PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
UNIVERSITY OF ECHAHID HAMMA LAKHDAR EL OUED FACULTY OF EXACT SCIENCES
Department of Computer Science

Domain : Mathematics and Computer science
Field : Computer science
Specialty : Information System

Design and development of an educational game for teaching preliminary concepts of mathematics to beginner students

Presented by:

SADALLAH Younes

MEDAOUI Rachid

MAAMOUN Mohamed Oussama

Supervised by:

Mr. Mohammed Charaf Eddine MEFTAH

Academic year: 2017-2018

Dedication

Words are not adequate to express my gratitude, respect and love to my dear parents whom have blessed and supported me throughout the ordeal of fulfilling our work.

Special dedication goes to my brothers (Lakhdhar and Haithem)

To All the family of Sadallah

To all my honest friends

(Rachid,Oussama,Youcef,Souhaib,Amine,Raouf, AbdelMalek,Chamso,Mohamed)

Thank you for the moments we shared together

Dedication

Words are not adequate to express my gratitude, respect and love to my dear parents whom have blessed and supported me throughout the ordeal of fulfilling our work.

Special dedication goes to my mother and my all brothers and sisters

To All the family of Medaoui

To all my honest friends

(Younes ,Oussama,Youcef,Souhaib,Abd al malek , Mohamed)

Dedication

Words are not adequate to express my gratitude, respect and love to my dear parents whom have blessed and supported me throughout the ordeal of fulfilling our work.

Special dedication goes to my mother and father and my all brothers and sisters.

To All the family of Maamoun

To all my honest friends

(Younes,Rachid,Youcef,Chmso,Seif,Lahdi,Abd Allah,Amel,Moustafa,Karim,Dhia)

Acknowledgments

Our greatest gratitude and our deepest thankfulness are to the One who should be praised first, to the Almighty ALLAH who gives us strength and guidance that enabled us to fulfill this study.

A very profound debt of gratitude is owed to our supervisor *Mr MEFTAH Mohamed Charaf Eddine* for his continuous encouragement, guidance, support, insightful criticisms and patience.

Also, we are thankful to all the examiners who spared us some of their precious and busy time to read, correct and criticize our work which is not immune from mistakes and imperfections.

We would like to acknowledge and express our heartfelt gratefulness to all our teachers without exception.

Abstract

According to constructivist theories, learning must be meaningful and actively engage students in decision-making and understanding. Due to this theory, we suggest to design this educational math game (because mathematics is difficult a little bit to learn its basics) for the students of the first year of primary school in order to change their manner of thinking.

Résumé

Selon des théories constructivistes, l'apprentissage doit être significatif et efficace pour engager les élèves dans la prise de décision et la compréhension. Grâce à cette théorie, on a proposé de faire ce jeu éducatif de mathématique (parce que le mathématique est difficile un tout petit peu pour apprendre les notions fondamentaux) pour les élèves de la première année primaire afin de changer leurs manière de pensée.

ملخص

وفقا لبعض النظريات، التعليم لكي يكون ذو معنى وفعالية وجب إشراك التلاميذ في اتخاذ القرار والفهم. بفضل هذه النظرية، اقترحنا تصميم لعبة تعليمية خاصة بالرياضيات (نظرا لصعوبة اكتساب بعض المبادئ الاساسية لهذه المادة) لتلاميذ السنة اولى ابتدائي من اجل تسهيل عملية استيعابهم لهذه المبادئ الأولية.

Summary

List of Figures

Chapter 2

Chapter 3

List of tables

Chapter 2

Chapter 3

General Introduction

Before two decades, the educational curriculum was based on the simple means (board, objects, copybooks ...etc.). When the technological revolution started, a lot of teachers tried to overlap the technology with education in order to improve the quality of education.

Games offer a unique structure to complement traditional teaching strategies and infuse teaching with energy, spark innovative thinking and provide diversity in teaching methods, games make learning concepts more palatable for students and supply learners with a platform for their creative thoughts to bounce around.

Problematic:

The second generation of mathematics curriculum of the first year primary contains a basic concepts (addition, subtraction, complementary) a little bit difficult for some students to understand it quickly.

So, we will try in this project to use the educational games as a method to facilitate accommodating of this concepts to the students.

Because of that, we choose a desktop application in java for learning math, so the main objectives of our application is:

- Amelioration the grade of students.
- Make some fun with their education way.
- Help teachers to make the courses more effective.
- Improve our programming skills.

According to these objectives we organized our memory, as illustrated by the following figure:

Chapter 1: The role of educational games.

Chapter 2: Comparison between educational games.

Chapter 3: Conception and realization.

Chapter I

The role of educational games

Introduction

We all know that children love playing games. Experience tells us that games can be very productive learning activities but how can we turn this love of games into rich mathematical learning experience? In the last years we noticed that most of students get bored from the abstract study in mathematics and their parents complain from the negative results.(1)

I.1 Suggestions to learn mathematics

A lot of teachers propose some ideas and solutions to fix this problem:

- Use the objects (such as match's fire, balls, hands …).
- Practice a lot of homework.
- Memorization some basic operations.
- Use the mathematical games.

I.2 Analyze of the suggestions

Suggestion 1: cannot use big numbers and objects cannot always be available

Suggestion 2: term "Homework" causes disgust to students and they will get bored quickly.

Suggestion 3: cannot memorize all the operations.

Suggestion 4: the best one and we will explain that in out project.

"Maths is kind of fun when you get to play some math games"

I.3 What is a mathematical game?

When considering the use of games for teaching mathematics, educators should distinguish between an 'activity' and a 'game'. Gough (1999) states that "A 'game' needs to have two or more players, who take turns, each competing to achieve a 'winning' situation of some kind, each able to exercise some choice about how to move at any time through the playing". The key idea in this statement is that of 'choice'. In this sense, something like Snakes and Ladders is not a game because winning relies totally on chance. The players make no decisions, nor do that have to think further than counting. There is also no interaction between players - nothing that one player does affects other players' turns in any way.(1)
Oldfield (1991) says that mathematical games are 'activities' in which:

- Involve a challenge, usually against one or more opponents.
- Are governed by a set of rules and have a clear underlying structure.
- Normally have a distinct finishing point.
- Have specific mathematical cognitive objectives. (1)

I.4 Levels of the games and how the games should be?

A good game provides engagement at cognitive, affective, and operative levels. That is, there must be *challenge* embedded with the game – if it's too easy, children will get bored and no learning will occur. The game must be enjoyable to play, and it must promote *interaction* and dialogue. There are many math games on the market that are basically drill and practice with the intention of building fluency with number facts. There are also an infinite number of traditional non-math based games that have a range of mathematical skills and processes embedded within them. (2)

I.5 The importance of mathematical games

Games give students opportunities to explore fundamental number concepts, such as the counting sequence, one-to-one correspondence, and computation strategies. Engaging mathematical games can also encourage students to explore number combinations, place value, patterns, and other important mathematical concepts. Further, they afford opportunities for students to deepen their mathematical understanding and reasoning. Teachers should provide repeated opportunities for students to play games, then let the mathematical ideas emerge as students notice new patterns, relationships, and strategies. Games can be an important tool for learning in elementary school mathematics classrooms:

- Playing games encourages strategic mathematical thinking as students find different strategies for solving problems and deepen their understanding of numbers.
- When played repeatedly, games support students' development of computational fluency.
- Games present opportunities for practice, often without the need for teachers to provide the problems. • Teachers can then observe or assess students and work with individuals or small groups of students.
- Games have the potential to allow students to develop familiarity with the number system and with "benchmark numbers" (such as 10s, 100s, and 1000s) and engage in computation practice, building a deeper understanding of operations.
- Games support a school-to-home connection. Parents can learn about their children's mathematical thinking by playing games with them at home. (3)

I.6 The benefits of educational games

The advantages of using games in a mathematical program have been summarized in an article by Davies (1995) who researched the literature available at the time.

- Meaningful situations - for the application of mathematical skills are created by games.
- Motivation - children freely choose to participate and enjoy playing.
- Positive attitude - Games provide opportunities for building self-concept and developing positive attitudes towards mathematics, through reducing the fear of failure and error.
- Increased learning - in comparison to more formal activities, greater learning can occur through games due to the increased interaction between children, opportunities to test intuitive ideas and problem solving strategies
- Different levels - Games can allow children to operate at different levels of thinking and to learn from each other. In a group of children playing a game, one child might be encountering a concept for the first time, another may be developing his/her understanding of the concept, a third consolidating previously learned concept.
- Assessment - children's thinking often becomes apparent through the actions and decisions they make during a game, so the teacher has the opportunity to carry out diagnosis and assessment of learning in a non-threatening situation.
- Home and school - Games provide 'hands-on' interactive tasks for both school and home.

- Independence - Children can work independently of the teacher. The rules of the game and the children's motivation usually keep them on task. (1)

Conclusion

For these reasons we are working to make a mathematical game in order to help students in their own study and resolve the problem of perception the mathematical concepts. Further, we are trying to collect all the basic skills and embody it

Chapter 2

Comparison between some existing educational games

Introduction

In this chapter we will compare between some educational games and show their drawbacks and advantages. Through this comparison we use the max of advantages and trying to avoid the drawbacks.

II.1 Game 1: Jungle Math (5)

II.1.1 The main Principle of the game

- Move a little monkey up and down in order to collect as many bananas as we can.
- Every period you will avoid an obstacle which contains a math question and you have to choose the correct answer.
- It continues with the same way

II.1.2 Drawbacks

- Using of big numbers in the first grade which are more difficult than the capacities of students
- In this game we can see that the fun part is dominant than the mathematical one.

II.1.3 Advantages

- It has a perfect design

Figure 1 : Jungle Math

II.2 Game 2: Fish Bowl Addition (6)

II.2.1 The main Principle of the game

One fish, two fish, add up all the fish! In this interactive addition game, kids help Birdee and Cuz-Cuz total up all the fish they've won.

II.2.2 Drawbacks

- there are not a lot of levels of operation to explain the concept.
- There is not a serval of mathematical ways

II.2.3 Advantages

- Availability of sound which can help the player by giving him the number.
- **Hint**: gives the player an instruction to reach to the correct answer.

Figure 2 : Fish Bowl Addition

II.3 Game 3: Subtraction Pizza Party(6)

II.3.1 The main Principle of the game

In this interactive math game, kids must drag pizza toppings off of the pizza to complete subtraction problems and give each customer exactly what they order. This first-grade math game uses digital manipulatives to help kids visualize the answer to problems in a concrete way, instead of relying on mental subtraction strategies.

II.3.2 Drawbacks

- Difficulty of understanding the basic of subtraction

II.3.3 Advantages

- It contains a lot of examples which are compatible with the reality.

