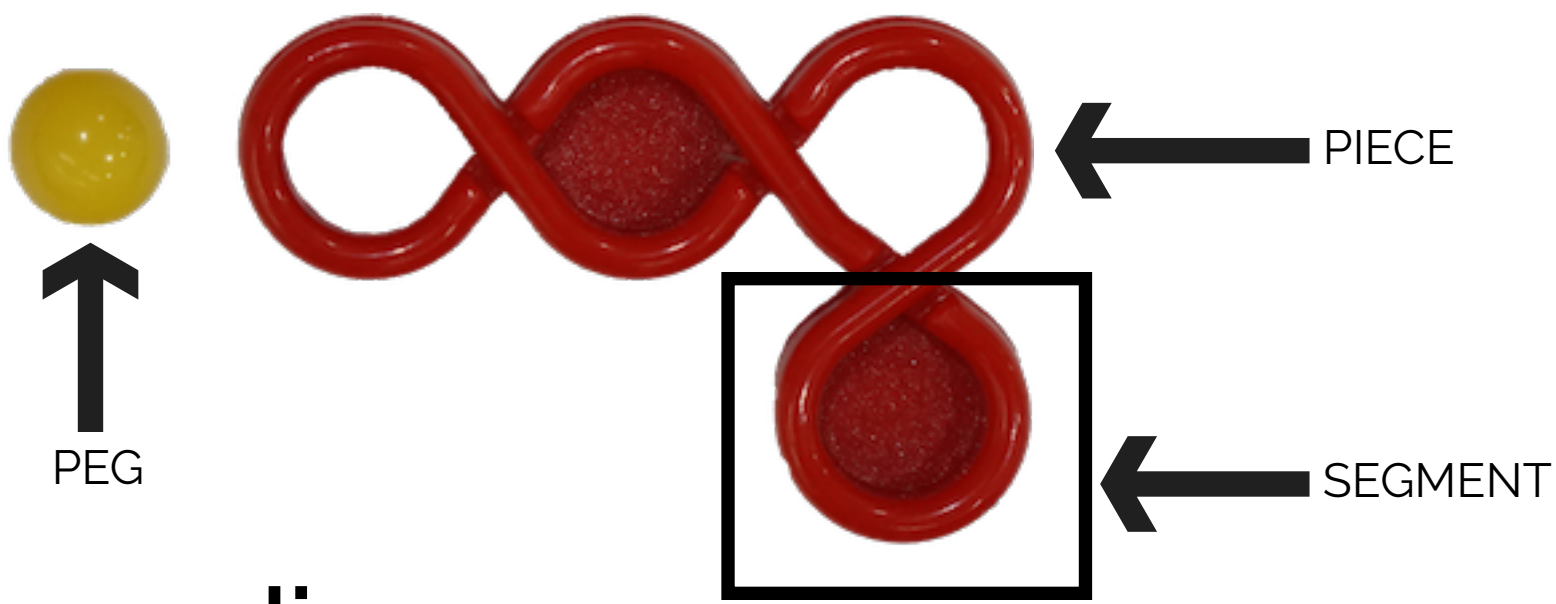
COMP1110 Assignment 2

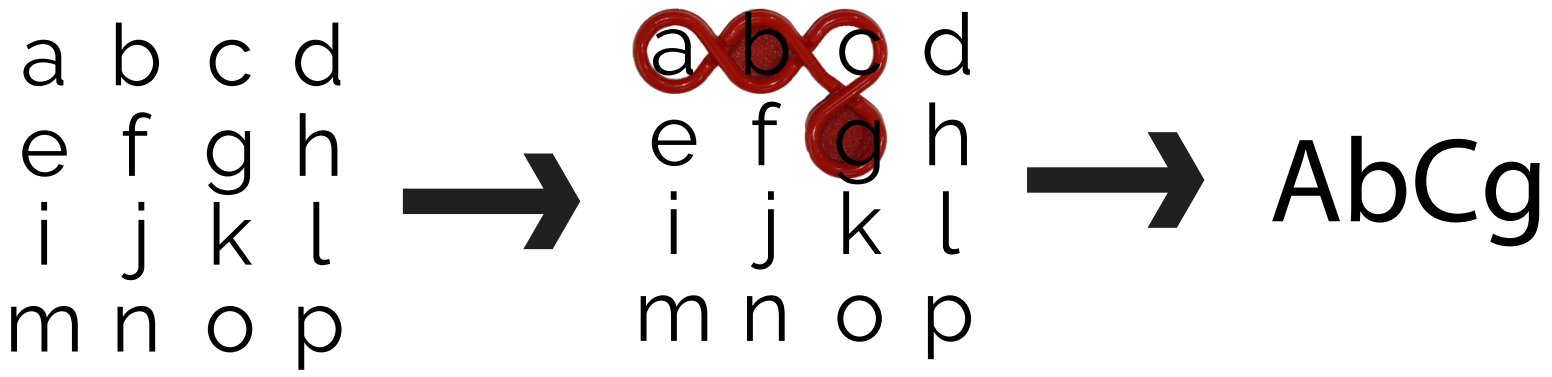
Saffron Bannister (u6062525), Lauren Nelson-Lee (u6378754) and Zhewen Li (Lily) (u6437091)
IQ-Twist
group thu14n

