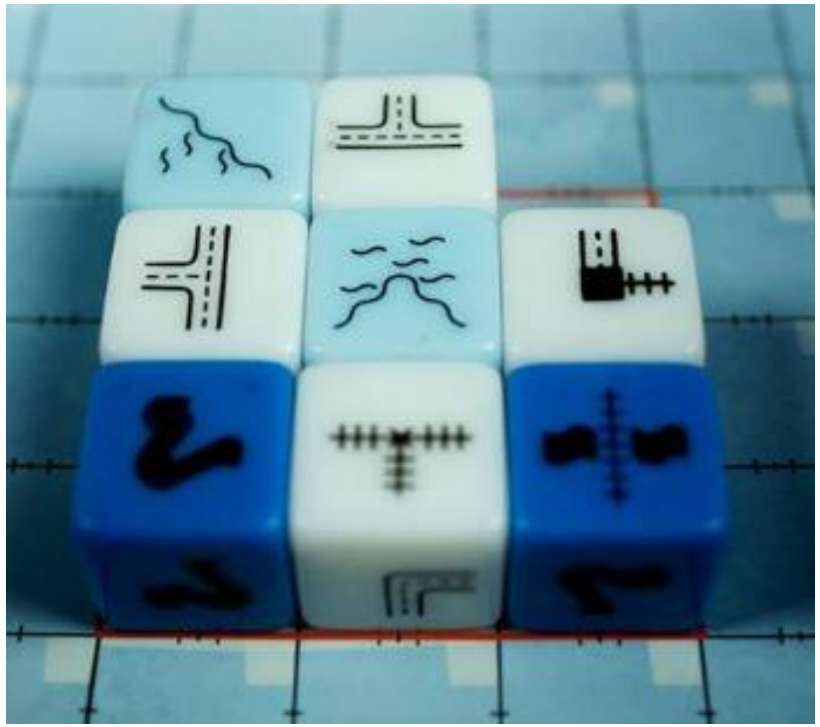
Railroad Ink: Deep Blue Edition



Group Wed09a

Wyman Wong - u6726234

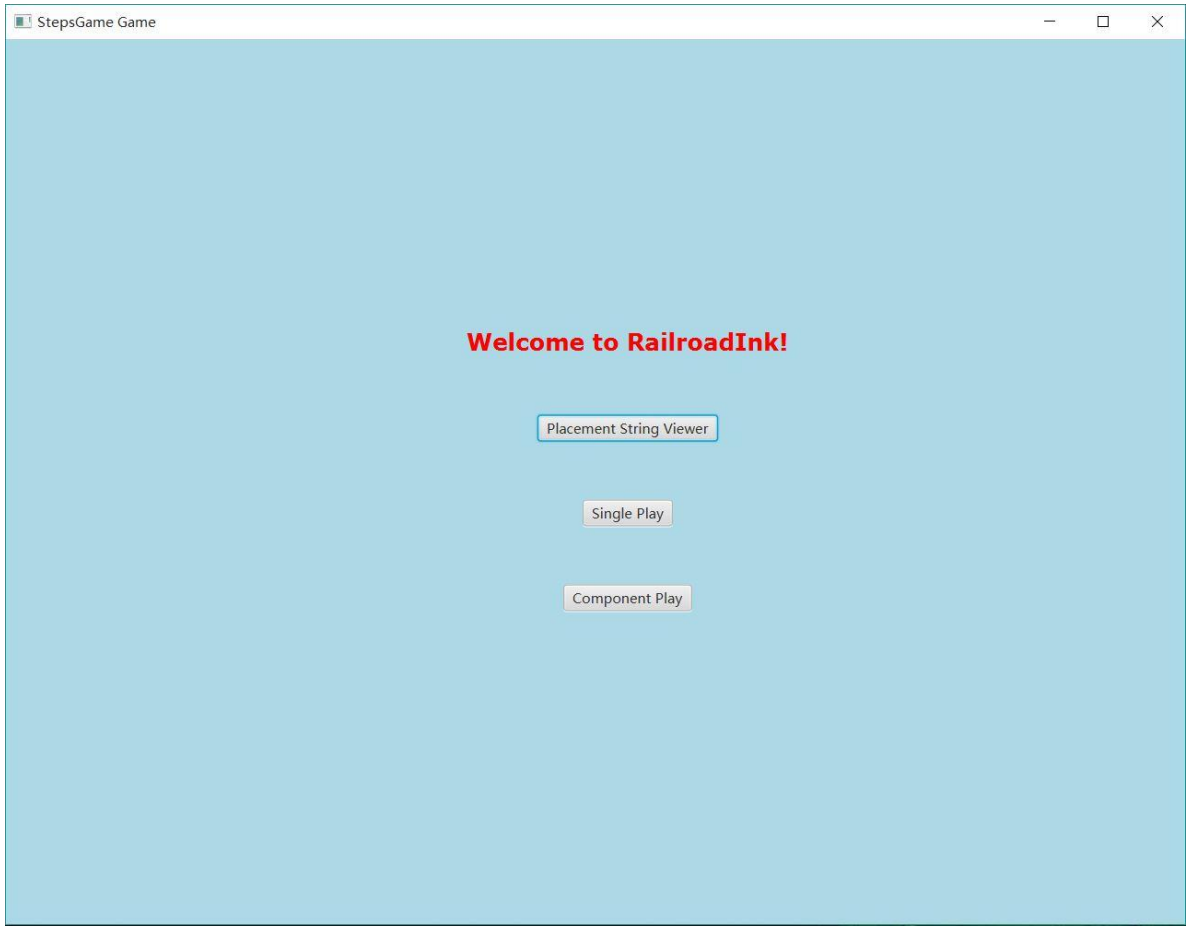
