COMP1110 ASSIGNMENT 2

FINAL PRESENATION

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

OVERVIEW

BY THE NUMBERS

Over
140
Commits

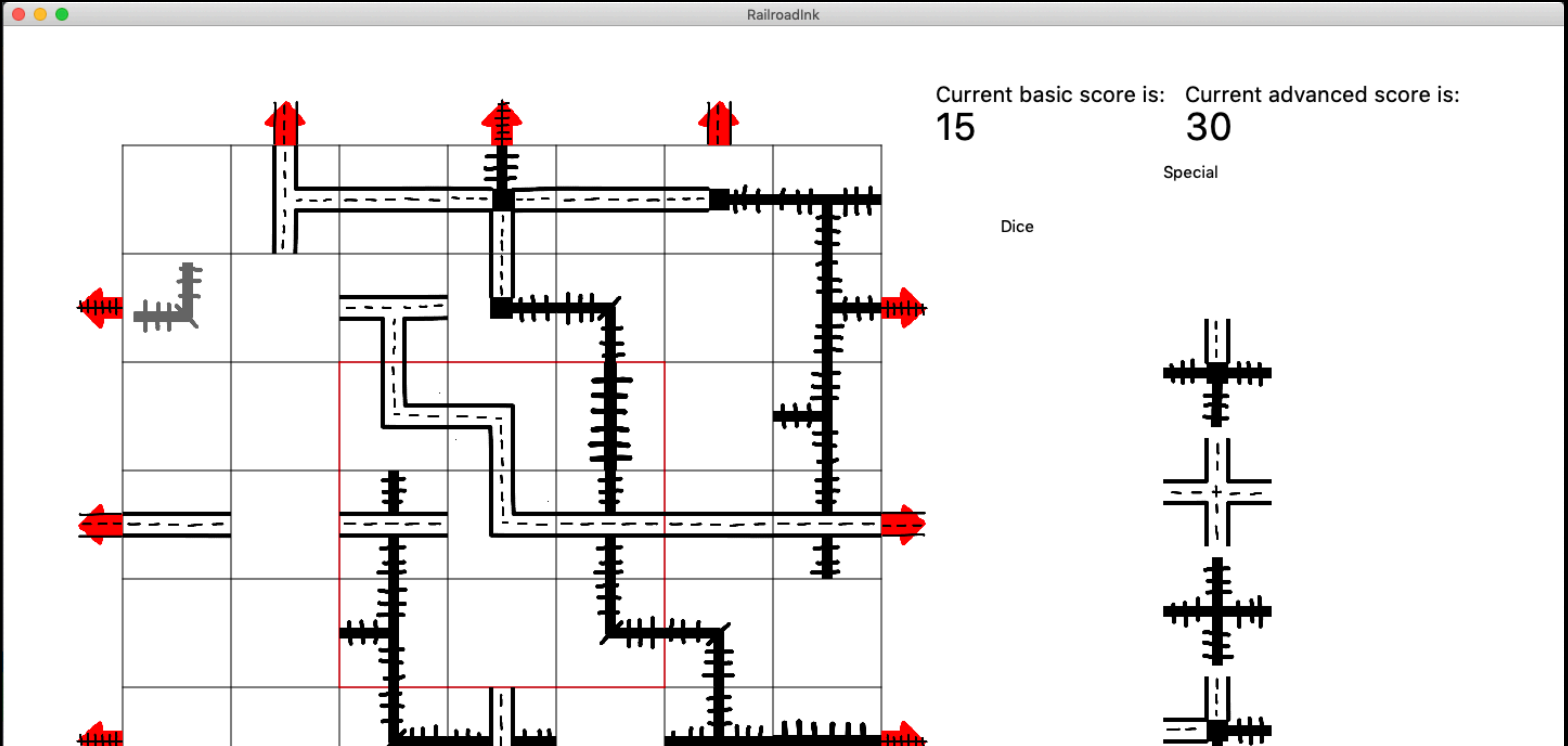
5
Java Class Files

302 MB
File Size

