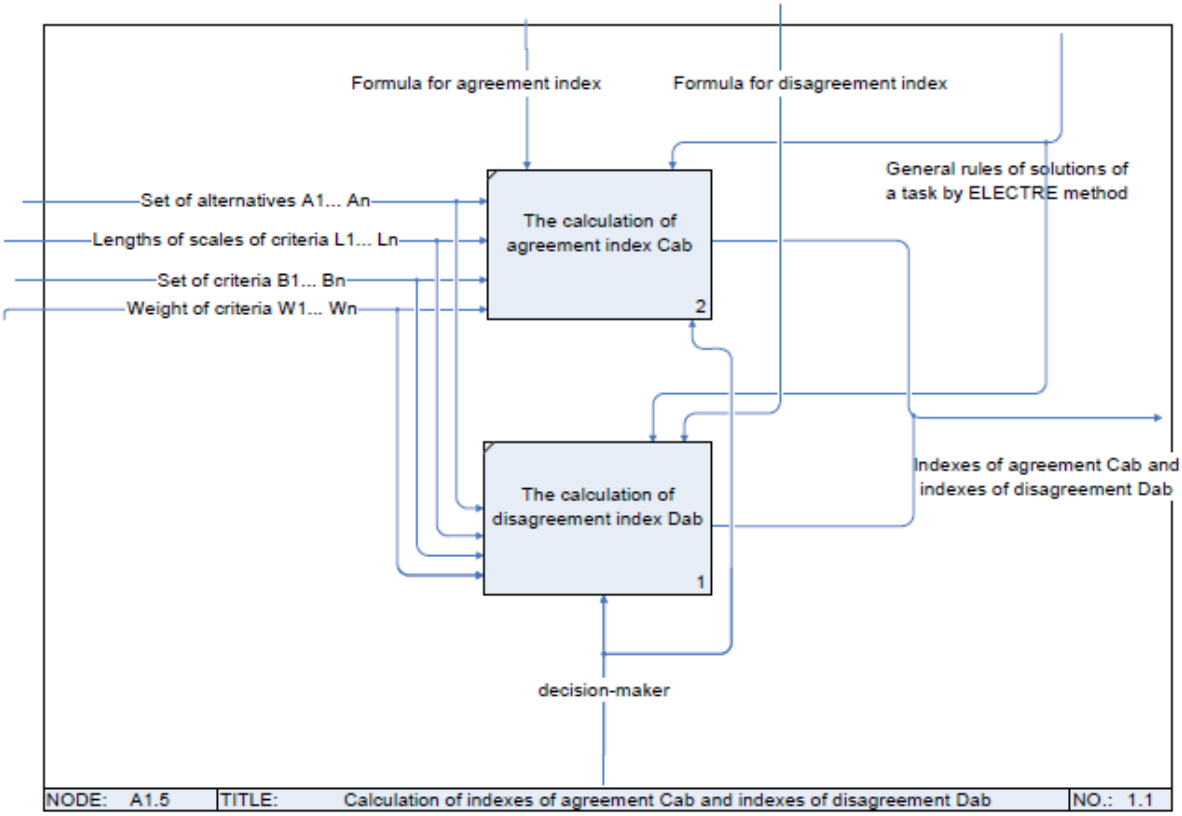


Figure 2 – Calculation of indexes

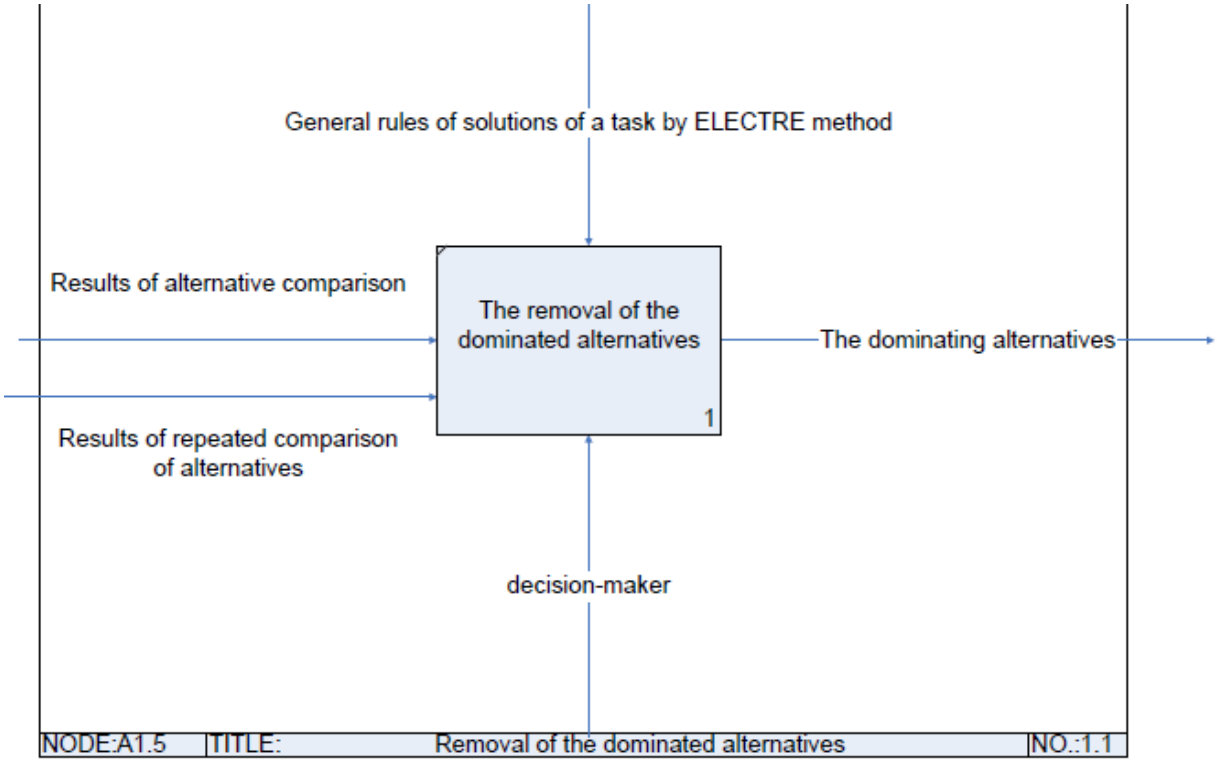


Figure 3 – Removal of the dominated alternatives

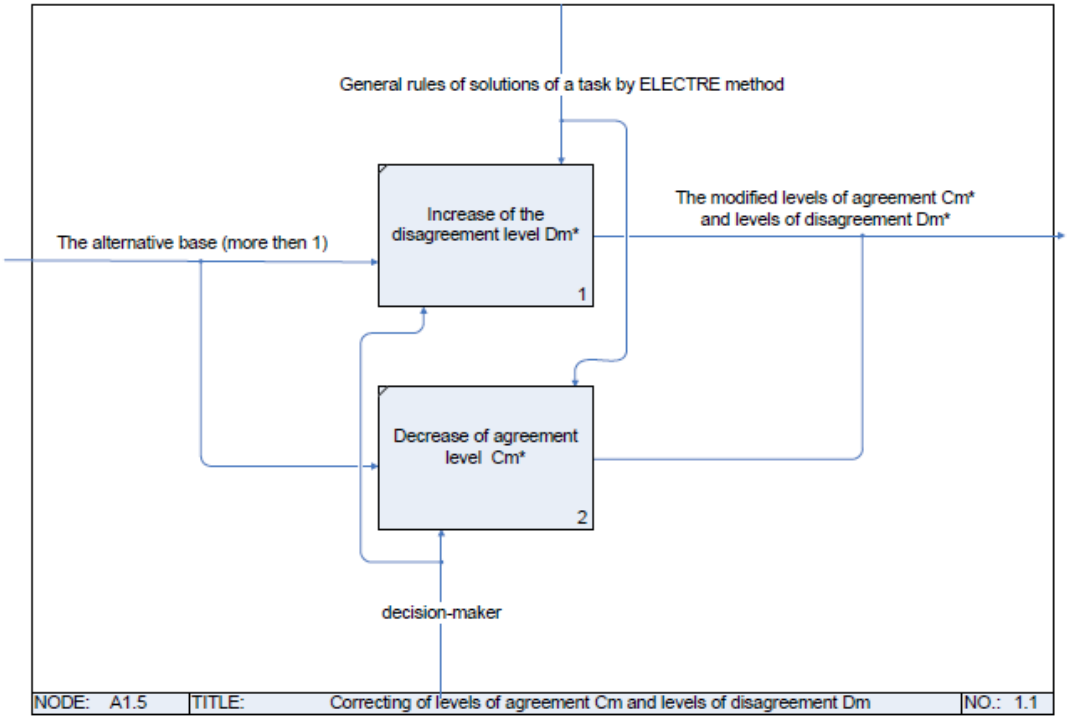


Figure 4 – Correcting of levels of agreement and disagreement

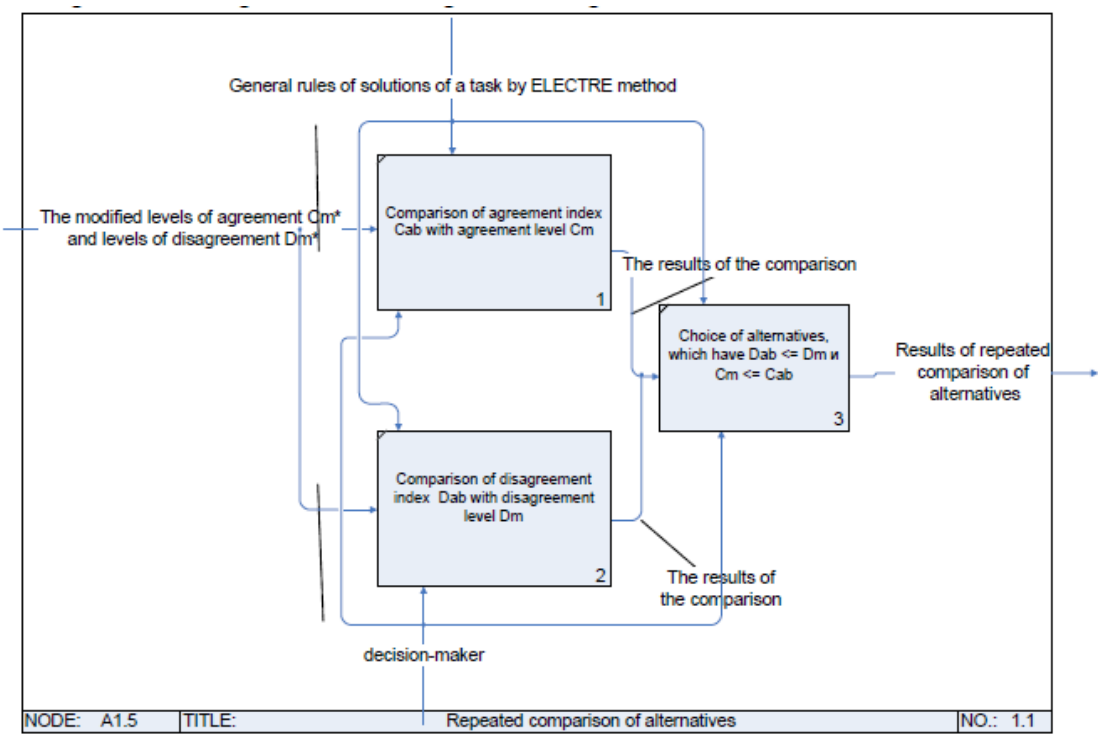


Figure 5 – Repeated comparison of alternatives

Table 1
Verbal - numeric scale of variables

Criteria	Linguistic evaluation	Numeric value
the speed of cargo delivery	Low	[0, 250]
	Average	[250, 500]
	High	[500, 750]
safety of cargo	Low	[0, 0,33]
	Average	[0,33, 0,66]
	High	[0,66, 1]
dependence on climatic conditions	Low	[0, 0,33]
	Average	[0,33, 0,66]
	High	[0,66, 1]
frequency of delivery	Low	[0, 33]
	Average	[33, 66]
	High	[66, 100]
costs of cargo shipping	Low	[0, 250]
	Average	[250, 500]
	High	[500, 750]
territorial availability	Low	[0, 0,33]
	Average	[0,33, 0,66]
	High	[0,66, 1]
reliability of observance of the schedule	Low	[0, 0,33]
	Average	[0,33, 0,66]
	High	[0,66, 1]
ability of transportation of different freights	Low	[0, 0,33]
	Average	[0,33, 0,66]
	High	[0,66, 1]

