

COMP1110 ASSIGNMENT 2

FINAL PRESENTATION

Co-developed by Carry Zhang (u6499267), Keyu Liu (u6342137), and Qixia Lu (u6805636)

OVERVIEW

BY THE NUMBERS

Over

140

Commits

5

Java Class Files

302 MB

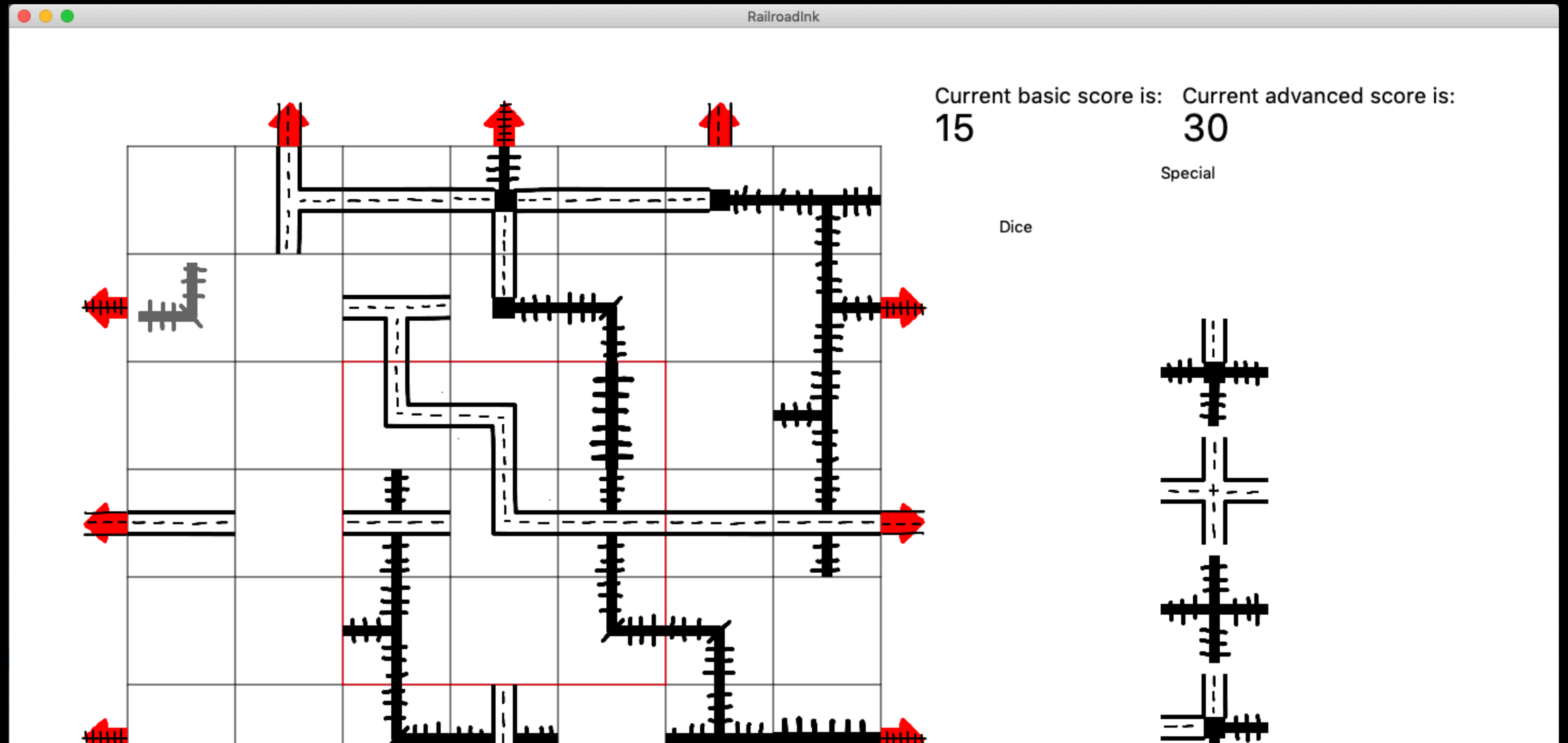
File Size

11

To-Dos Completed

USER INTERFACE

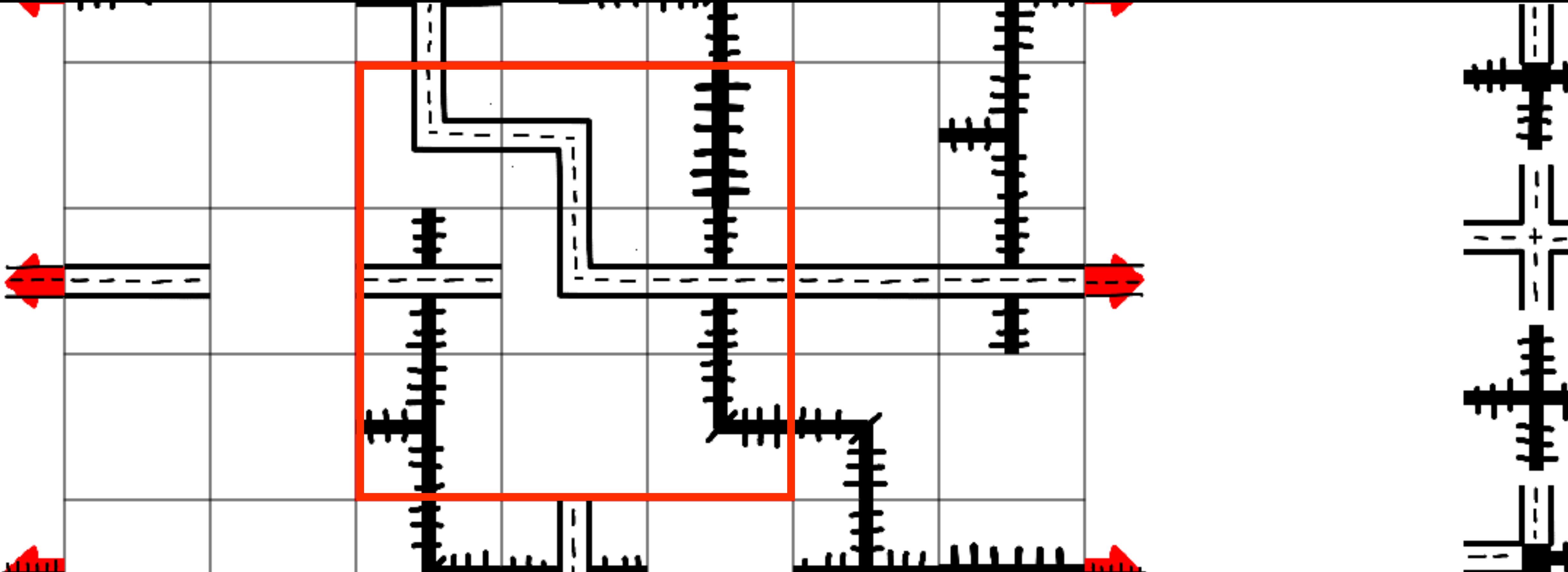