11
To-Dos Completed

USER INTERFACE

THE POWER OF SIMPLICITY



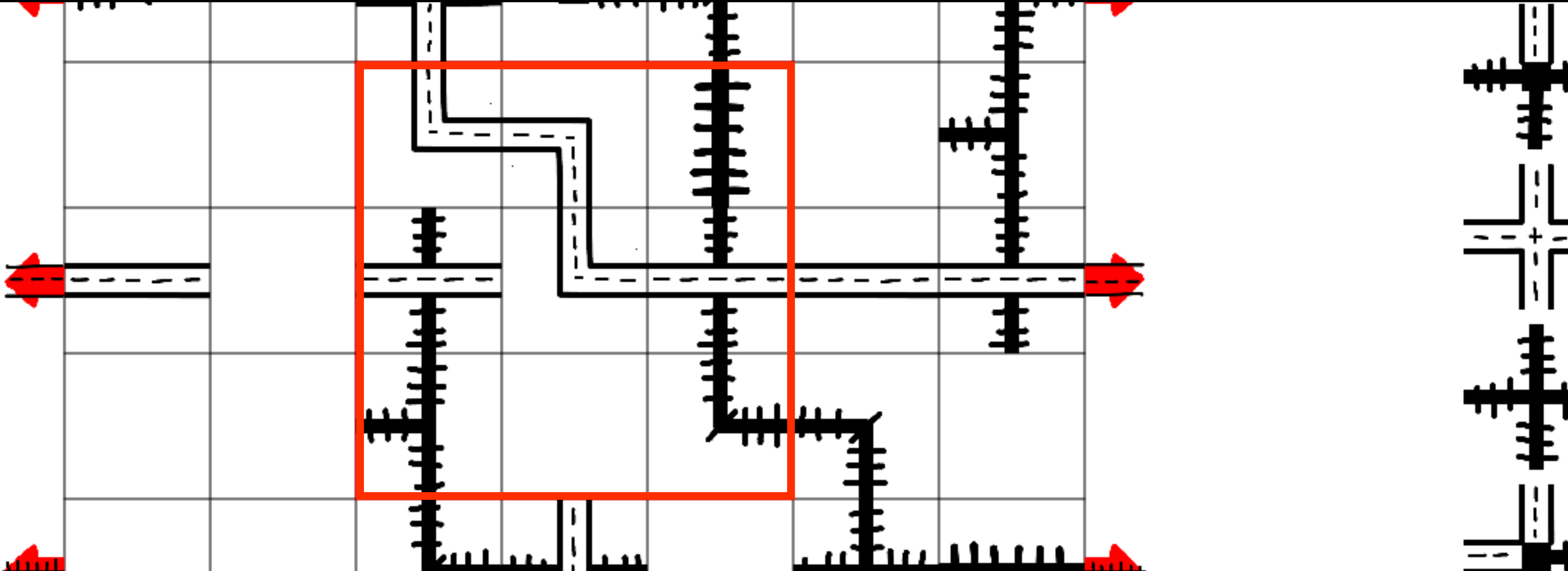
Current basic score is:
15

Current advanced score is:
30

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.



