



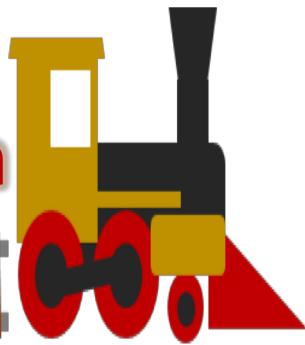
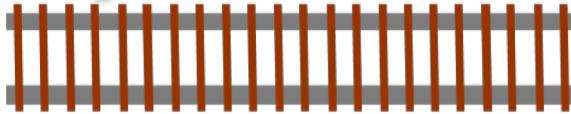
Group members:

Thoraya Al-Sabti (u6136358)

Samuel Brookes (u5380100)

Baozhong Tan (u6126217)

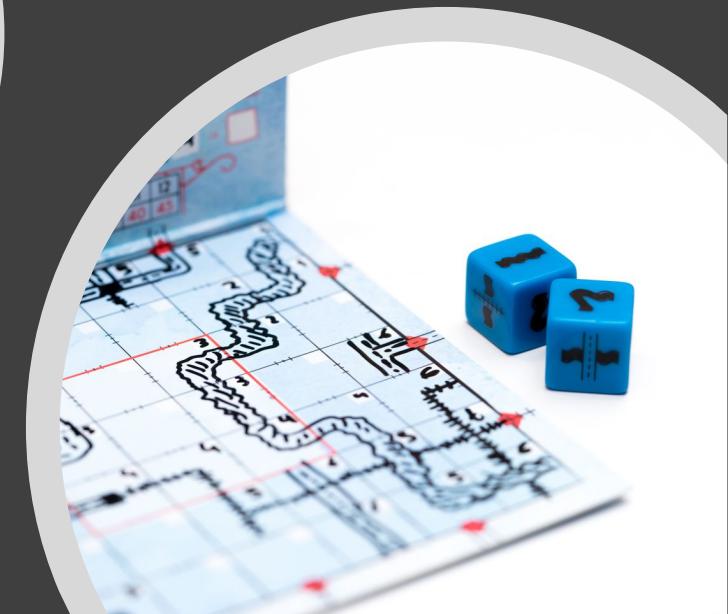
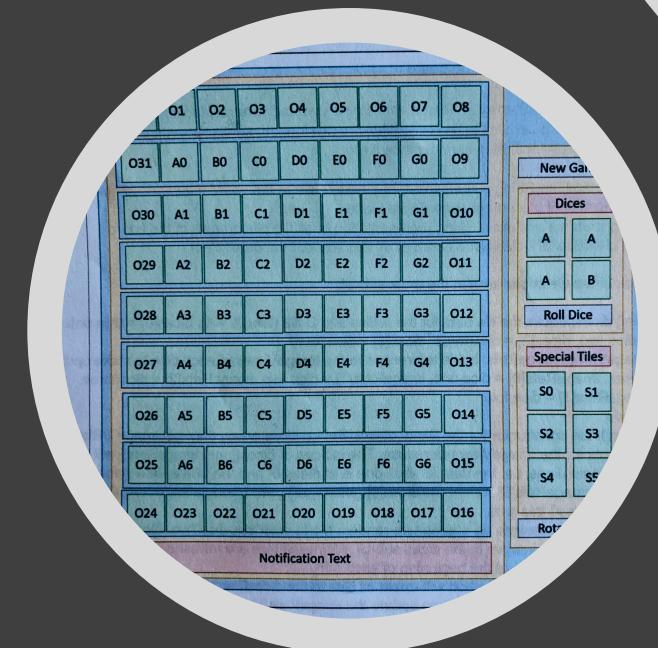
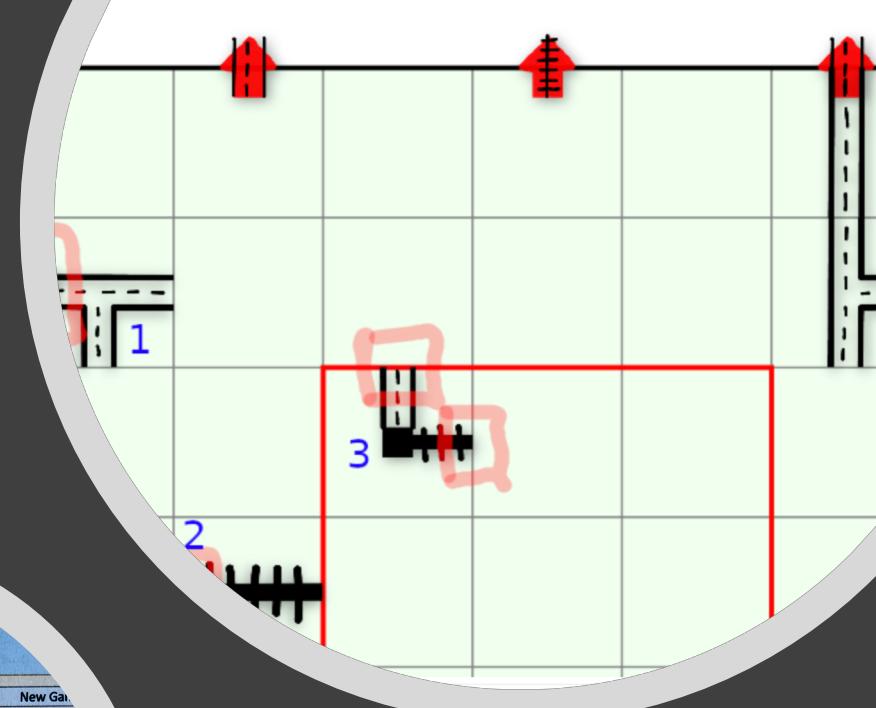
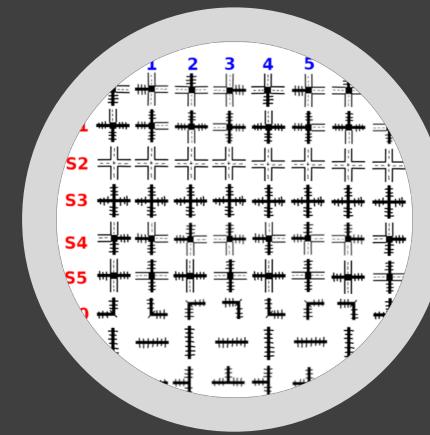
Design Approach