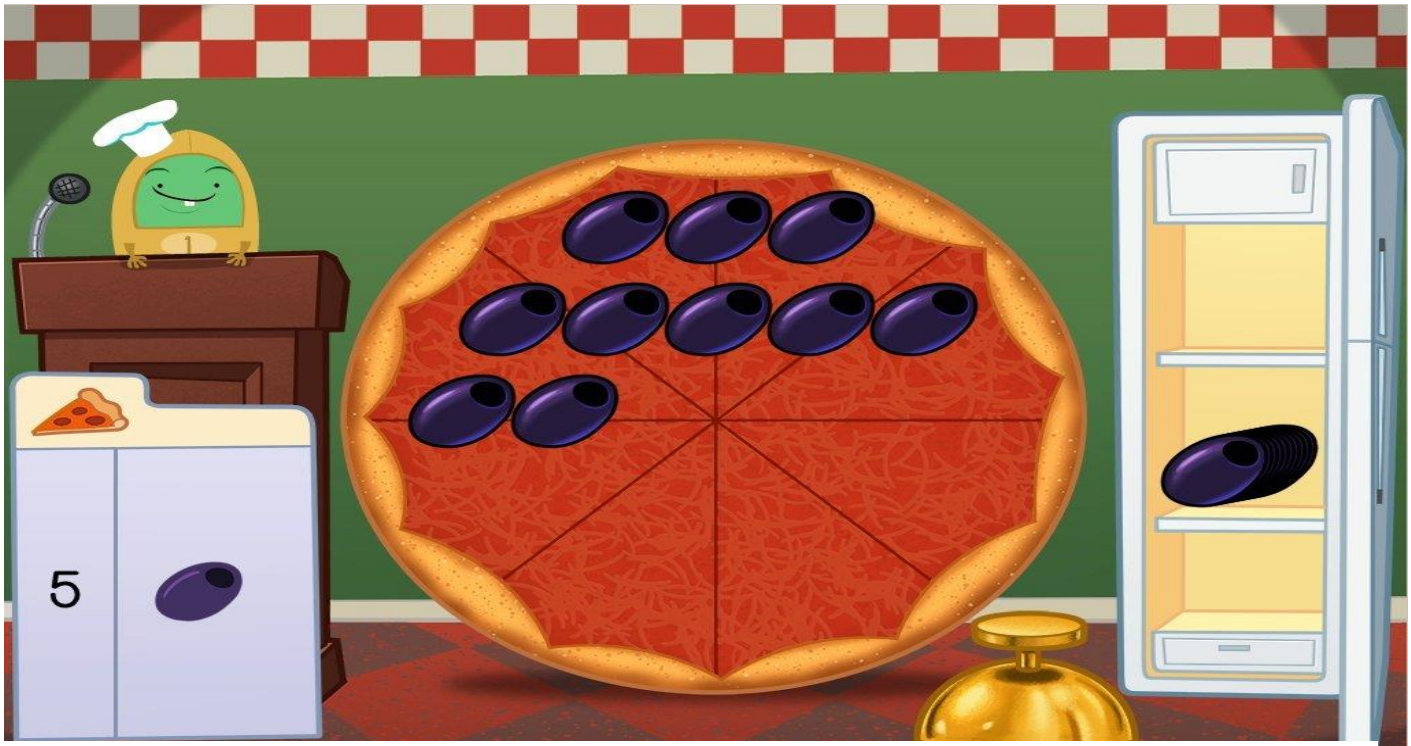

Figure 3 : Subtraction Pizza Party

II.4 Game 4: Counting in the kitchen (6)

II.4.1 The main Principle of the game

Click on the group that matches the number you hear before time runs out.

II.4.2 Drawbacks

- All the levels are the same
- Cannot learn a lot of the basics of calculations from it

II.4.3 Advantages

- it makes the player more quickly to calculate
- It is simple and it does not need a lot of skills

Figure 4 :Counting in the kitchen

II.5 Game 5: Numbie(6)

II.5.1 The main Principle of the game

Tap and drag the box with the number of fruit's pieces you listen.

II.5.2 Drawbacks

- lack of mathematical calculations (addition, subtraction … etc.).

II.5.3 Advantages

- perfect design. The same as the first one.

Figure 5 :Numbie

II.6 Game 6: Math Addition(6)

II.6.1 The main Principle of the game

Tap and drag the box with the number of fruit's pieces you listen.

II.6.2 Drawbacks

- It should be the last level
- there are not touchable things to collect between numbers and it very simple design

II.6.3 Advantages

- usage of small numbers (less than ten)

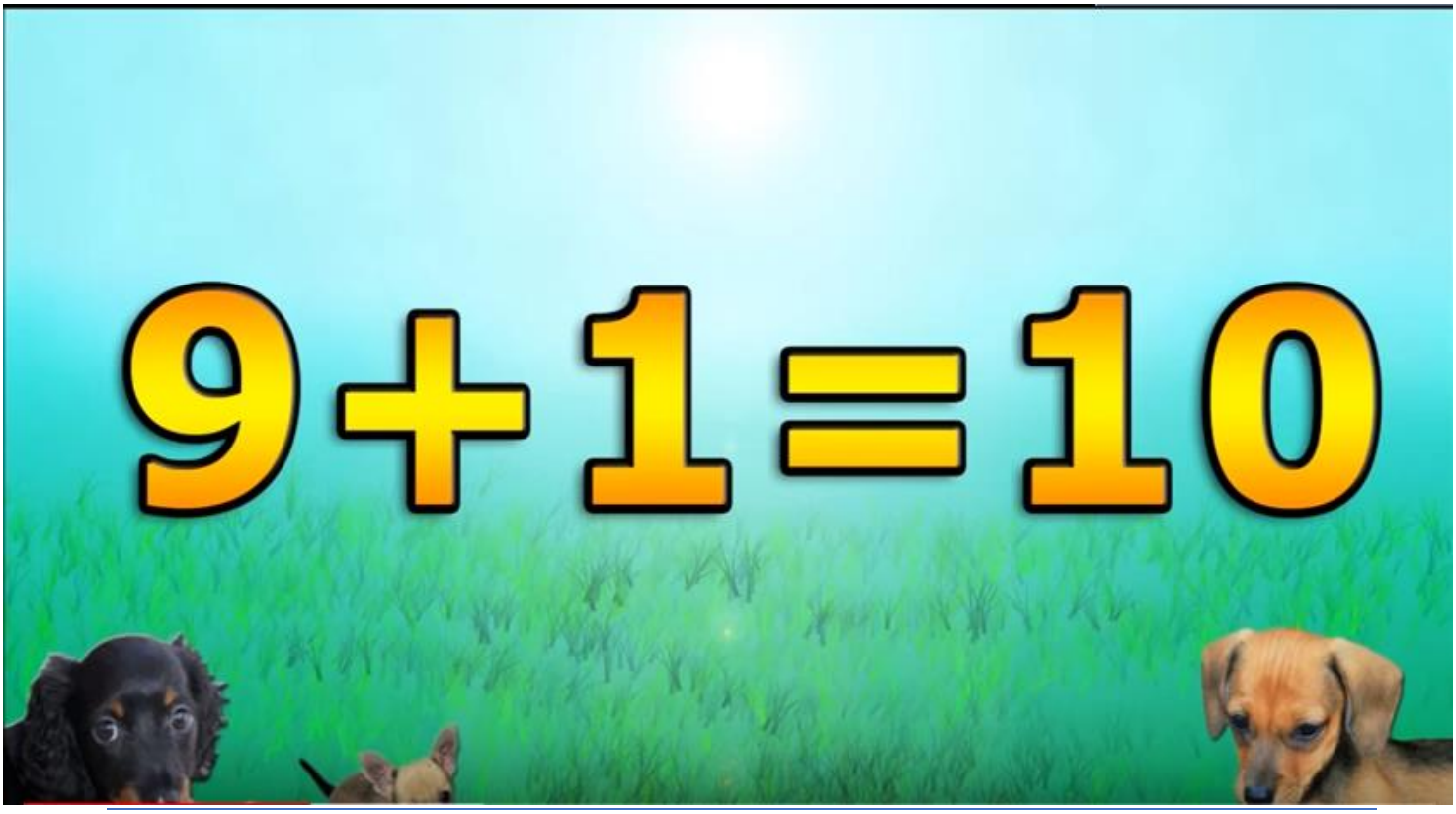

Figure 6 : Math Addition

II.7 Game 7: Baby Learn Numbers with Funny Food Math(6)

II.7.1 The main Principle of the game

Put things in the appropriate position and show result

II.7.2 Drawbacks

- There is not any relation between calculation and this game because there are not a lot mathematical calculation.

II.7.3 Advantages

- **it** has a good design

Figure 7 : Baby Learn Numbers with Funny Food Math.

II.8 Summary of advantages and drawbacks

	Existing games
Advantages	• gorgeous designs(Numbie) • it has hints(Fish Bowl Addition) • it has sounds(Fish Bowl Addition) • usage of a developed environments(Counting in the kitchen)
Drawbacks	• a lot of fun instead of learning(jungle math) • sometimes there are noise because of the sound • the most of games have not a database(Math addition)

Table 1 : advantage and drawbacks

Conclusion：

From this comparison (advantages and drawbacks of each game) we will try in the next chapter to do the best conception of our game.

Chapter 3

Conception and realization

Introduction

In this chapter we will explain some diagrams and the methods which we used in our project in addition to the environment.

III.1 Conception

III.1.1 needs analysis

We contacted a director (Mr Ben Salem MICHARA) of Zin El-Abidin CHKIRI primary school in Guemar - ELoued, first of all he accepted the idea and encouraged us to continue our project. Secondly, he gave us the conditions of this game (variation of levels, math skills ..etc.) according to the educational curriculum.

So, from:

- Our meeting with the director and teachers.
- The official public school curriculum.
- Existing educational games.
- Some research on the trends of the children of HMI

We are trying to follow these conditions and advices in order to product a good quality of game.