On the basis of the aprioristic information in this case taken from [5] criteria estimates of different types of transport which are represented in a linguistic form taking into account chosen a term sets (Table 1) are defined. As a result on each type of transport we will receive sets of linguistic values of estimates by criteria for different types of transport (Table 2).

Table 2
Table of estimates of options

	Automotive (A)	Railway (B)	River (C)	Sea (D)	Air (E)
the speed of cargo delivery (km per hour)	Low (35)	Low (15)	Low (15)	Low (17)	High (500)
safety of cargo (marks)	High (0,7)	High (0,8)	Average (0,5)	Average (0,5)	High (0,9)
dependence on climatic conditions (marks)	Average (0.6)	Low (0,3)	Average (0,6)	Average (0,6)	Average (0,5)
territorial availability (marks)	High (1)	Average (0,6)	Average (0,5)	Average (0,5)	Average (0,5)
costs of cargo shipping (rub per km)	Low (15)	Low (50)	Average (150)	Average (200)	Большие (500)
frequency of delivery (times per day)	High (70)	Low (30)	Low (10)	Low (10)	Low (10)
reliability of observance of the schedule (marks)	Low (0,3)	Average (0,5)	Average (0,4)	Average (0,4)	High (0,9)
ability of transportation of different freights (marks)	Average (0,4)	Average (0,5)	Average (0,4)	Average (0,4)	Average (0,5)

We will assume what experts have also chosen weight and length of scales, presented in Table 3.