IQ-Twist

We all know what this game is by now but this is how we implemented it:
Pieces and pegs are represented as classes, with associated enums to represent their shape

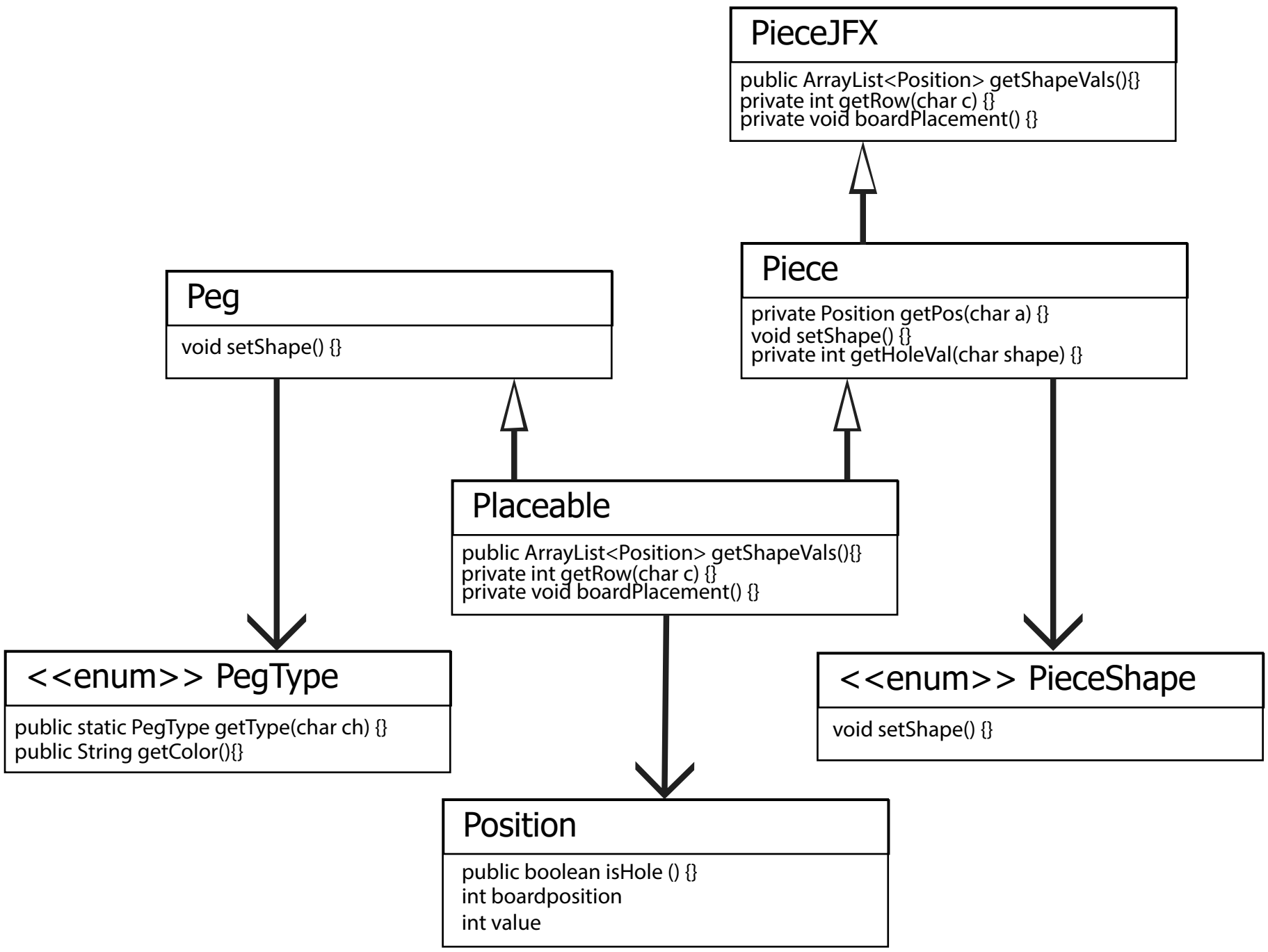


Piece encoding



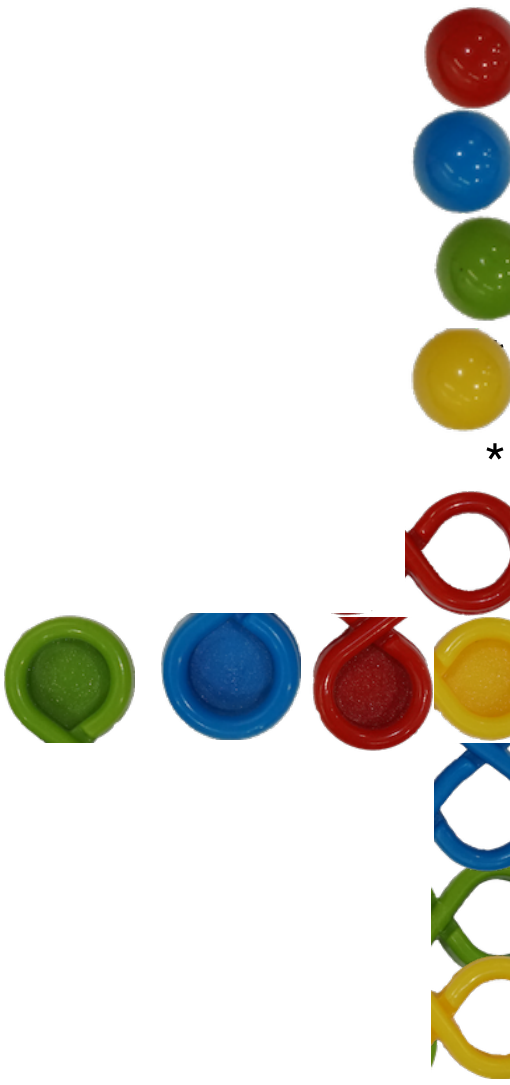
Imagine pieces as existing on a 4 x 4 space where each space is represented by a character.

Then, in order to differentiate between a hole or solid segment you then use capital or lowercase letters to represent the encoding (capital for hole and lowercase for filled)



Board Simulation

The board is internally simulated as a size 32 array of ints, with each int representing a state for that spot on the board, like this:

