Snake Game

Developed by Abdukarim Avazi

[Github@Snake Game]

ACKNOWLEDGEMENT

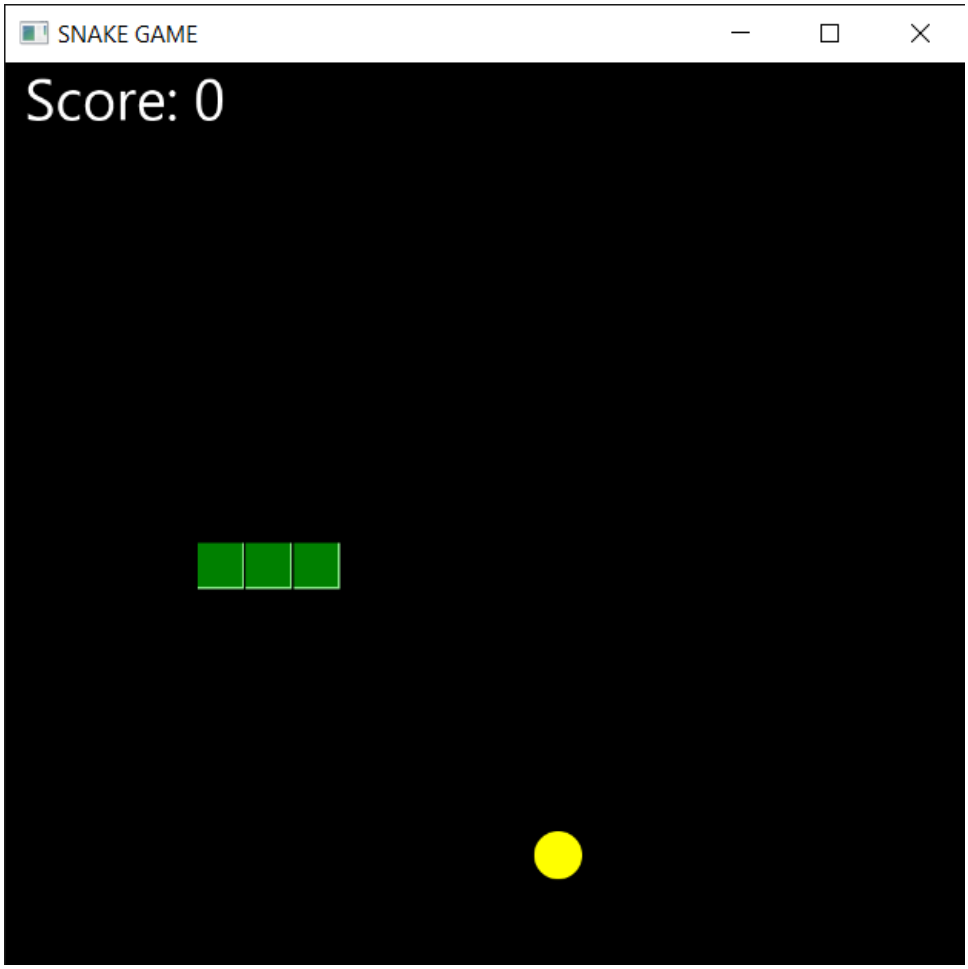
First of all, I would like to thank my lecturer Nurlan Shaidullayev for helping me to acquire some basic knowledge of “Java Programming Language”. At the same time, he gave me the opportunity to learn something new related to our module like constructors, methods, arrays, JFrames etc. Beside from my lecturer, I like to thank my other classmates for helping to understand the assignment related questions more clearly. They gave their best for completing this report on time. I thank them for their efforts.

INTRODUCTION

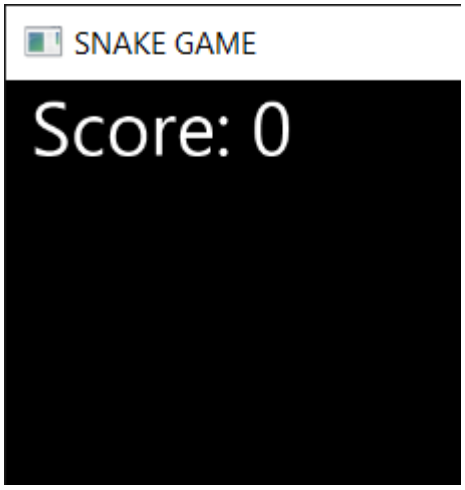
This assignment is based on developing a GAME using “Java Programming Language”. For that we used GUI (Graphical User Interface) in this development so that it will become more users friendly to interact. Besides, Snake helps to relieve fatigue at the time when you are boring.

EXPLANATIONS

Step 1. When we open a new the program we will see a window, where you should start play snake game. For playing, user should hit buttons up, down, left, right.



Step 2. In that window also you can see one more thing, score: 0. It will increase when snake eats a ball, and also the length of snake also gets longer.It is easy to understand and all thing clearly for users, because nothing else which can make people confused.



Step 3. This is the last window in my project which shows than my project is completed. It shows GAME OVER, happens when the snake eats themself or when he passes border of window.

