**Spring 2022 – CSE1242 Computer Programming II Project Report**

**Java Game**

**Authors**

**İlker Keklik 150120074**

**Yusuf Duman 150120023**

**Course**

**CSE1242 Computer Programming II, Spring 2022**

**Date**

**May 09, 2022**

**Problem Definition**

This project is a Java Game that created with using JavaFX. In this game there are varius types of tile used in 4x4 square Pane. In each tiles, there is a pipe path that uses for traveling the ball. If the path is in the right place from StarterTile to EndTile it means the game is finished and it displays an animation which the ball goes to EndTile from StarterTile by following the path. Then opens next level to be resolved. We completed required parts of the project.

**Implementation Details**

We designed our algorithm like that:

We have 3 classes wich extends Imageview and implements Tile interface which we created. Tile interface has get and set methods to get up, down, right, left variables for each image in the game level. According to input file, we check which image to add into given box number and we add images into objects which is initialized from one of our 3 classes (StarterTile, UsualTile and EndTile). In this way, our level is ready.

Then we coded SetDragDetect and SetFreeDrop methods. Then we added functionality of dragging into objects that are allowed to move and added functionality of being able to dropped on into Empty Free tiles using those methods.

Every imageview has its boolean up,down,right,left properties and they are true or false depanding on which direction the image is. For example if a image has way inside from left to right, it has right = true , left = true ; up = false, down = false.

Using this we checked if the games is finished after every move. In our isFinished method we controlled if imageviews up, down, right, left properties match. For example, if starterTile is vertical it has only down=true so in our control algorythm, it checkes the box below the starterTile and if it sup property is true, they match and this loop continues until EndTile.

If loop arrives EndTile Level is completed so next level is uploaded.

**javafx.scene.image.ImageView**

**<<interface>>**

**Tile**



**StarterTile**

**UsualTile**

**EndTile**

1)

|  |  |
| --- | --- |
|  | *<<interface>>*  *Tile* |
| +  +  +  +  +  +  +  + | *getUp():boolean*  *getDown():boolean*  *getRight():boolean*  *getLeft(():boolean*  *setUp(up:boolean):void*  *setDown(down:boolean):void*  *setRight(right:boolean):void*  *setLeft(left:boolean):void* |

2)

|  |  |
| --- | --- |
|  | StarterTile |
| -  -  -  - | up:boolean  down:boolean  right:boolean  left:boolean |
| +  +  +  +  +  +  +  +  + | StarterTile(tileImage:Image, up:boolean, down:boolean, right:boolean, left:boolean)  getUp():boolean  getDown():boolean  getRight():boolean  getLeft(():boolean  setUp(up:boolean):void  setDown(down:boolean):void  setRight(right:boolean):void  setLeft(left:boolean):void |

3)

|  |  |
| --- | --- |
|  | EndTile |
| -  -  -  - | up:boolean  down:boolean  right:boolean  left:boolean |
| +  +  +  +  +  +  +  +  + | EndTile(tileImage:Image, up:boolean, down:boolean, right:boolean, left:boolean)  getUp():boolean  getDown():boolean  getRight():boolean  getLeft(():boolean  setUp(up:boolean):void  setDown(down:boolean):void  setRight(right:boolean):void  setLeft(left:boolean):void |

4)

