CMPE 160
Introduction to Object Oriented Programming
Project 02 Project Report

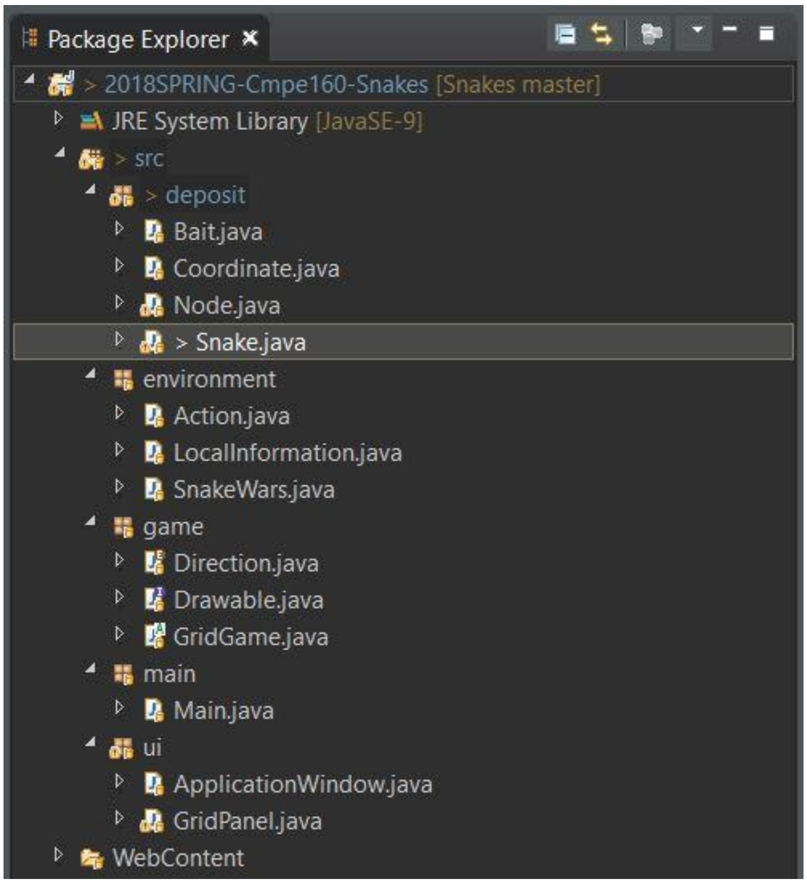
Author: Ibrahim Ozgurcan Oztas
2016400198

Tables of Contents:

1 – Introduction

This project is given to test out skills to implement several classes and use them to create an object-oriented application. On the other hand, our understanding of List<E> class is also tested. With all these features, this program creates a Snake, which is a LinkedList in my program, and a Bait that a Snake can eat and grow. At a certain size, Snake reproduces and increase its quantity. The main objective is every snake can't pass through neither its own body nor other snakes' bodies. Also, all snakes have to move inside the game borders. Hence, there is only one Bait that all snakes can eat, therefore all snakes have to run to the Bait.

2 – Class Hierarchy and Class Description


Package Explorer

```
> 2018SPRING-Cmpe160-Snakes [Snakes master]
    JRE System Library [JavaSE-9]
    > src
        > deposit
            Bait.java
            Coordinate.java
            Node.java
            > Snake.java
        environment
            Action.java
            LocalInformation.java
            SnakeWars.java
        game
            Direction.java
            Drawable.java
            GridGame.java
        main
            Main.java
        ui
            ApplicationWindow.java
            GridPanel.java
    WebContent
```

This picture is a simple illustration of class hierarchy to show the interactions between each class.

** In package deposit, there are 4 classes namely Bait.java, Coordinate.java, Node.java, and Snake.java. In Node.java, I've implemented the simple structure of Node that is different from Java's Node class. My Node.java extends Coordinate.java class that has 2 integer fields "x" and "y" to represent the location of each node in game. In Bait.java class which extends Node.java, there is only a draw() method that makes the bait drawn in the graphic user interface. And the last class, Snake.java, I've implemented a LinkedList<Node> that holds the snake's parts.

** In package environment, there are 3 classes namely Action.java, LocalInformation.java, and SnakeWars.java. Action.java is implemented to synchronize the actions between the game engine and each snake. There is an Enum named Type that has 4 action types named as MOVE, EAT, STAY and REPRODUCE. In LocalInformation.java, all necessary information what snake needs is constructed such as free directions, map's current phase, snake's own location etc. In SnakeWars.java, the game engine is regulated. Every action of every snake occurs if action is necessary or it can be done by snake. All local information is created in this class via a method.

** In package game, there is 1 abstract class named GridGame.java, 1 interface named Drawable.java and 1 enum named Direction.java. In GridGame.java, this is where the game engine exists. All game is controlled inside this class at a changeable frame per second. In Drawable.java, there is only one method that gives ability to the classes which have implemented Drawable.java be drawn in graphic user interface. In Direction.java, there is only four values namely DOWN, UP, RIGHT and LEFT.