Table 3
Criteria weights and scale lengths

Criteria	Criteria weight	Length of scales
the speed of cargo delivery	[7, 8, 9]	[450 500 550]
safety of cargo	[6 ,7, 8]	[0.5, 1, 1.5]
dependence on climatic conditions	[4, 5, 6]	[0.5, 1, 1.5]
territorial availability	[5, 6, 7]	[0.5, 1, 1.5]
costs of cargo shipping	[7, 8, 9]	[450 500 550]
frequency of delivery	[5, 6, 7]	[50 100 150]
reliability of observance of the schedule	[7 ,8, 9]	[0.5, 1, 1.5]
ability of transportation of different freights	[7 ,8, 9]	[0.5 , 1, 1.5]

4 The decision procedure

The calculation of the agreement and disagreement indexes is performed by using fuzzy table FuzzyCalc [7] (Fig. 6 and 7).

2		A	B	C	D	E
3	A	-	0,505917	0,867286	0,867286	0,361369
4	B	0,505917	-	1,01183	1,01183	0,596259
5	C	0,379438	0,144548	-	0,867286	0,361369
6	D	0,379438	0,144548	0,867286	-	0,361369
7	E	0,650465	0,560122	0,867286	0,867286	-

Figure 6 – The matrix of agreement indexes

17		A	B	C	D	E
18	A	-	0,3	0,1	0,1	0,93
19	B	0,4	-	0	0,004	0,97
20	C	0,6	0,3	-	0,004	0,97
21	D	0,6	0,3	0,1	-	0,966
22	E	0,6	0,9	0,7	0,6	-

Figure 7 – The matrix of disagreement indexes

We will set for the reviewed example levels of an agreement C and disagreement D: =0,582571, D=0,309467. Values, which are smaller then the level of an agreement C and bigger then the level of disagreement D, are rejected. Alternatives A, B, C, D, that is respectively automobile, railway, river and sea means of transport have entered a kernel of the dominating alternatives.

A certain disadvantage of the ELECTRE method is participation the person who is making the decision in formation of kernels. One more version of the solution of the considered task which is caused by existence of uncertainty in basic data and results (Fig. 8) is possible. As matrixes of coefficients of an agreement and disagreement can be considered as a matrix of the fuzzy values characterizing uncertainty of a situation, for an assessment of alternative decisions fuzzy entropy [8] which is determined by a classical formula of Shannon [9] can be used.

As entropy is an assessment of level of uncertainty, the best decision has to have the minimum value of entropy. Calculations confirm already received results – the best is the alternative B. Thus, two various versions of the solution of the same task have resulted in coinciding results that is in full accordance with methodology of the theory of stability according to which the result of data processing invariant concerning a method of processing corresponds to reality [6].

	A	B	C	D	E	max	Entropy
A	-	0,505917	0,867286	0,867286	0,361369	0,867286	1,50294
B	0,505917	-	1,01183	1,01183	0,596259	1,01183	0,998159
C	0,379438	0,144548	-	0,867286	0,361369	0,867286	1,72492
D	0,379438	0,144548	0,867286	-	0,361369	0,867286	1,72492
E	0,650465	0,560122	0,867286	0,867286	-	0,867286	1,36436

Figure 7 – Entropy calculation

5 Conclusions

On the example of the considered problem of the choice of a transport type for logistic system it is proved that the accounting of uncertainty in expert estimates allows to make a choice of a type of transport more reasonable. The received results give the grounds to speak about an opportunity of

practical application of the modified ELECTRE method in the conditions of uncertainty of expert estimates. Receiving coinciding results by various and independent methods confirms their reality.

In future it will be interesting to apply the results of this work in real cases and to create the automotive program to use the modified ELECTRE method for interregional integration.

Acknowledgement: This work was supervised by Prof., PhD Vladimir Chernov, from Vladimir State University, Faculty of Information Technology, RUSSIA.

References

[1] Lubentsova V. S. *Mathematical models and methods in logistics,* Samara: Samar. un-t, 2008.
[2] Roy B. *Multicriteria Methodology for Decision Aiding*, Dordrecht : Kluwer Academic Pulisher, 1996.
[3] Sokolova M.L., Chernov V.G. *The logistics management a transportation system «point of departure – point of destination» in real time regime*, Prague: Publishing Center of The International Scientific Association "Science & Genesis", 2014.
[4] Sokolova M.L., Chernov V.G. *The logistics management a transportation* Sibiu: Publishing Center of International Students Conference on Informatics ICDD, 2014.
[5] Tutygin A. G., V. B Boxes. *Advantages and shortcomings of a method of the analysis of hierarchies*, 2010.
[6] Deshmukh, K.C., Khot, P.G., *Generalized Measures of Fuzzy Entropy and their Properties*: World Academy of Science, Engineering & Technology, 2011.
[7] Chernov V. G. *Solution of business challenges means of indistinct algebra,* Moscow: Diagramm, 1998.
[8] Chernov V. G. *Entropy criterion of decision-making in the conditions of full uncertainty, Moscow:* Management information systems, 2014.
[9] Deshmukh, K.C., Khot, P.G. *Generalized Measures of Fuzzy Entropy and their Properties:* World Academy of Science, 2011.
[10] Zogal S.E. *Methods of decision making and expert choice*, Tambov: TGTU, 2010.

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

International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2017, May 25-27
Sibiu, Romania

Framework for Segmentation of Digital Images

Aleksandar Stojak, Eva Tuba, Milan Tuba III

Abstract

Digitalization significantly changed everyday life in many aspects, one of which is photography. Digital images are widely used in variety of fields. There are numerous sources from which a digital image can be produced and also a great number of applications for this type of data. Depending on the purpose of specific digital image it is important to clarify and emphasize all valuable data in order of reducing human error in evaluation and interpretation. That is achieved through various image processing methods, one of which is image segmentation. In this paper a framework for image segmentation is being presented. Frameworks scope of work is being tested on several benchmark examples using multilevel threshold with two most successful criterias: Kapur's entropy and Otsu's between-class variance. Framework's utility is presented through comparison of applied methods in terms of execution time, optimal number of threshold and objective function values.

1 Introduction

Digital images are widely used throughout great number of areas of human life, ranging from everyday life photography, in capturing important events, to recording and detection in numerous scientific fields. It has proven to be a valuable tool because it comes with wide range of benefits that are inherited from the digital era we live in. Digital image represents a set of digital values called pixels. They hold a numerical value which is the brightness of a given color. Every image comes with a fixed number of rows and columns of pixels and it is known as image resolution. Thereby, every digital image is a two dimensional matrix of numbers. Nowadays, every computerized machine that is able to record something, can give out the output in a form of a digital image. That is why digital images have many sources and can be utilized in different areas, such as astronomy, meteorology, forensics, biology, diagnostics, etc [1]. Depending on the source, digital image will have certain properties and will have anomalies of various types and amounts. The amount of valuable information that can be read from it correlates with subjective view of an observer and the quality of the recorded image. Since every digital picture has certain properties, regarding the source of it and the task at hand, having ability to work and adjust criteria to the particular task is very important. Framework is a flexible set of tools that can tackle generic tasks, while leaving the room for user-written code. Framework for image segmentation is capable of processing any digital image, regardless of the source. Being able to expand and modify tools in the framework, within given boundaries, makes it a good choice for all types of digital image manipulation.