Figure 1: selections from the educational curriculum

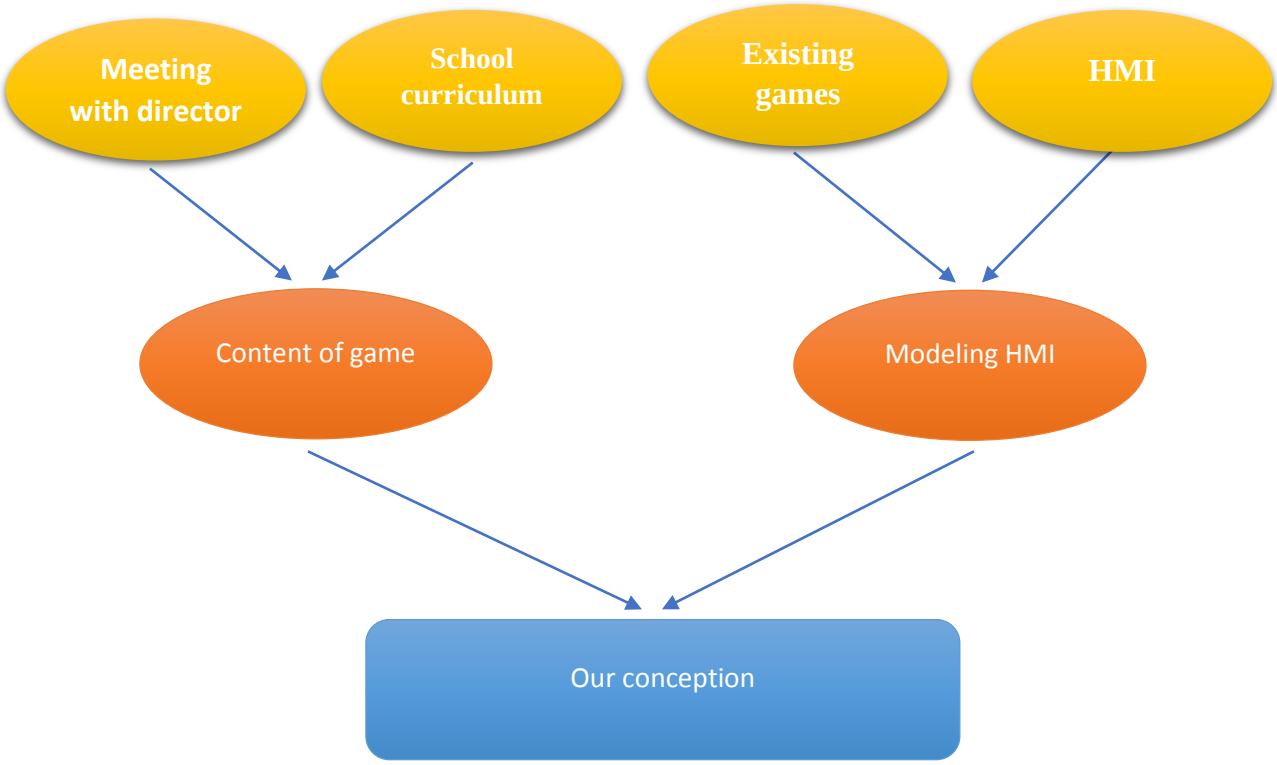

Figure 2: schema of deatailed design

III.1.2 MVC Architecture

Swing API architecture follows loosely based MVC architecture in the following manner.

- Model represents component's data.
- View represents visual representation of the component's data.
- Controller takes the input from the user on the view and reflects the changes in Component's data.

- Swing component has Model as a separate element, while the View and Controller part are clubbed in the User Interface elements. Because of which, Swing has a pluggable look-and-feel architecture.

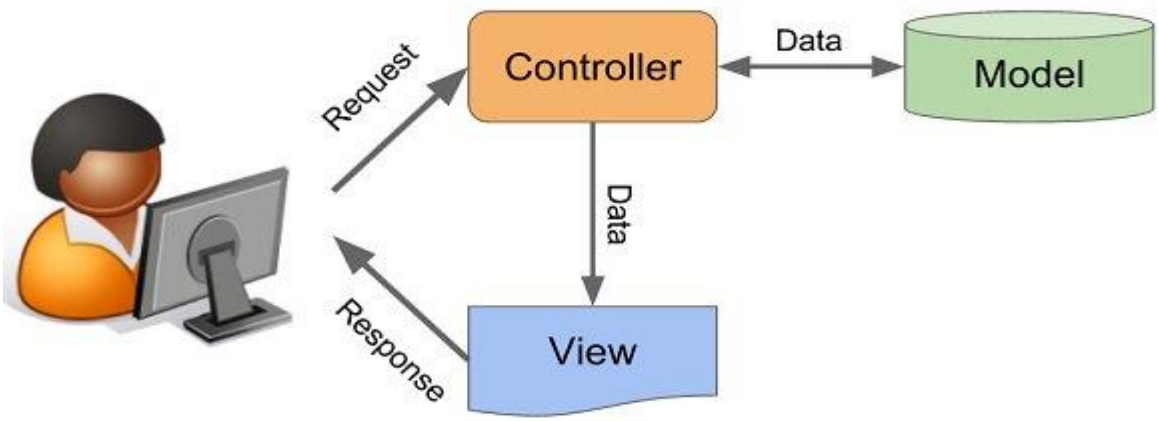

Figure 3: model-view-controller

III.1.3 How to play?

- The student should sign in the game (enter the username and password).
- These two data will be saved in the database.
- Before he gets started, the student will choose one lesson (Addition, Subtraction, Complementary).
- For each lesson there are four levels (except the complementary there are just three levels).
- To pass from a level to another one student must collect eight points.
- If he does not raise his score to eight points , he will be obliged to restart the level until he the condition.

III.1.4 Diagrams

III.1.4.1 Use case diagram

use case diagram is the primary form of system/software requirements for a new software program under developed. Use cases specify the expected behaviour (what), and not the exact method of making it happen (how).(8)

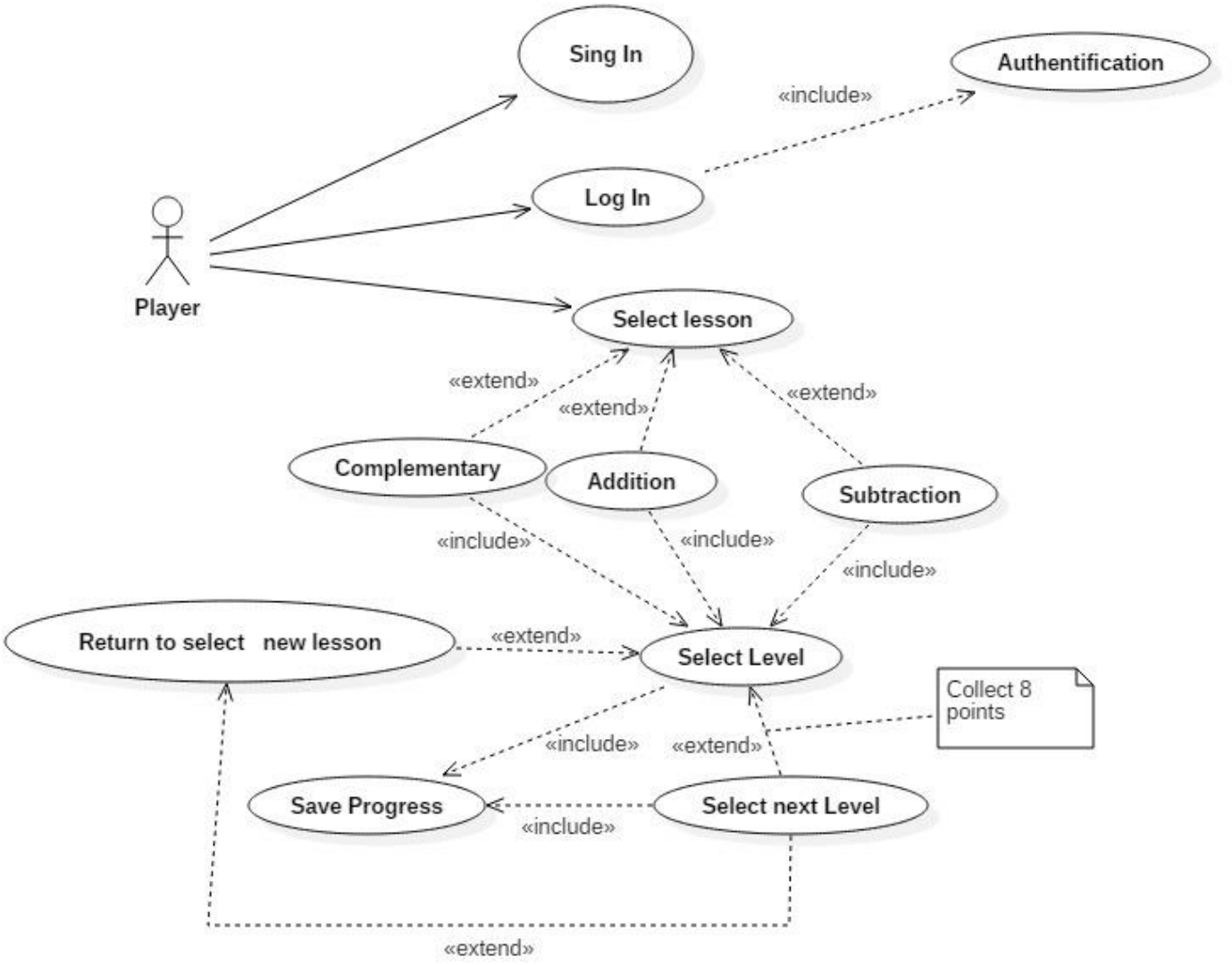

Figure 4: Use Case Diagram

III.1.4.2 Activity diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.(8)