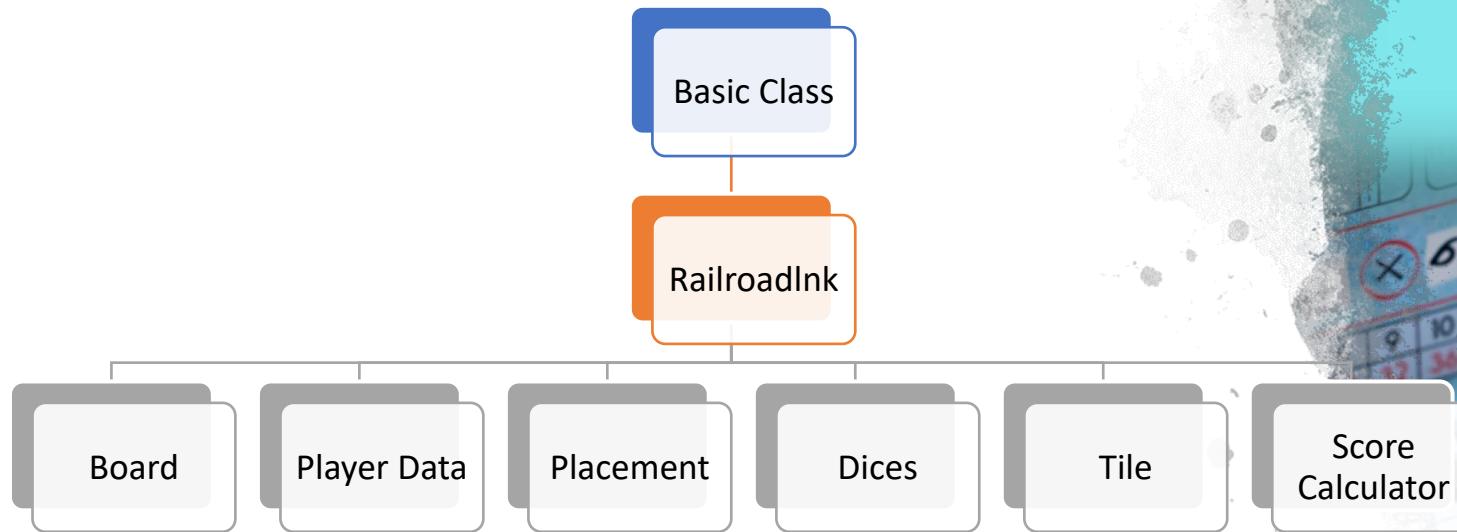
- Define the problem statement.
- Creating the basic classes.
- Coding the classes and methods.
- Setup the game.