2 Image processing

Since digital image is a two dimensional matrix of values, digital image processing is alteration of the values in the matrix. By increasing the value of all pixels, the whole image becomes brighter, while decreasing makes it darker. The change during either process depends on the number of pixels that are changed and the difference between the initial values and new ones. Image processing is often applied on

a group of pixels by certain criteria, which depends on the goal of the process. Those goals can be segmentation for feature highlighting, information extraction [2], denoising etc. Every recording and capture is prone to various types of unwanted change. It can be a product of interference from internal or external factors, and it can appear as noise or abnormality. Using processing of digital image for the removal of noise is always favorable before any other type of processing. Denoising is a type of enhancement that can ensure more clearer understanding and reading of the information contained in the digital image. Type of enhancement depends on the type of desired information, so the task of every enhancement technique is to provide an image which is better than the original in terms of fulfilling the task. Two categories of image processing is defined by the domain on which we are modifying the image: Spatial Domain and Frequency Domain.
Processing in spatial domain is alteration of pixels directly on the image, while frequency domain processing is transforming the image into mathematical model which will be altered, then reversed back into original but processed form. Most commonly, the mathematical model is Fourier transform. Histogram is widely used model for image processing. It represents a probability distribution of variables which in case of digital image are pixel intensity variables. For multilevel thresholding histogram information is used.

3 Segmentation

Segmentation is a process that alters a digital image in the areas of interest, making it more easier for reading and interpreting valuable information that it contains. There is no standardized technique for segmentation since there are numerous ways it can be achieved [3]. Depending on the case, every method of segmentation yields measurable results and can be modified to better suit the given purpose.
Most used methods for segmentation are intensity-based (thresholding), edge-based, region based and clustering. When using any of the segmentation techniques, there are two basic properties of gray-scale values that can be a base for segmentation process: discontinuity and similarity. Discontinuity is defined as sudden changes in gray-scale levels, which indicate isolated points or edges on an image. Similarity is a property that is naturally tending to be used for grouping and region based segmentation.
There are three basic properties that are used for segmentation: color, texture and motion [4]. Majority of segmentation processes use a combination of data from one or more properties.
The simplest segmentation technique is intensity threshold, which is based upon selection of a pixel value that will be a reference point for adding or subtracting from a desired region of segmentation. If a pixel value is greater or lower than the assigned threshold it is included or excluded from the interest group. This type of technique is naturally suited for application on image histogram. Multilevel threshold is frequently used technique that is based on multiple thresholds selected by using known criteria, such as Shanon's entropy, Tsallis' maximum entropy, Kapur's entropy, Otsu's between-class variance, [5] etc.
Region growing is grouping of pixel regions based on the predefined similarity criteria followed by spatial adjacency of pixels. It starts with selection of seed candidates from the area of interest. Properties from the seed pixels are used for similarity criteria when checking the neighboring pixels. The similarity criteria can be 1) the absolute intensity difference between checked pixels in a certain spatial range, 2) the absolute intensity difference between the average intensity of the growing region and seed candidate or 3) the difference between standard deviation of intensity in random neighborhood pixels and that of a specified neighborhood of the candidate pixel [4].
Split and Merge algorithm approach is based on homogeneity criterion. Starting with the whole image or the part of it, it is being split into smaller parts if it does not contain the given homogeneity. On the other hand, part that contains the given homogeneity is merged with the area that shares the same homogeneity value [3].

4 Multilevel Thresholding

Image thresholding is widely used segmentation technique which is based on alteration of an image by manipulation of the image histogram. The more defined thresholds are the segmentation of the image will

be better. Although the selection of favorable thresholds seems like a trivial task, the image histogram can contain a number of peaks, some emphasized more than others, but still crucial since the information contained in them can be lost due to simple picking of the most prominent ones [2]. Kapur's and Otsu's objective functions are both good criteria for determining the optimal number of threshold values in the benchmark examples. Although they might not be applicable in every case they are a good starting point for further processing.

4.1 Kapur's Method

For Kapur's method the threshold criteria is formulated as follows: Let an image I contain n pixels with gray levels in the set (0, 1, ..., L - 1). Let $h(i)$ present the number of pixels at gray level i, and $p_{\mathrm{i}} = h(i)/n$ be the probability of occurrences of gray level i in the image I. The subdivision of an image into k+1 classes can be used for the calculation of k optimal thresholds (t_0,t_1,...,$t_{k\text{-}1}$). The optimal thresholds are obtained by maximizing the objective function

$$f(t) = H_0 + H_1 + \ldots + H_m \tag{1}$$

where the entropies H_i are defined by

$$\begin{aligned} H_0 &= -\sum_{i=0}^{t_1-1} \frac{P_i}{\omega_0} \ln \frac{P_i}{\omega_0}, \qquad \omega_0 = \sum_{i=0}^{t_1-1} P_i \\ H_1 &= -\sum_{i=t_1}^{t_2-1} \frac{P_i}{\omega_1} \ln \frac{P_i}{\omega_1}, \qquad \omega_1 = \sum_{i=t_1}^{t_2-1} P_i \\ &\vdots \\ H_m &= -\sum_{i=t_m}^{L-1} \frac{P_i}{\omega_m} \ln \frac{P_i}{\omega_m}, \qquad \omega_m = \sum_{i=t_m}^{L-1} P_i \end{aligned} \tag{2}$$

4.2 Otsu's Method

Otsu's approach is the maximization of the between-class variance and it is described as follows: Let it be that an image I can be represented by L gray levels. The probabilities of pixels at level i are denoted by p_i, so $p_i \geq 0$ and $p_0 + p_1 + \ldots + p_{L\text{-}1} = 1$. Cumulative probabilities for classes A_i, i = 0, 1, ... , k can be defined as

$$w_0 = \sum_{i=0}^{t_0-1} p_i, \;\; w_1 = \sum_{i=t_0}^{t_1-1} p_i, \;\; \ldots, \;\; w_k = \sum_{i=t_{k-1}}^{L-1} p_i \tag{3}$$

where t_j are the thresholds between these classes. For k + 1 classes A_i, (i = 0, 1, ..., k) the goal is to maximize the objective function