Senyuan Zhang - u6571614

Zichuan Ding - u6156881

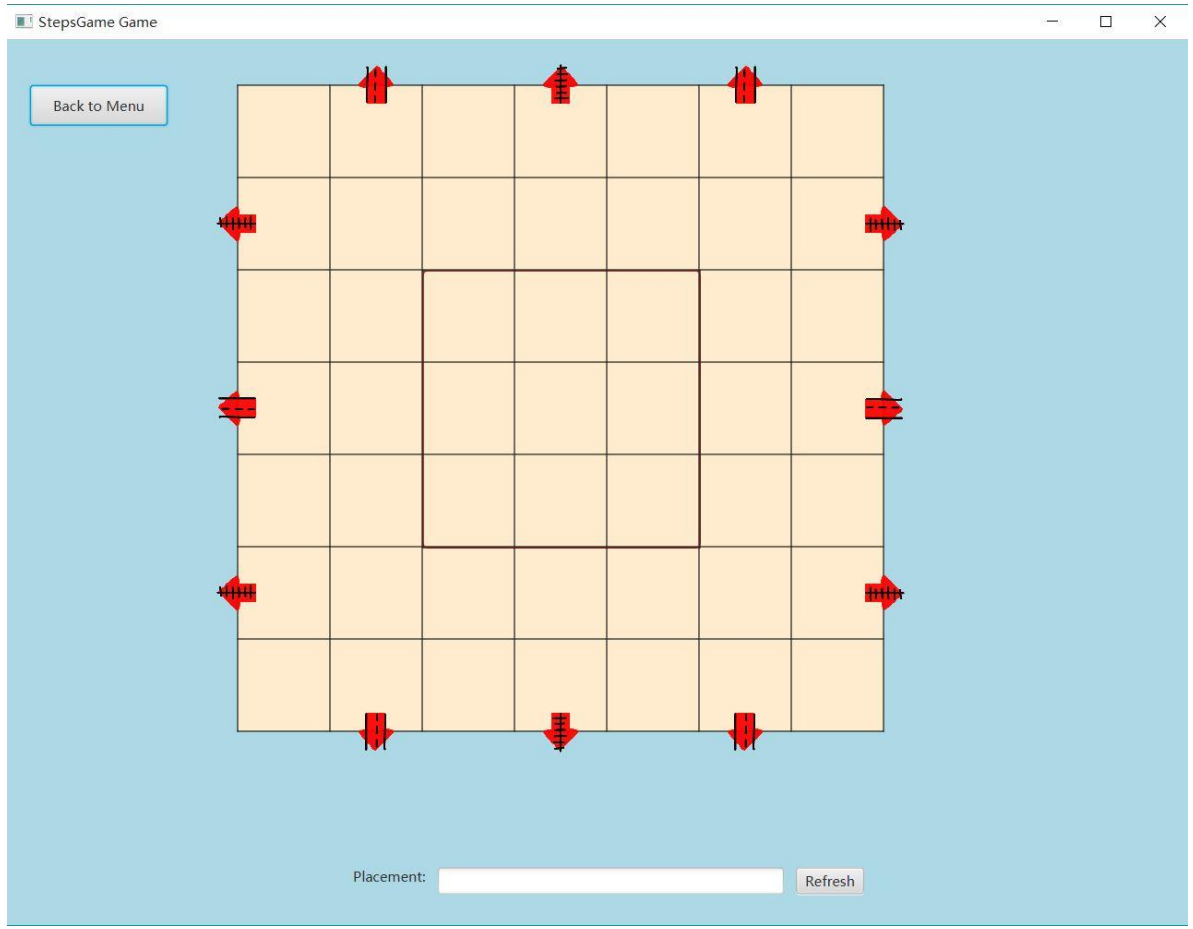
Source: https://www.horrible-games.com/railroad-ink-deep-blue-edition/

