

COMP1110-ass2-tue12V

Yafei Liu(u6605935)

Xing Meng(u6483085)

Bowen Tan(u6822656)

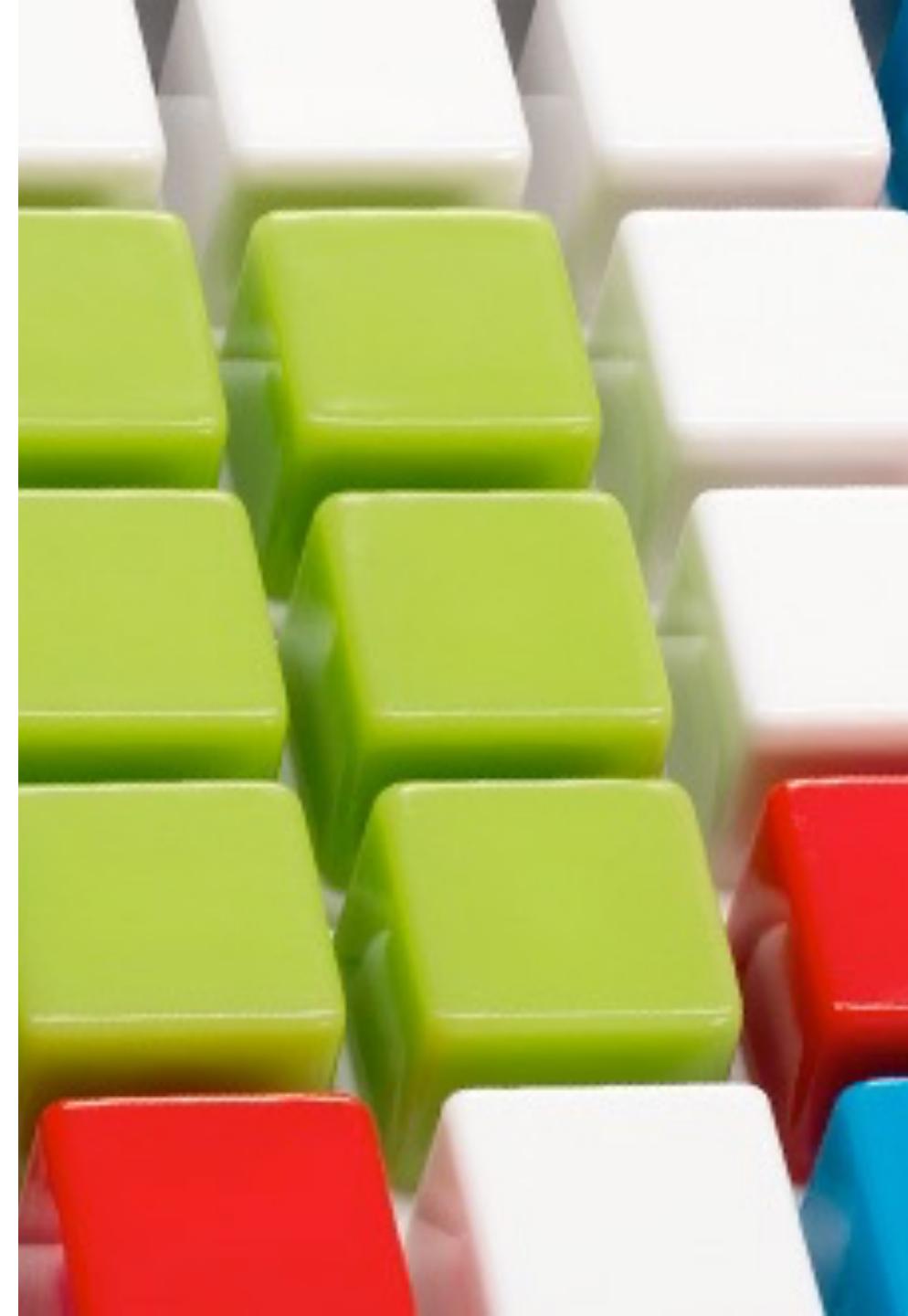


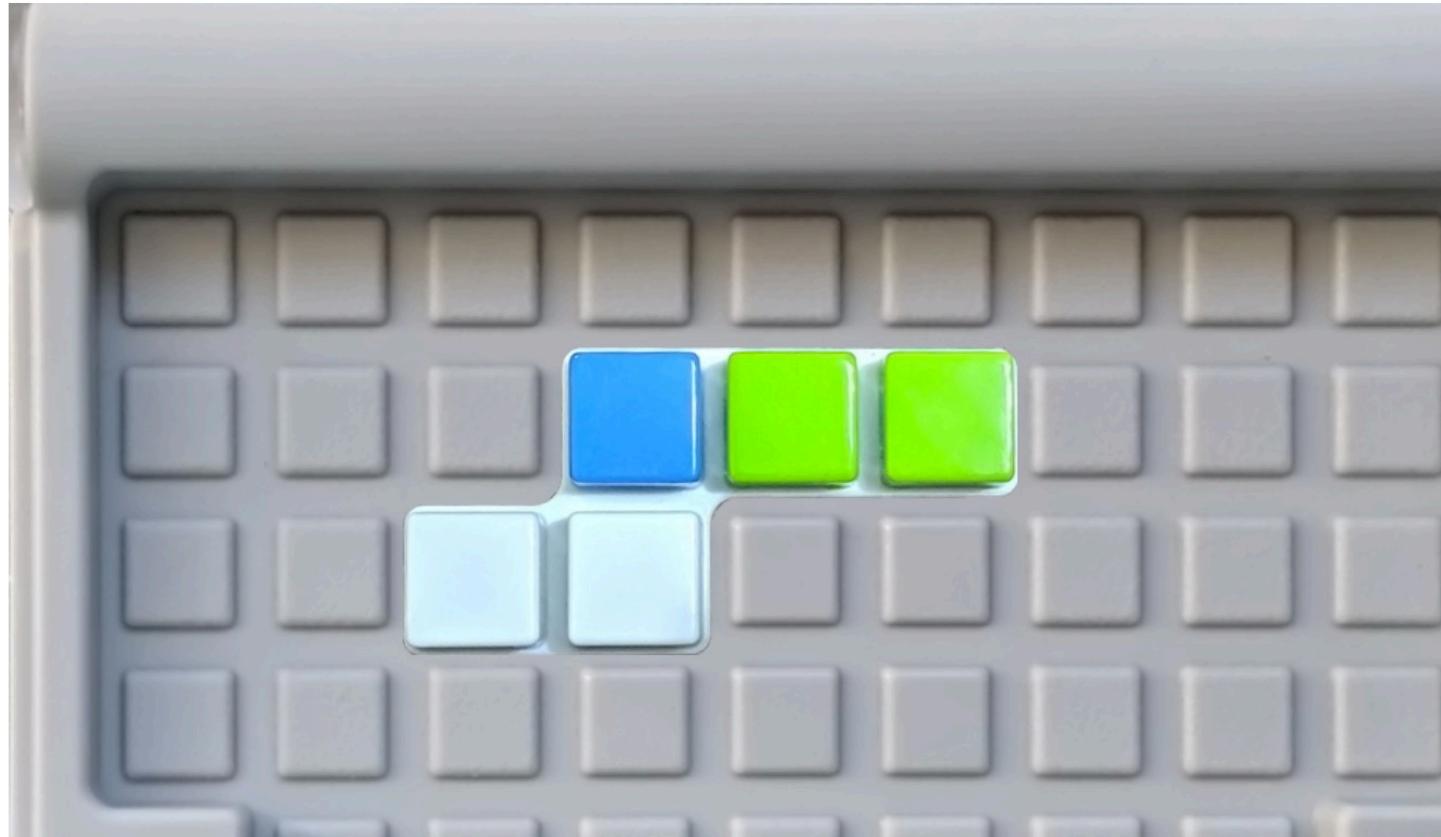
Overview of the game

This project focuses on a puzzle-fixing intellectual game called IQ-focus. It contains various kinds of pieces and requires the player to make every piece cover the whole board. By making this main project, we used both the basic java language and the java-fx method.

Skeleton

- ① Color: classifies the whole colors of the pieces into six kinds including GREEN, BLUE, RED, WHITE, NONE, B. (here NONE stands for the square of board that is not covered by any piece; B stands for the left and right bottom of the board, no piece can cover this two parts.)
- ② Rotation: we defined for kinds of rotation which are 0, 1, 2, 3 according to introductions on GitLAB.
- ③ Piece&PieceType: As there are various kinds of pieces in this game, we created two classes called Piece and PieceType to classify them according to the first letter of the string from a to j represents the pieces.
- ④ Location: This class is for us to set and renew the value of position X and position Y of each pieces.





Task 4

- Before starting the game, we used the method of JAVA-FX to show the board. As the pieces will be placed on it in the next part, so after we run the project we are able to see the board first, which simplified our tasks in task4.
- Next, we inserted the images of those 40 kinds pieces, which will be used in the next JAVA-FX parts.(in the picture its the placement of b210)

