The 11th International Scientific Conference
eLearning and Software for Education
Bucharest, April 23-24, 2015
10.12753/2066-026X-15-027

MiniGL: GAME AND LEARNING

Adian FLOREA, Elimelec BURGHELEA, Arpad GELLERT
Computer Science and Electrical Engineering Department, "Lucian Blaga" University of Sibiu,
Emil Cioran Street No. 4, Sibiu, Romania
adrian.florea@ulbsibiu.ro, elimelec.burghelea@gmail.com, arpad.gellert@ulbsibiu.ro

Delilah FLOREA
Informatics and Computer Science Department, "Samuel von Brukenthal" National College of Sibiu,
Huet Square, No. 5, Sibiu, Romania
delilah_florea@yahoo.com

Abstract: *"Learning may be boring! Try different games in the morning!" We just paraphrased a radio formula in order to explain that it is not need to force students to learn, but we can offer them attractive ways of acquiring knowledge, for example a game. Because nowadays almost every person has a smartphone, and students use smartphones more than PCs, we build games available to mobile devices (smartphone, tablets). Thus, even if they play simple exercises, their brain is working. Furthermore, by a mechanism of reward, they are motivated and want to play again in order to improve their performance, skills, understand solving methods and discover new things. This project targets a large audience. It contains easy games for children, hard games for adults, but those share a thing in common: learning in a pleasant way. It also targets new developers who would like to make their first game application, but starting from scratch is hard for them. By easily adding new features they will be motivated to do something more. Others can play various games while testing their knowledge and can improve their performance because this application provides some training levels. It doesn't matter how they use this game, in the end everybody wins. The purpose of this paper is both to develop enhanced skills of computing, attention and memorizing for games' users and to create programming skills on portable media devices for coders, namely developing the first game for smartphone by a beginner. We implemented in Java under Eclipse framework some educational games for math learning and for testing the cognitive capacity of memory. The applications run on Android mobile phone or on desktop computers having Android emulator with a newer version than 2.2 (Froyo). Our choice is motivated by the fact that, according to market research 49.2% of mobile devices sold worldwide in 2012 have used Android System Architecture. The implemented games are time-dependent straight influencing the score. The MiniGL tool is currently used by students and teachers at Java and Games Programming laboratories from academia especially belonging to "Samuel von Brukenthal" National College and "Lucian Blaga" University (LBUS), both from Sibiu. MiniGL assures the following features specific to almost all educational tools: free availability for use, open-source, flexibility, extensibility and partial portability. The application's source code can be found at https://github.com/coloredflare/MiniGL. The results of a survey on second year students of the Computer Science and Electrical Engineering department LBUS about MiniGL usefulness shows that our students consider MiniGL a valuable learning tool.*

Keywords: *Android; Mobile Learning; Games; Programming; Open-Source.*

I. INTRODUCTION

There is one thing that teachers and learners have to agree upon: learning is boring. Adults may want to learn because they know they need to be prepared for life, but youngsters, the so called "digital natives" or "Net-generation" [1], want to acquire knowledge and technical skills without effort and fast. Nowadays, the competence of software application development has become a highly desirable skill for many jobs not only in IT [2] but also in many technical and non-technical areas. On this issue insisted very much even the visionary Steve Jobs, who said: “everybody in this country should learn to program a computer, because it teaches you how to think” [3]. More and more universities are teaching programming languages for people without technical knowledge.

The mobile communications revolution supported by rapid advances in technology helps both teachers and students in learning process [4] but also, on all social categories by improving their daily lives. Moreover, hardware and software technology applied to educational process became mandatory in nowadays knowledge society and education represents a fundamental element for knowledge improvement and economy growth [5]. Developing the infrastructure of universities (wireless networks in campuses, services and support), the ubiquitous computing concept, contributed to the learning process of students and reduced the involved costs. The students' skills in handling computing tools such as portable media devices or smartphones (the tech-savy students concept), using of social media services, allowed them access to information more quickly, enabling communication between them and with the teachers, facilitating interaction with educational tools regardless of geographic area. We would emphasize the situation at our class when the students use smartphones for online reading and then solve the assignments, or for running applications made in Flash tutorials or using other technologies to describe the basic concepts of computer science domain.