Overview

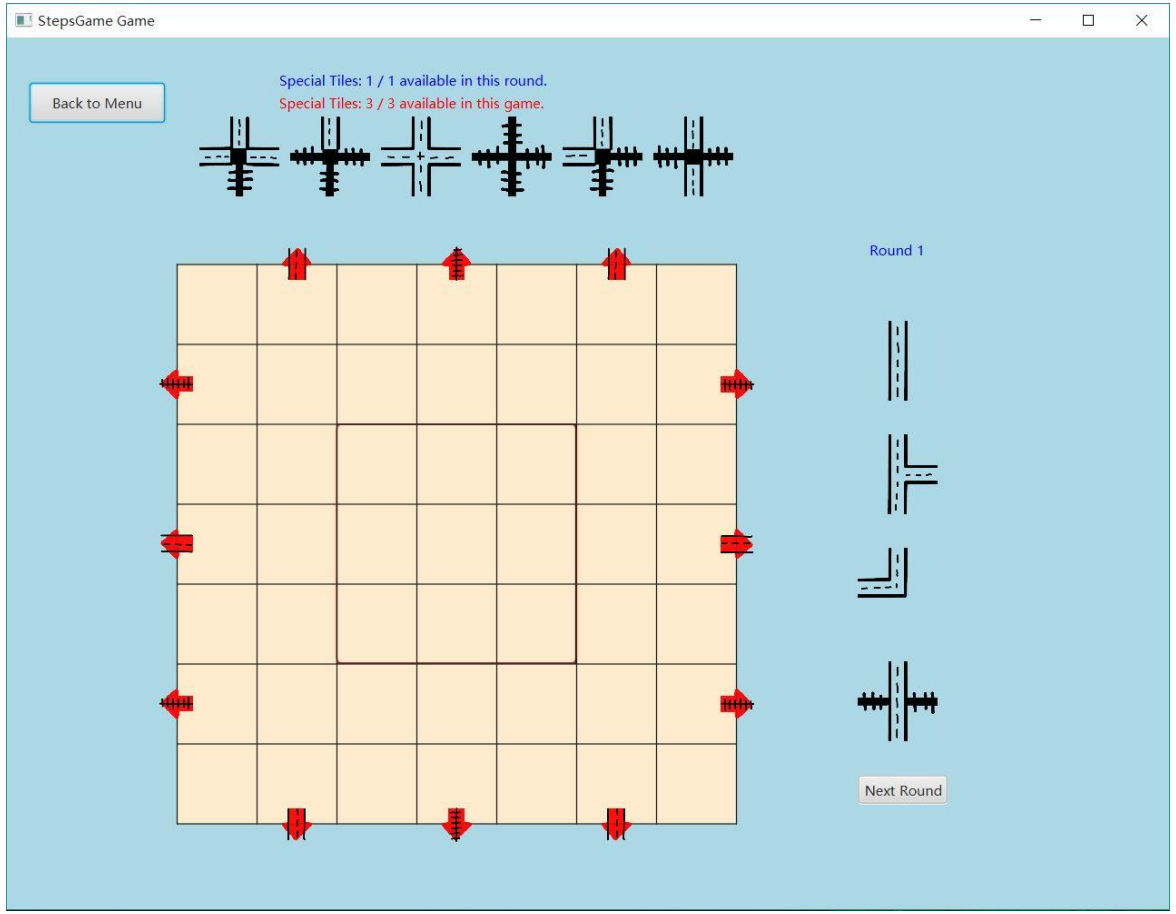
- Implement task 1 – task 11
- Low coupling program (Object-oriented)
- Well encapsulating code (Object-oriented)
- Good user experience
- About 2,800 lines of code, 1,600 lines of functional code