USER INTERFACE

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

USER INTERFACE

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

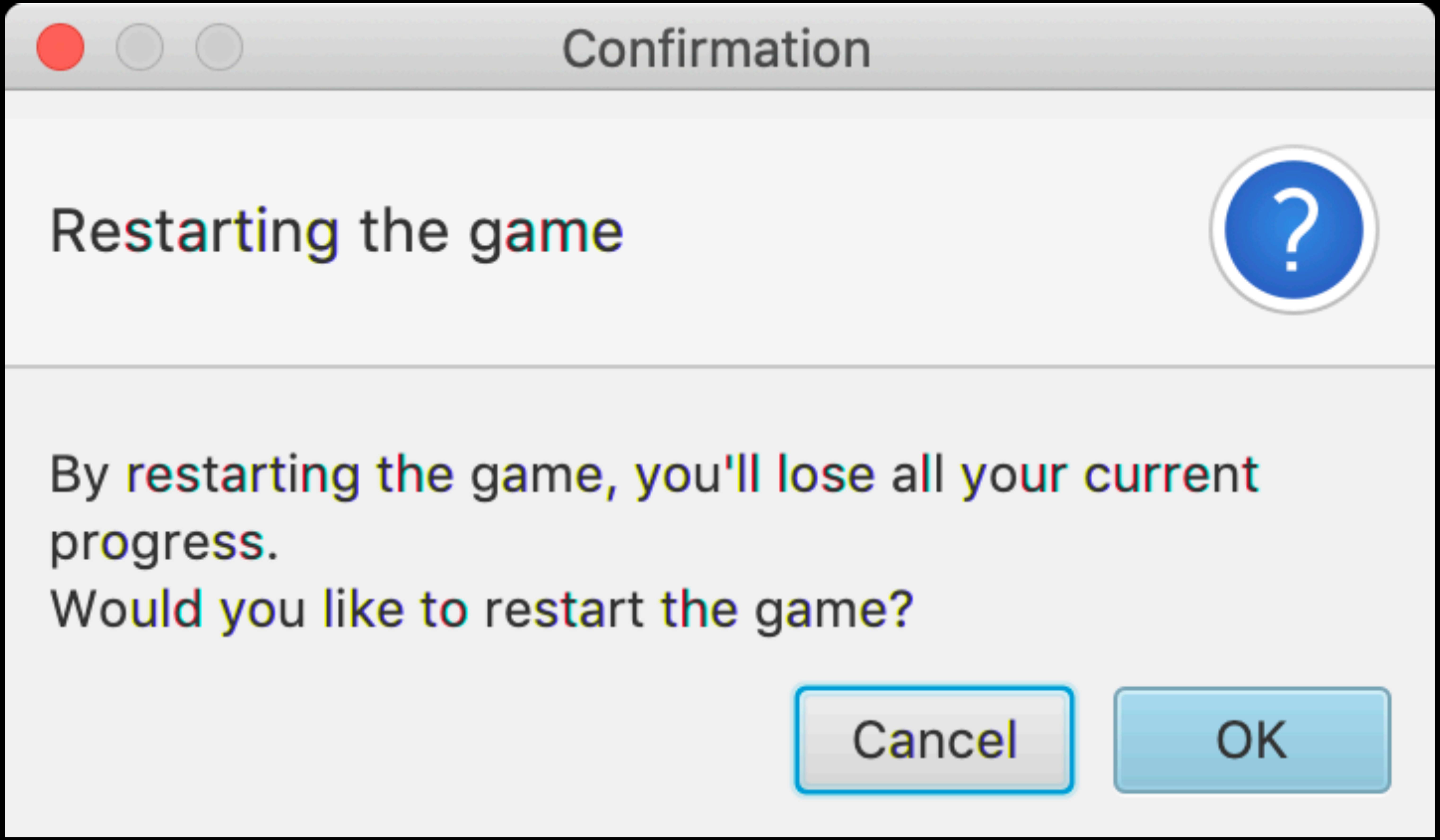
FOCUS ON WHAT MATTERS

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