The issue that arises is how to use smart, touch sensitive devices, in order to enrich the students' learning experience and increase the efficiency of teaching process, encourage them to learn [6]. In our opinion, we do not need to force students to learn, but we can offer them attractive ways of acquiring knowledge, for example a game. Researchers in cognitive sciences have pointed out that human intelligence is not fixed and the best way for developing consists in tackling new and difficult tasks, where might struggle and fail. Along with the exercises, we give them short examples of how they can solve an exercise. No one is going to read pages of theory anyway. This is very comfortable because modern students learn efficiently through discoveries, rather by implementing and not by reading instructions from the manual or by listening to teachers [7].

The motivation of this article is two-fold: providing a learning games platform for user and a development platform that can be edited freely for beginner programmers. Also, for students it is a great way to learn new things about programming by developing their own game. We implemented in Java under the Eclipse integrated development environment (IDE) educational games for math learning and for testing the cognitive capacity of memory. The applications run on Android mobile phones or on desktop computers having Android emulator with a newer version than 2.2. Our choice is motivated by the fact that, according to market research 49.2% of mobile devices sold worldwide in 2012 have used Android System Architecture [8]. Besides, the Android platform is open source and due to its portability attracts a large community of software developers. The MiniGL tool is currently used by students and teachers at Java and Games Programming laboratories from “Samuel von Brukenthal” National College and “Lucian Blaga” University (LBUS), both from Sibiu. MiniGL assures features specific to educational tools: free for use, open-source, flexibility, extensibility and partial portability. From educational viewpoint, our application provides the following advantages:

- Develops enhanced skills of computing, attention and memorizing, for game users.
- Acquires programming competences on portable media devices, for game developers.
- Improves students' math results.
- Based on advanced hardware-software technologies creates a pleasant learning alternative to the traditional education system.
- Serious games inculcate the lifelong learning concept in the formal teaching of students.

All these benefits are translated into professional and transversal key competences, in the training of students, and tracking of their vocational route. These skills are the problem solving, decision taking, the emotionally self-control, critical thinking, creativity and initiative.

The organization of the rest of this paper is as follows. In section 2 we shortly review the Related Work in the field of educational software and mainly of game-based learning. Section 3 describes the requirements in order to play the game as user or to modify the existing project, as a developer. In section 4 we present the architecture and how it can be extended adding new features.

Section 5 shows different game genres. In section 6, we assessed on students from LBUS the MiniGL impact on learning process. Finally, section 7 concludes the paper and suggests future work directions.

II. RELATED WORK

There is a big project that aims to provide education to anyone for any topic. That project is KhanAcademy (http://www.khanacademy.org). Unfortunately, mobile users do not have native access to their content. To compensate this lack of types of games on the Khan Academy platform, our MiniGL project tries to give Android users a variety of exercise types, with direct access to the source code. Android market is full of math games. A search by word "math" gives more than 200 results. It may look that is overkill to make another game like that, but all those games have only the basic exercises and they don't have any explanation of how to solve a particular exercise and if someone doesn't know how to solve that exercise, he will skip it. Our work aims to cover many types of learning exercises (not only math) and to implement different genres like trivia, matching game and so on. Also, it provides simple suggestions of developing your own game.

There are also other papers that discuss about eLearning. One of them [9] is about an educational web game framework (developed in Javascript using HTML5 web technologies) that helps young children to get a sense about quantities, seen as rational numbers. In [10] is presented a computer game for training approximate number system in order to improve math abilities of peoples suffering of dyscalculia.

Our previous experience about developing games is described in [11]. We implemented an eLearning platform [6] for improving the teaching and learning process in somewhat abstract domains, such as computer architecture or object oriented programming, with the help of games: a crossword puzzle and a collaborative jigsaw puzzle, the last one supporting multiplayer mode for up to 16 simultaneous players, being simple, fast, fun and reliable. Our application allows geographically distributed students to concurrently and collaboratively play the same game.