THE POWER OF SIMPLICITY



USER INTERFACE

RUN FOR RED

Highlighted centre area provides a simple illustration of bonus score area, so you can focus on getting that high score.



AIM HIGH

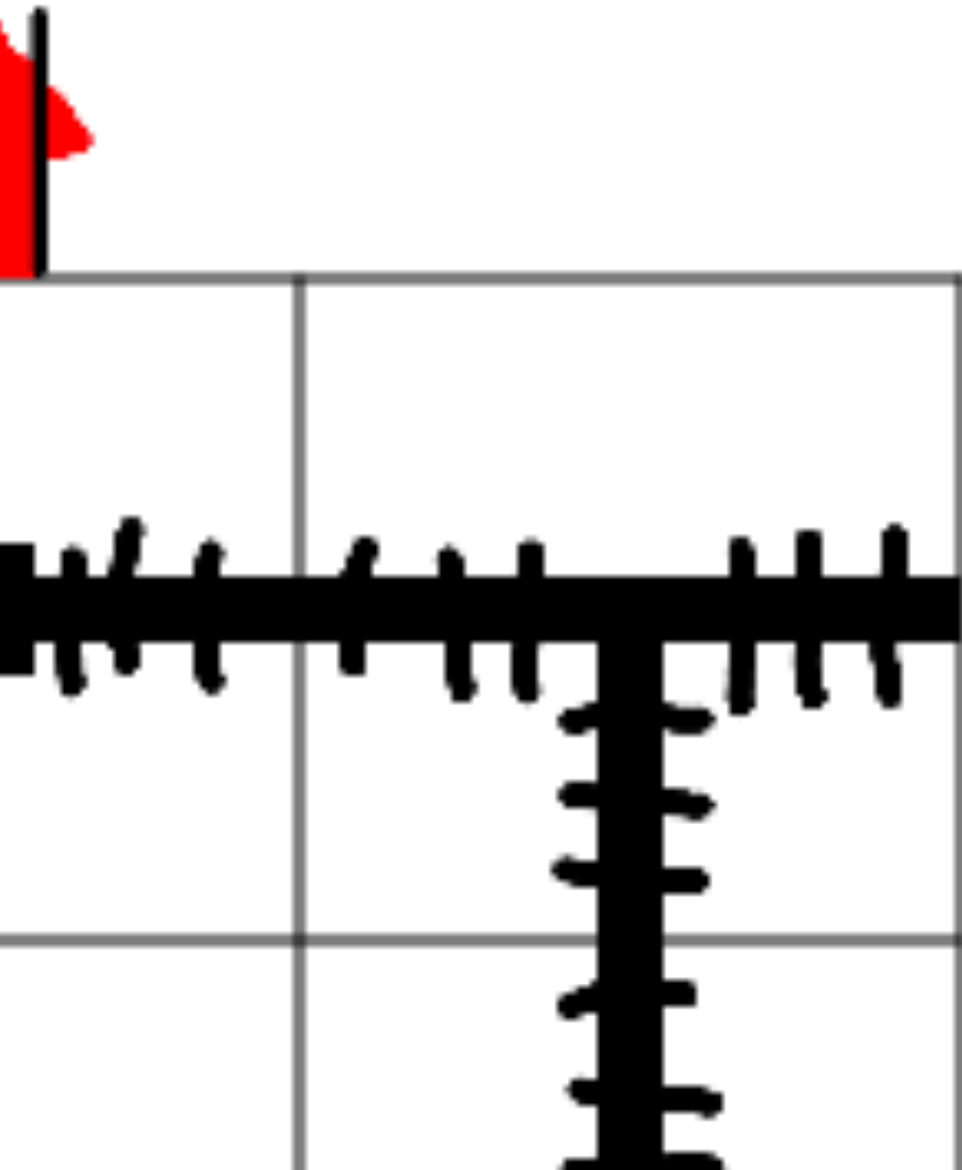
With real-time updated score counter, including both basic score and advanced score.