Define the problem statement.

- Define the working principle of the game.
- Understand the Rules, Board, Tiles, Legal placements and Scoring.
- Understand the encoding game .



Creating the basic classes.



```

{ //If single player, create only one PlayerData object and put it in
  playerData.put(1, new PlayerData(player: 1, new Board(), new Dices()
  playerData.get(1).diceData.rollDice());
}
else
{ //If two player (incl. computer opponent), create two PlayerData objects
  playerData.put(1, new PlayerData(player: 1, new Board(), new Dices(), null)
  playerData.put(2, new PlayerData(player: 2, new Board(), new Dices(), null)

  //Roll the dice of player one
  playerData.get(1).diceData.rollDice();

  //Copy the dice data into player two's PlayerData object
  playerData.get(2).diceData.copyPlayerDices(playerData.get(1).diceData);

  if(gameMode == 'c')
  { //add pointer to player data for computer opponent if gameMode is 'c'
    ComputerOpponent = new ComputerOpponent(playerData.get(2));
  }
}
}

```

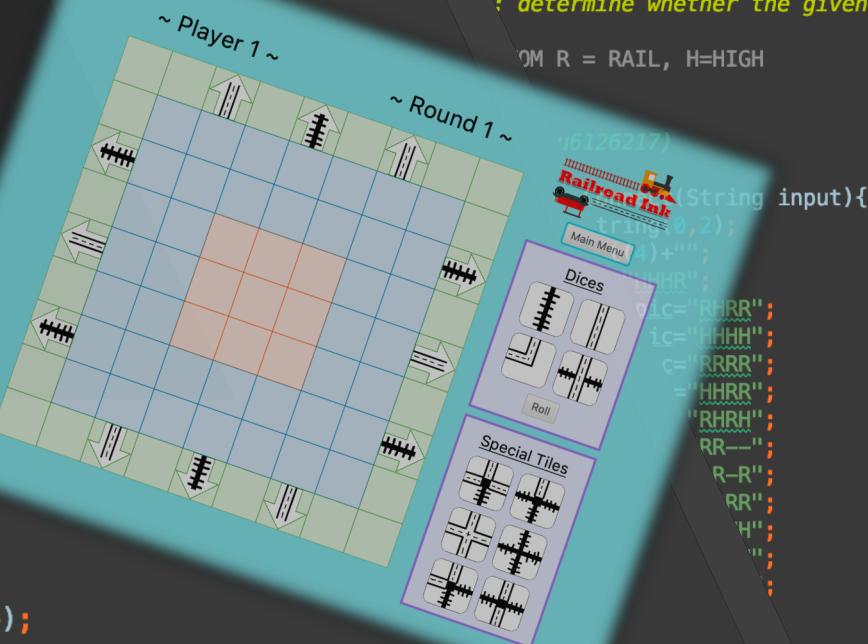
Coding the classes and methods to figure out the tasks.

```

/* Set up local reference variables for this launch.
* Each time the game board is reloaded, the correct player
* data is accessed from the playerData HashMap because the
* player field keeps track of which player's turn it is.
*/
boardRef = playerData.get(player).boardData;
diceRef = playerData.get(player).diceData;
specialRef = playerData.get(player).specialData;

//set up gameContainer
boardProper = new GridPane();
setUpGameInfo();
setUpBoard();
txtNotification = new Text();
formatText(txtNotification, size: 30, bold: true, underline: false);
VBox gameContainer = new VBox();
formatBox(gameContainer, Color.LIGHTBLUE, spacing: 10, border: false);
gameContainer.getChildren().addAll(gameInfo, boardContainer, txtNotification);