Some approaches employing the Android mobile phone platform to teach Java programming are described in [12], [13]. In the former the author presents a course designed to teach C/C++ programmers how to develop object-oriented Java software using Android platform. Unlike us, Riley involves knowledge of C/C ++ from students side. Also, although it focuses on curriculum design proposing some practical applications, in the paper there are no simple examples of source code required for novice programmers of applications for mobile devices.

In [8] the authors present a critical analysis of m-Learning projects emphasizing on mobile computing, the main mobile operating systems and their main features: the architecture together with four categories of learning systems technologies (generic tools and applications, web and quiz environments, frameworks and mobile applications).

III. HARDWARE AND SOFTWARE REQUIREMENTS

3.1 From User Point of View

Since this game was developed for Android it is recommended to have an Android touchscreen phone or an Android emulator, a newer version than 2.2. The game should run just fine on almost all configurations, even on low cost devices, but on an emulator is running considerably slowly. For emulator a performant PC, such as quad-core 2.4 GHz Intel P4 Xeon with 4GB RAM memory is necessary.

3.2 From Developer Point of View

This project targets devices with Android operating system. Because Android native applications are written in Java [14], this game is mostly made in Java. However, some layouts may be developed in Html with Css and Javascript. Also, the design is made with Xml. If someone wants to modify or add new features to this game, it can be freely done. This project is MIT licensed. The open source programming environment is Eclipse that integrates C, C++ and Java tools targeting different operating systems and processors. The portability and flexibility of Eclipse recommend it for creation of desktop, mobile and web applications. The whole project is hosted at GitHub: https://github.com/coloredflare/MiniGL. This repository must be cloned on the developer's hard disk and imported it in Eclipse. For this the Eclipse platform and setup for Android development is necessary. The Eclipse and Android SDK Tools can be downloaded separately or as a bundle. See [15] for Android Developer Tools plugin, https://www.eclipse.org/downloads/ for Eclipse platform and http://developer.android.com/sdk/index.html for ADT Bundle. A Java Development Kit must be also installed from: http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html.

IV. MiniGL SOFTWARE ARCHITECTURE

4.1 Basic Features

Main is the core class of this project and extends the basic class *Activity*. When the game is launched actually this activity is started. The *onCreate* method is the first that is called. Even if they are written in Java, Android applications do not have the classic main method. This allows the developer to control the application from the very start. There are other methods that are special to an activity that are called when the user leaves the application, pause it or resume. Beside these methods, there are methods that start games, change the game mode or show the settings, but those are explained later.

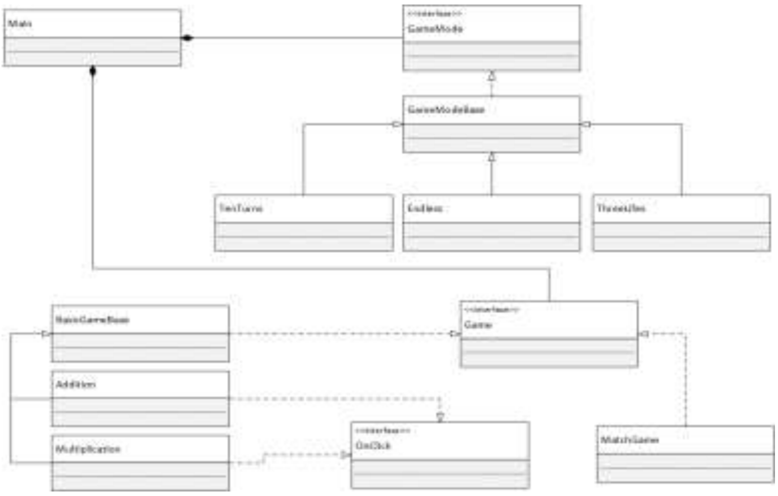

Figure 1. Application class diagram

There are many interfaces. One of them is *Game*. Any game type (*Addition*, *Multiplication*, *MatchGame*, etc.) must implement this interface. Another interface is *GameMode* which must be implemented by any game mode class as it can be seen below.

```
public void launchGame(View view) {
        int gameId = view.getId();
        switch (gameModeId) {
                case 0:                 gameMode = new Endless();break;
                case 1:                 gameMode = new TenTurns();break;
                …
        }
        switch (gameId) {
                case R.id.addition:     game = new Addition(gameMode, this);break;
                case R.id.match_game:   game = new MatchGame(gameMode, this);break;
                ….
        }
        game.start();
}
```