CODE EXPLANATION

In object-oriented programming, for example, an object is a self-contained entity that consists of both data and procedures to manipulate the data. In other way, object oriented is the software engineering concept where it is represented using the “OBJECTS”. Below are the objected oriented samples we used in this “Java Programming Language”:

Sample 1: In this part of code was written the characteristics of snake as you can see speed, width and others.

```
public class Main extends Application {
    // variable
    static int speed = 5;
    static int foodcolor = 0;
    static int width = 20;
    static int height = 20;
    static int foodX = 0;
    static int foodY = 0;
    static int cornersize = 25;
    static List<Corner> snake = new ArrayList<>();
    static Dir direction = Dir.left;
    static boolean gameOver = false;
    static Random rand = new Random();

    public enum Dir {
        left, right, up, down
    }
```

Sample 2. In this peace of code we are controlling snake's width, height, cornersize when he eat each ball, and appears new ball. Moreover, there is also written code of buttons which controls the direction of snake.

```
public void start(Stage primaryStage) {
    try {
        newFood();

        VBox root = new VBox();
        Canvas c = new Canvas(width * cornersize,
height * cornersize);
        GraphicsContext gc = c.getGraphicsContext2D();
        root.getChildren().add(c);

        new AnimationTimer() {
            long lastTick = 0;

            public void handle(long now) {
                if (lastTick == 0) {
                    lastTick = now;
                    tick(gc);
                    return;
                }

                if (now - lastTick > 1000000000 /
speed) {
                    lastTick = now;
                    tick(gc);
                }
            }

        }.start();
```

```
    Scene scene = new Scene(root, width * cornersize,
height * cornersize);

// control
        scene.addEventFilter(KeyEvent.KEY_PRESSED, key
-> {
            if (key.getCode() == KeyCode.W) {
                direction = Dir.up;
            }
            if (key.getCode() == KeyCode.A) {
                direction = Dir.left;
            }
            if (key.getCode() == KeyCode.S) {
                direction = Dir.down;
            }
            if (key.getCode() == KeyCode.D) {
                direction = Dir.right;
            }

        });

        // add start snake parts
        snake.add(new Corner(width / 2, height / 2));
        snake.add(new Corner(width / 2, height / 2));
        snake.add(new Corner(width / 2, height / 2));
```

```
scene.getStylesheets().add(getClass().getResource("").
toExternalForm());
        primaryStage.setScene(scene);
        primaryStage.setTitle("SNAKE GAME");
        primaryStage.show();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

ASSUMPTION

Although we tried to develop this program more user friendly, it has some limitations too. Some of the limitations we have given in details: We did not add any registration function (only GUI) for users. Only the registered users can execute this program. All the details of the registered users have been inserted into the text file. For further implementation of this ATM program, developers need to connect with database. Here we are only using “TEXT FILE”. We did not add any button for deleting the registered users from this program. If admin want to delete the users, he or she needs to open the text file “PASSWORD.TXT” and delete from there. In future we can add more functions like scanning user’s finger print for verification. We can also add the function for third party payment (Users can pay their bills through atm machine).

REFERENCE

BRYANT, J., 2012. JAVA 7 FOR ABSOLUTE BEGINNER. NEW YORK: APRESS. DREAMCODE.NET, 2014. DREAMCODE.NET. [Online] Available at: http://www.dreamincode.net/forums/topic/206344-basic-gui-in-java-using-jframes/ [Accessed 5 December 2014]. EXAMPLES.JAVACODEGEEKS, 2014. EXAMPLES.JAVACODEGEEKS. [Online] Available at: http://examples.javacodegeeks.com/desktop-java/swing/jbutton/set-action-command-forjbutton/ [Accessed 5 December 2014]. HOME&LEARN, 2014. HOME&LEARN. [Online] Available at: http://www.homeandlearn.co.uk/java/read_a_textfile_in_java.html [Accessed 5 December 2014]. JAVA2S, 2014. JAVA2S.COM. [Online] Available at: http://www.java2s.com/Questions_And_Answers/Java-File/text-file/password.htm [Accessed 5 December 2014]. NETBEANS.ORG, 2014. NETBEANS.ORG. [Online] Available at: https://netbeans.org/kb/articles/gui-functionality.html [Accessed 7 December 2014]. ORACLE.COM, 2014. docs.oracle.com. [Online] Available at: http://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.html [Accessed 5 December 2014]. PROGRAMCREEK.COM, 2008. PROGRAMCREEK.COM. [Online] Available at: http://www.programcreek.com/2011/03/java-write-to-a-file-code-example/ [Accessed 5 December 2014]. REALAPPLETS.COM, 2014. REALAPPLETS.COM. [Online] Available at: http://www.realapplets.com/tutorial/actionexample.html [Accessed 5 December 2014]. STACKOVERFLOW, 2011. stackoverflow. [Online] Available at: http://stackoverflow.com/questions/4716503/best-way-to-read-a-text-file [Accessed 6 December 2014].