USER INTERFACE

STAY IN CONTROL

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

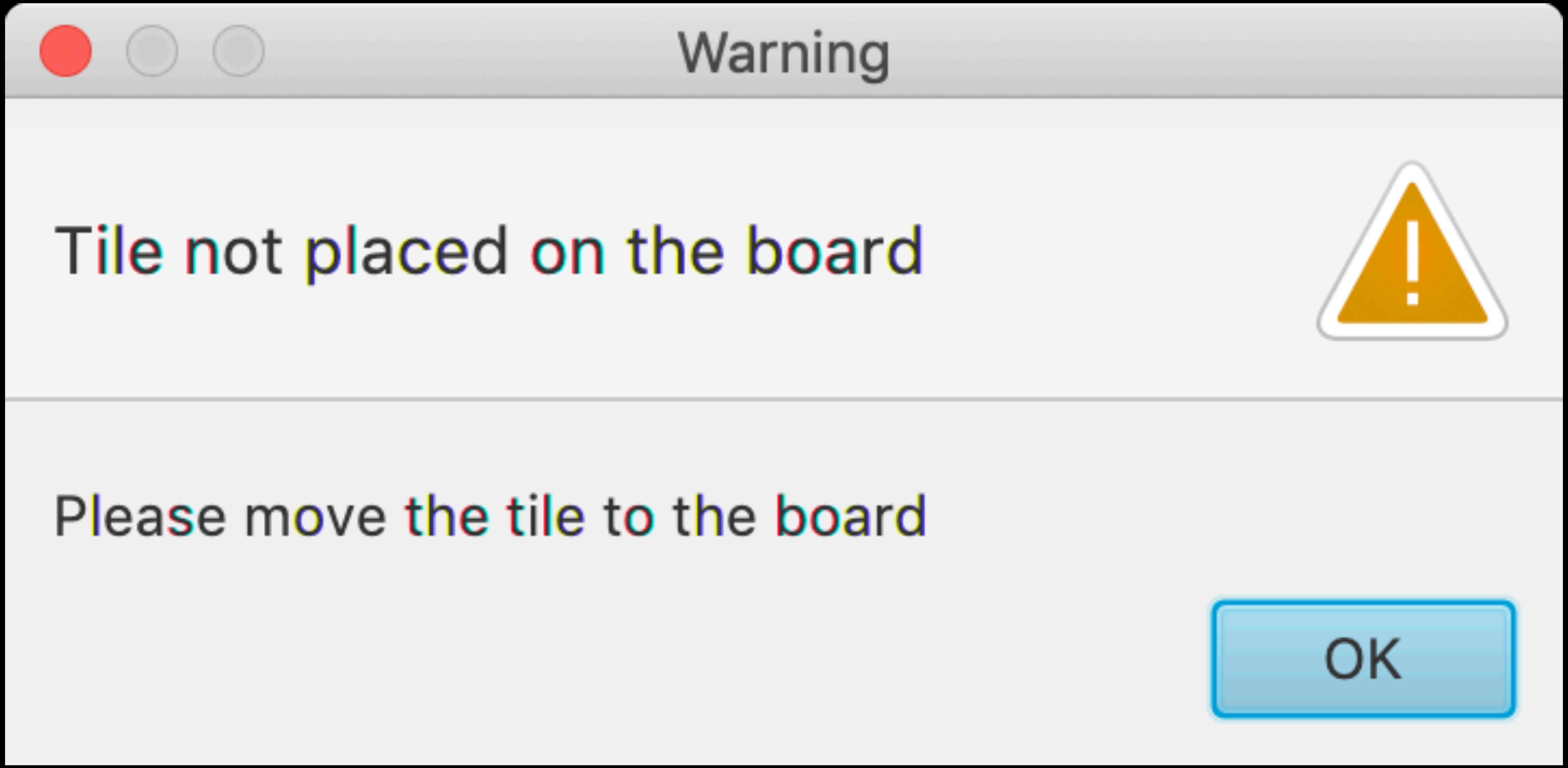

Confirmation

Restarting the game

By restarting the game, you'll lose all your current progress.
Would you like to restart the game?