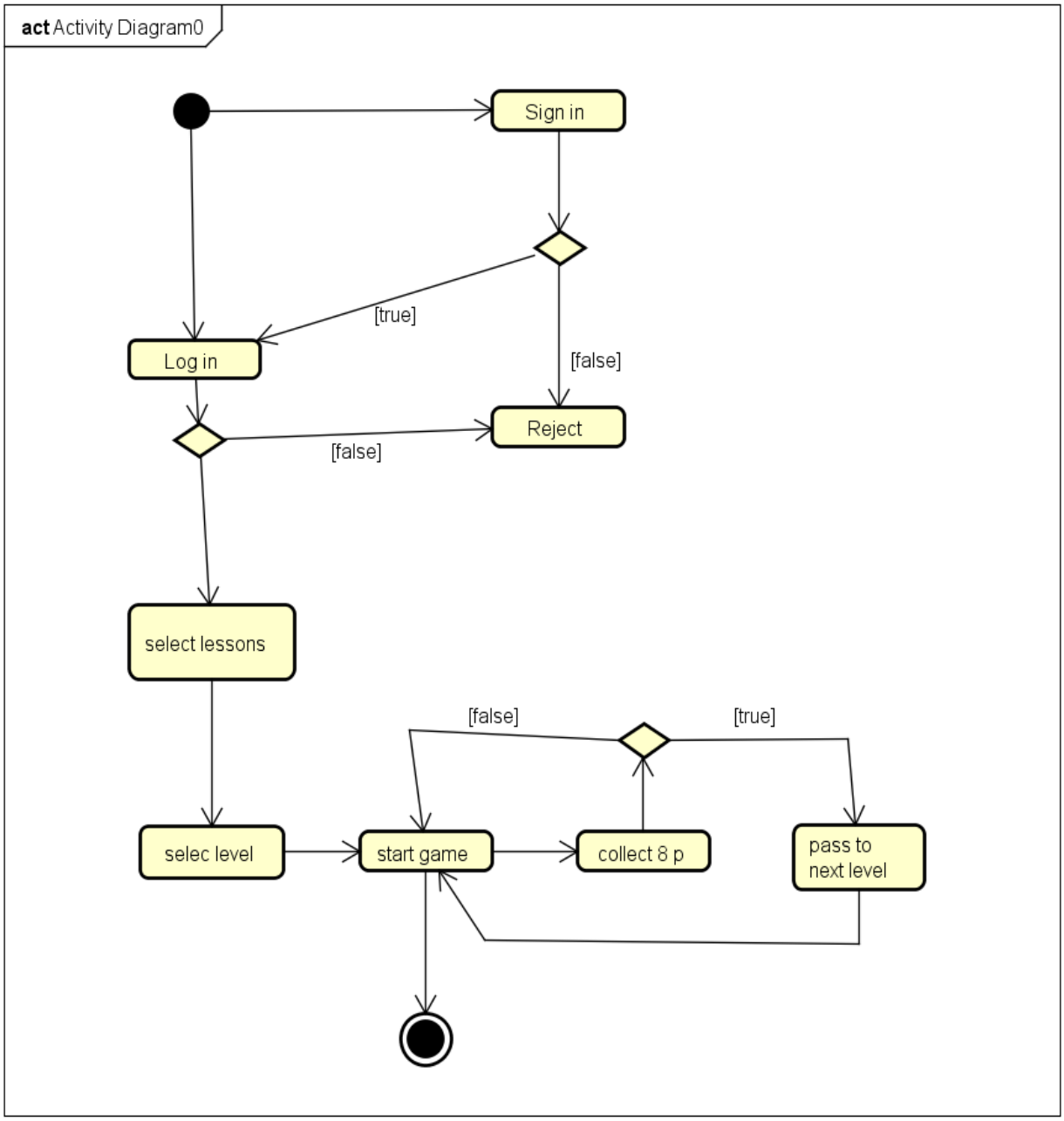

Figure 5: Activity diagram

III.1.4.3 Sequence diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time.A sequence diagram is a good way to visualize and validate various runtime scenarios. **(8)**

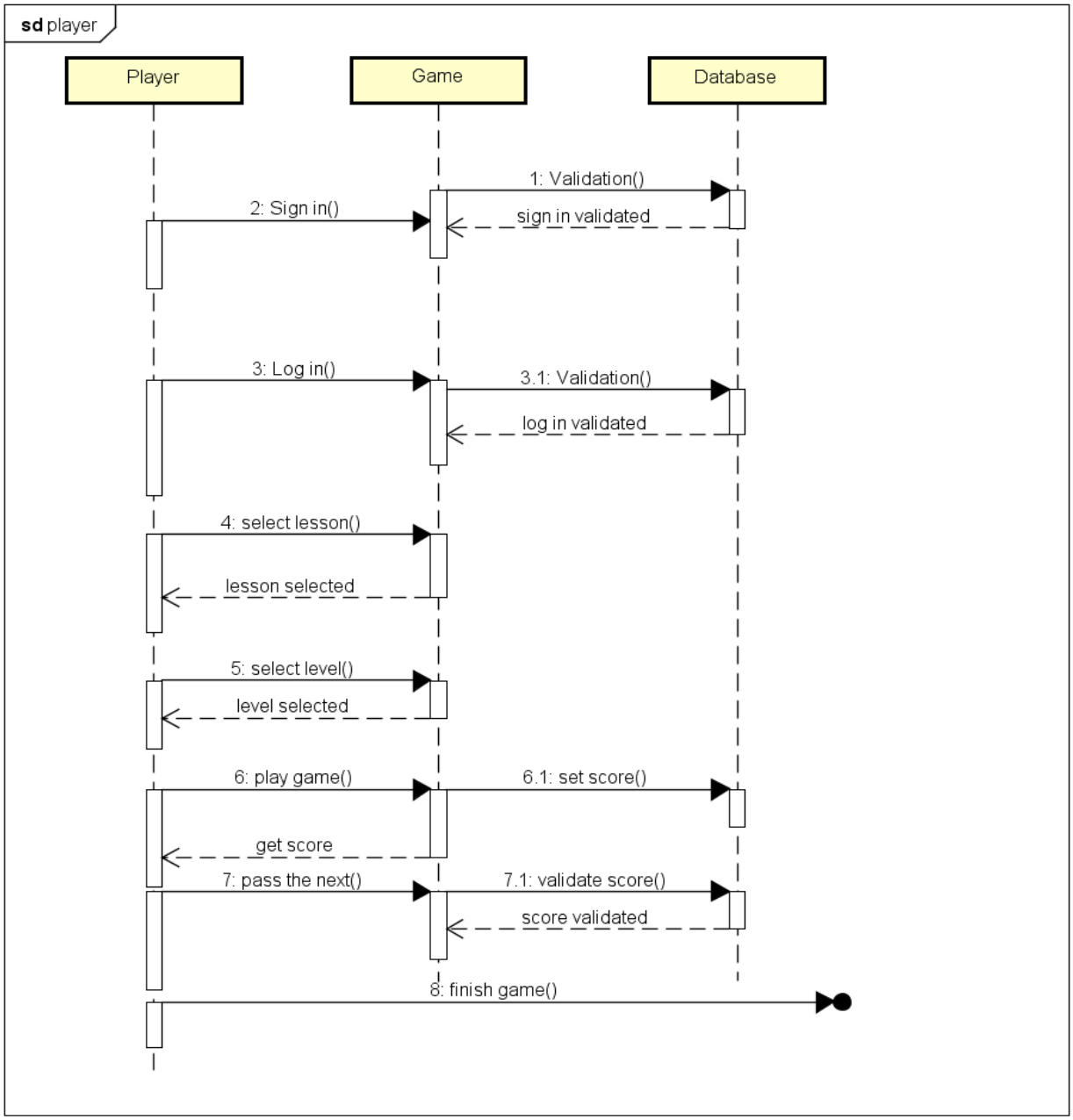

Figure 6: Sequence diagram

III.2 Realization

III.2.1 Java Swing (7)

Swing API is a set of extensible GUI Components to ease the developer's life to create JAVA based Front End/GUI Applications. It is built on top of AWT API and acts as a replacement of AWT API, since it has almost every control corresponding to AWT controls. Swing component follows a Model-View-Controller architecture to fulfill the following criterias.

- A single API is to be sufficient to support multiple look and feel.
- API is to be model driven so that the highest-level API is not required to have data.
- API is to use the Java Bean model so that Builder Tools and IDE can provide better services to the developers for use.

III.2.1.1 Swing Features

III.2.1.1.1 Light Weight

Swing components are independent of native Operating System's API as Swing API controls are rendered mostly using pure JAVA code instead of underlying operating system calls.

III. 2.1.1.2 Rich Controls

Swing provides a rich set of advanced controls like Tree, TabbedPane, slider, colorpicker, and table controls.

III. 2.1.1.3 Highly Customizable

Swing controls can be customized in a very easy way as visual apperance is independent of internal representation.

III. 2.1.1.4 Pluggable look-and-feel

SWING based GUI Application look and feel can be changed at run-time, based on available values.

III.2.1.2 Swing Containers

Following is the list of commonly used containers while designed GUI using SWING.(4)

Sr.No.	Container & Description
1	**Panel** JPanel is the simplest container. It provides space in which any other component can be placed, including other panels.
2	**Frame** A JFrame is a top-level window with a title and a border.
3	**Window** A JWindow object is a top-level window with no borders and no menubar.

Table 1: description of swing

III.2.2 *Derby Database*

III.2.2.1 What is the Derby database?

Apache **Derby**, an Apache **DB** subproject, is an open source relational **database** implemented entirely in Java and available under the Apache License, Version 2.0. Some key advantages include: **Derby** has a small footprint -- about 3.5 megabytes for the base engine and embedded JDBC driver.

III. 2.2.2 Derby JDBC driver

Derby consists of both the database engine and an embedded JDBC driver. Applications use JDBC to interact with a database. Applications running on JDK 1.5 or earlier, must load the driver in order to work with the database.

-In an embedded environment, loading the driver also starts Derby.

-The Derby driver class name for the embedded environment is *org.apache.derby.jdbc.EmbeddedDriver*.

-In a Java application, typically load the driver with the static *Class.forName* method or with the *jdbc.drivers* system property. For example:

```
Class.forName ("org.apache.derby.jdbc.EmbeddedDriver");
Java
 -Djdbc.drivers=org.apache.derby.jdbc.EmbeddedDriver applicationClass
```

+If the application runs on JDK 1.6 or higher, then we do not need to explicitly load the EmbeddedDriver. In that environment, the driver loads automatically.(1)

III.2.3 Methods

In the first frame there are five methods:

III.2.3.1 Method for connection with the database

Code

```
try {

con=DriverManager.getConnection("jdbc:derby://localhost:1527/gameStu
dent;create=true;user=root;password=root");
      } catch (SQLException ex) {

      }
```

I

II.2.3.2 Method for insertion a new user

Code

```
try {
      PreparedStatement st = con.prepareStatement("insert into student
(name,password,score) values (?,?,?)");
      st.setString(1, user);
      st.setString(2, pass);
      st.setInt(3,1);
      st.execute();
      JOptionPane.showMessageDialog(null, "ok");

       System.out.println("ok");
   } catch (SQLException ex) {
       Logger.getLogger(logIN.class.getName()).log(Level.SEVERE, null, ex);}
```

III. 2.2.3 Method for authentication

when the user enters his name and password, this method will get his information and send it to the database than check if it is an existing account or a fake one, if the operation completed successfully, the user can start playing, if it is not he will receive an alert window (create another account or confirm your information).

Code

```
 try {

con=DriverManager.getConnection("jdbc:derby://localhost:1
527/gameStudent;create=true;user=root;password=root");
      } catch (SQLException ex) {
      }

      try {
         PreparedStatement st1
=con.prepareStatement("select name, password from
student where name = ? and password = ?");
         st1.setString(1, user);
         st1.setString(2, pass);
         res = st1.executeQuery();
         while(res.next()) {
            c++;
         }
```

III. 2.2.4 Method for insertion the score

this method gives an initial score (0) to the new user. After each succeed level, the score will increment.