```



```

connect.add(true);
String pica = testConnect(a);
String picb = testConnect(s);
if(a.charAt(3)==s.charAt(3)&&a.charAt(2)>s.charAt(2))
  boolarr.add(pica.charAt(1)==picb.charAt(3)||pica.charAt(1)=='-');
else if(a.charAt(3)==s.charAt(3)&&a.charAt(2)<s.charAt(2))
  boolarr.add(pica.charAt(3)==picb.charAt(1)||pica.charAt(3)=='-');
else if(a.charAt(2)==s.charAt(2)&&a.charAt(3)>s.charAt(3))
  boolarr.add(pica.charAt(0)==picb.charAt(2)||pica.charAt(0)=='-');
else if(a.charAt(2)==s.charAt(2)&&a.charAt(3)<s.charAt(3))
  boolarr.add(pica.charAt(2)==picb.charAt(0)||pica.charAt(2)=='-');
} else if(start.contains(a.substring(2,4))) connect.add(true);
else
  connect.add(false);
}

if(!connect.contains(true)) boolarr.add(false);
if(start.contains(s.substring(2,4))) isolate.add(true);
else isolate.add(false);

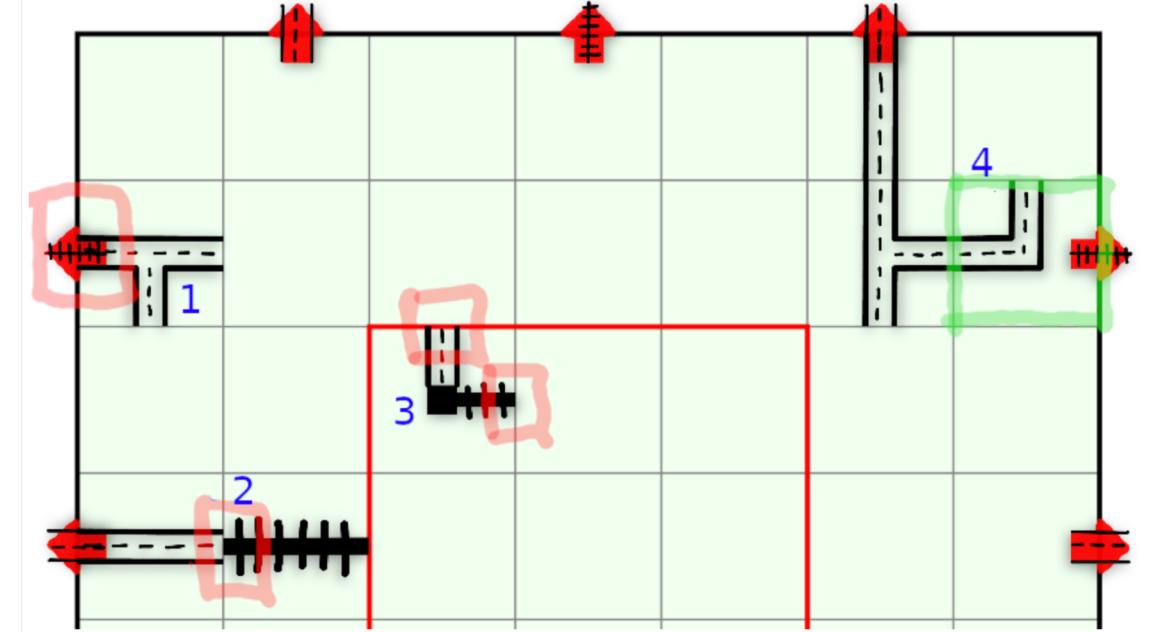
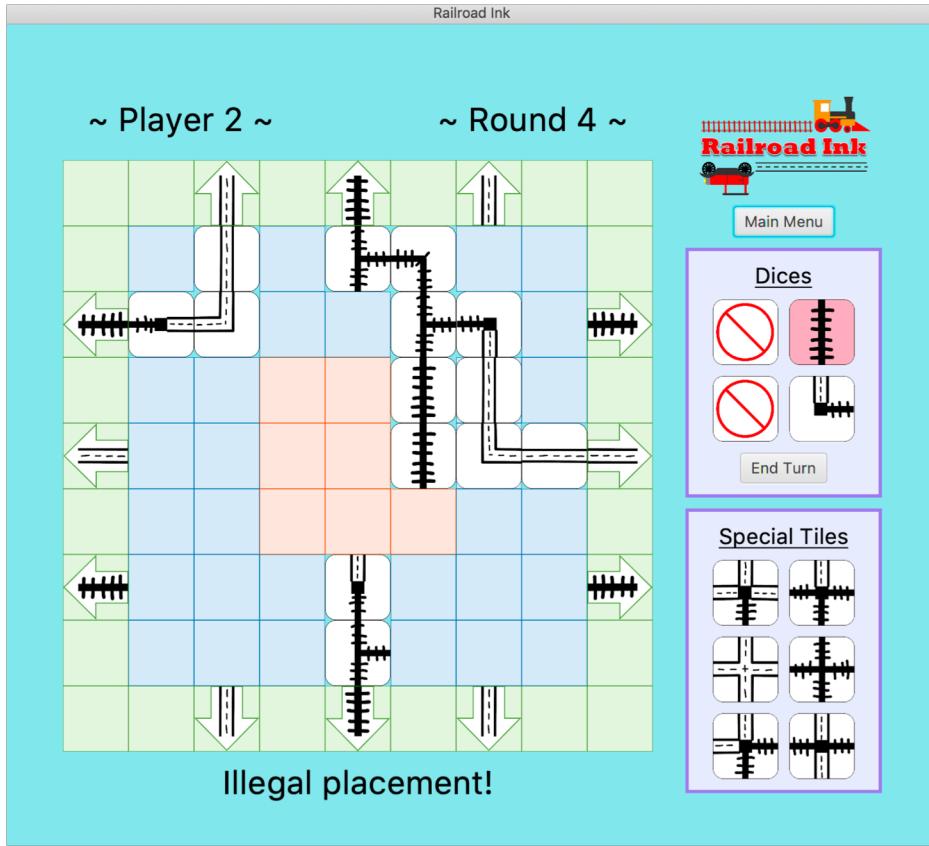
late.contains(true)) boolarr.add(false);

contains(false)) return false;
true;
: determine whether the given placement sequence is valid