IN THE KNOWN

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



Warning

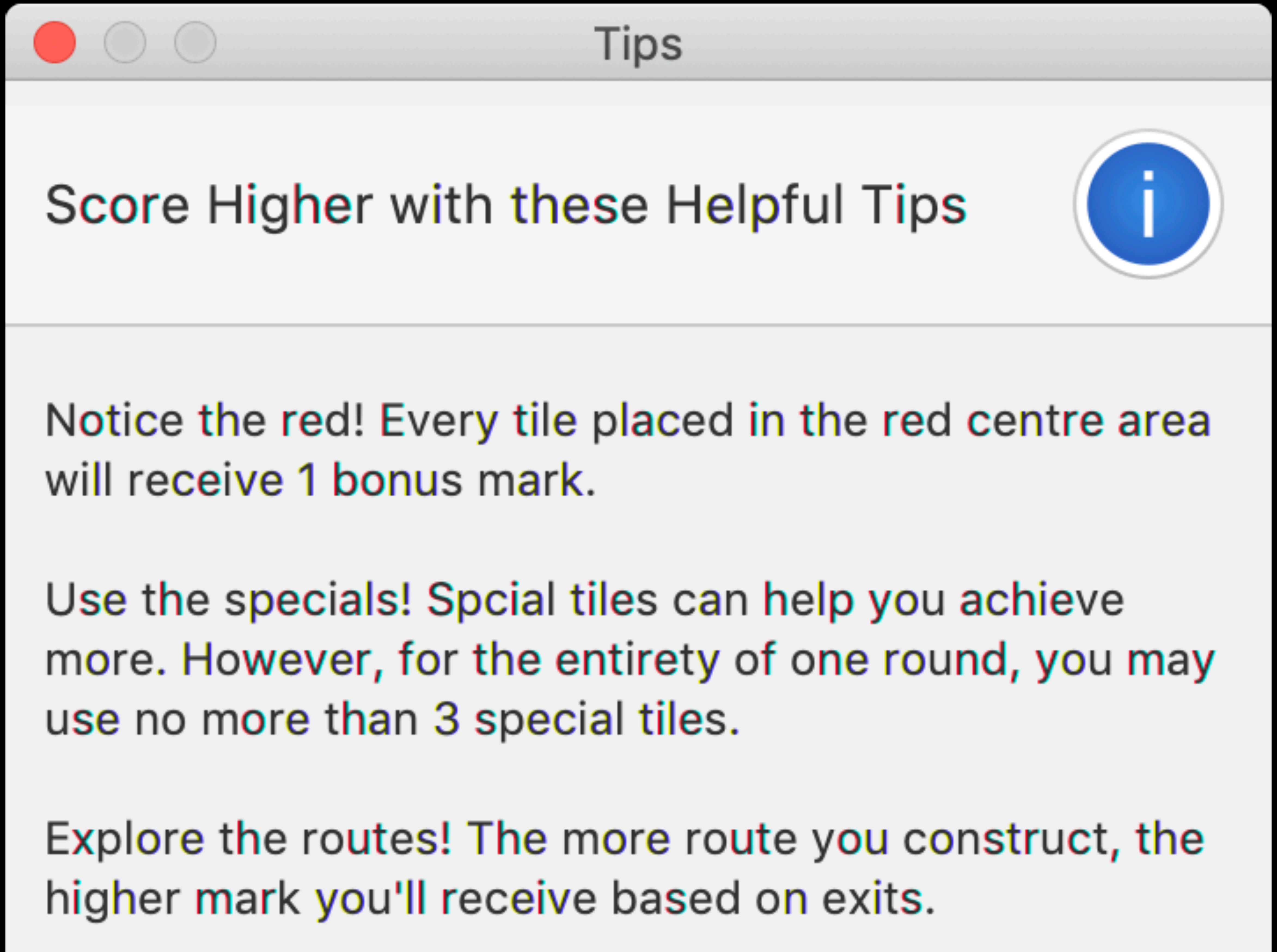
Tile not placed on the board

Please move the tile to the board

USER INTERFACE

BE A PRO

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


Tips

Score Higher with these Helpful Tips

Notice the red! Every tile placed in the red centre area will receive 1 bonus mark.

Use the specials! Spcial tiles can help you achieve more. However, for the entirety of one round, you may use no more than 3 special tiles.

Explore the routes! The more route you construct, the higher mark you'll receive based on exits.

DESIGN

THE REPLACE METHOD

TILE IDENTIFICATION, CENTRALISED.

Developed by Qixia Lu (u6805636)

THE REPLACE METHOD

NAME & RENAME THE SHAPE

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

Placement.java
Board.java
Viewer.java

```
        map.put("A3", "HHH0");
        map.put("A4", "H0H0");
        map.put("A5", "H00H");
        map.put("B0", "H0R0");
        map.put("B1", "HR00");
        map.put("B2", "HRHR");
        map.put("HE", "H000");
        map.put("RE", "R000");
    }
```

HERE AND THERE

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

```
    public String replace(String a){
        String State = map.get(a.substring(0,2));
        StringBuilder replace = new StringBuilder(State);
        char indexUp = State.charAt(0);
        char indexRight = State.charAt(1);
        char indexDown = State.charAt(2);
        char indexLeft = State.charAt(3);
        switch (a.charAt(4)){
            case '1':
                replace.setCharAt( index: 0, indexLeft);
```