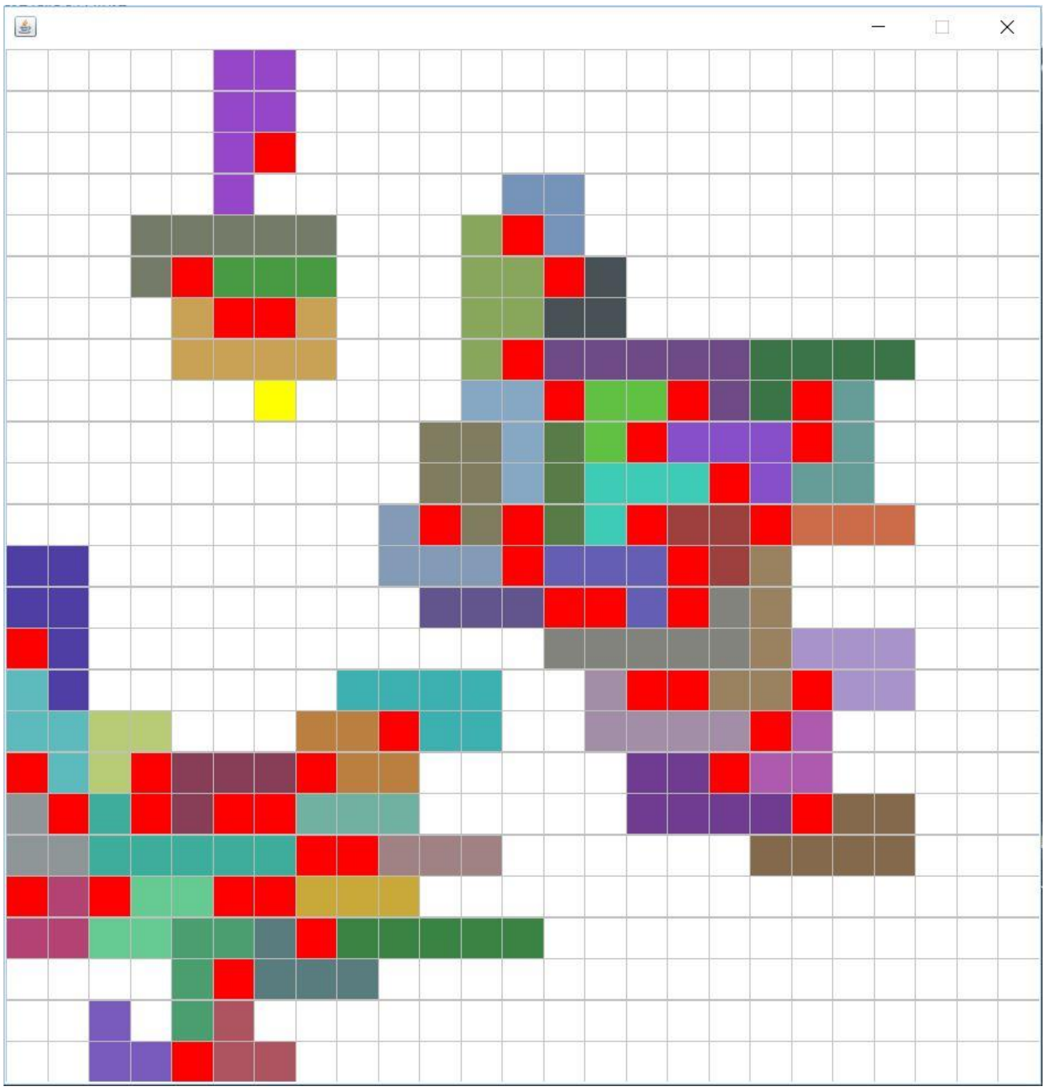
** In package main, there is only one class which is Main.java that executes the game. Also, there is the project description inside this as PDF.