```

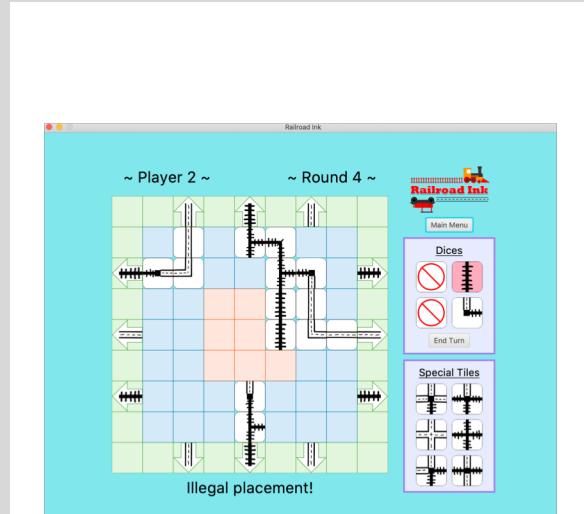
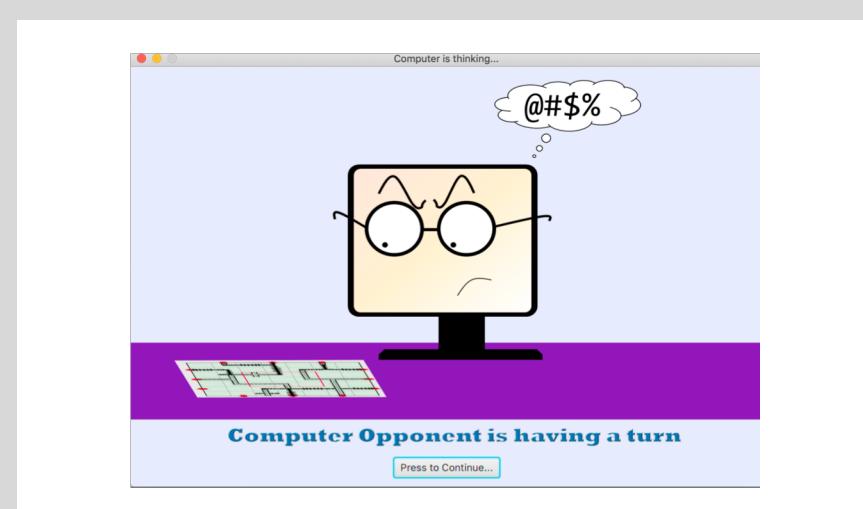
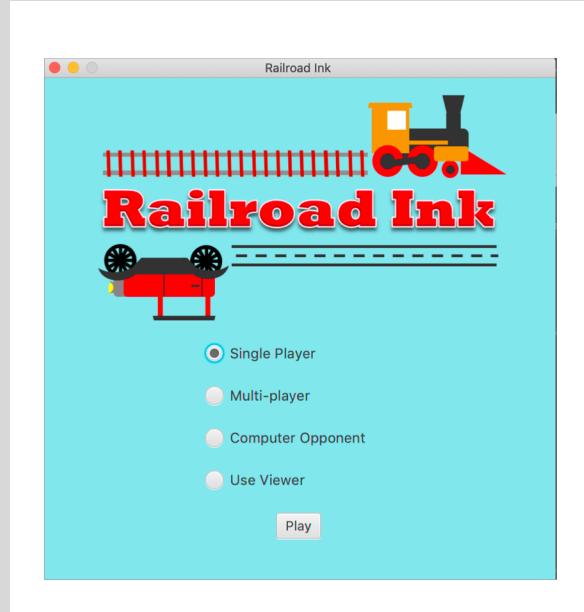
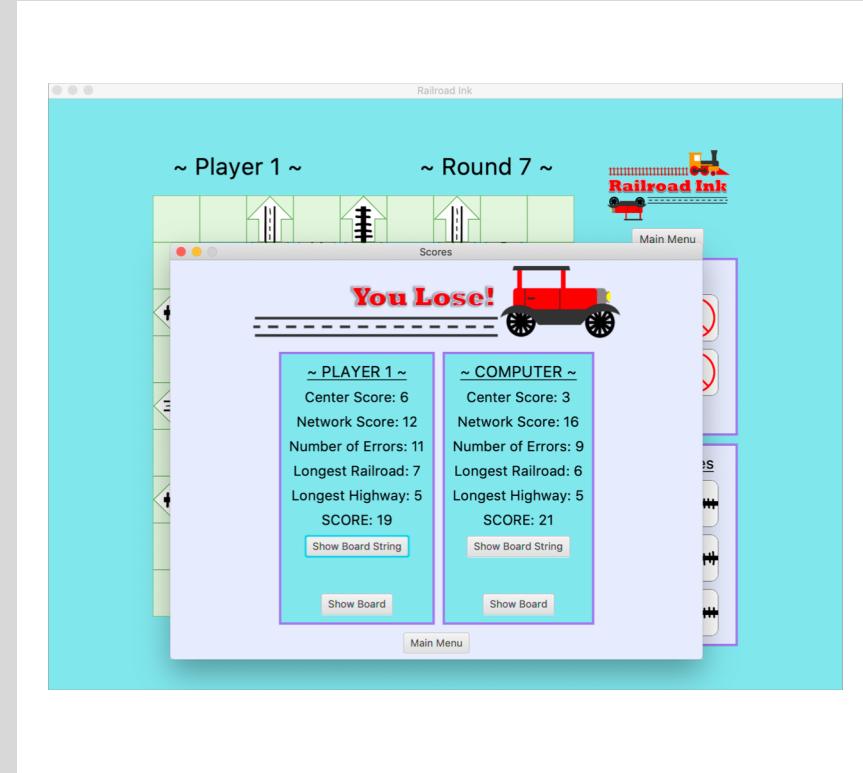
OM R = RAIL, H=HIGH

(16126217)
Railroad Ink (String input){
 String s = input.substring(0,2);
 if(s.equals("RR"))
 return true;
 else if(s.equals("RHRR"))
 return true;
 else if(s.equals("HRRH"))
 return true;
 else if(s.equals("HRRR"))
 return true;
 else if(s.equals("RRHH"))
 return true;
 else if(s.equals("RRH"))
 return true;
 else if(s.equals("RHR"))
 return true;
 else if(s.equals("RR--"))
 return true;
 else if(s.equals("R-R"))
 return true;
 else if(s.equals("RR"))
 return true;
 else if(s.equals("H"))
 return true;
 else
 return false;



Legal Placements.