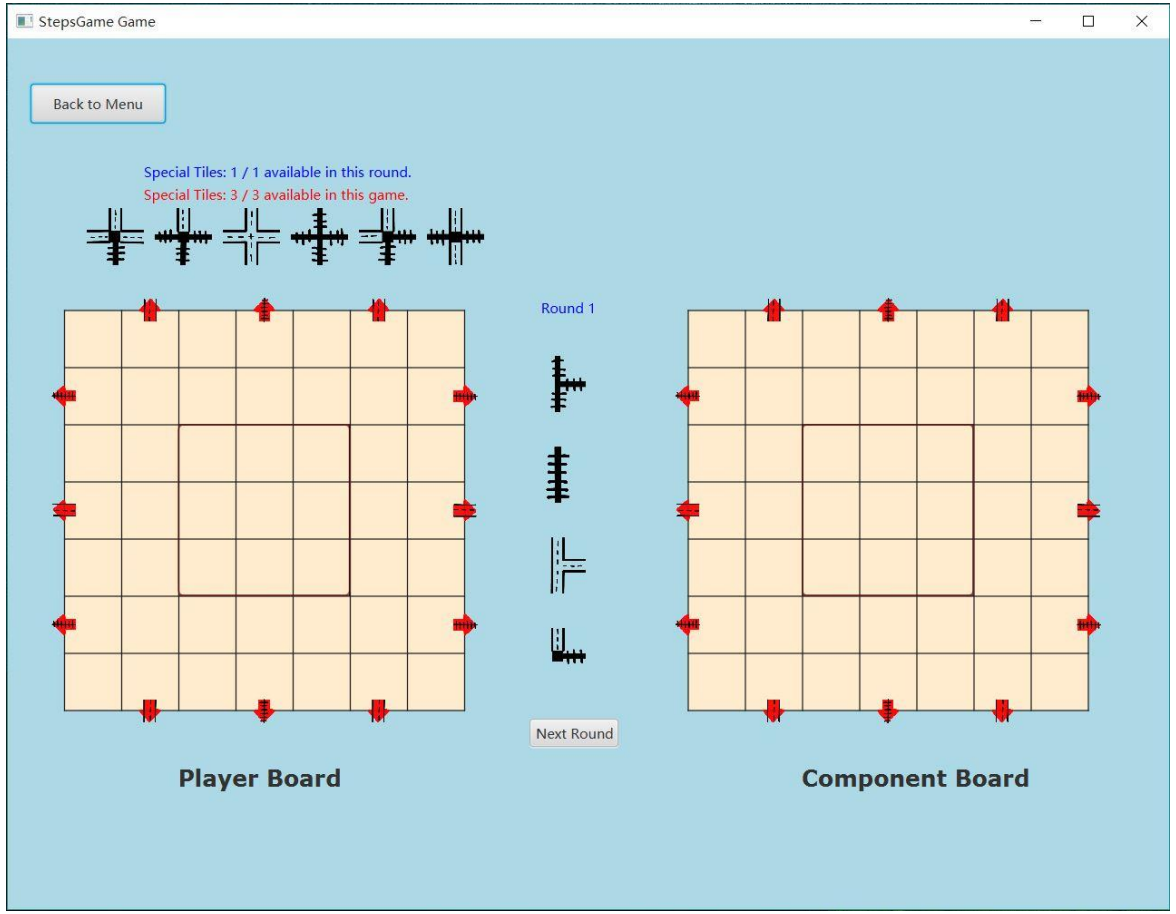
Menu Page



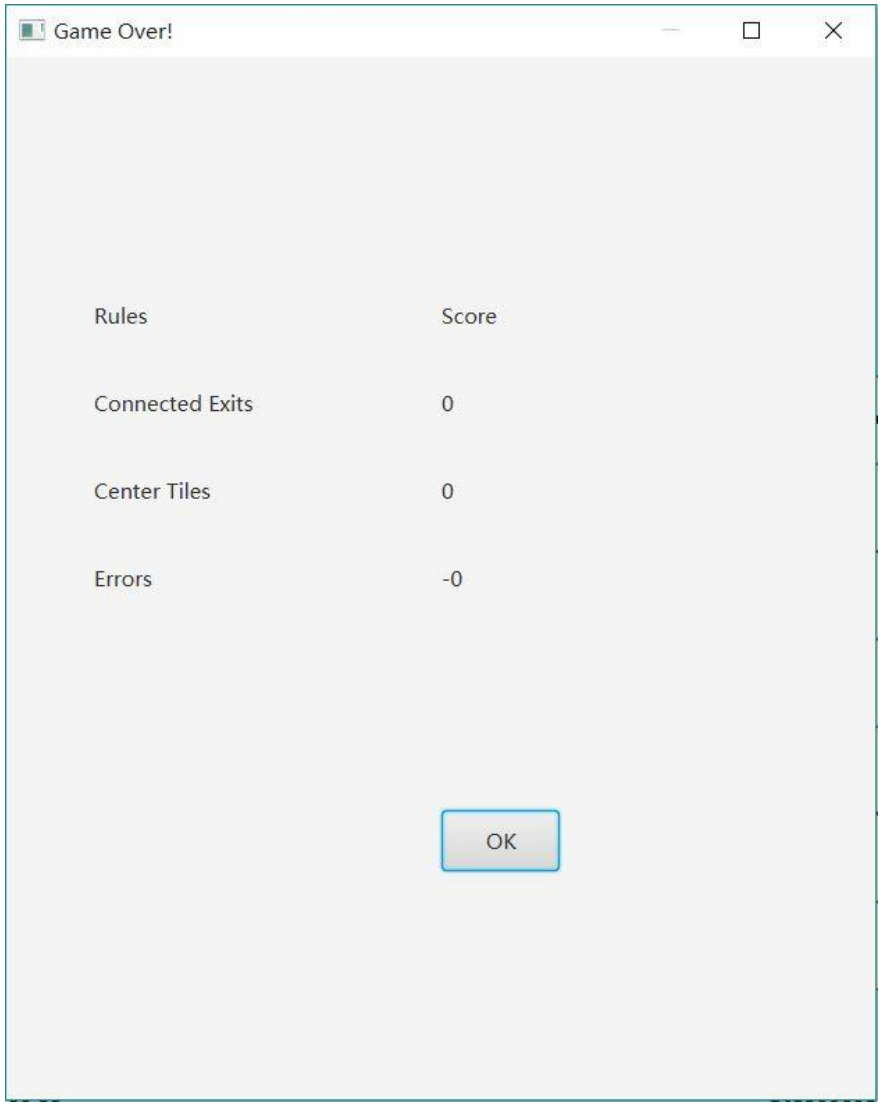
Placement Viewer



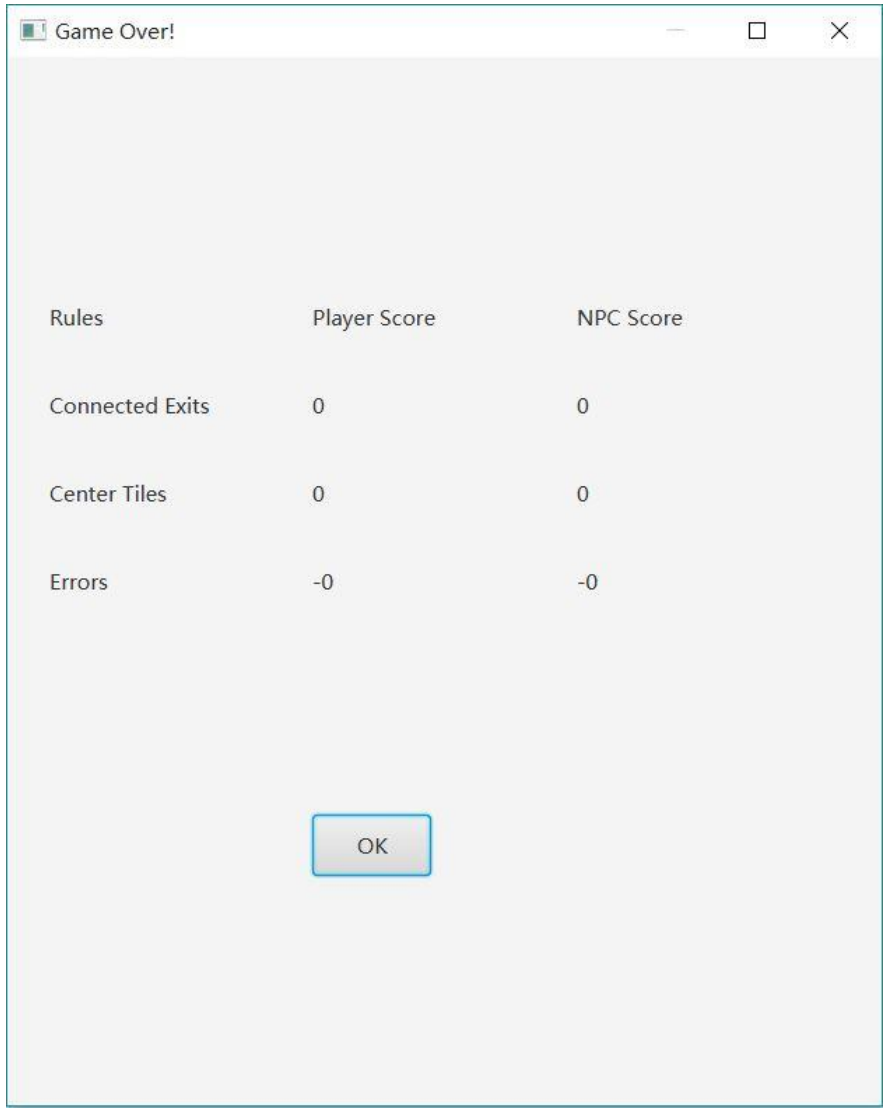
Single Play