THE REPLACE METHOD

TO THE EDGE

Finding the edge connection based on its shape.

```
        Placement p = new Placement();
        if (pL == '0' || pL == '6' || pR == 'A' || pR == 'G'){
            //use the replace method to find whether the tile side has touched the edge of the board
            if (pL == '0'){
                if (p.replace(t).charAt(3) != '0'){
                    count++;
                }
            }
            if (pL == '6'){
                if (p.replace(t).charAt(1) != '0'){
                    count++;
                }
            }
            if (pR == 'A'){
                if (p.replace(t).charAt(0) != '0'){
                    count++;
                }
            }
            if (pR == 'G'){
                if (p.replace(t).charAt(2) != '0'){
```

DESIGN

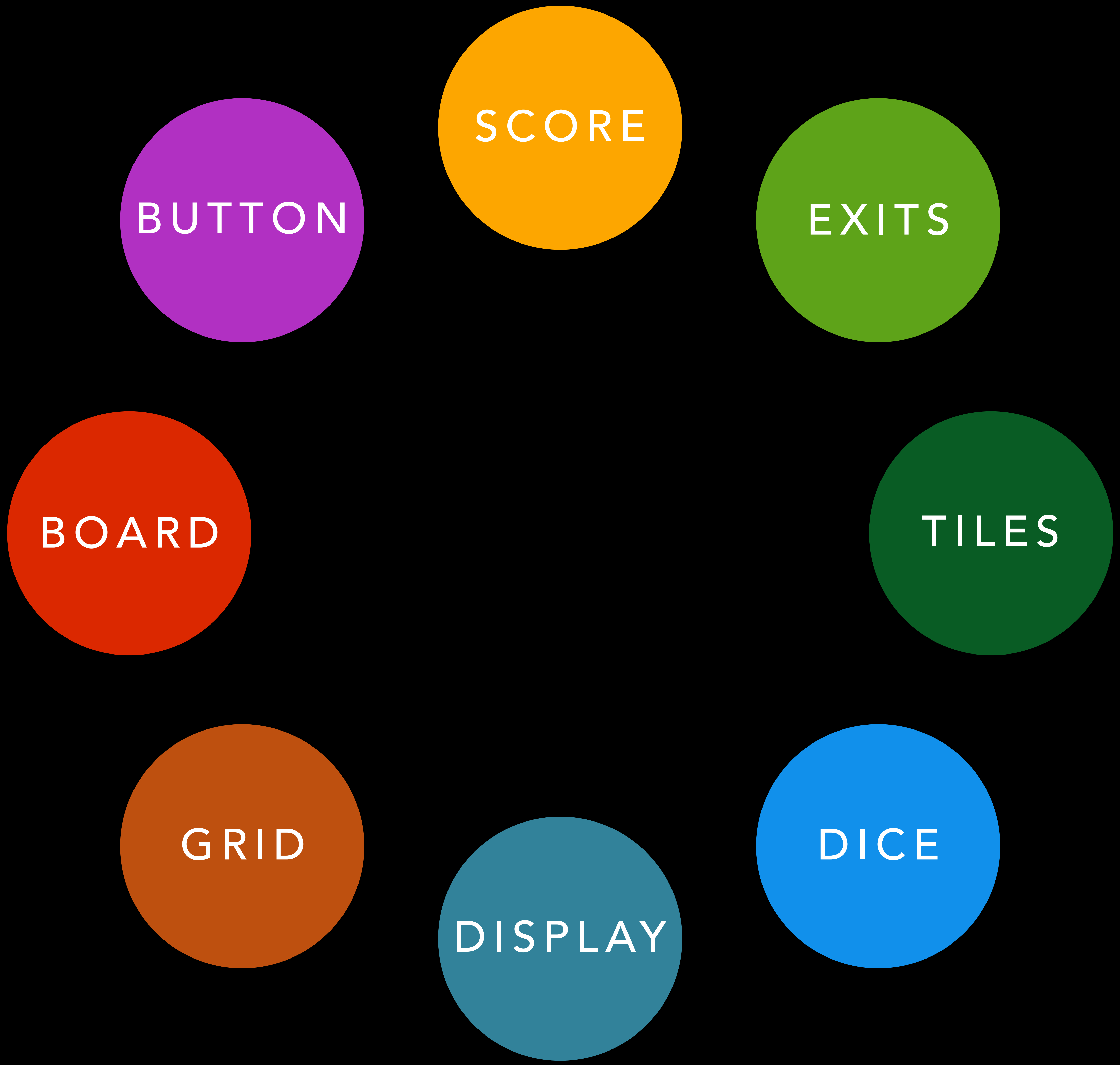
GROUPING

FOR AN ORGANISED JFX

Developed by Carry Zhang (u6499267)