Screen Shot of game



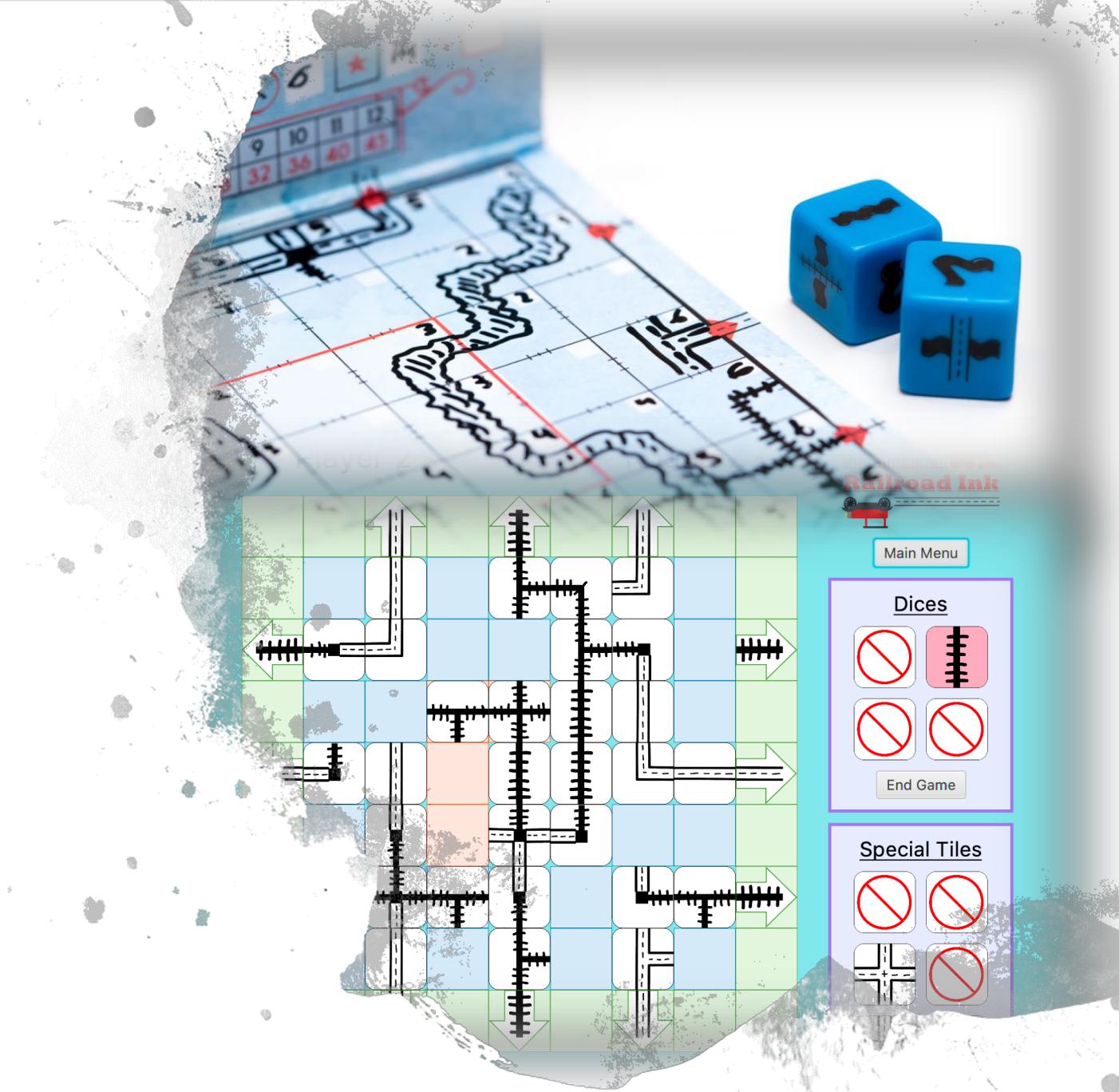
Interesting Aspects

- **What is interest in solving this game?**
 - 1) Analysis: understand the working principle and the logical of the game.
 - 2) Design: design the solution of the game, architecture, functions, classes, user interface.
 - 3) Coding: writing the code of the game solution.
 - 4) Testing: test the program including the classes, which help to improve the code.