Code

```
try {

            con =
DriverManager.getConnection("jdbc:derby://localhost:1527/gameStudent;create=tru
e;user=root;password=root");
                PreparedStatement st = con.prepareStatement("update student set
SCORE=? where NAME=?");

                st.setInt(1, re);
                st.setString(2, user);
                st.executeUpdate();
            } catch (SQLException ex) {
              System.out.println(ex);
               }
```

III. 2.2.5 Method for getting the score

this method gets the score for an existing user who played before

Code

```
if(c>0) {
              PreparedStatement s = con.prepareStatement("select score from student where
name = ?");
               s.setString(1, user);
              ResultSet rs = s.executeQuery();
              while(rs.next()) {
                 Score = rs.getInt("SCORE");
              }

              Welcom.score=Score;

            name = user;
            password = pass;
            NiveauAdd01.user = name;
             Bienvenue b = new Bienvenue();
             b.setVisible( true);
             this.hide();
         }
         else {
            JOptionPane.showMessageDialog(null, "this account does not exist");
            user ="";
            pass ="";
         }
```

III.2.4 Levels

III. 2.4.1 Levels of addition

III. 2.4.1.1 Level No 1

This is the simplest level in addition, in this level there are group of balls in the left side and others in the right side, the user have to enter the correct answer from keyboard (the sum of all the balls) if the answer is correct he will be able to pass to the next step, if it is not he will stay in the same step until he enter the correct answer.

Code

```
public void result (int r)
        {
                if(number1+number2==r){
            int re = Welcom.score+=2;
             System.out.print(user);
                        oprationImages[3] = new
ImageIcon("images/"+r+".gif").getImage();
                        nextButton.setEnabled(true);
       x=true;
                }else{
                        y=false;
                }
     this.repaint();
}
public void result (int r)
        {
                if(number1+number2==r){
            int re = Welcom.score+=2;

             System.out.print(user);
                        oprationImages[3] = new
ImageIcon("images/"+r+".gif").getImage();
                        nextButton.setEnabled(true);
       x=true;
                }else{
                        y=false;
                }
     this.repaint();
}
```

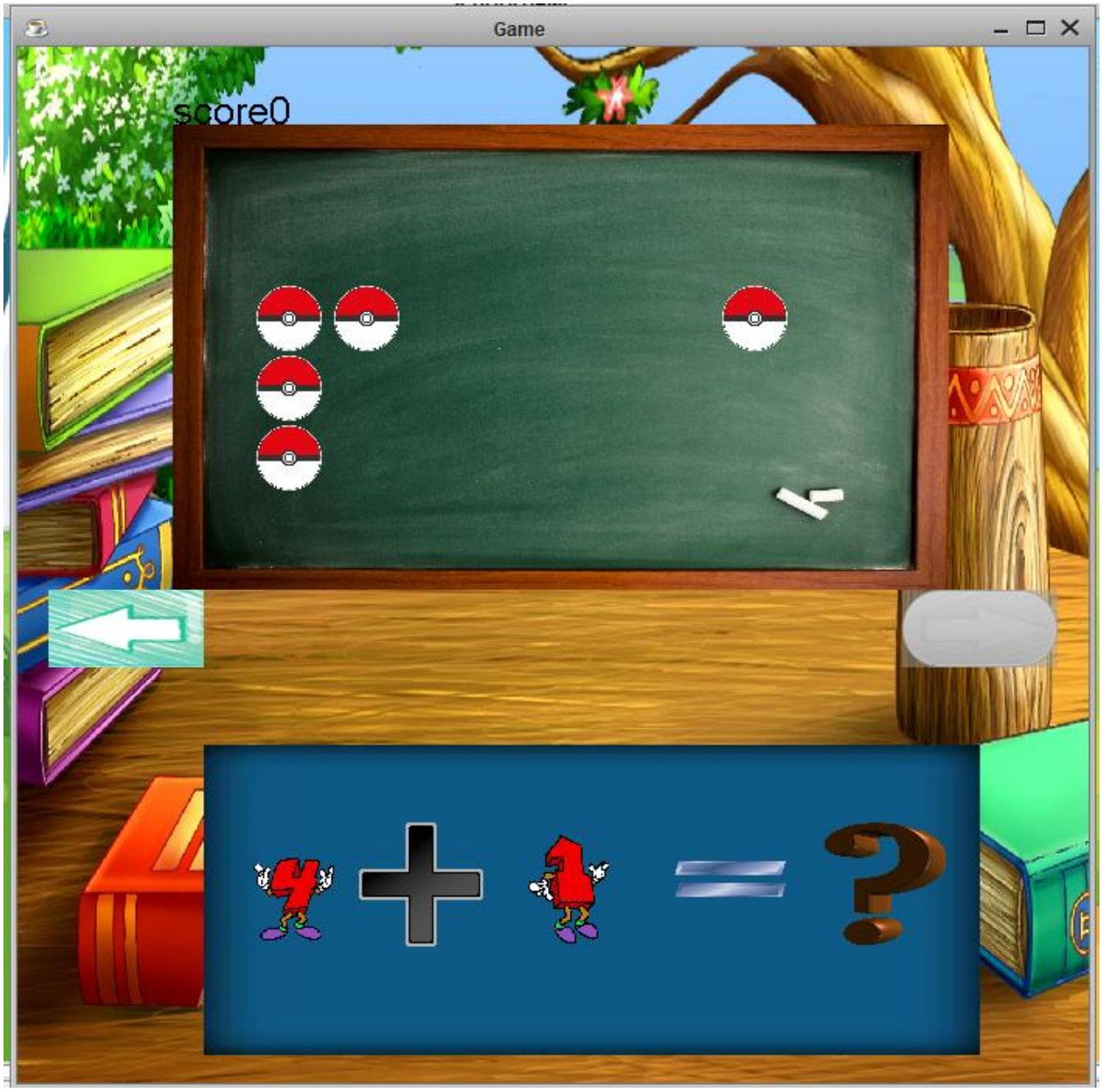

Figure 7: example from first level of addition

III. 2.4.1.2 Level No 2

in this level there is a domino stone and choices in the bottom, the user have to choose

code

```
if(NumberDomino[domino]==NumberDomino[1])
 {
                        g2d.drawImage(NumberImage[6] , START+20 , yop , SIZE ,SIZE , this);
                        g2d.drawImage(NumberImage[7] , START+(SIZE+20)*2 , yop , SIZE ,SIZE , this);
                        g2d.drawImage(NumberImage[5] , START+(SIZE+20)*4 , yop , SIZE ,SIZE , this);
                }
                if(NumberDomino[domino]==NumberDomino[2]) {
                        g2d.drawImage(NumberImage[7] , START+20 , yop , SIZE ,SIZE , this);
                        g2d.drawImage(NumberImage[6] , START+(SIZE+20)*2 , yop , SIZE ,SIZE , this);
                        g2d.drawImage(NumberImage[5] , START+(SIZE+20)*4 , yop , SIZE ,SIZE , this);
}
```

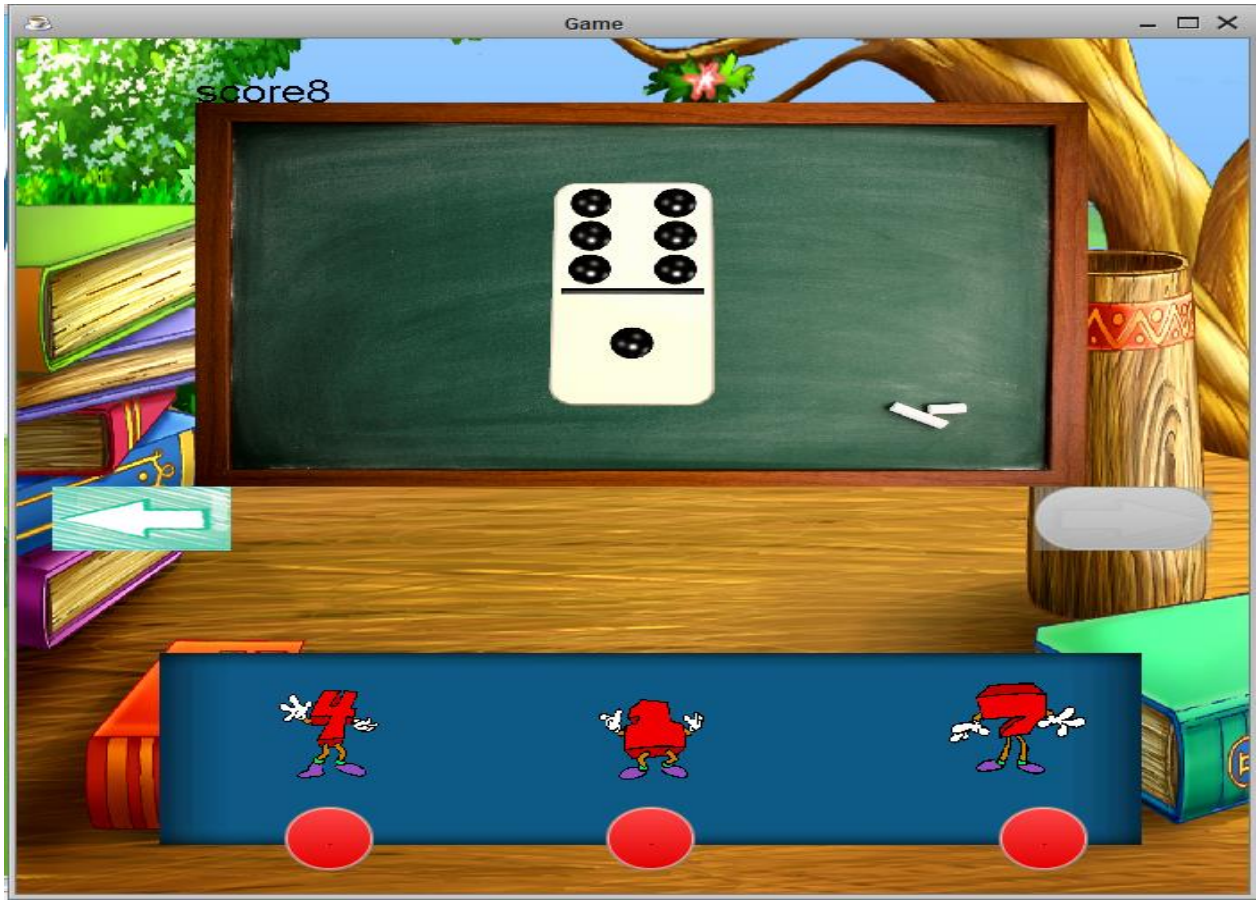

Figure 8: example from Second level of addition

the correct choice corresponding with the number of points in the domino stone.

III. 2.4.1.3 Level No3

in this level , there is a result in the left side and the operations in the right one , beside each operation there is a color . User have to choose the correct color on the bottom of the panel.