RailroadInk

Current basic score is: **14** Current advanced score is: **29**

Special

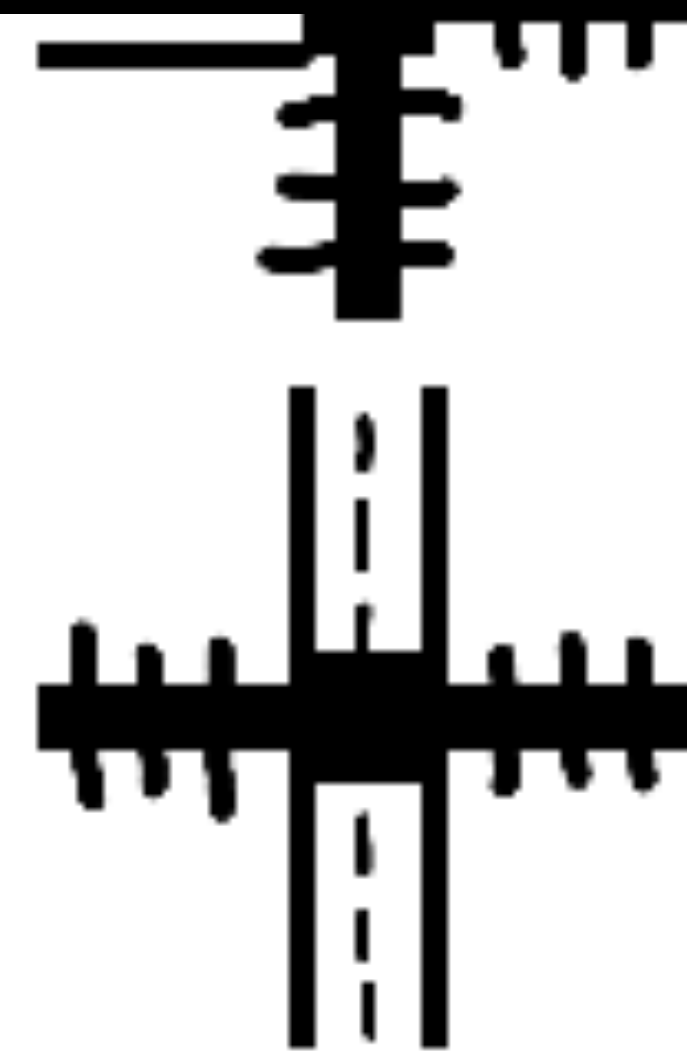
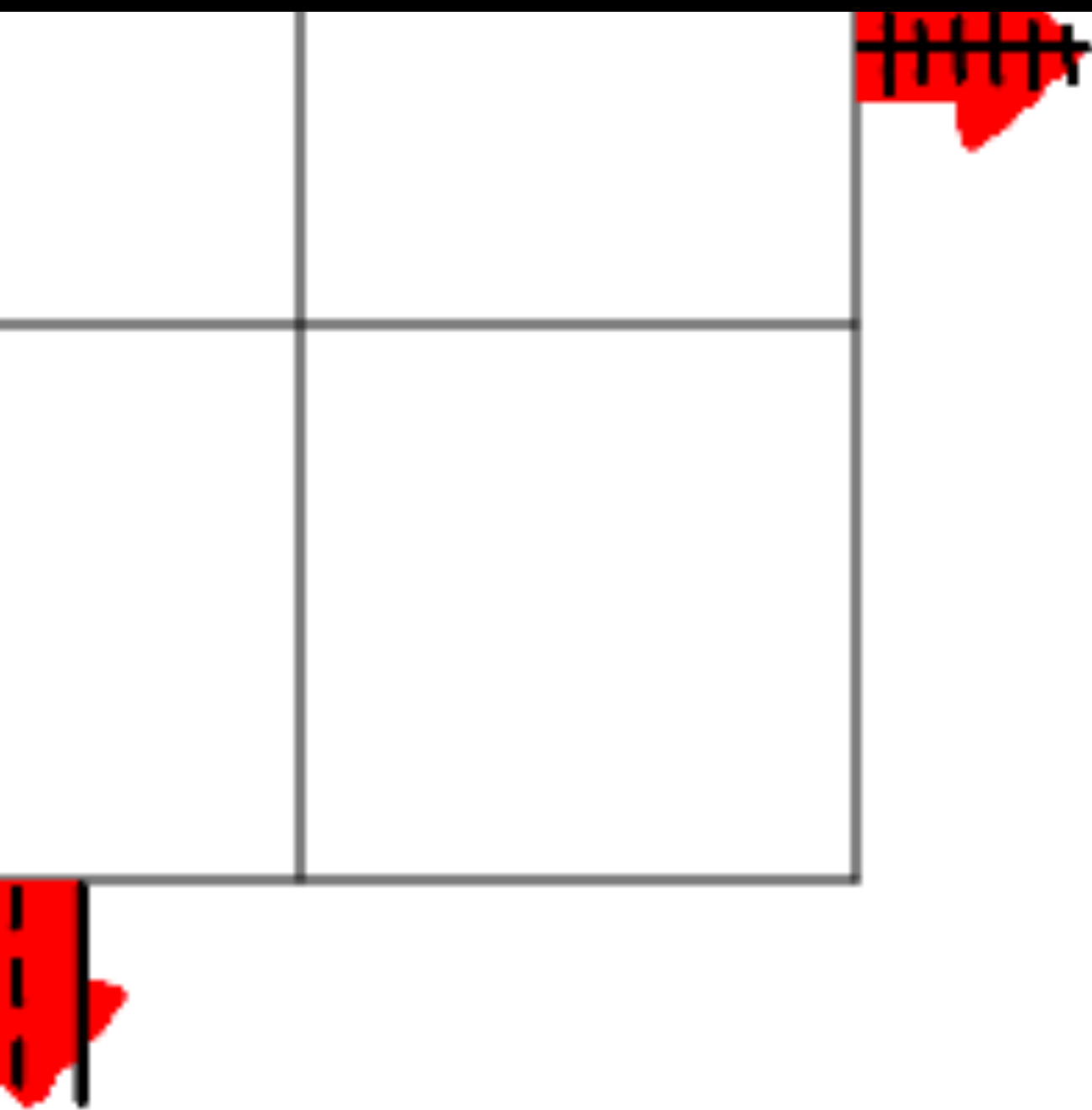
Dice



USER INTERFACE

POWER TO YOU

Adequately-placed easily-accessible buttons with great features including a placement viewer.