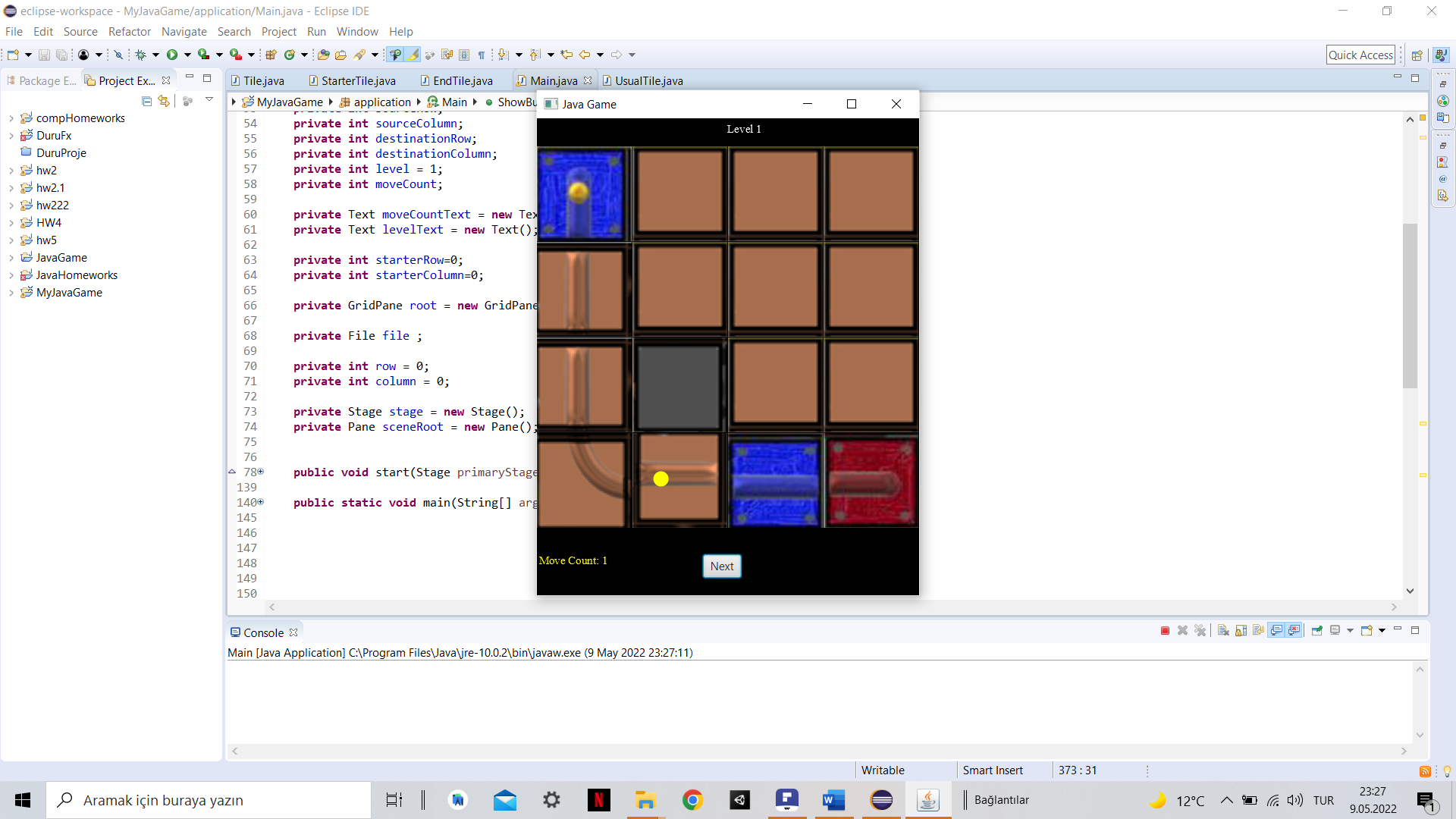
|  |  |
| --- | --- |
|  | UsualTile |
| -  -  -  - | up:boolean  down:boolean  right:boolean  left:boolean |
| +  +  +  +  +  +  +  +  + | EndTile(tileImage:Image, up:boolean, down:boolean, right:boolean, left:boolean)  getUp():boolean  getDown():boolean  getRight():boolean  getLeft(():boolean  setUp(up:boolean):void  setDown(down:boolean):void  setRight(right:boolean):void  setLeft(left:boolean):void |

5)