Code

```
if(NumberResault[result]==NumberResault[1]) {
                    btn1.addActionListener(e -> {MainNiveauAdd3.Winer();});
                    MainNiveauAdd3.B=true;

}
      if(NumberResault[result]==NumberResault[2]) {
                    btn3.addActionListener(e -> {MainNiveauAdd3.Winer();});
                    MainNiveauAdd3.B=true;

}
     if(NumberResault[result]==NumberResault[3]) {
                    btn2.addActionListener(e -> {MainNiveauAdd3.Winer();});
                    MainNiveauAdd3.B=true;

}
      if(NumberResault[result]==NumberResault[4]) {
                    btn3.addActionListener(e -> {MainNiveauAdd3.Winer();});
                    MainNiveauAdd3.B=true;
}
```

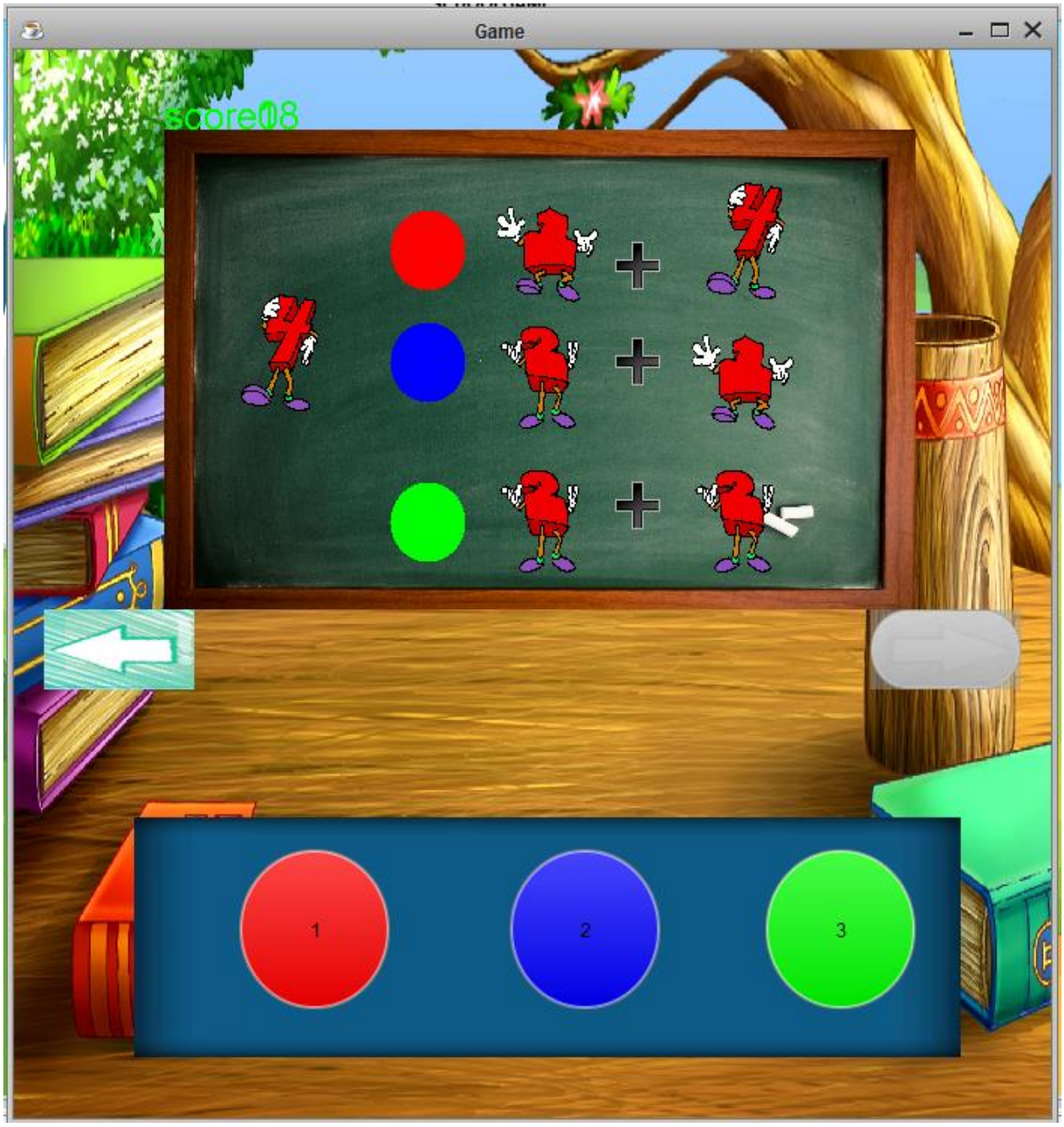

Figure 9:example from third level of addition

III. 2.4.1.4 Level No 4

This is the last level and the hardest one ,the user have to enter the sum between the two numbers existing on the bottom of screen.

Code

```
public void result (int r){
                    if(number1+number2==r) {
                              oprationImages[3] = new
ImageIcon("images/"+r+".gif").getImage();
                              nextButton.setEnabled(true);
                     int re = Welcom.score+=2;}
                    this.repaint();}
public void result (int r){
                    if(number1+number2==r) {
                              oprationImages[3] = new
ImageIcon("images/"+r+".gif").getImage();
                              nextButton.setEnabled(true);
                     int re = Welcom.score+=2;}
                    this.repaint();}
```

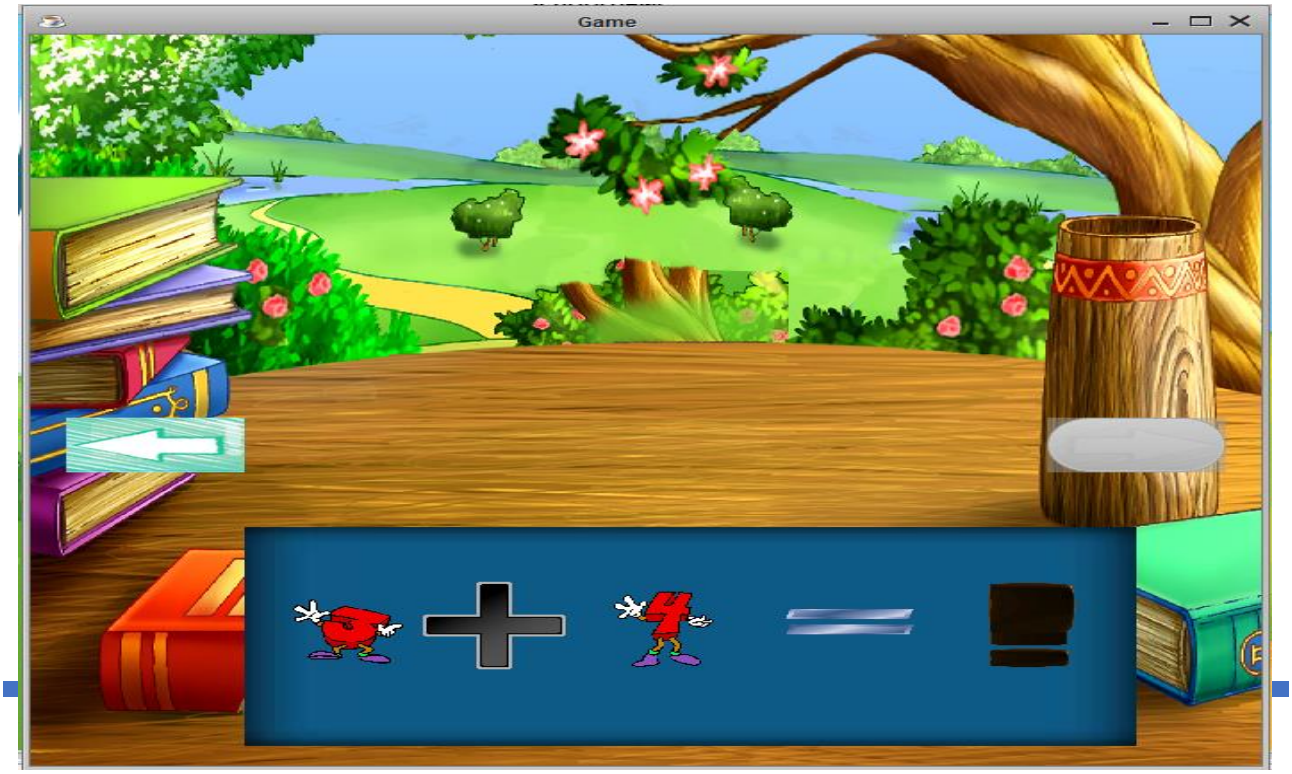

Figure 10 :example from last level of addition

III.2.4.2 Levels of subtraction:

III. 2.4.2.1 Level No 1

in this level there are group of balls in the center of screen and operation in the bottom, the user have to enter the subtraction between the two numbers with the help of balls.

Code

```
public void result ( int r)
          {
              if(number1 - number2==r){
                t=true;

              oprationImages[3] = new
ImageIcon("images/"+r+".gif").getImage();
                   nextButton.setEnabled(true);
                 ImageIcon p =new
ImageIcon("images/spong.gif");
                     spong=p.getImage();

                      int re = Welcom.score+=2;
              Connection con;
              try {
                con =
DriverManager.getConnection("jdbc:derby://localhost:152
7/gameStudent;create=true;user=root;password=root");
                   PreparedStatement st =
con.prepareStatement("update student set SCORE=?
where NAME=?");

                   st.setInt(1, re);
                   st.setString(2, user);
                    st.executeUpdate();
              } catch (SQLException ex) {
                 System.out.println(ex);
              }
              }
      }
```