This is round 1

Generate

Restart

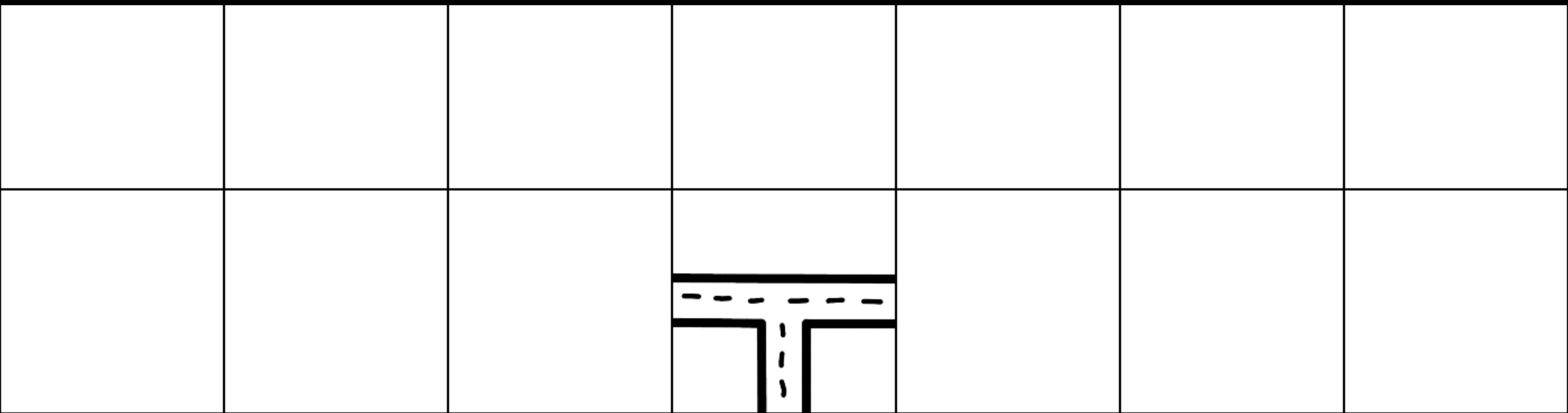
About

Viewer

Tips

CHECK IT OUT

Built-in pop-up placement viewer provides a quick reference, without adding distractions to the current game board.



Placement:

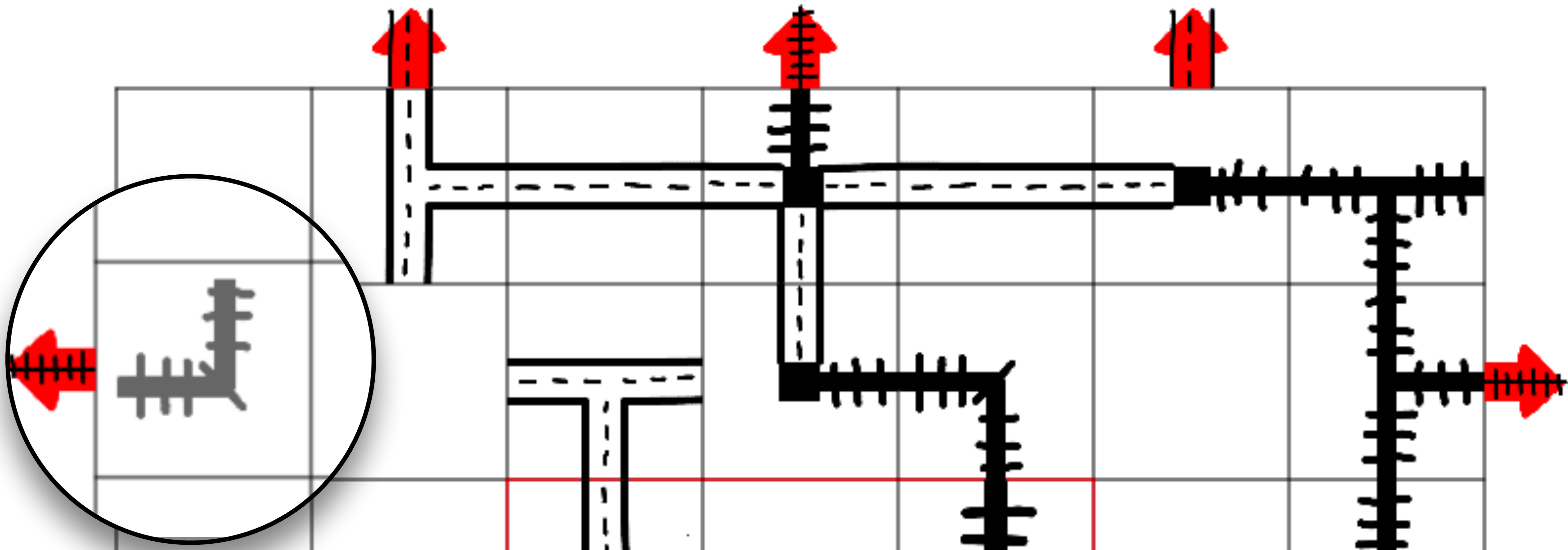
A3G31

Refresh

USER INTERFACE

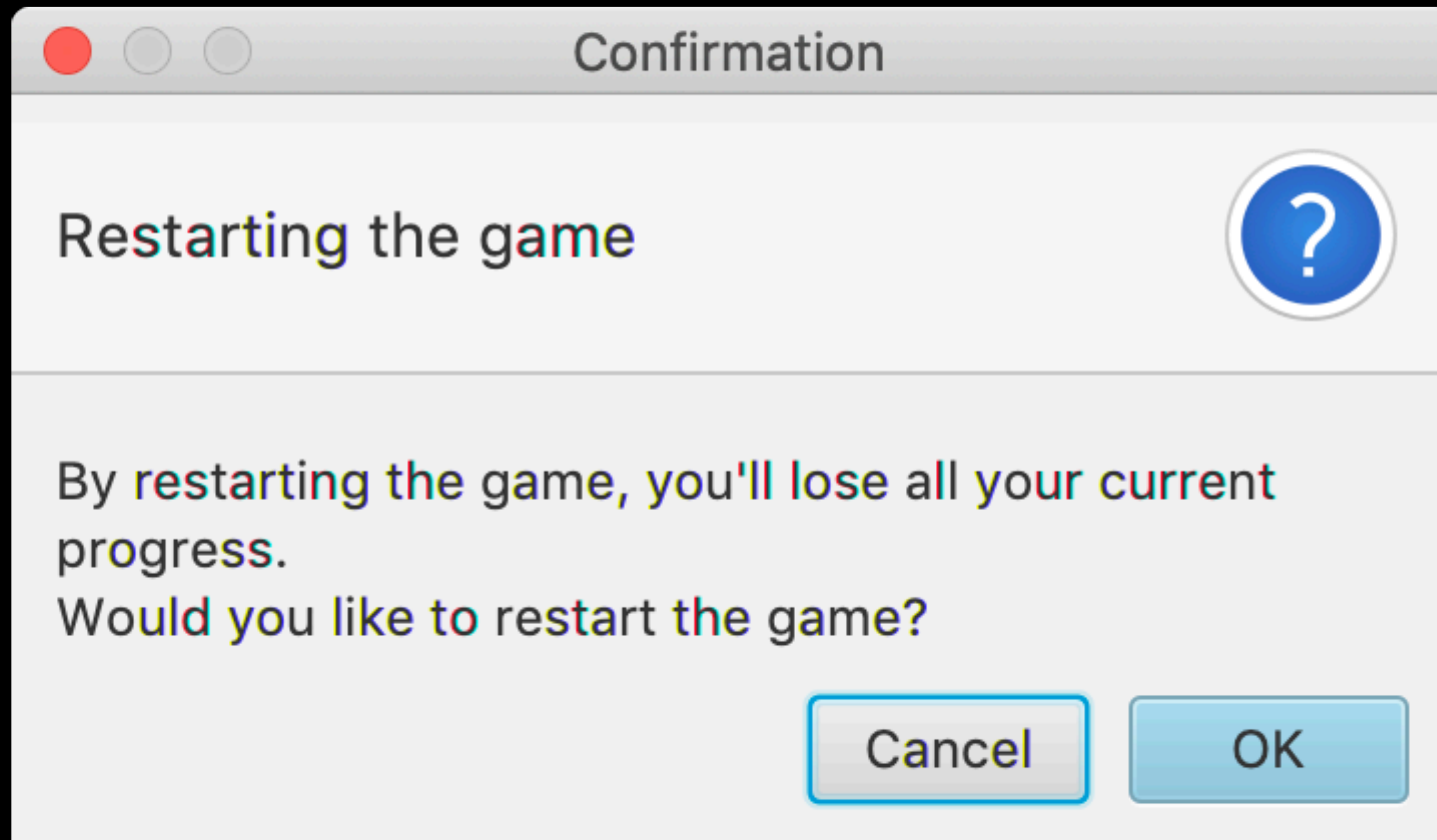
FOCUS ON WHAT MATTERS

Selected pieces are coloured differently so you can drop that perfect placement confidently.