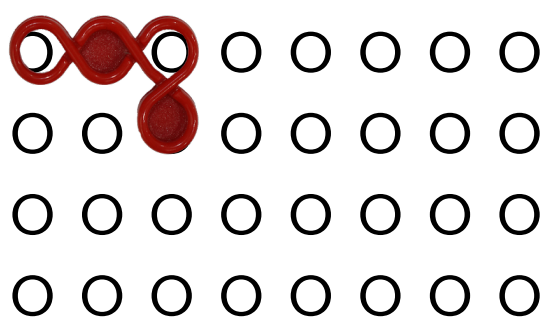


- 1 = red peg
- 2 = blue peg
- 3 = green peg
- 4 = yellow peg

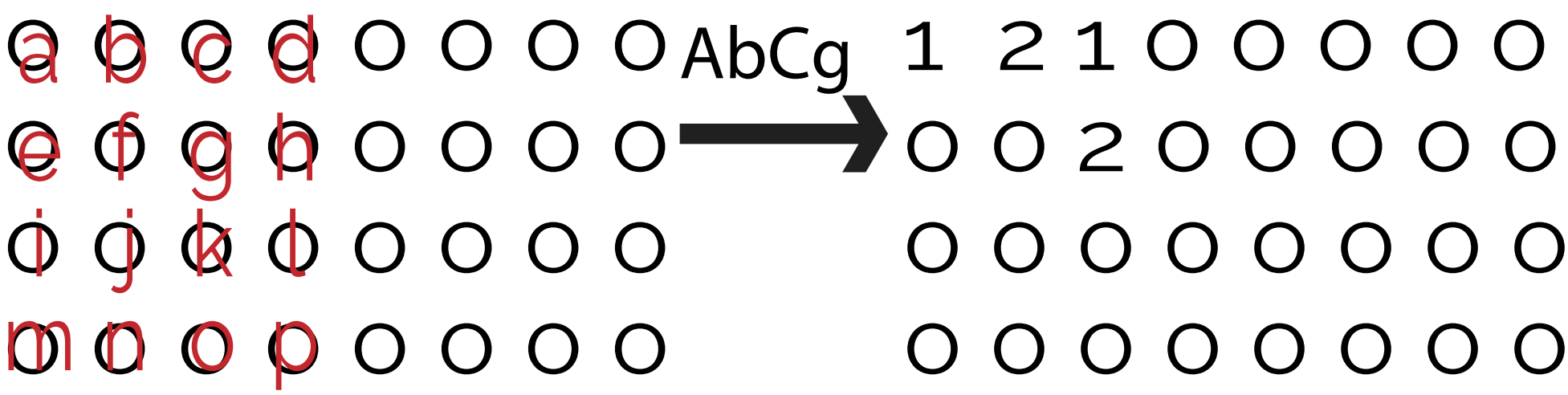
0 = empty space
1 = red hole
2 = filled space
3 = blue hole
4 = green hole
5 = yellow hole

O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O

Here is a size 32 int array representing an empty board
Imagine placing piece a in rot 0 at the spot at 1A (a1A0)



We do this by imagining our previous 4 x 4 space starting at the top left position where we want to place our piece, and placing the segments as indicated



Using this implementation we can check whether piece placement is valid
(by checking against the existing boardstate),

Viable piece placements

Using the board simulator, generates a set of all valid next placements.

Solutions

Gets the set of viable next placements, then recursively adds those to the placements until you have a set of complete solutions.
Has a supporting class to determine symmetry.

Starting placement

Starts with a goal and works backwards, placing pegs in the available holes. The number of pegs is based on a difficulty parameter.