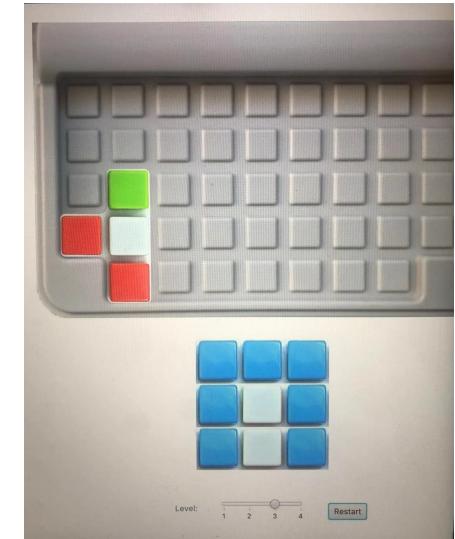
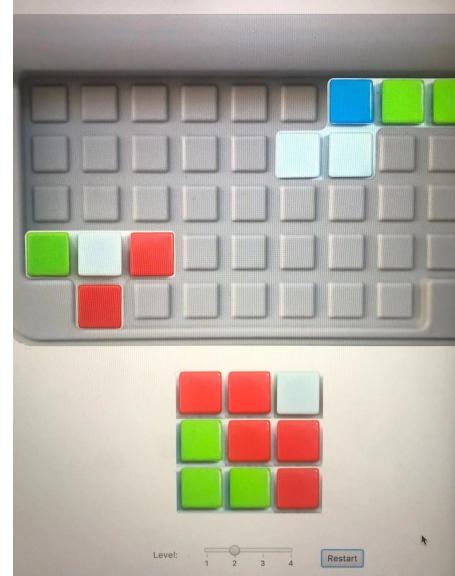
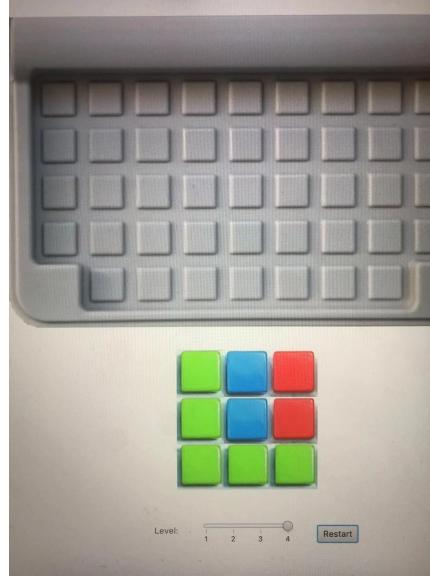


Task2, 3, 5, 9

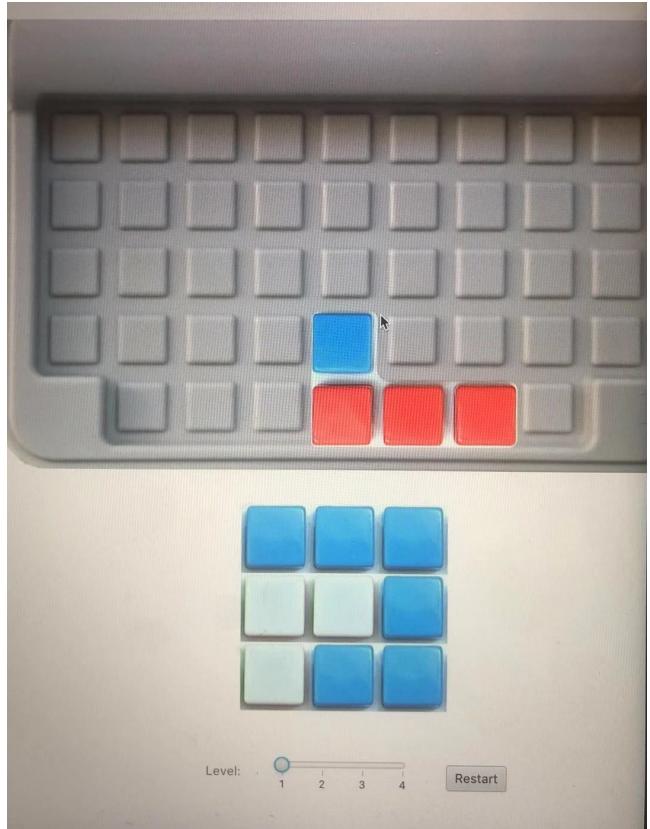
- Task2&3: To determine if placement well formed, which means that to confirm that there is no piece placed out of the board. In this part we mainly used if and while statements.
- Task5: Based on the concept of well formed in task2 and task3, this part is for judging whether a piece overlaps another one.
- Task9: It asks to determine all the solutions to the different challenges of the game. Similarly it could be solved by if statements after creating a set of string.

Task 8

- This task requires us to implement challenges according to the asserts that have been provided.
- First of all, we made a difficulty control bar at the bottom of the game. The minimum difficulty is 1 and maximum difficulty is 4.
- Then we created a variable called button used for restarting a game by different difficulties when press it.
- Finally, we wrote a method called newGame to start a new game with a difficulty.



Task 10



This part requires to provide hints for a challenge, give the next piece in solution that you have not placed. We considered that if the player finish all the pieces, then the game do not provide him any hints cause there is no hint any more. It show a good job text at draggable class instead if the player complete the whole game. So we firstly wrote a function to determine whether the game has been finished. Showing the hints by press key SLASH and hide hints by release SLASH.

- In conclusion, most of the tasks have been done well and could finish running in the boundary of time.
- But we think some tasks still contains parts that could be fixed. As an example, we did not make the function works in task 11 works.
- By working on this project, we enhanced the learning and cooperating ability . We also got more familiar to programming.