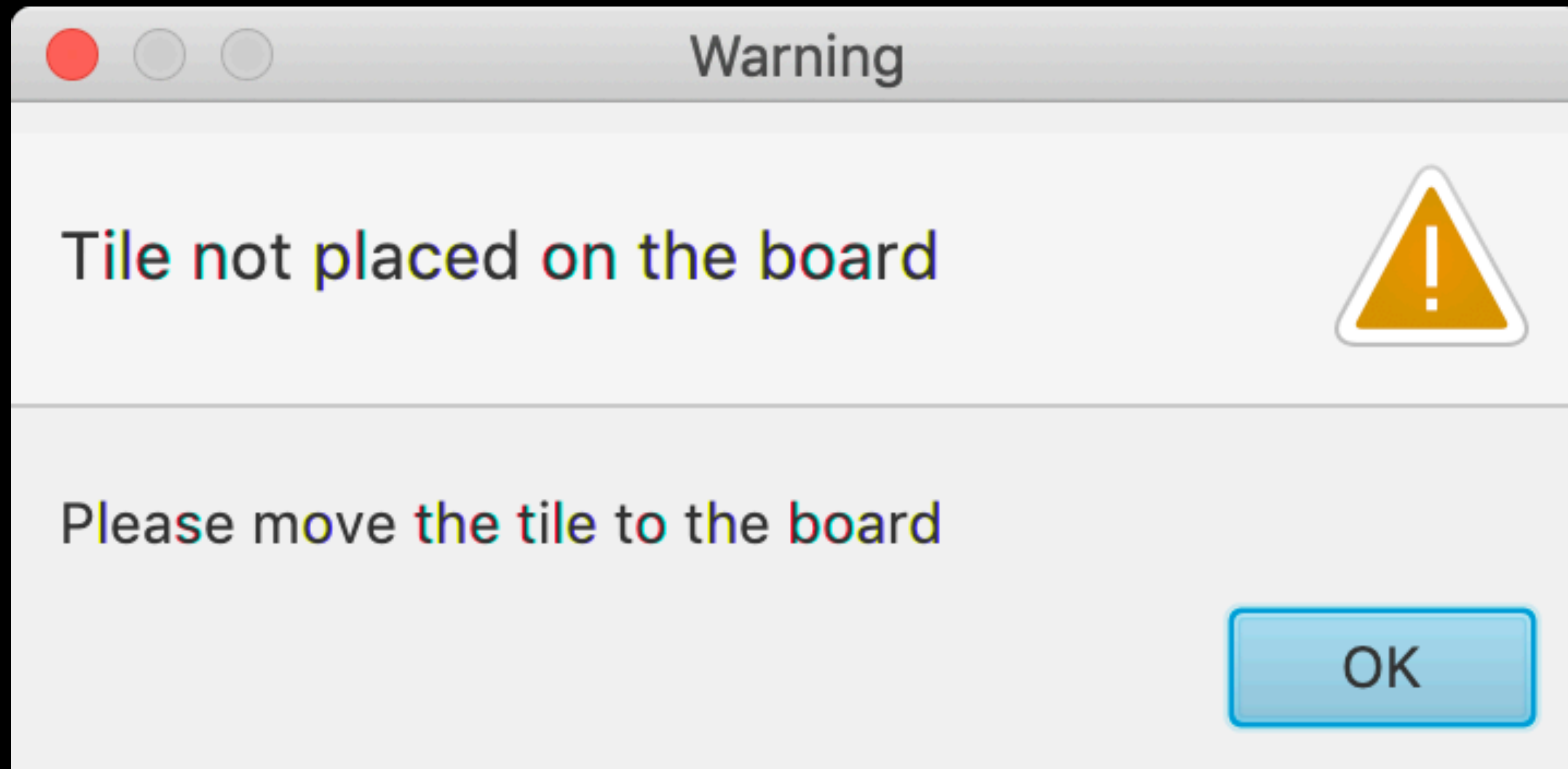
STAY IN CONTROL

Before restarting the game, a confirmation pop up will appear double-checking whatever you would like to restart.



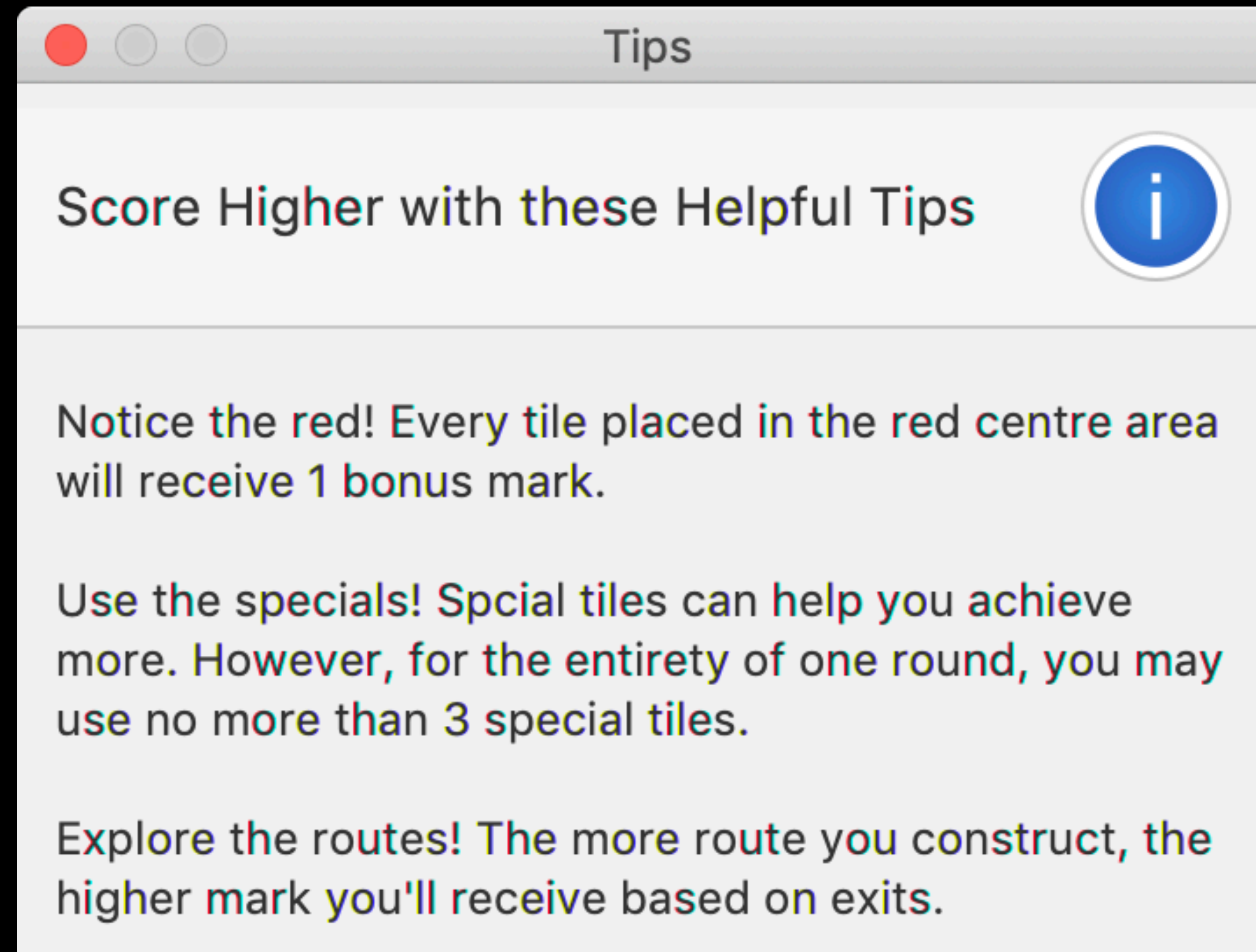
IN THE KNOWN

Warning messages give users the power of knowing what's illegal about the placement, rather than a simply cancelling the operation.



BE A PRO

Get insights from the creator with the tips pop-up, listing the significant for a even better result.