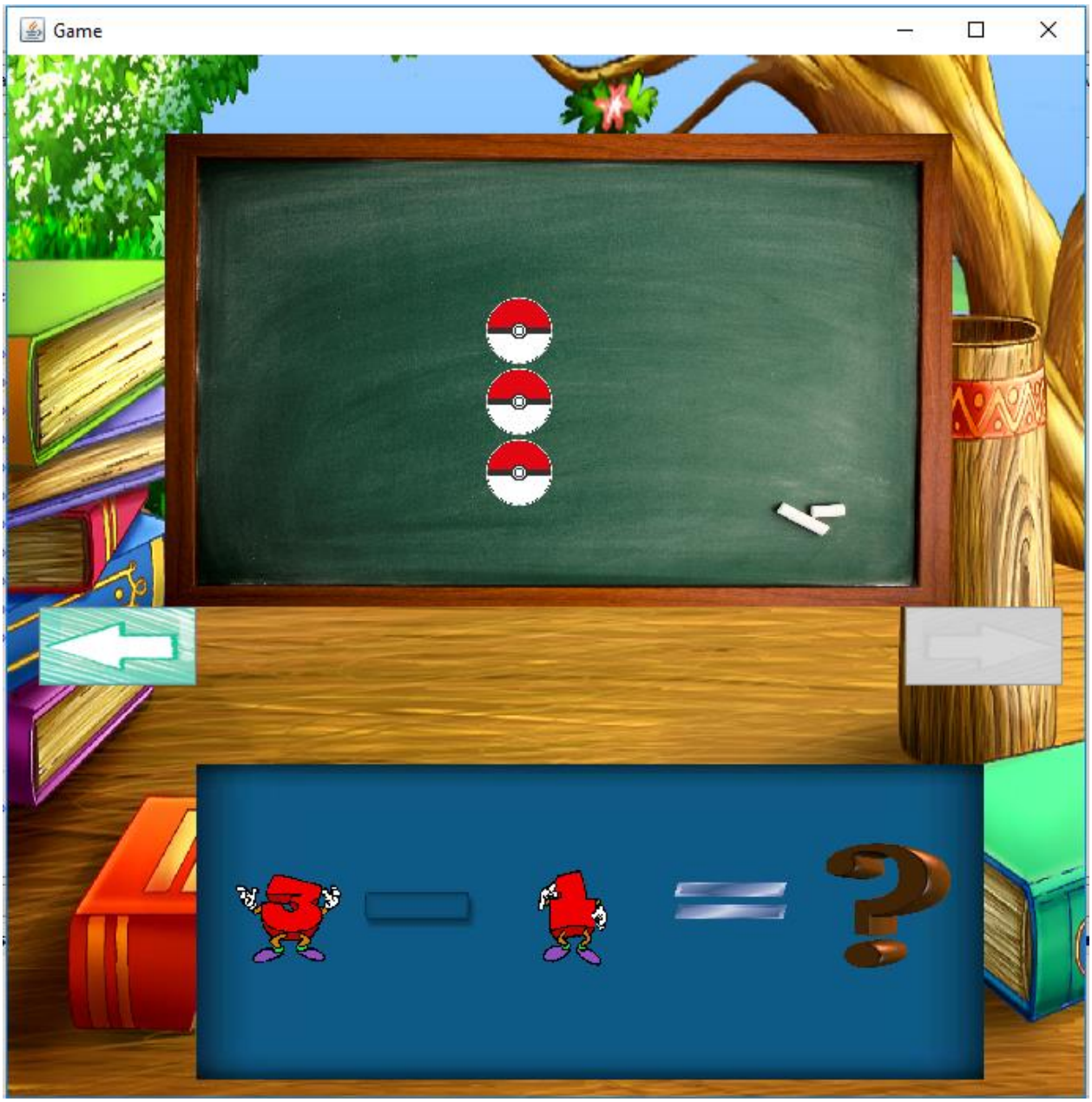

Figure 11: example from first level of subtraction

III. 2.4.2.2 Level No 2

In this level, there is a result in the left side and the operations in the right one , beside each operation there is a color . User have to choose the correct color on the bottom of the screen.

Code

```
if(NumberResault[result]==NumberResault[2]) {
                btn3.addActionListener(e ->
{MainNiveauAdd3.Winer();});
                MainNiveauAdd3.B=true;
        }
            if(NumberResault[result]==NumberResault[3])
{
                btn2.addActionListener(e ->
{MainNiveauAdd3.Winer();});
                MainNiveauAdd3.B=true;
        }
```

Figure12 : example from the second level of subtraction

III. 2.4.2.3 Level No 3

In this level, there is two groups of shapes and a minus sign between them. The Here is to enter the correct answer using the calculator existing in the bottom of the screen.

Code

```
public void result ( int r)
        {
       if(number1 - number2==r){
             t=true;

              oprationImages[3] = new
ImageIcon("images/"+r+".gif").getImage();
                 nextButton.setEnabled(true);
         }
```

Figure13: example from the third level of subtraction

III 2.4.2.4 Level No 4

This is the hardest level ,the user have to enter the subtraction between the two numbers existing on the bottom of screen.

Code

```
public void result ( )
        {
            if(number1 - number2==fois){

        System.out.println("n1:"+number1+ "" + number2);

                nextButton.setEnabled(true);
         }


public void result ( )
        {
            if(number1 - number2==fois){

        System.out.println("n1:"+number1+ "" + number2);

                nextButton.setEnabled(true);
         }
```

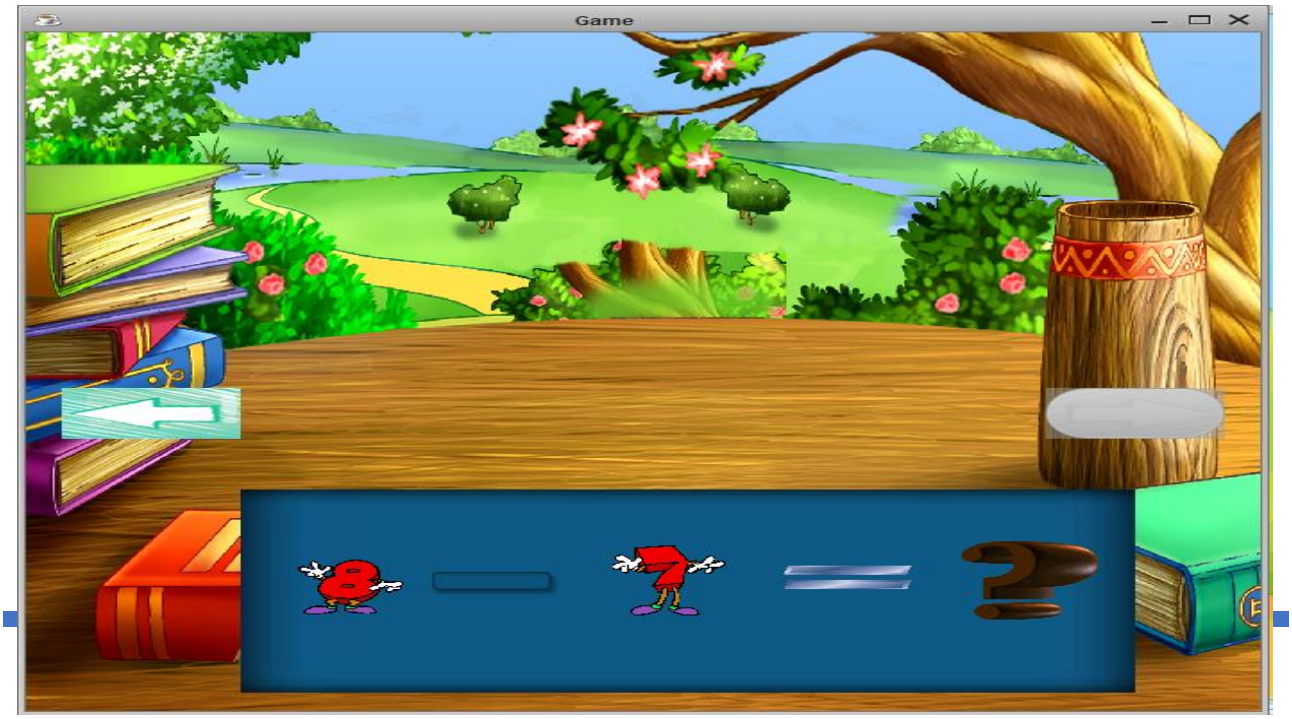

Figure14 :example from the last level of subtraction

III.2.4.3 Levels of complementary:

III. 2.4.3.1 Level No 1

there are two groups of shapes and an operation in the bottom, the user have to enter the complementary of the result

Code

```
public void result ( int r)
        {
    if(number1 - number2==r){
            t=true;
          oprationImages[3] = new
ImageIcon("images/"+r+".gif").getImage();
              nextButton.setEnabled(true);
        }
         }
```

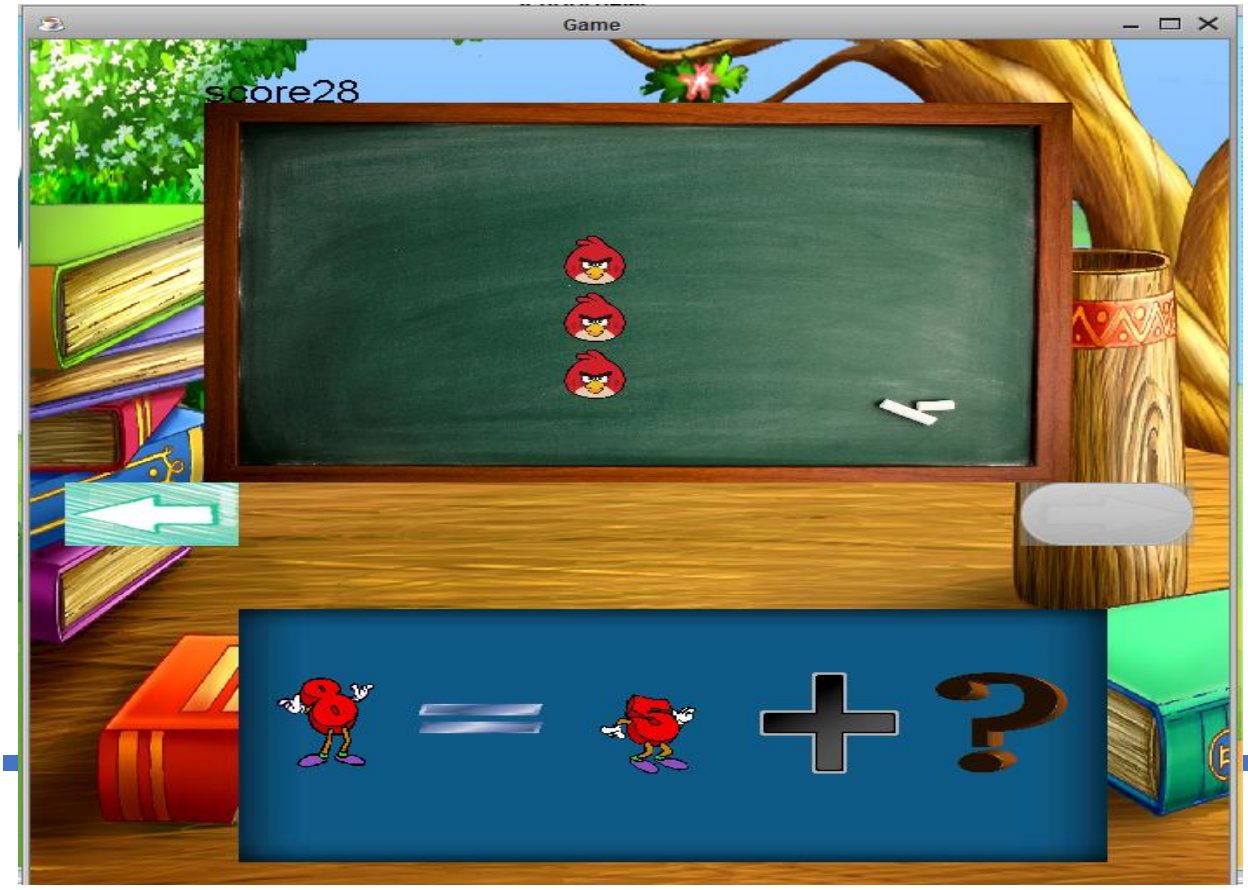

Figure15 :example from the first level of complementary

III. 2.4.3.2 Level No 2

it’s more complicate than the first one , in this level there is just one group of shapes (which represent the result) and the user have to know the complementary according to the result