Beside *Game* and *GameMode*, there are interfaces like *onClick*, *onHold* and *onSwipe*. They are implemented only if that game uses such kinds of event. Those actions are first captured in Main –

that’s why *Main* has those methods: *onClick* (View view), *onHold* (View view) and so on – and then passed to the actual game.

4.2 Extending MiniGL

To add a new game, or a new game mode, a class that implements *Game*, respectively *GameMode* must be created. Then the game can be implemented as it is wished, but when *start()* is called, the game has to start, and when *stop()* is called, the game must end and save score or state. Then to the layout for the main activity, which is located in */res/layout/*, can be added a button, a text view or an image that represents the game. That view must be set to call a method when the user clicks on it. That method must be defined in *Main*. It has to be *public*, *void* and must have a *View* as parameter. Now starting the game is as easy as the next two lines of code:

```
game = new Addition(gameMode, this);
game.start();
```

This is how an *Addition* game is started. For a certain game, *Addition* must be replaced with the class of that game. For game mode, the same steps are necessary, but the game mode must be set to the custom game mode:

```
gameMode = new Endless();
```

For a certain game mode, *Endless* must be replaced with the class of that game mode.

The following code sequence represents a part of *MatchGame.click_card (View view)* method emphasizing the removing cards from view in case of two are matching.

```
public class MatchGame implements Game {
        public int ct=0, nrOfCards=6, x,y;
        public String z, t, card[]={"Frontcard1", "Frontcard2", ….};
        Button btnarray[] = new Button[6];
        ….
        public void click_card(View v){
                        ct++; nrOfCards--;
                        if(ct==1) {
                                x = v.getId();
                                switch(x) {
                                case R.id.button1: {
                                        Button btn1 = (Button) findViewById(R.id.button1);
                                        btn1.setText(card[0]);
                                        z=(String)btn1.getText(); break;
                                }
                                ….
        }
```

V. GAME GENRES

5.1 Trivia

As the application menu reveals (see Figure 2), there are different game modes, each one affecting how long, how many mistakes or how many exercises are in the trivia game. The *Endless* game mode lets play forever. *Three Lifes* lets play until three mistakes. *Ten Turns* gives only 10 exercises. Finally, *One Minutes* lets play only for a minute.

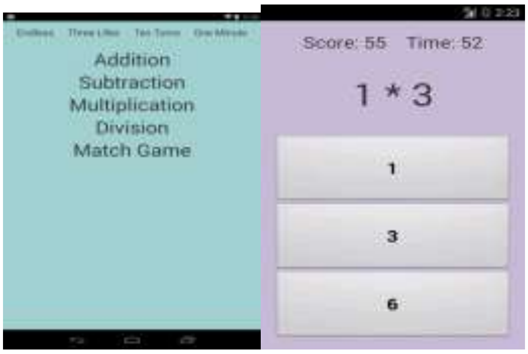

Figure 2. MiniGL user interface. Playing Multiplication

The implemented games are computing games (associated with mathematical operators) and memory games [16] based on matching two playing cards having the same symbol (graphical or text). In the first category, the user will see an exercise. The types of exercises are multiple-choice single answer (only one being correct). Usually, all the exercises and answers are randomly generated and it is unlikely to find the same exercise again soon, because there are generated hundreds of exercises. The user cannot memorize that much, so his/her brain is used to solve it. Obviously that, playing regularly, the user will improve in time its performance. Every time the user chooses the correct answer, he gets some reward points. The score is based on how much time has passed since the last answer, but it cannot be lower or higher than a limit.