DESIGN

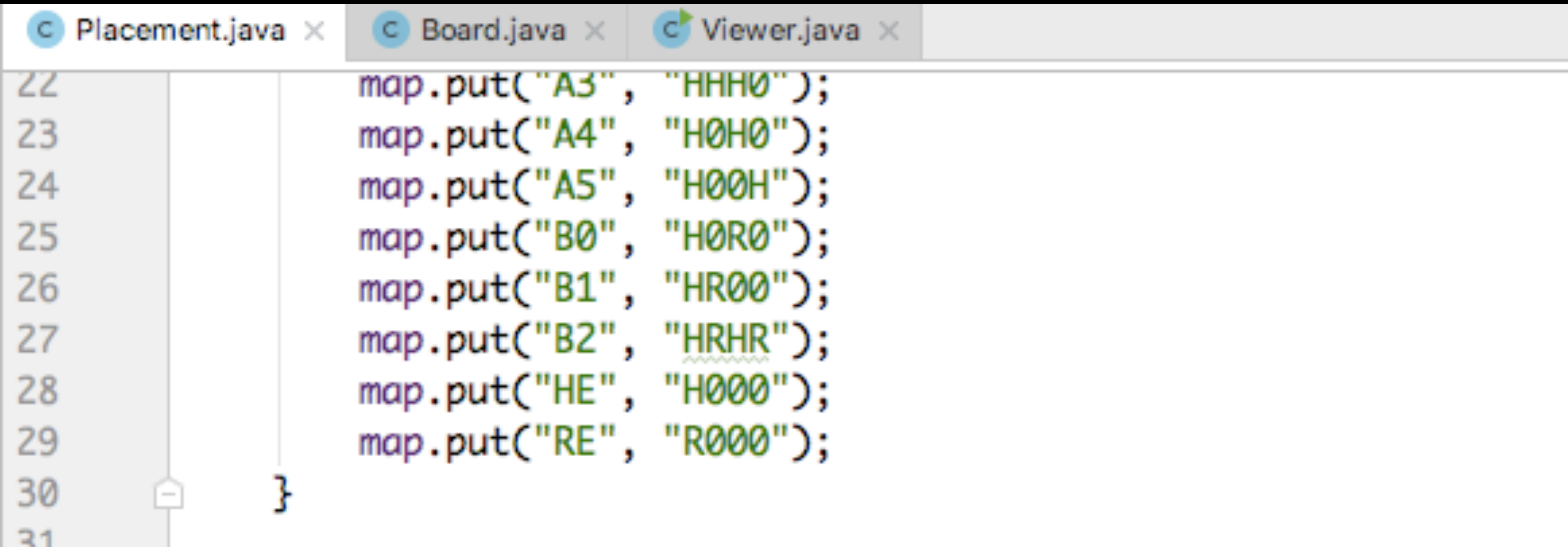
THE REPLACE METHOD

TILE IDENTIFICATION, CENTRALISED.

Developed by Qixia Lu (u6805636)

NAME & RENAME THE SHAPE

Identifying the connection with ease by renaming the shape based on its exits, for a centralised tile management.



```
C Placement.java x C Board.java x C Viewer.java x
22 map.put("A3", "HHH0");
23 map.put("A4", "H0H0");
24 map.put("A5", "H00H");
25 map.put("B0", "H0R0");
26 map.put("B1", "HR00");
27 map.put("B2", "HRHR");
28 map.put("HE", "H000");
29 map.put("RE", "R000");
30 }
31
```

HERE AND THERE

The method has been used across the entire assignment, including key tasks such as Task 5 and Task 8.

```
33 public String replace(String a){
34     String State = map.get(a.substring(0,2));
35     StringBuilder replace = new StringBuilder(State);
36     char indexUp = State.charAt(0);
37     char indexRight = State.charAt(1);
38     char indexDown = State.charAt(2);
39     char indexLeft = State.charAt(3);
40     switch (a.charAt(4)){
41         case '1':
42             replace.setCharAt(index: 0, indexLeft);
43             replace.setCharAt(index: 1, indexDown);
```


THE REPLACE METHOD

TO THE EDGE

Finding the edge connection based on its shape.

```
439 Placement p = new Placement();
440 if (pL == '0' || pL == '6' || pR == 'A' || pR == 'G'){
441     //use the replace method to find whether the tile side has touched the edge of the board
442     if (pL == '0'){
443         if (p.replace(t).charAt(3) != '0'){
444             count++;
445         }
446     }
447     if (pL == '6'){
448         if (p.replace(t).charAt(1) != '0'){
449             count++;
450         }
451     }
452     if (pR == 'A'){
453         if (p.replace(t).charAt(0) != '0'){
454             count++;
455         }
456     }
457     if (pR == 'G'){
458         if (p.replace(t).charAt(2) != '0'){
459             count++;
460         }
461     }
462 }
```

DESIGN

GROUPING

FOR AN ORGANISED JFX

Developed by Carry Zhang (u6499267)

GROUPING

VARIOUS SPECIALISED GROUPS

BUTTON

SCORE

EXITS

BOARD

TILES

GRID

DISPLAY

DICE

```
private final Group display
private final Group grid
private final Group board
private final Group button
private final Group score
private final Group exits
private final Group tiles
private final Group dice
```

GROUPING

INDIVIDUALLY CONTROLLED

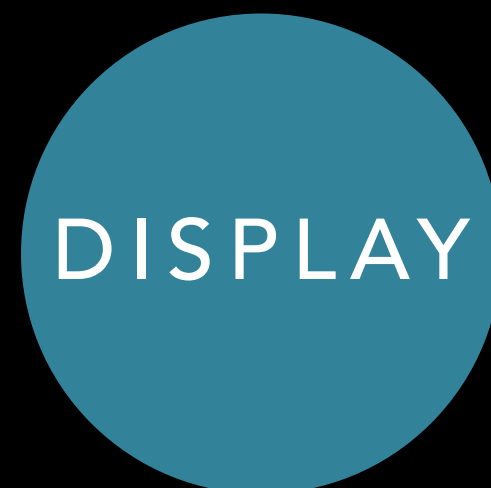
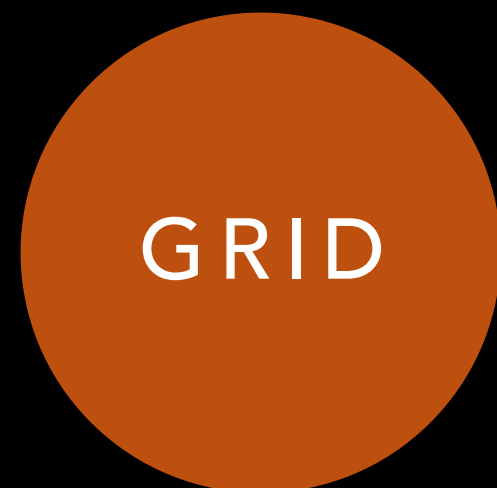
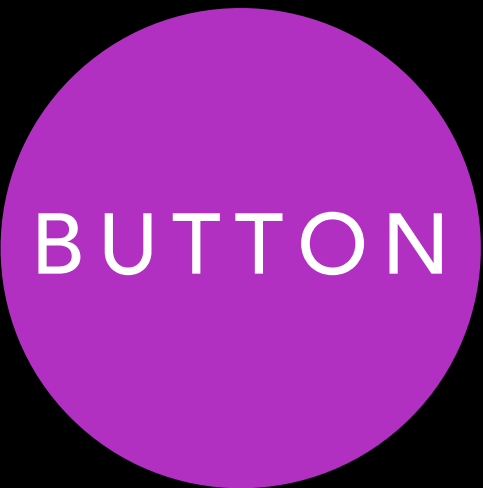