Component Play

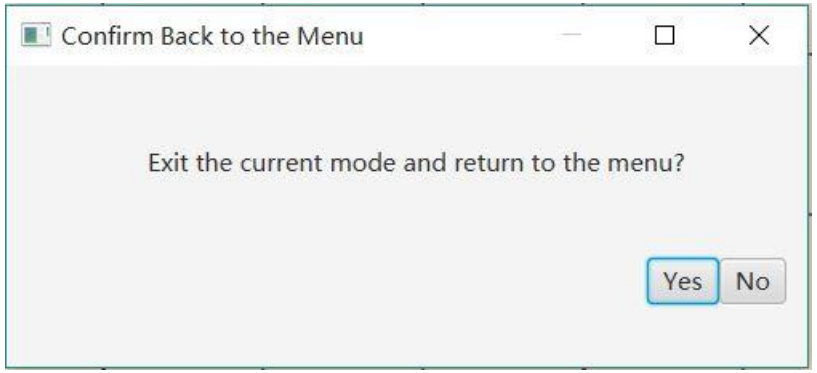


Single Score

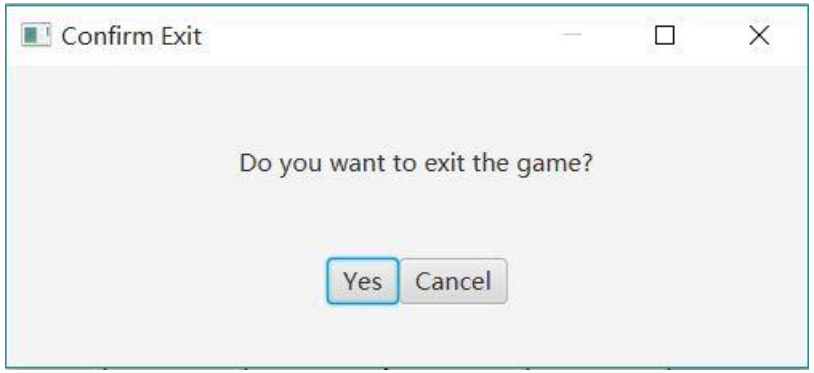


Component Play

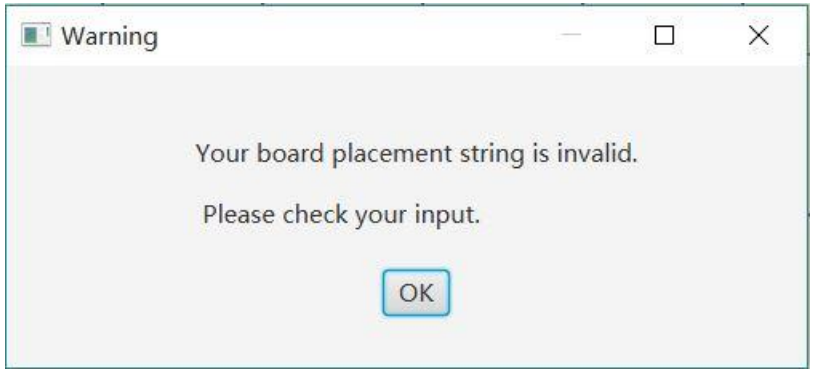
Return Alert



Exit Alert

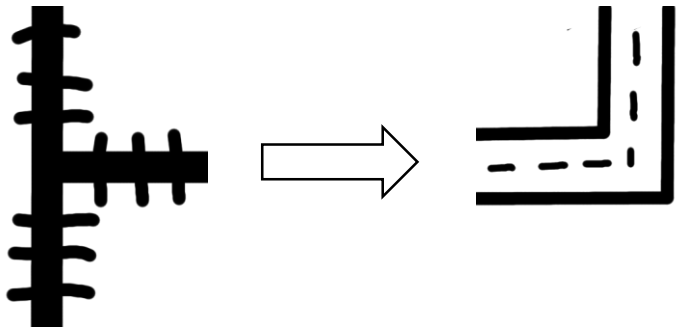


Invalid String Alert



Difficulty 1

How to determine the validity of the connected dice

Is  Valid?