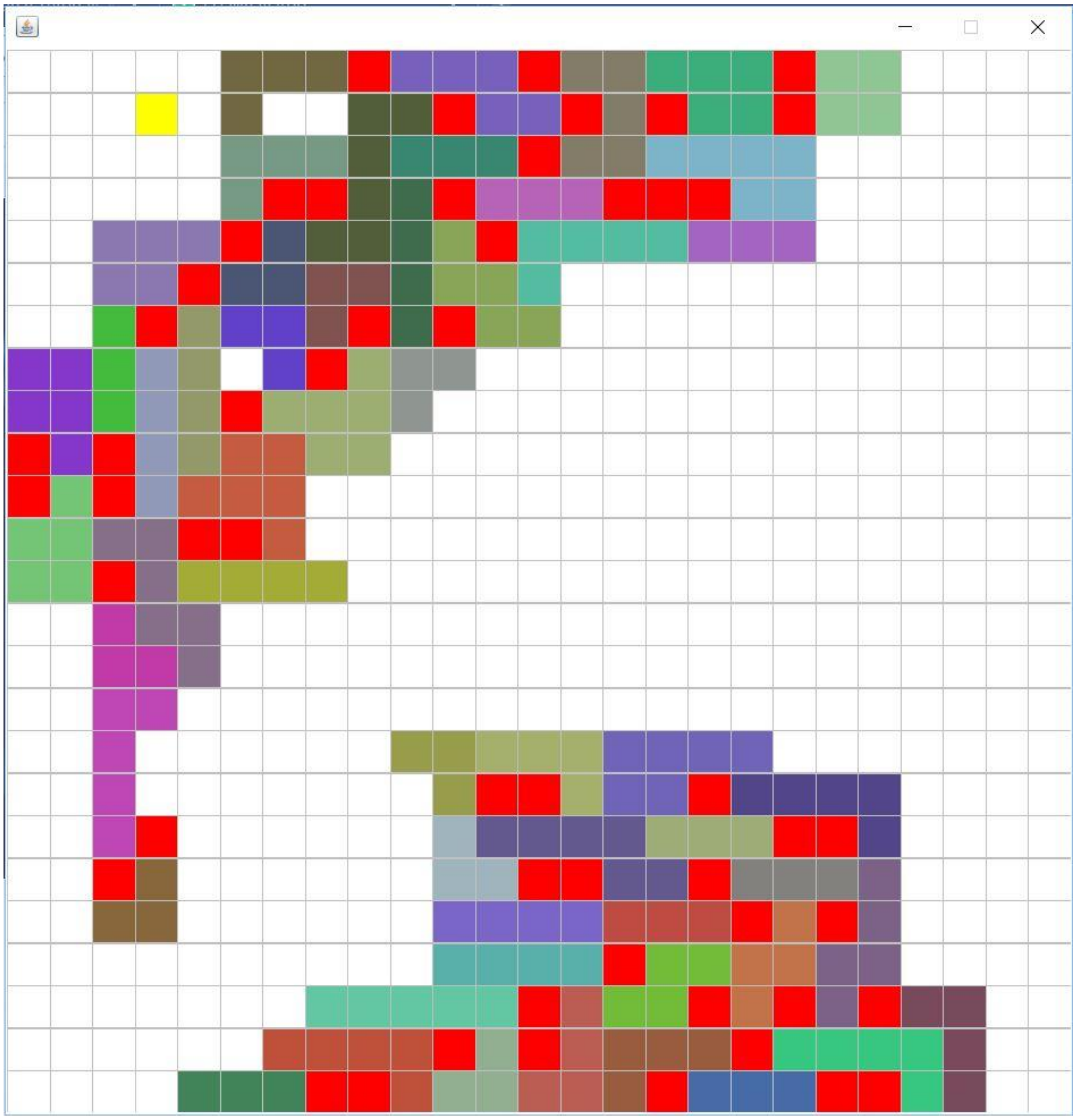
** In package ui(user interface), there is 2 classes namely ApplicationWindow.java and GridPanel.java. In ApplicationWindow.java, I've implemented JPanel and GridPanel. In GridPanel.java, all graphic components are set and used to illustrate the game to the screen.

3 – AI Implementation

In my program, I’ve devised an AI that simply handles the snake’s basic actions. At the given size, snake does reproduce by simply copying the values of last 4 node to a new LinkedList<Node> and returns a new Snake with the new LinkedList<Node>. At a certain condition which is when the absolute distance between snake’s head and the bait is 1, the snake performs the eat action. Also, with the location of bait, snake moves to the bait. It checks the possible difference between its head and the bait with x and y components and acts towards a sensible direction. Therefore, at a certain time, the snake will reach the bait. But all else fails, which means there is no option for snake to act, it stays where it is.

4 – Project Illustration





Note 1: The red squares are heads of the snakes, the yellow square is the Bait, the other colors are representing the snakes’ bodies.

Note 2: The game speed and the magnitude of the dimensions can be adjusted in Main.java such as;

```
SnakeWars game = new
SnakeWars(gridWidth, gridHeight,
gridSquareSize, gameSpeedpersecond);
```

5 – Conclusion

In the conclusion, I can say that this project helped me to understand the basis of OOP and the depth meaning of List class and the situations which I can use. And also it gave me a simple idea of GUI(graphic user interface) and how to use it.

6 – Self-Criticism

In this project, I've seen that my understanding of such concepts is inadequate and inefficient. I've always seen the complicated ways of solving the problem whereas the simplest ways are just in front of my sight with my bare eyes. For that, in the coding process, I've deleted my project core more than 5 times and always make myself discouraged and helpless. I've made my vision blurred and vague. After all, the best thing that this project had made me learn is the fact that everything could be done from scratch by scratch only if I were willing to do it in that way. For that, I am grateful that this kind of a project is given. I believe that I can code with much more self-confidence and clarity. And I want to say thank you whomsoever had contributed to create a project like that one for the things I've learned during the process.