Code

```
public void result ( int ){
      if(number1 - number2==r){
            oprationImages[3] = new
ImageIcon("images/"+r+".gif").getImage();
                  nextButton.setEnabled(true);
            x=true;
          } }
```

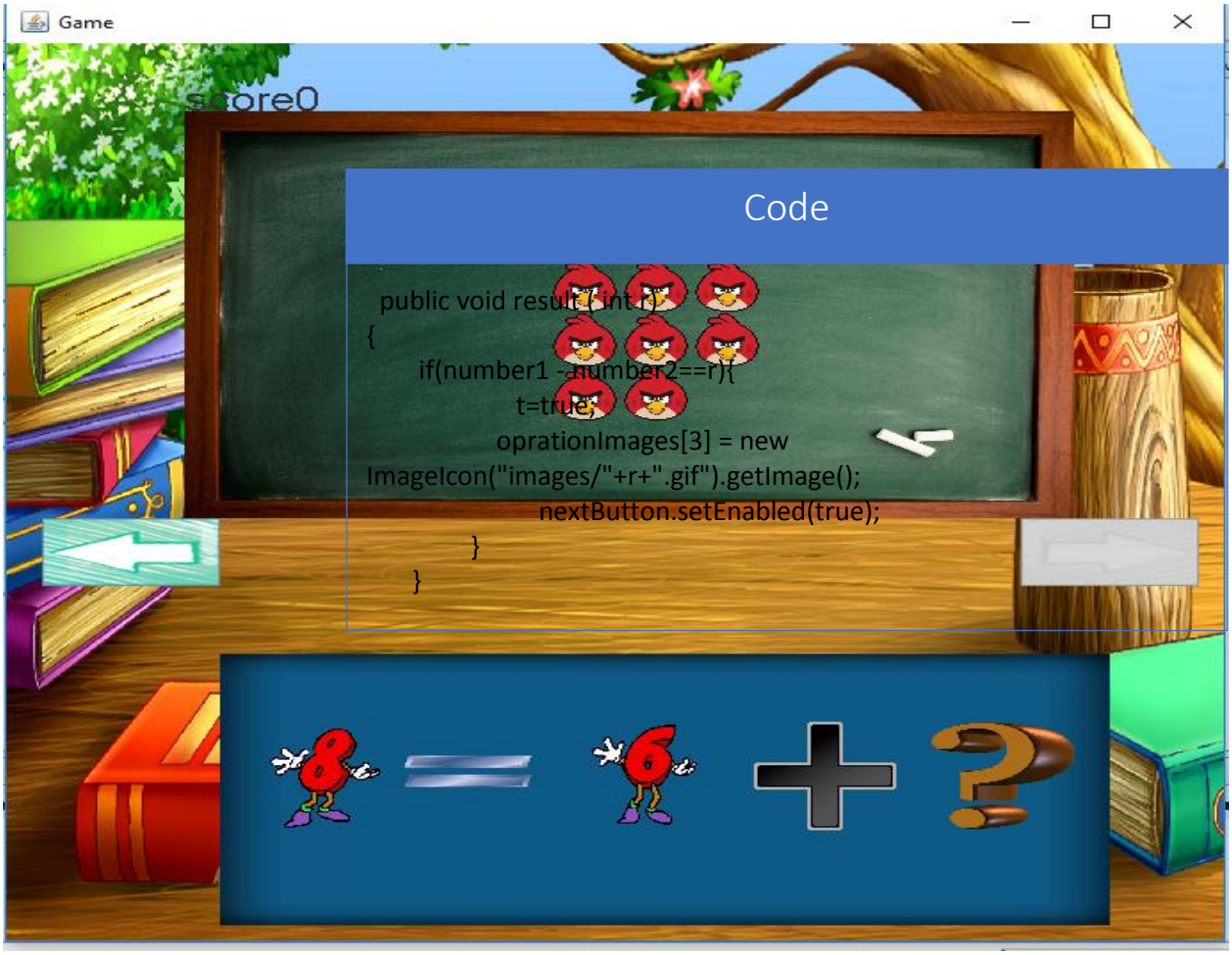

Figure16 :example from the second level of complementary

III. 2.4.3 Level No 3

This is the hardest level ,the user have to enter the complementary between the two numbers existing on the bottom of screen

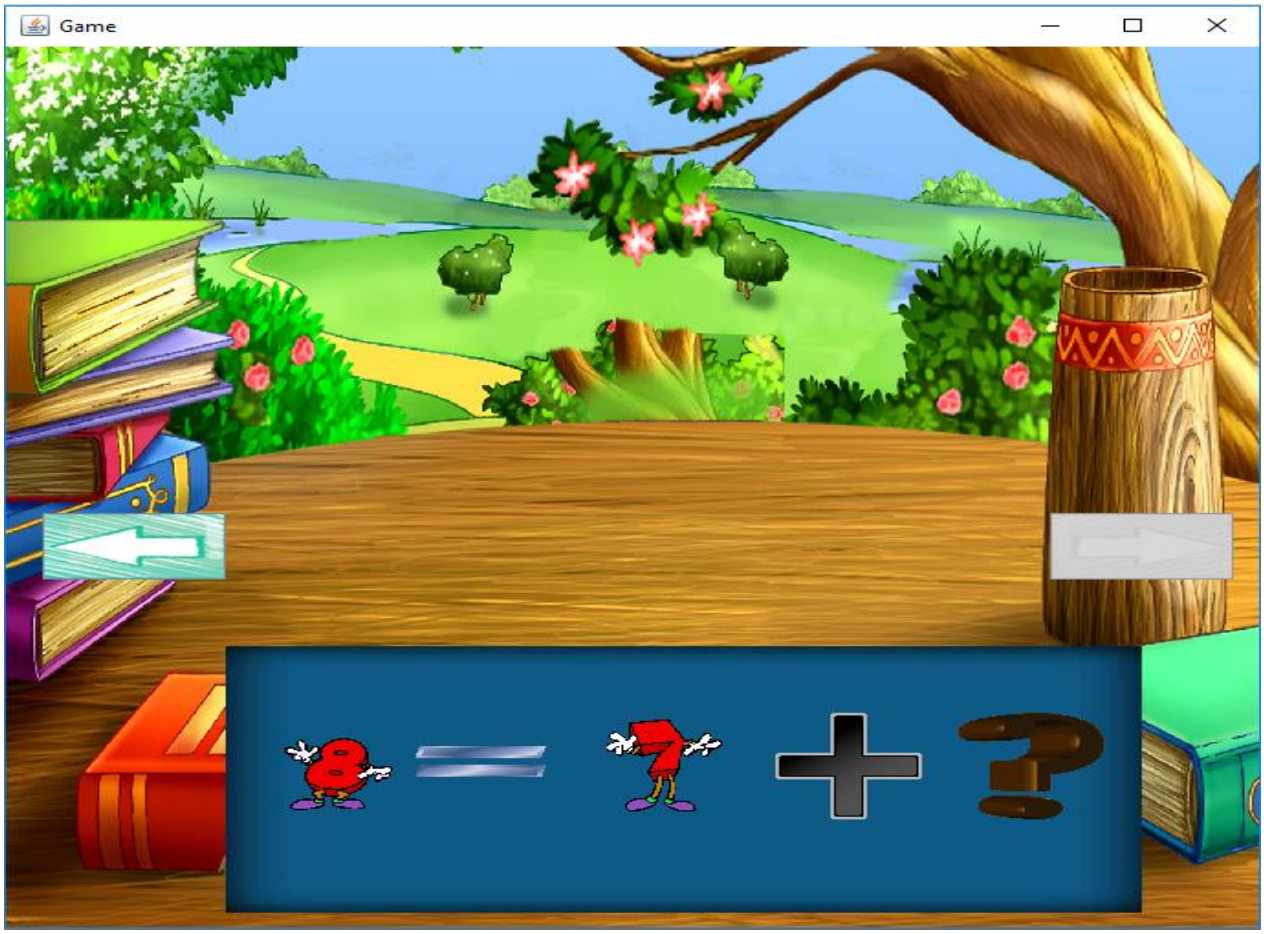

Figure17 :example from the third level of complementary

Conclusion:

In this chapter, we have described the process of implementing of our game by specifying the development environment and we have finished the implementation of all the cases. So, we can say that we produced the last version of the game.

III.3Validation

In the end of this project which we designed an educational game to simplify the methods of understanding.

Existing games	Our game

Advantages	• gorgeous designs • it has hints • it has sounds • usage of a developed environments	• contains all math skills • progression between a level and the other • there are a variation designs in the game • our game has a database
Drawbacks	• a lot of fun instead of learning • sometimes there are noise because of the sound • the most of games have not a database	• there is not a sound to help the player • usage of a simple environments • it has a lot of frames

Table 2: comparison between our games and existing game

General conclusion:

In the chapters, we explained the steps of conception and realization of this application, all of these based on the UML to medialization.

This project helped us to use our skills and knowledge and improve it, especially in games development. Thus, we learned more about Java SWING and Derby Database.

The main future goals of project are:

-Add more levels with a lot of fun.

- Generalization our game for all the education levels.

-Making this game on all other platforms (android, web).

-Develop our source code and make it more readable.

-Publish our game in all schools around our country.

References

[1]: Aldridge, S. & Badham, V. (1993). Beyond just a game. Pamphlet Number 21. Primary Mathematics Association.

[2]: Davies, B. (1995). The role of games in mathematics. Square On. Vol.5. No. 2

[3]: Gough, J. (1999). Playing mathematical games: When is a game not a game? Australian Primary Mathematics Classroom. Vol 4. No.2

[4]: https://db.apache.org/derby/

[5]: https://www.education.com/

[6]: https://www.ixl.com/math

[7]: https://www.tutorialspoint.com/swing/

[8]: https://www.smartdraw.com/