5.2 Matching Game

The basic idea of this game is to match two elements of the same type (a memory game). All elements are randomly taken from a database and randomly put on the screen. Elements can also be generated randomly if it is a math game. The aim is to group these elements in pairs within a minimum number of steps. It develops attention and willingness to resolve configurations in the shortest time. After starting the game, the user will see on the screen six or twelve randomly generated playing cards that contain combinations of text and numbers (see Figure 3).

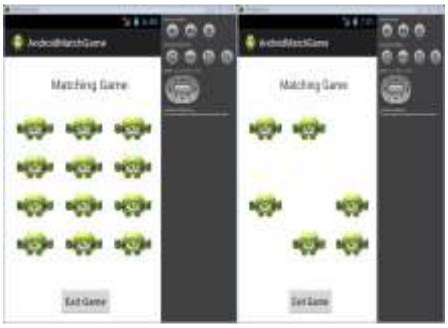

Figure 3. MiniGL user interface. Playing Matching Game

During runtime, the cards should be grouped, such that there are pairs of two cards that contain the same text to be associated with each other. First only the back of the cards is shown that does not help at all. In order to see the text the user has to tap them. The first tap will show the content of the tapped card. The second tap will review the content and if it is matching with the first, both cards will disappear. Otherwise, the cards will return in the initial position such that the user will see "*Backcard*", which is the backside of playing cards. Then the user has to flip again the cards. The game continues until there are no cards left. At every new game, the configuration of the cards is changed and their position will be different. The user must concentrate, to develop an enhanced attention regarding the previously unveiled cards. To be able to finish in the smallest number of turns, which means getting the best score, the student has to memorize each card's position and content from the first time he/she has seen it. Because they are randomly generated, in time the user's memory and attention will improve. Every time the user matches two cards, he receives some points. At the end, the user receives an additional bonus based on how much time has passed since the game started and how many moves he/she used.

Even though each time the cards are placed in new positions, as the user plays more, the interest for solving the generated configuration as quickly as possible grows more. The game fosters the attention related to the arrangement of playing cards and of their positioning on the screen. Any speculation regarding their dispersion will fail because they are changing at every new play. The Matching Game has been tested in the classroom. The students showed interest in testing it and the possibility of porting on Android mobile phone aroused the everyone desire to have it and play with it.

VI. THE MiniGL IMPACT ON LEARNING

Besides the goal of this paper – that of presenting the MiniGL tool and enhancing the skills of computing, attention and memorizing for gamers, and to create programming skills on portable media devices for coders, – we were interested in attempting to assess the impact of MiniGL impact on learning. We suppose that a dynamic presentation directly completed with examples of source code implementation is a more effective teaching tool than static presentations of Java, Eclipse or Android environment. Therefore, we requested feedback from our students. We offered the opportunity to experiment playing / software developing with MiniGL within a lab setting for 30 students from second year of the Computer Science and Electrical Engineering department. At that moment of time, the students had already been introduced to the Java programming under the Eclipse IDE via classroom lectures and homework assignments.

After a period required for experimentation (approximately 30 minutes), students were asked to answer a four question survey using a 5 point Likert scale where 1 represents strongly disagree and 5 represents strongly agree. The survey and results appear in Table 1. Some valuable suggestions for improve our work were presented in Table 2 and few positive comments we illustrated in Table 3.

Table 1. 5-Point student survey of MiniGL

Question	***Average Result***
In general, this tool helped me to pay higher attention when I solve exercises fast and enhanced my memorizing skill.	4.15
I learned to write programs for my smartphone that makes me very proud.	4.50
By playing MiniGL was fun but also motivated me to win.	3.95
Coding with MiniGL has clarified for me the differences between of using an emulator or writing programs directly on the device.	4.10
I like MiniGL but it fits better on laptops or iPad as it has a bigger screen. It allows me to read better being more effective.	2.50

Table 2. Suggestions from students for improving MiniGL

Difficult viewing animation (too fast, too slow, ...). I suggest some changes in layout.
It cannot save game configuration or level reached in order to resume the game later.
Some games are too easy. The game must be stopped and restarted without exiting from it. Graphics is well done but the animation may be improved.