SCORE

The real-time score display is powered by individually controlled Score group by clearing it each time a new placement is added to the board, and stay updated with the method `updateScore`

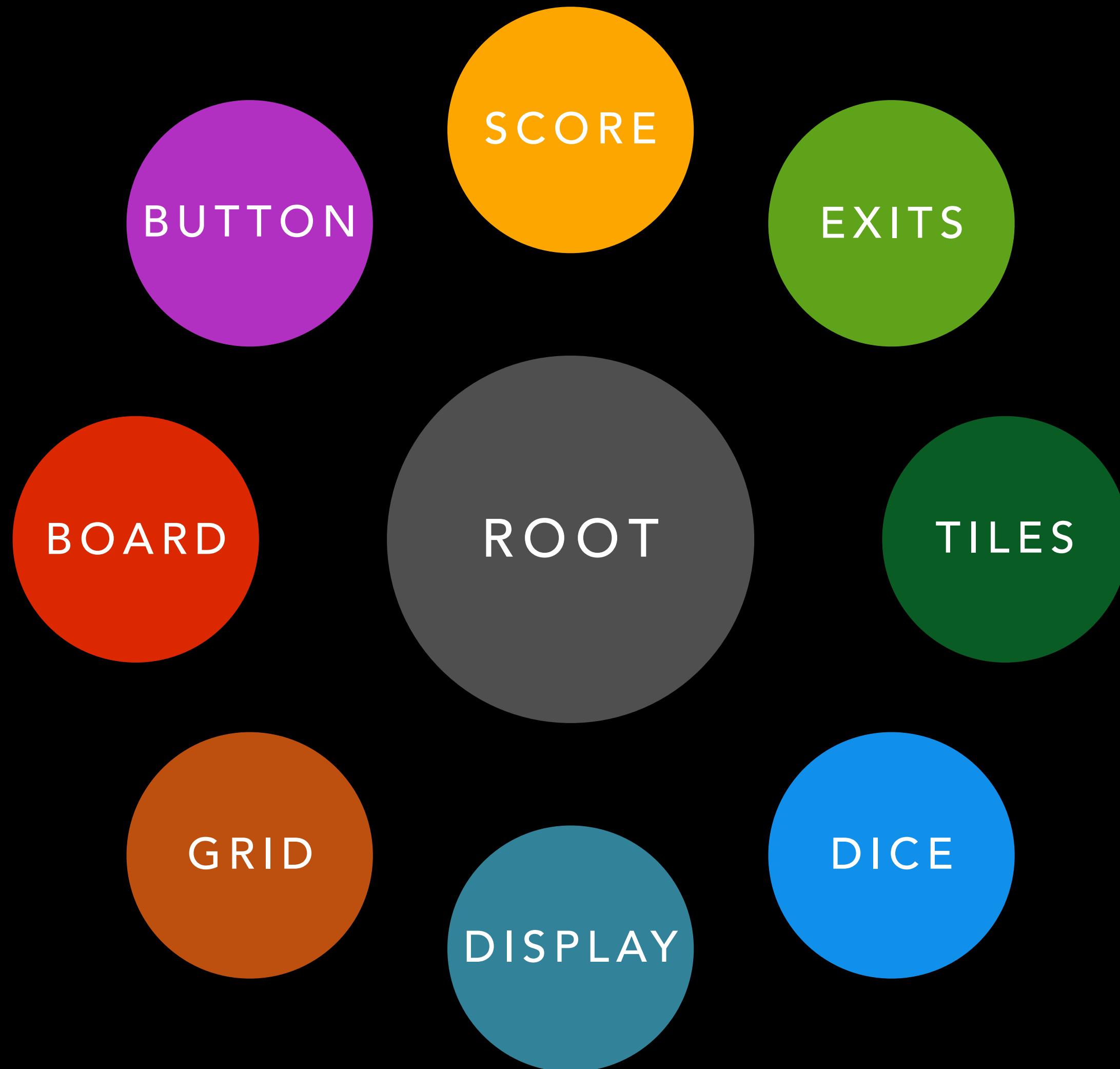
GROUPING

VARIOUS SPECIALISED GROUPS



GROUPING

AND ONE ROOT TO RULE IT ALL



```
private final Group root
Scene scene = new Scene(root...
root.getChildren().addAll(...
```

Demo

SUMMARY

EXTRAS & WRAPPING UP...

- Java-doc standard documentation for the Game class
- User-friendly interface with great functionality
- Object oriented approach used throughout
- Accumulated method to calculate bonus score move by move in **getAdvancedScore** method *Developed by Keyu Liu (u6342137)*

```
/**
 * Move the piece to the new position
 * @param movementX & movementY distance the mouse has moved since its original position
 */

void drag(double movementX, double movementY) {
    setLayoutX(getLayoutX() + movementX);
    setLayoutY(getLayoutY() + movementY);
}
```

ANY QUESTIONS?

QUESTIONS & ANSWERS