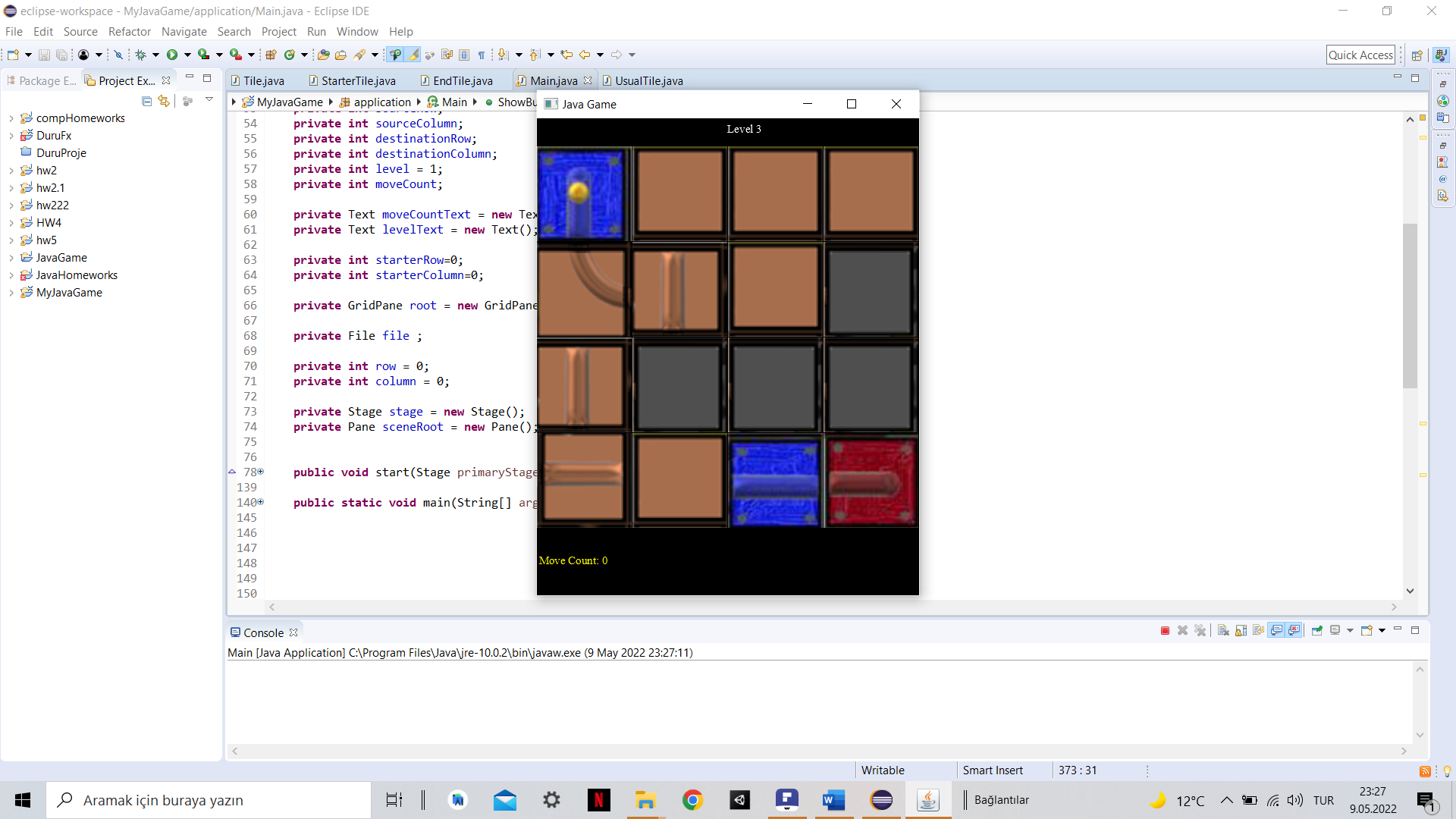
|  |  |
| --- | --- |
|  | Main |
| -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - | arrayList: ArrayList<Tile>  rows: ArrayList<Integer>  cols: ArrayList<Integer>  button: Button  source: ImageView  destination: ImageView  sourceRow: int  sourceColumn: int  destinationRow: int  destinationColumn: int  level: int  moveCount: int  moveCountText: Text  levelText: Text  starterRow: int  starterColumn: int  root: GridPane  file: File  row: int  column: int  stage: Stage |
| +  +  +  +  +  +  +  +  +  +  +  +  +  + | start(primaryStage: Stage):void  main(args: Strings[]):void  SetDragDetect(imageView: ImageView):void  SetFreeDrop(imageView:ImageView):void  Move():void  isFinished():boolean  startAnim ():void  findViewFromCoordinates(col:int,row:int):Tile  LoadLevel(path: String):void  SetLevel():void  ShowButton():void |

**Test Cases**

The animation ball goes in correct path.



It uploads the level correctly according to the input file.



Move count works properly and Next button appears once game is finished.