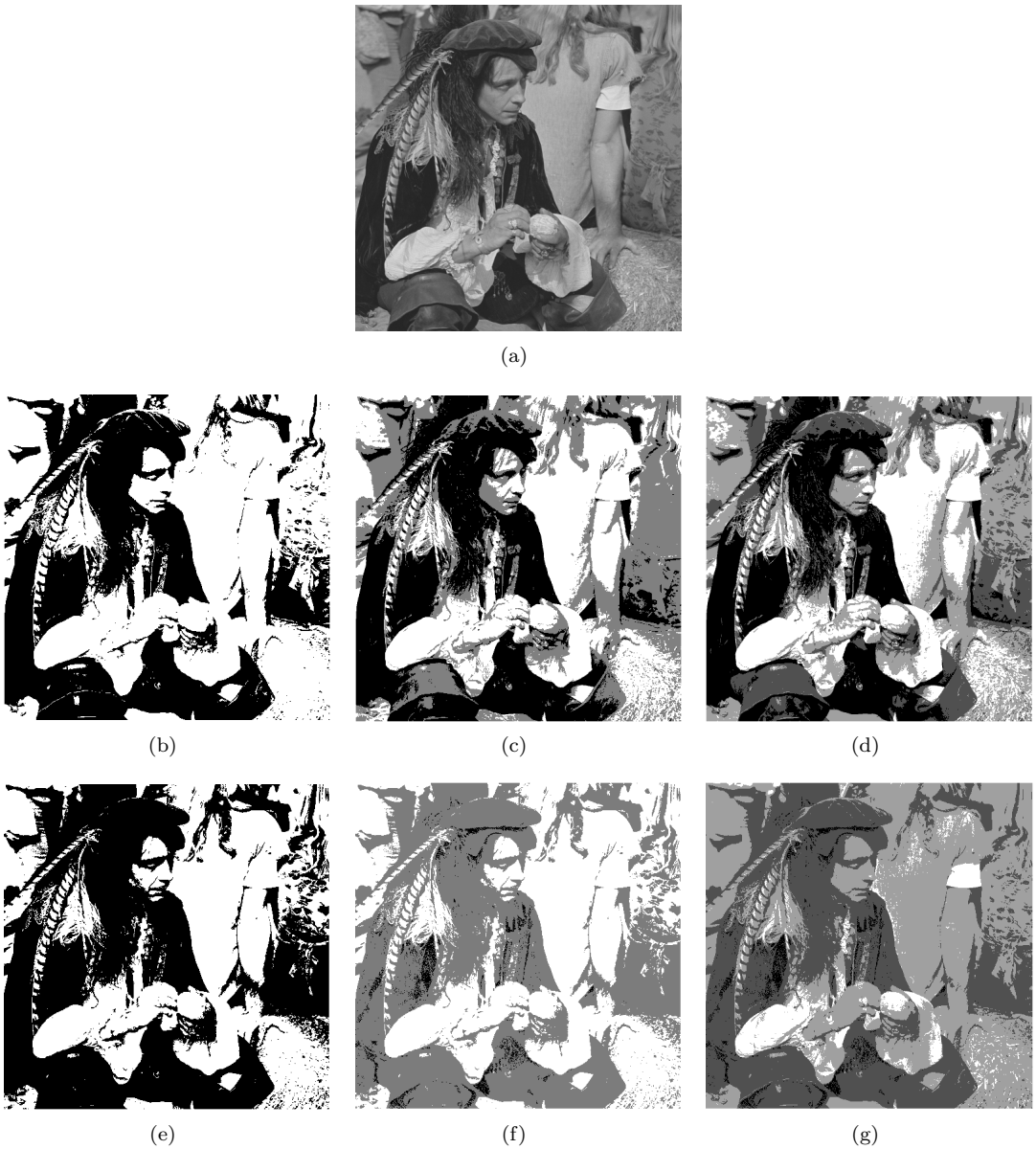
$$f(t_0, t_1, \ldots, t_{k-1}) = \sum_{i=0}^{k} \sigma_i \tag{4}$$

where the sigma functions are defined by:

$$\sigma_0 = w_0 \left(\sum_{i=0}^{t_0-1} \frac{ip_i}{w_0} - \sum_{i=0}^{L-1} ip_i \right)^2 \; \sigma_1 = w_1 \left(\sum_{i=t_0}^{t_1-1} \frac{ip_i}{w_1} - \sum_{i=0}^{L-1} ip_i \right)^2 \; \sigma_k = w_k \left(\sum_{i=t_{k-1}}^{L-1} \frac{ip_i}{w_k} - \sum_{i=0}^{L-1} ip_i \right)^2 \tag{5}$$

5 Experimental Results

The proposed framework for image segmentation of benchmark examples was implemented in Matlab R2015a. Experiments were performed on the platform with Intel R CoreTM i7-3770K CPU at 4GHz, 8GB RAM, Windows 10 Professional OS. Images are all of size 512×512. Different images were chosen to test the performance of the proposed framework.
Proposed methods for multilevel thresholding are Kapur's and Otsu's methods. They are applied separately for comparison of results in terms of optimal number of thresholds, execution times and objective function value. Test images are shown in Figure 1, Figure 2, Figure 3 and Figure 4. Optimal number of thresholds aswell as objective function values and execution times for both methods are shown in Table 1 and Table 2.



(a)

(b) (c) (d)

(e) (f) (g)

Figure 1: Segmentation of Pirate test image: (a) Original image, (b-d) Otsu's method ((b) 1, (c) 2 and (d) 3 thresholds), (e-g) Kapur's method ((e) 1, (f) 2 and (g) 3 thresholds). Source: *http* : *//www.imageprocessingplace.com/root/*$_{f}$*iles*$_{V}$*3/image/*$_{d}$*atabases.htm*