Class Tile()

To initialize, store each die

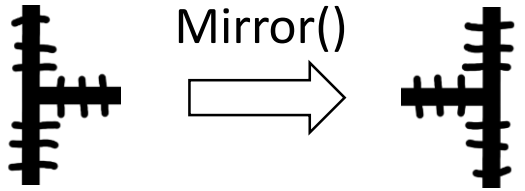
Store Initial pattern

```
private static final int[][] ORIGIN_EDGE_STATUS = {
        //S0 original shape
        {HIGH, HIGH, RAIL, HIGH},
        //S1 original Shape
        {HIGH, RAIL, RAIL, RAIL},
        //S2 original Shape
        {HIGH, HIGH, HIGH, HIGH},
        //S3 original Shape
        {RAIL, RAIL, RAIL, RAIL},
        //S4 original Shape
        {HIGH, RAIL, RAIL, HIGH},
        //S5 original Shape
        {HIGH, RAIL, HIGH, RAIL},
```

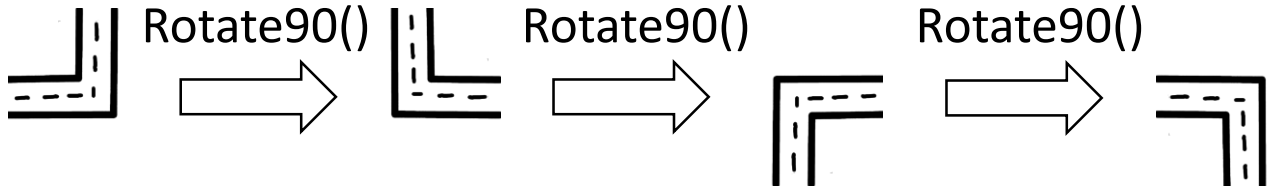
Class Tile()

To change the pattern of each die

Mirror()



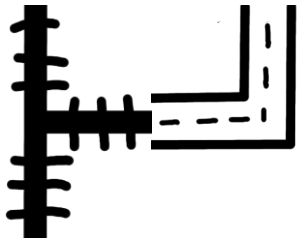
Rotate 90, 180, 270 degree()



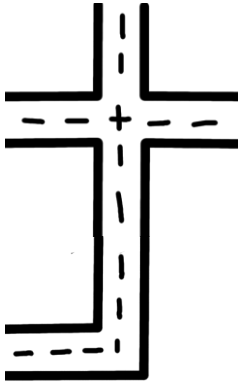
Step 1
Is it legal?

Illegal or legal

Illegal



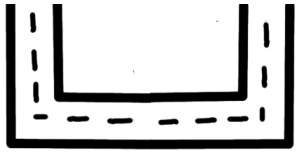
legal



Step 2
Is it connected?