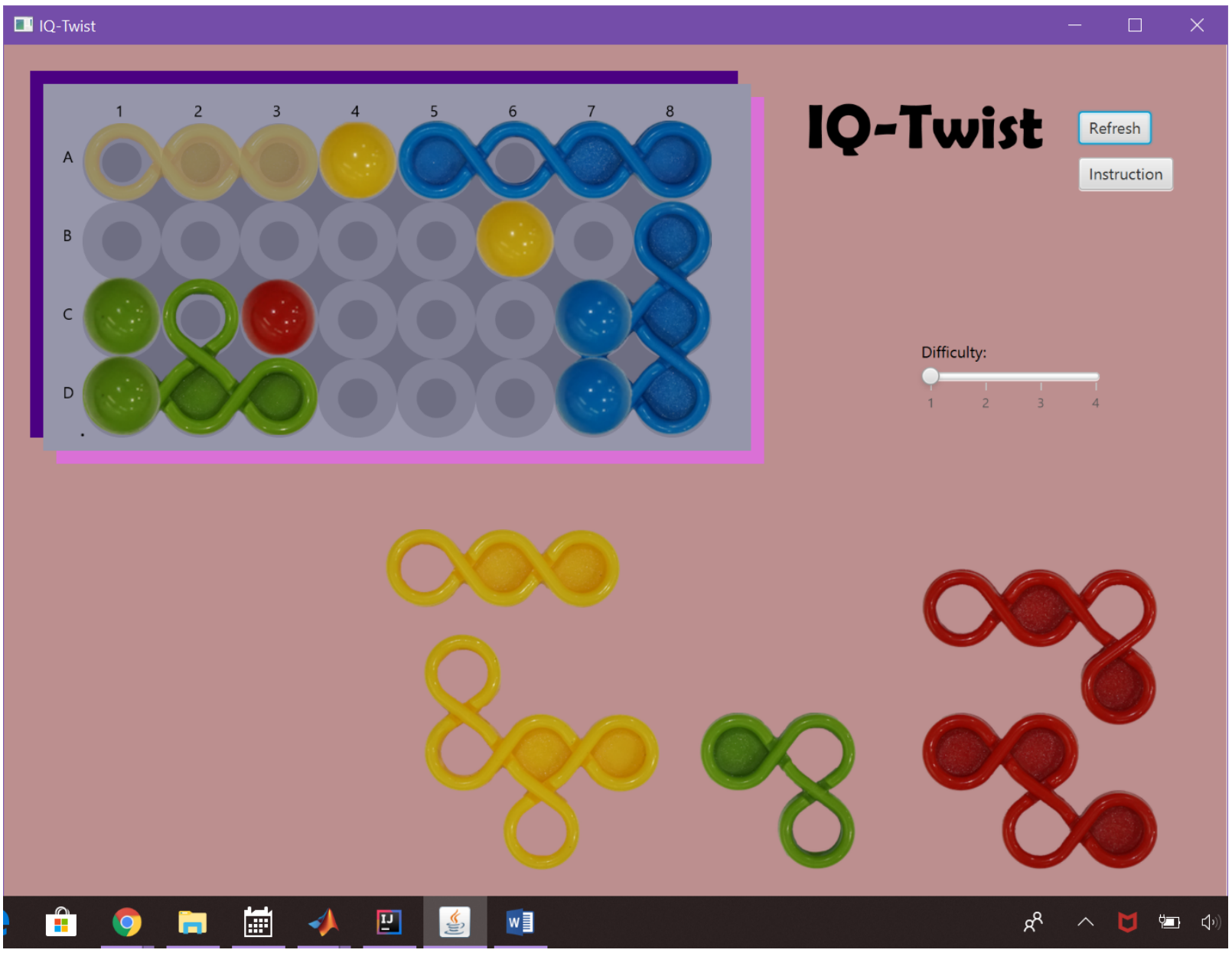
Viewer

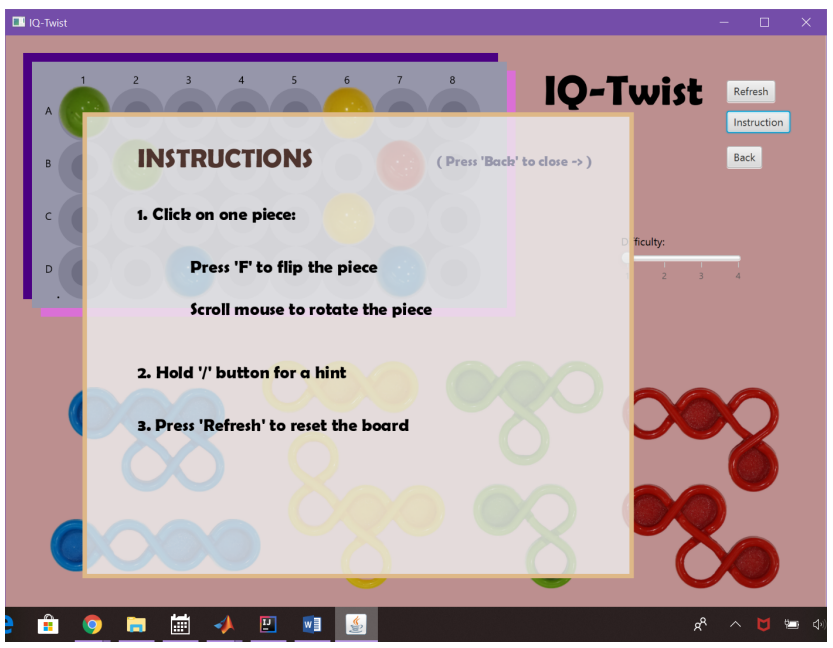
A simple viewer that displays a piece placement. Used for testing other methods.