GROUPING

VARIOUS SPECIALISED GROUPS



```
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

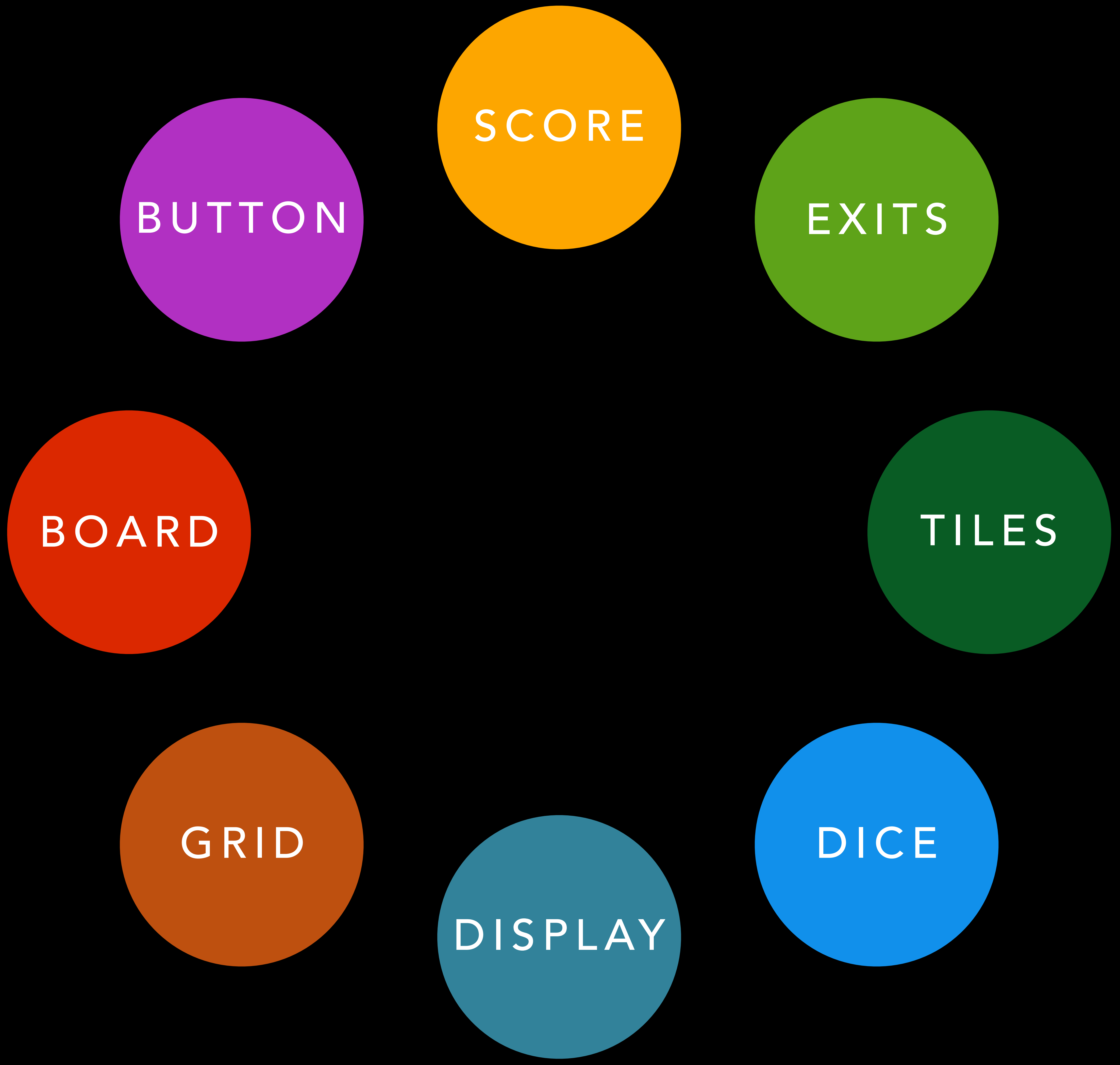
INDIVIDUALLY CONTROLLED

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()`

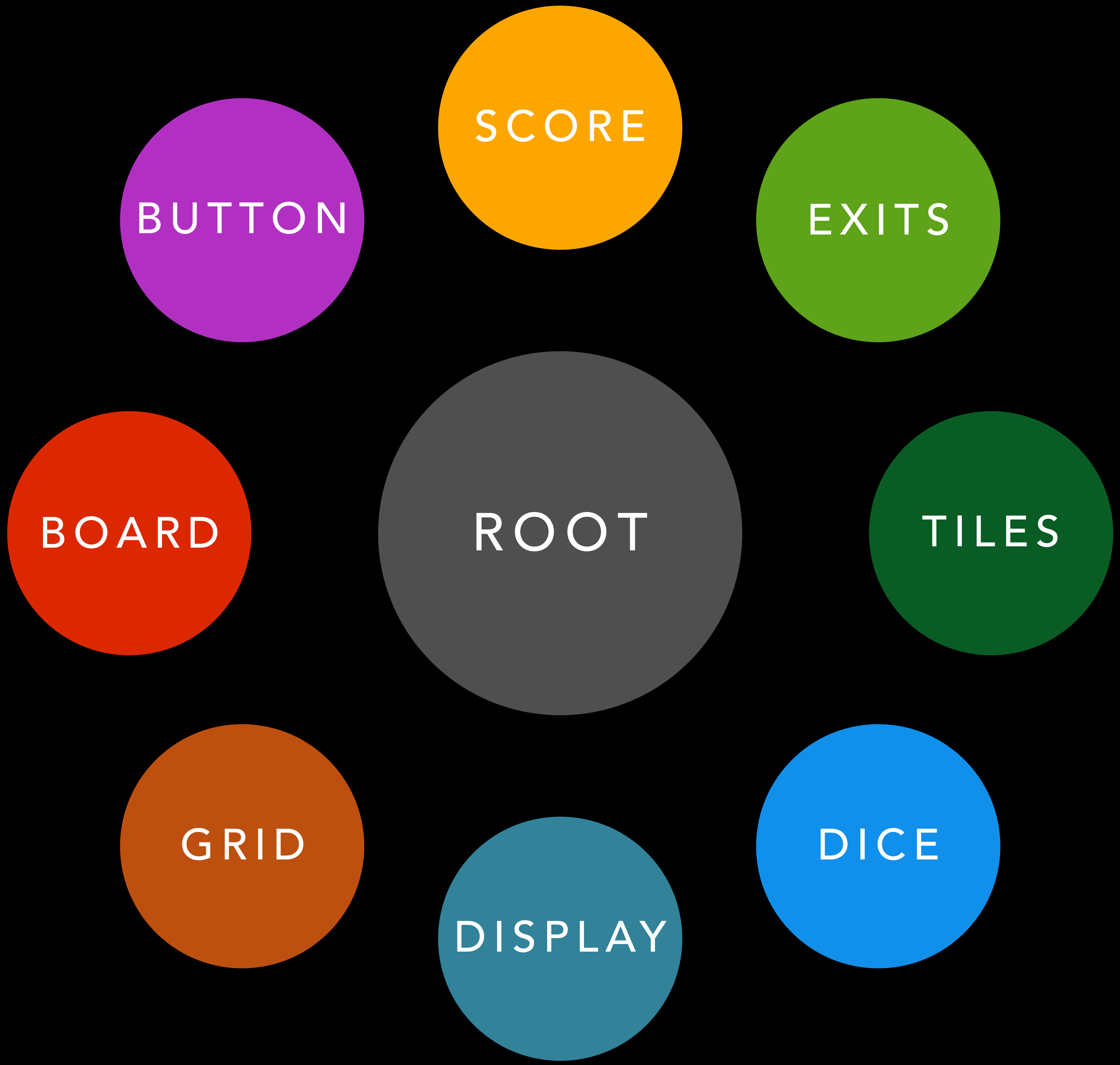
GROUPING

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
- Partly inspired by the code from Assignment 1

```java
/**
 * 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);
}

/**
 * Check if the piece is on the board
```

ANY QUESTIONS?

QUESTIONS & ANSWERS