Interesting Aspects

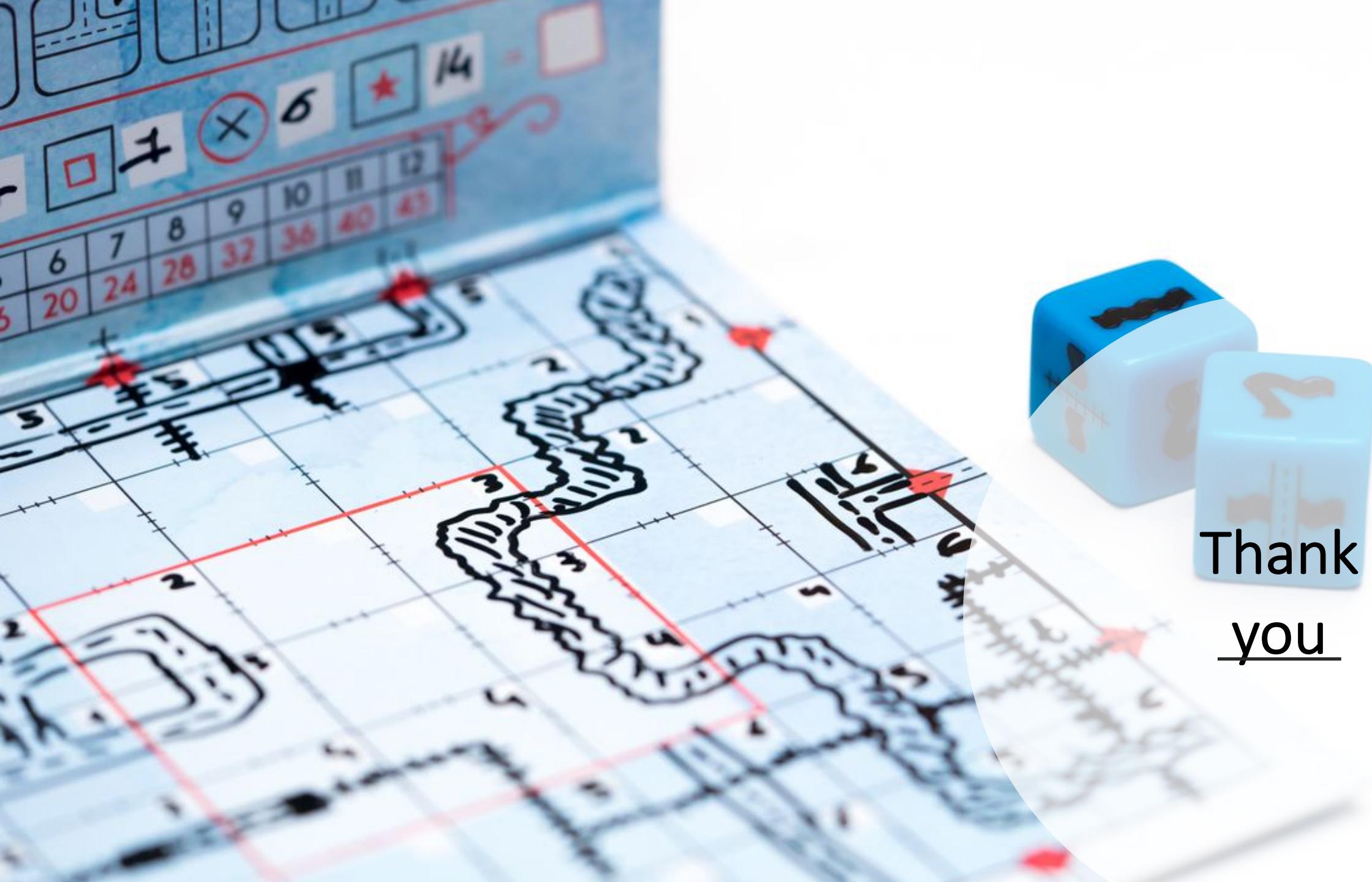
Why this game interesting?

- Concentration
- Logic
- Problem Solving
- Planning



*Run The
Game*





Thank
you