FOOP Final Project

B07902082, 黃哲維
B08902001, 龍品瑞
B08902004, 陳詩淇
B08902134, 曾揚哲

July 1, 2021

Contents

1 Work Division

黃哲維：研究Java Swing, 書面報告
龍品瑞：研究Java Swing, gameflow, Roads
陳詩淇：Roads, Hero, gameflow, 書面報告
曾揚哲：Vehicles, gameflow, graphics, 書面報告

2 UML - Relation between Classes

1. App
   It holds the Main method and it creates the window which is for the game's further implementations.
2. Window
   It sets up the attributes of our game's window and the panel it is currently running.
3. StartPanel
   Note that all 3 of our Panel classes extend the JPanel class from Java Swing package which we use to primarily produce the on screen animations.
   This is the first panel shown to the game once the program is running where the player can choose to press enter to start or simply click the start button. There is a separate class for the start button called "StartButton" that is extended from JButton.
4. GamePanel
   This is where we put our gameflow and set up the input.
   After clicking the start buttor, we enter the playing part of our game which is performed on this Panel. The main character "Hero", moving "Blocks", and "Roads" are presented on this panel, continuously doing their actions by a Timer method.
5. EndPanel
   This is the game over screen which displays the score and also has a JButton called "restart" for the player to click on to retry the game.
6. Road
   It represents one row on our window, which might be a road, land, or water. Roads and lands are passable while water isn't. These different "Road" objects can hold different "Blocks" on it due to their type.
7. Block
   Block includes a "Tree", a running "Car", or a moving "Bridge" (actually it is a driftwood). Each "Block" object has its own velocity and direction.
8. Tree
   It is one kind of a "Block" is generated only on a Land. It does not move in the game or in the real world.
9. Car
   It is one kind of a "Block" and is generated only on a Road. It can be generated randomly from 4 different colors, also since we don't want to see a car driving backwards, we use different images when the car appears from the left or from the right.
10. Bridge

It is one kind of a "Block" and is generated only on "Water". Each bridge has its own weight and is randomly decided when generated.

11. Hero
    This is the main character. The function to decide if it collide a block is in this class.
12. MoveAction
    This is the class where the actions are done after we press a key. The possible actions include moving the main character, hit by a car or fall into the water and go to the "EndPanel".
13. ScoreBoard
    This is where we calculate the total steps our main character has moved forward.

3 Advantages of Our Design

1. We separate each panel so that it would be easy if we'd like to further extend some other features.
2. Open sourced code so that we do not need to worry about the OCP aspects.
3. By playing this game, we are able to see Professor Lin even when the semester is over, which is really enjoyable.

4 Disadvantages of Our Design

1. Our project doesn't entirely follow OCP design, which has been mentioned throughout the whole semester.
2. Some arguments are tested and hardcoded so it would be difficult to fit other window sizes.

5 Packages

1. Java Swing on JDK 16 [1]

6 How to Play

Run `make` and the program would run.
As a traditional game, the player can confirm to start the game by pressing enter or using the cursor to click start. To control the hero, the keys, W, A, S, D are used to move up, left, down, right respectively. If the player crashes into an obstacle such as a car, or falls into the river, the game is over and the game over screen is displayed along with the score. The player may press the play button to retry the game from the beginning.

References

[1] Java Swing: https://docs.oracle.com/en/java/javase/16/docs/api/index.html