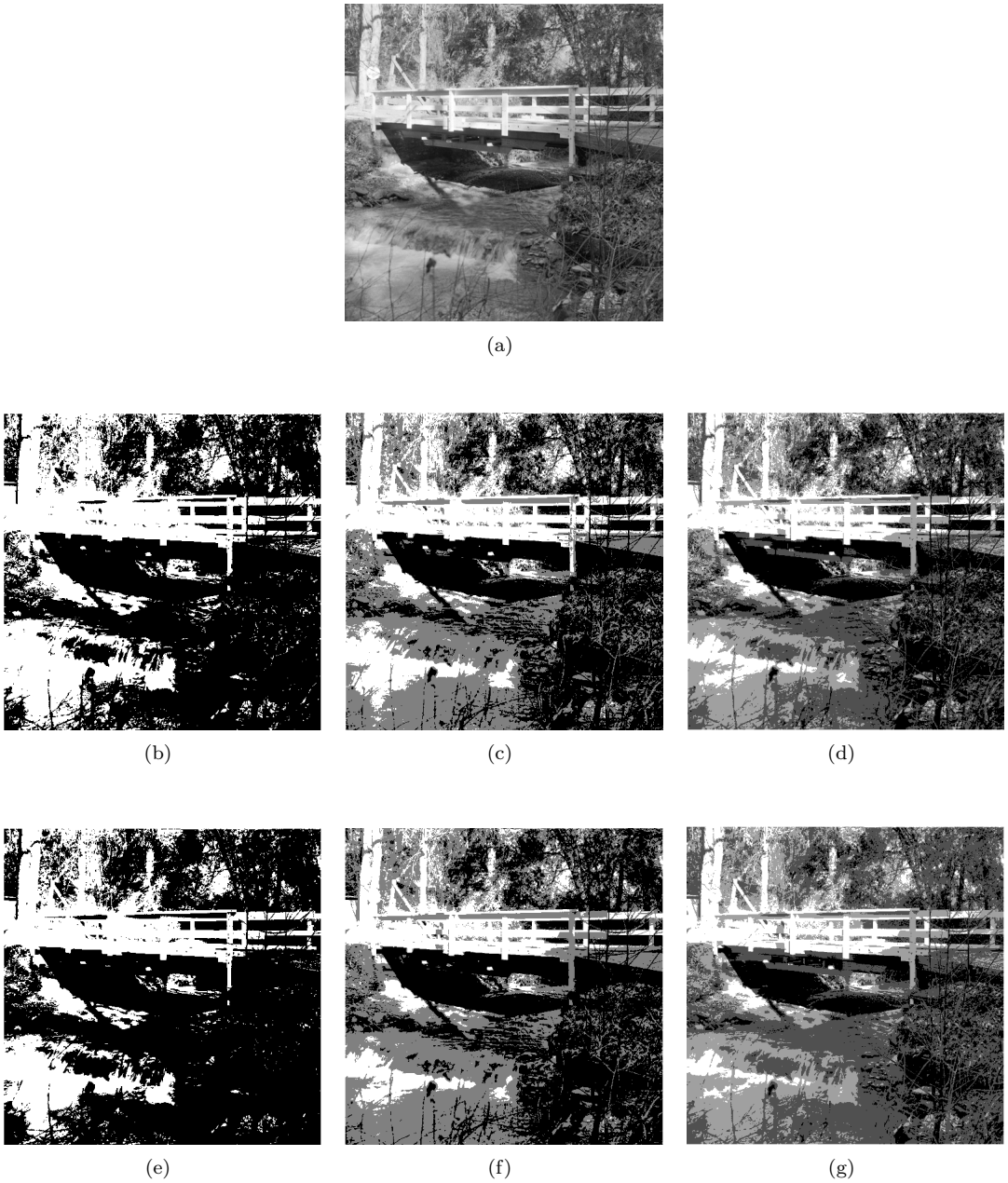


Figure 2: Segmentation of Bridge test image: (a) Original image, (b-d) Otsu's method ((b) 1, (c) 2 and (d) 3 thresholds), (e-g) Kapur's method ((e) 1, (f) 2 and (g) 3 thresholds). Source: *http* : *//www.imageprocessingplace.com/root/*$_{f}$*iles*$_{V}$*3/image/*$_{d}$*atabases.htm*

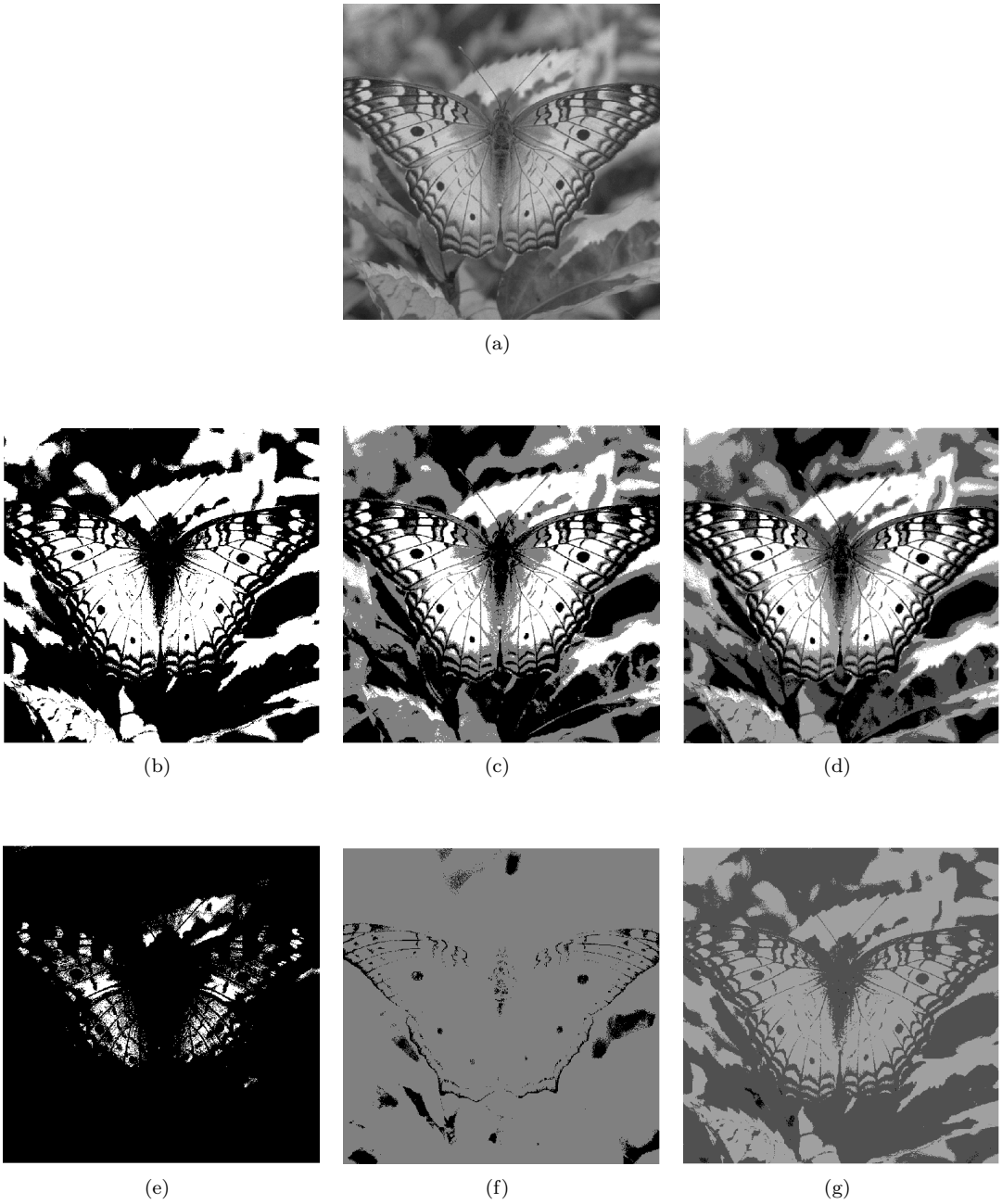


Figure 3: Segmentation of Butterfly test image: (a) Original image, (b-d) Otsu's method ((b) 1, (c) 2 and (d) 3 thresholds), (e-g) Kapur's method ((e) 1, (f) 2 and (g) 3 thresholds). Source: *http* : *//www.imageprocessingplace.com/root/*$_{f}$*iles*$_{V}$*3/image/*$_{d}$*atabases.htm*