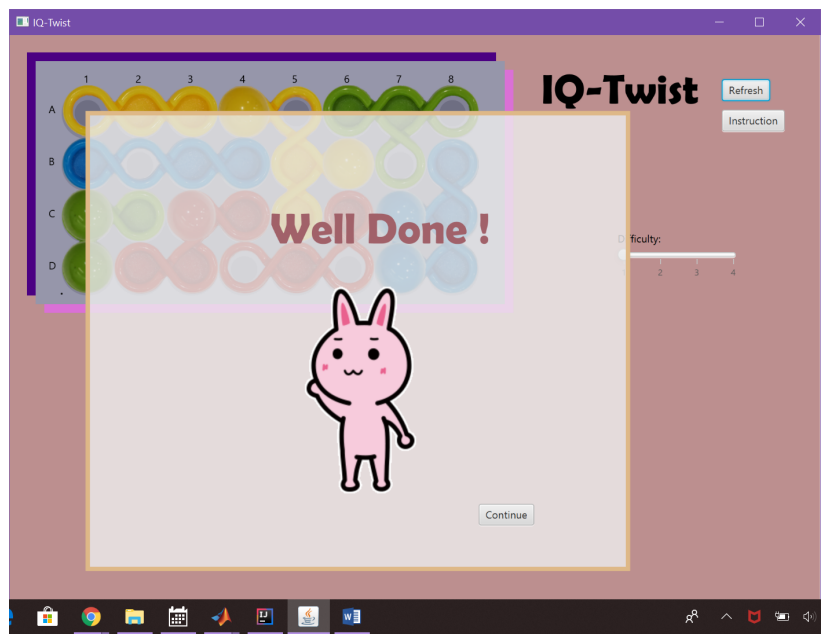
Tests

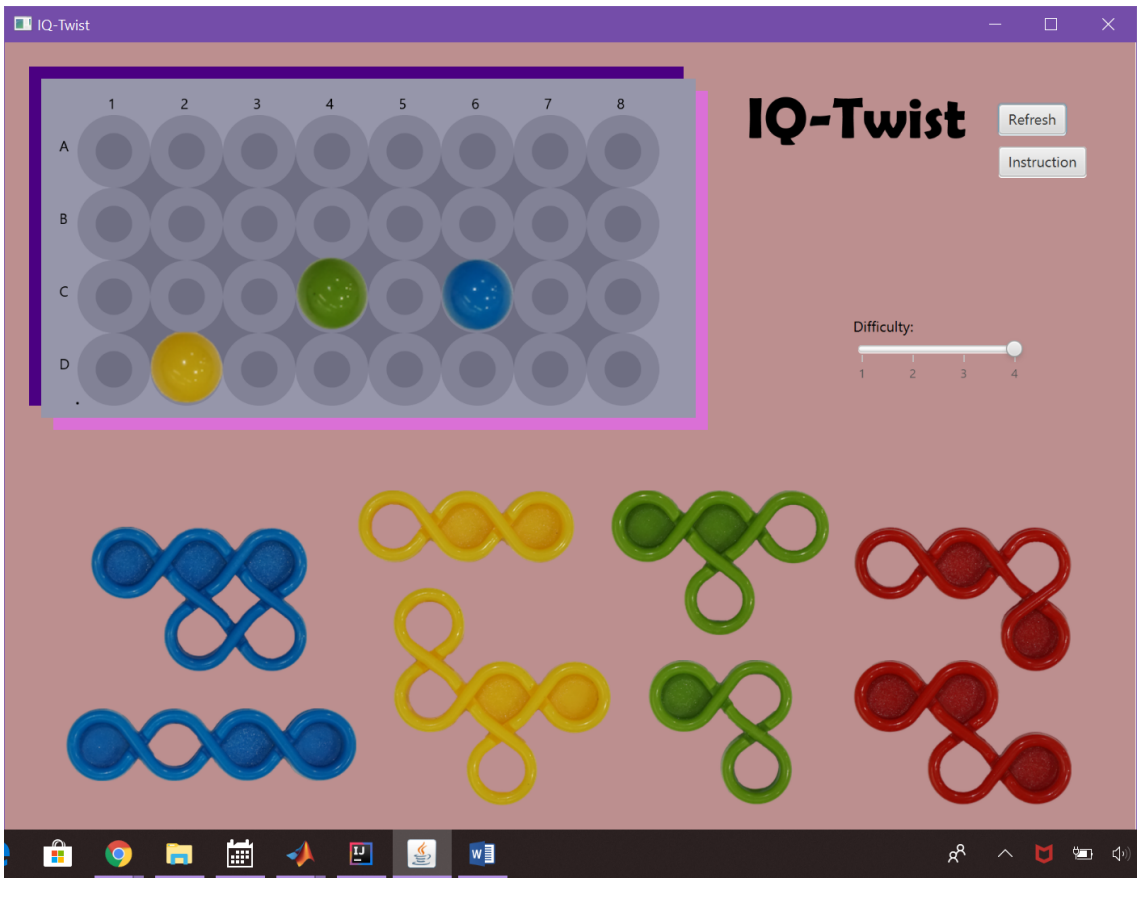
Unit tests were written for many methods that were not tested by the provided framework.
Viewer could also be used to help with testing.