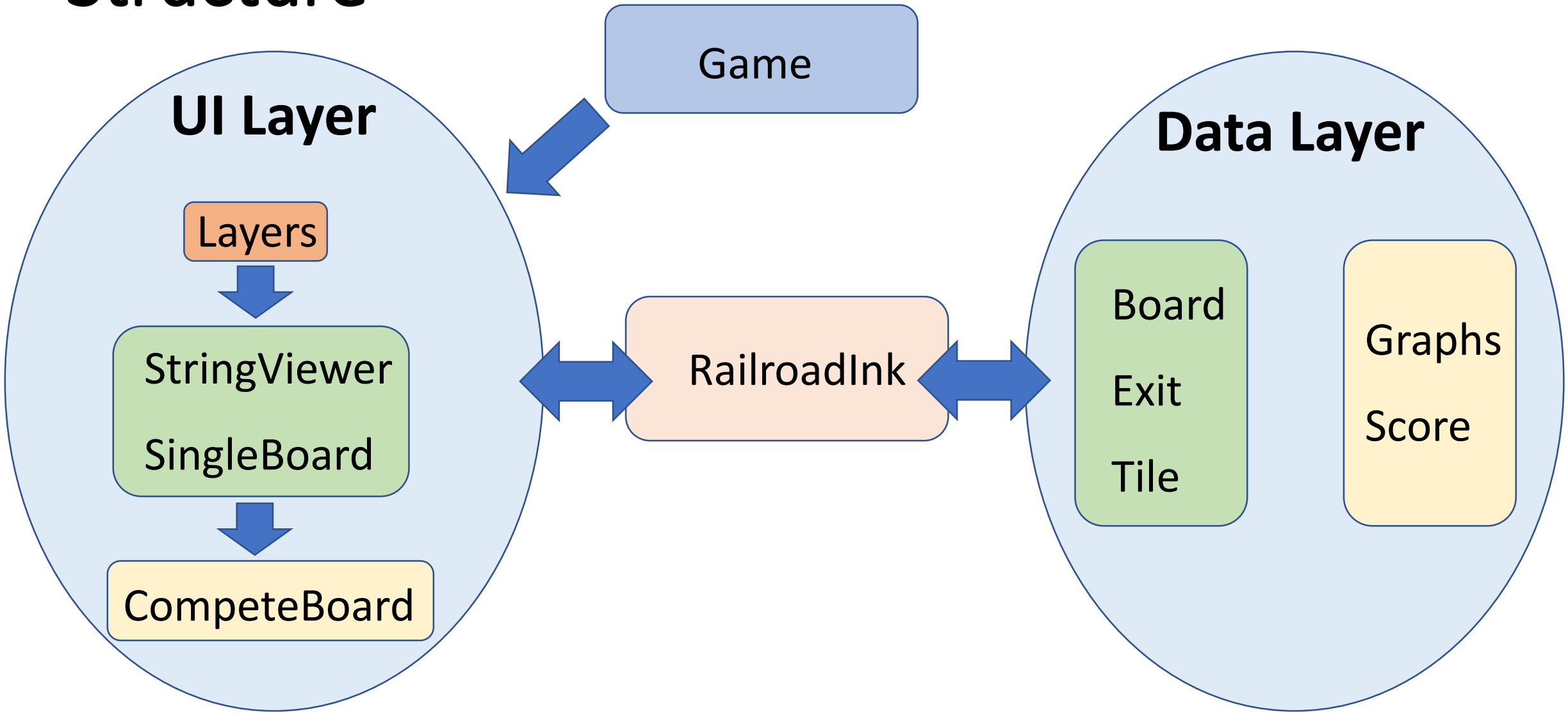
Connected or not

Connected



Unconnected

Structure


Game
UI Layer
Layers
StringViewer
SingleBoard
CompeteBoard
RailroadInk
Data Layer
Board
Exit
Tile
Graphs
Score

Layers Constructor

```
Layers(Scene menuScene, int layoutX, int layoutY, int fitHeight, int fitWidth) {
    this.menuScene = menuScene;
    this.layoutX = layoutX;
    this.layoutY = layoutY;
    this.fitHeight = fitHeight;
    this.fitWidth = fitWidth;

    root.getChildren().add(controls);
    scene = new Scene(root, VIEWER_WIDTH, VIEWER_HEIGHT);
    scene.setFill(Color.LIGHTBLUE);
    makeBoard(layoutX, layoutY, fitHeight, fitWidth);
    makeRedRectangle(layoutX, layoutY, fitHeight, fitWidth);
    makeExit(layoutX, layoutY, fitHeight, fitWidth);
    makeReturnButton();
}
```

String Viewer Constructor

```
StringViewer(Scene menuScene, int layoutX, int layoutY, int fitHeight, int fitWidth) {
    super(menuScene, layoutX, layoutY, fitHeight, fitWidth);
    this.fitHeight = fitHeight;
    this.fitWidth = fitWidth;

    makeStrBoardControl();
}
```

```
SingleBoard(Scene menuScene,
            int boardLayoutX, int boardLayoutY,
            int specialLayoutX, int specialLayoutY,
            int choiceLayoutX, int choiceLayoutY,
            int fitHeight, int fitWidth) {
    super(menuScene, boardLayoutX, boardLayoutY, fitHeight, fitWidth);

    root.getChildren().add(ctrlGroup);
    root.getChildren().add(imgGroup);
    root.getChildren().add(roundImgGroup);

    singleBoard = new Board();
    makeSpecialBoard(specialLayoutX, specialLayoutY, fitHeight, fitWidth);
    makeChoiceBoard(choiceLayoutX, choiceLayoutY, fitHeight);
}
```

SingleBoard Constructor

```
CompeteBoard(Scene menuScene,
                  int boardLayoutX, int boardLayoutY,
                  int specialLayoutX, int specialLayoutY,
                  int choiceLayoutX, int choiceLayoutY,
                  int fitHeight, int fitWidth) {
    super(menuScene, boardLayoutX, boardLayoutY,
            specialLayoutX, specialLayoutY,
            choiceLayoutX, choiceLayoutY,
            fitHeight, fitWidth);

    makeComponentBoard(boardLayoutX, boardLayoutY, fitHeight, fitWidth);
}
```

CompeteBoard Constructor

Design for Task 8

```
class Graphs {
    private int graphId;
    private int exitNum;
    Node head;

    class Node {
        Tile tile;
        Node up, right, down, left;
    }
}
```

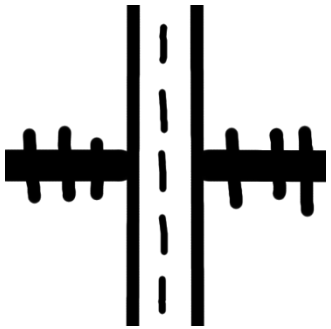
```
class Score {
    private ArrayList<Graphs> mapList;
}
```

Design for Task 8

Initialization:

1. Assign graphId = 0 for each exit and tile.
2. Assign graphId = -1 and use four edges graphId.

   (Specifically for tile B2)



Useful Tips

- Global variables with constant values (“static final”)
- imgView.setPickOnBounds(ture);
- imgView.setDisable(true);