Table 3. Important feedback from students

Very effective! I was having trouble understanding the multithreading, inter-process communication and object oriented programming concepts (encapsulation, inheritance and polymorphism), but MiniGL helped me to fix these concepts.
I became much motivated when using the MiniGL. It transformed me from a passive content consumer to an active contributor.
I consider MiniGL very useful. It allows students to learn smartphone programming (Android Java) and the teachers can explain concepts related to applications software development and their implications in hardware.
Great job! I start love programming.

VII. CONCLUSIONS

Studies show that the degree of concentration of students drops after approx. 10-15 minutes of teaching. Thus, the application of modern interactive teaching methods becomes imperative. A major challenge in teaching is to keep students motivated during the lesson. Thus, we resorted to application development as different memory and computing games, which capture the users' attention, given the random generation of playing cards or the random operations and numbers that appear in the exercises.

The application is far from being complete. We will continue to add new exercises / games, maybe others will help too. Also, we would like to port this game to other mobile platforms like Windows Phone or iPhone/iPad. Further developments of the application could contain a memory (or database) with scores generated and establishing a hierarchy with the obtained best scores. Furthermore, we intend to extend our application to facilitate multi-user playing allowing geographically distributed students to concurrently and collaboratively play the same game.

A further improvement may be made by inserting a new parameter, which is the number of cards that composes the test. It could be a source of difficulty in the application. After the user solved the memory test with a smaller number of cards, he/she is able to move to a higher level, by entering a greater number of cards that will compose the test on the screen. Another idea for improvement would be to generate texts, previously introduced in database from various fields of interest, which subsequently during game will match in pair. In this way, the game will aim not only keeping attention focused, but also for learning new things.

Reference Text and Citations

[1] *Android Developer Tools*, 2014, http://developer.android.com/tools/sdk/eclipse-adt.html, (accessed 02 April 2014).

[2] Anghel, T., Florea, A., Gellert, A., Florea, D., 2014, *Developing Online Collaborative Games for e-Learning Environments*. In New Horizons in Web Based Learning, pp. 221-230, Springer Berlin Heidelberg.

[3] Anghel, T., Florea, D., Florea, A., 2009, *Encouraging student's learning potential with Testing Assistant*, The 7th National Conference for Virtual Learning, Romania.

[4] Axelrod-Contrada, J., 2014, *Computer Coding Now Seen As Vital Job Skill*, The Boston Globe.

[5] Chirilă, C.B., 2014, *Educational resources as web game frameworks for primary and middle school students*, The 10th eLearning and Software for Education Conference, Romania.

[6] Gliga, F., Ivănescu, A., 2014, *Computer training of approximate number system could improve math abilities*, The 10th eLearning and Software for Education Conference, Romania.

[7] Florea, D., Florea, A., 2013, *Matching Game*, Proceedings of the International Symposium Creativity and Technology in Addressing the Educational Process, Romania.

[8] Ivanovic, M., Jain, L.C., 2014, *E-Learning Paradigms and Applications, Agent-based Approach*, Eds. Springer.

[9] Mednieks, Z., Dornin, L., Meike, G.B., Nakamura, M., 2012, *Programming Android: Java Programming for the New Generation of Mobile Devices*, Second Edition, O'Reilly Media.

[10] Peñalvo, F.J., Palacios, R.C., Lytras, M., 2012, *Outcomes of International Research Projects on Technology Applied to Education*, Journal of Universal Computer Science, vol. 18, no. 1, pp. 1-4, Austria.

[11] 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.

[12] Prensky, M., 2001, *Digital Natives, Digital Immigrants, Part II: Do They Really Think Differently?*, On the Horizon, NCB University Press, vol. 9, no. 6, pp. 1-6.

[13] Ryan, T., 2014, *Raising the Bar: An interview with Carol Russell*, ACM eLearn MAGAZINE: Education and Technology in Perspective.

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

[15] 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.

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