The Game Interface

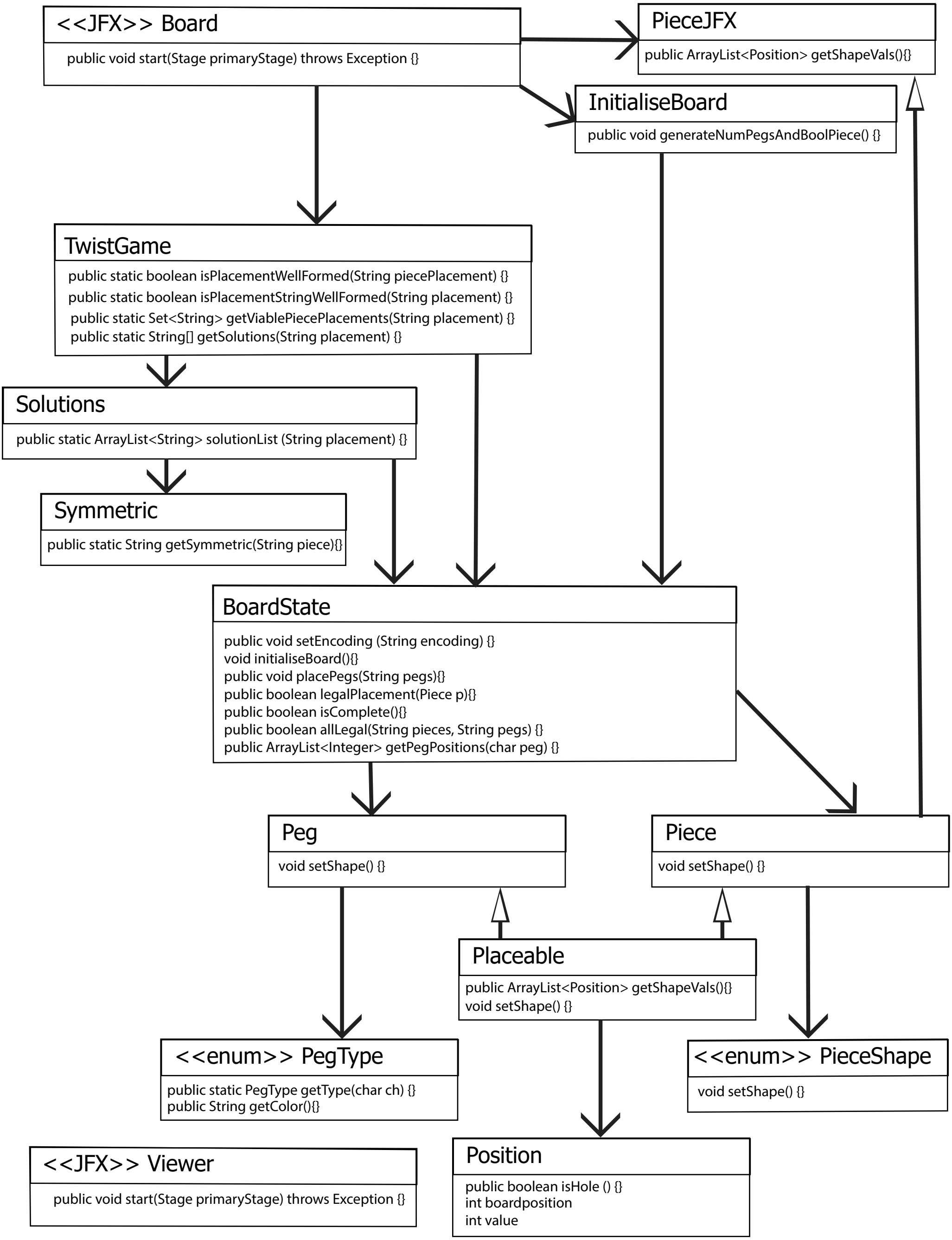








Structure



NOTE: internal methods and get/set methods have been omitted