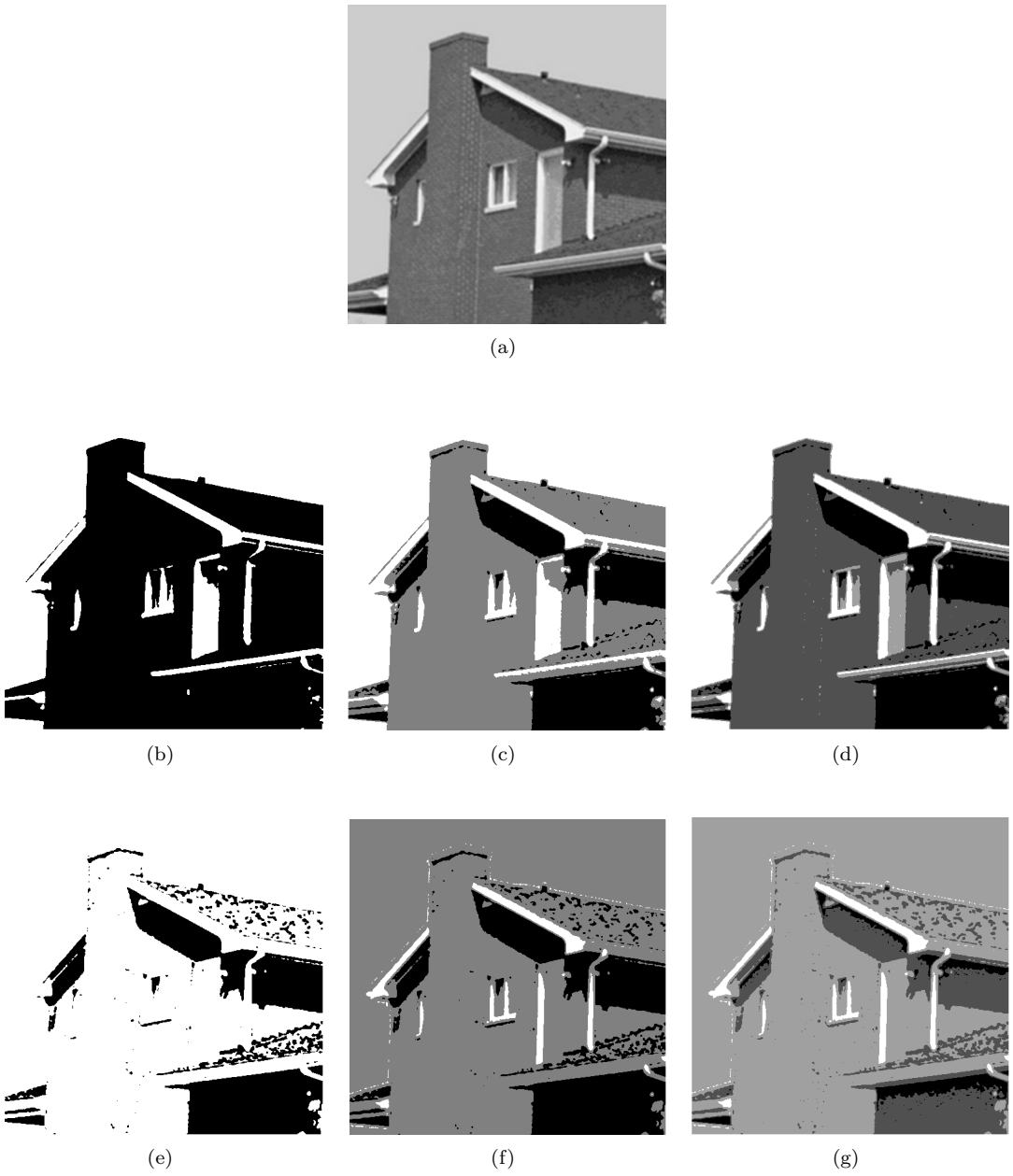


Figure 4: Segmentation of House test image: (a) Original image, (b-d) Otsu's method ((b) 1, (c) 2 and (d) 3 thresholds), (e-g) Kapur's method ((e) 1, (f) 2 and (g) 3 thresholds). Source: *http* : *//www.imageprocessingplace.com/root/*$_{f}$*iles*$_{V}$*3/image/*$_{d}$*atabases.htm*

Table 1: Objective function values, thresholds and execution time for Otus's method

Image	Objective	Thresholds	Time (s)
Pirate	1708.1202	108	0.0135
	1994.5267	86, 141	1.8566
	2115.3056	71, 114, 155	192.7879
House	2860.5239	148	0.0149
	3162.4930	83, 156	2.0422
	3234.8681	82, 131, 182	189.0305
Butterfly	1272.2718	122	0.0166
	1553.0734	98, 151	1.9671
	1669.2783	81, 118, 160	185.4724
Bridge	2048.3912	127	0.0146
	2503.1582	93, 159	1.9946
	2686.2754	75, 123, 179	205.6786

Table 2: Objective function values, thresholds and execution time for Kapurs's method

Image	Objective	Thresholds	Time (s)
Pirate	8.6528	125	0.0243
	12.2363	36, 125	1.9894
	15.5869	36, 112, 179	130.9157
House	7.4076	96	0.0239
	10.7304	95, 208	2.9741
	13.6167	47, 97, 208	227.0668
Butterfly	8.3460	213	0.0346
	11.6418	27, 213	1.9743
	14.8455	27, 120, 213	231.9157
Bridge	9.3720	139	0.0216
	12.8954	98, 172	1.9967
	16.1613	64, 126, 187	229.0295

What can be derived from the data is that Otsu's method is faster in term of execution time for finding one or two thresholds, but that time is not much impactfull since it is an approximate difference in hundreds of a second, while when the goal is three thresholds Otsu's method is significantly better in three out of four cases. What is observable from the Figures is that the images with Otsu's thresholds tend to be slightly darker but have more distinguishable segmented regions which results in less data loss. That is verifiable by threshold values in the Tables which are lower and less dispersed in Otsu's case.

6 Conclusion

In this paper a framework for image segmentation is presented. Using existing utilities, multilevel thresholding is applied and Otsu's and Kapur's methods are used for optimal threshold measurement. Using benchmark examples it is determined that the framework is capable of successful image segmentation and comprehensive analysis presented in comparison of both methods in terms of optimal threshold value, execution time and objective function value. Future work may include algorithm for optimization of search for optimal number of thresholds as well as improved interface.

Acknowledgment: "This work was supervised by Professor *Milan Tuba*, from *Faculty of computer science, University Jhon Naisbitt* ".

References

[1] Tuba, Eva and Tuba, Milan and Simian, Dana Handwritten digit recognition by support vector machine optimized by bat algorithm. *24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision,(WSCG 2016)*, 369–376, 2016. ,

[2] Tuba, Milan Multilevel image thresholding by nature-inspired algorithms - A short review. *The Computer Science Journal of Moldova*, 318–338, 2014. ,

[3] Stojak, Aleksandar and Tuba, Eva and Tuba, Milan Framework for abnormality detection in magnetic resonance brain images. *24th Telecommunications Forum (TELFOR)*, 1–4, 2016. ,

[4] Solomon, Chris and Breckon, Toby *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*, John Wiley & Sons, 2011. ,

[5] , Tuba, Milan and Bacanin, Nebojsa and Alihodzic, Adis Multilevel image thresholding by fireworks algorithm. *Radioelektronika (RADIOELEKTRONIKA), 2015 25th International Conference*, 326–330, 2015. ,

[6] Saad, Mohd and Abu-Bakar, SAR and Muda, Sobri and Mokji, M and Abdullah, AR Fully automated region growing segmentation of brain lesion in diffusion-weighted MRI *IAENG International journal of computer science*, 155–164, 2012. ,

[7] Rana, Rachana and Bhadauria, HS and Singh, Annapurna, Study of various methods for brain tumour segmentation from MRI images. *International Journal of Emerging Technology and Advanced Engineering*, 338–342, 2013. ,

[8] Nikolic, Marina and Tuba, Eva and Tuba, Milan Edge detection in medical ultrasound images using adjusted Canny edge detection algorithm. *24th Telecommunications Forum (TELFOR)*, 1–4, 2016. ,

[9] Korolija, Stefan and Tuba, Eva and Tuba, Milan An Algorithm for Medical Magnetic Resonance Image Non-Local Means Denoising. *International Journal of Signal Processing* 138–145, 2016. ,

[10] Saad, Norhashimah Mohd and Abu-Bakar, SAR and Muda, Sobri and Mokji, Musa Automated segmentation of brain lesion based on diffusion-weighted MRI using a split and merge approach. *Biomedical engineering and sciences (IECBES), 2010 IEEE EMBS conference*, 475–480, 2010. ,

ALEKSANDAR STOJAK
John Naisbitt University
Bulevar umetnosti 29
SERBIA
E-mail: *aleksandarstojak@gmail.com*

EVA TUBA
John Naisbitt University
Bulevar umetnosti 29
SERBIA
E-mail: *etuba@ieee.org*

MILAN TUBA III
John Naisbitt University
Bulevar umetnosti 29
SERBIA
E-mail: *milantuba@gmail.com*

LIST OF AUTHORS

Paul - Gheorghe BARBU
"Lucian Blaga" University
Faculty of Engineering
10, Victoriei Bd., Sibiu, 550024
ROMANIA
E-mail: barbu.paul.gheorghe@gmail.com

István BIALKÓ
Spiru Haret University
Faculty of Mathematics and Informatics
str. Ion Ghica, nr. 13, CP 030045, Sector 3, Bucharest
ROMANIA
Email: bialko.istvan@gmail.com

Petrică BOTA
"Lucian Blaga" University
Faculty of Science
Dr. I. Rațiu St., No. 5-7, Sibiu, 550012
ROMANIA
E-mail: petrica.bota2@yahoo.com

Cristian-Ionuț CAZAN
Petroleum-Gas University
iTIMF
39 Bldv Bucharest, Ploiești
ROMANIA
E-mail: cristicazan94@gmail.com

Cristiana CONSTANTINESCU
University of Craiova
Faculty of Science, M.A.N.A Research Laboratory
A.I Cuza Street, No. 13, CP 200585, Craiova, Dolj
ROMANIA
Email: constantinescu.cmaria@gmail.com

Gheorghe-Cătălin CRIȘAN
Lucian Blaga University
Faculty of Science
Str. Dr. I. Rațiu, No.5-7, Sibiu, 550012
ROMANIA
E-mail: crisan.gheorghecatalin@gmail.com

Elian DORAN
Lucian Blaga University
Faculty of Science
Str. Dr. I. Rațiu, No.5-7, Sibiu, 550012
ROMANIA
E-mail: elian.doran@ulbsibiu.ro

Ioana Teodora DUMA
Babes-Bolyai University
Faculty of Mathematics and Computer Science
Mihail Kogalniceanu nr.1, 400084, Cluj-Napoca
ROMANIA
E-mail: ioanad1220@gmail.com

Adrian-Cristian FLOREA
”Samuel von Brukenthal” National College, Sibiu
Informatics and Computer Science
ROMANIA
E-mail: acflorea99@gmail.com

Katalina GRIGOROVA
University of Ruse
Department of Informatics and Information Technologies
8, Studentska str., Ruse 7017
BULGARIA
E-mail: k.grigorova@gmail.com

Svetoslav HADZIIVANOV
University of Ruse
Department of Informatics and Information Technologies
8, Studentska str., Ruse 7017
BULGARIA
E-mail: warrolen@gmail.com

Dragoș HODINA
Lucian Blaga University
Faculty of Science
Str. Dr. I. Rațiu, No.5-7, Sibiu, 550012
ROMANIA
E-mail: hodina.dragos@gmail.com

Ionuț HODINA
Lucian Blaga University
Faculty of Science
Str. Dr. I. Rațiu, No.5-7, Sibiu, 550012
ROMANIA
E-mail: ionut.hodina@gmail.com

Flavius Adrian HOLERGA
Theoretical Highschool "Axente Sever"
Mathematics-Informatics
Mediaș, Sibiu
ROMANIA
E-mail: flavius.holerga@gmail.com

Monica-Andreea IONESCU
Petroleum-Gas University
iTIMF
39 Bldv Bucharest, Ploiești
ROMANIA
E-mail: ionescu.monica.andreea@gmail.com

Borislav KOSHAROV
University of Ruse
Department of Informatics and Information Technologies
8, Studentska str., Ruse 7017
BULGARIA
E-mail: bosakmaw@gmail.com

Dorina LUCA
Institute of Mathematics and Computer Science of The Academy of Sciences of Moldova
5 Academiei, street, Chisinau MD-2028
MOLDOVA
E-mail: dorina.luca@math.md

Andrei MUREŞAN
Babes-Bolyai University
Faculty of Mathematics and Computer Science
Mihail Kogalniceanu nr.1, 400084, Cluj-Napoca
ROMANIA
E-mail: muresanandrew@gmail.com

Ionuț-Stelian NICOARĂ
Petroleum-Gas University
iTIMF
39 Bldv Bucharest, Ploieşti
ROMANIA
E-mail: yonutz9337@gmail.com

Alex NEGRU
Lucian Blaga University
Faculty of Science
Str. Dr. I. Rațiu, No.5-7, Sibiu, 550012
ROMANIA
E-mail: alex.negru@ulbsibiu.ro

Alexandru PINTEA
National College Emil Racoviță
Mathematics & Informatics
9-11 Kogalniceanu, Cluj-Napoca
ROMANIA
E-mail: al_pintea@yahoo.com

Iulia Georgiana RÎNEA
”Samuel von Brukenthal” National College, Sibiu
Informatics and Computer Science Department
ROMANIA
E-mail: iulia_rinea@yahoo.com

Robert SĂNDICĂ
Lucian Blaga University
Faculty of Science
Str. Dr. I. Rațiu, No.5-7, Sibiu, 550012
ROMANIA
E-mail: novaspace_robi@yahoo.com

Maria SOKOLOVA
Vladimir State University
Faculty of Information Technology
Str. Gorky, 87, Vladimir
RUSSIA
E-mail: sok.masha2011@yandex.ru

Aleksander STOJAK
Faculty of Computer Science
Bulevar umetnosti 29
SERBIA
E-mail: aleksandarstojak@gmail.com

Eva TUBA
John Naisbitt University
Graduate School of Computer Science
Bulevar Umetnosti 29
SERBIA
E-mail: etuba@ieee.org

Milan III TUBA
Megatrend University
Faculty of Computer Science
Bulevar Umetnosti 29
SERBIA
E-mail: milantuba@gmail.com

Milan TUBA
Megatrend University
Faculty of Computer Science
Bulevar Umetnosti 29
SERBIA
E-mail: tuba@np.ac.rs

Tatiana VERLAN
Institute of Mathematics and Computer Science of The Academy of Sciences of Moldova
5 Academiei, street, Chisinau MD-2028
MOLDOVA
E-mail: tverlan@math.md

SPONSORS (in alphabetical order):