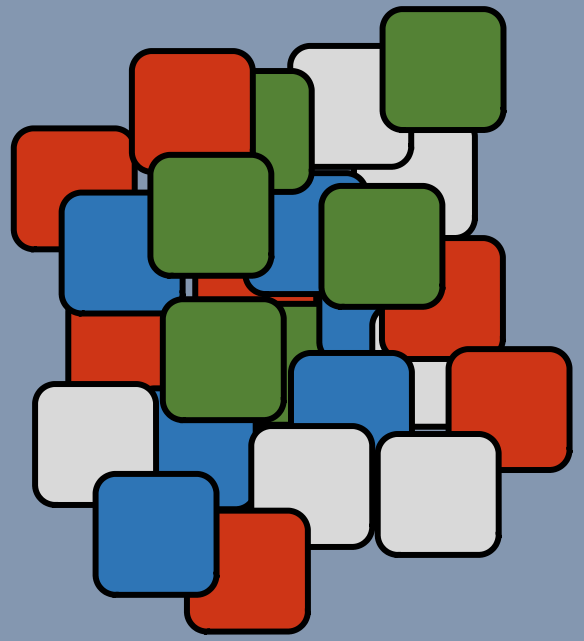


IQ-FOCUS :

JAVA Game Development
Assignment

Our Team

Li Anbang

BIT

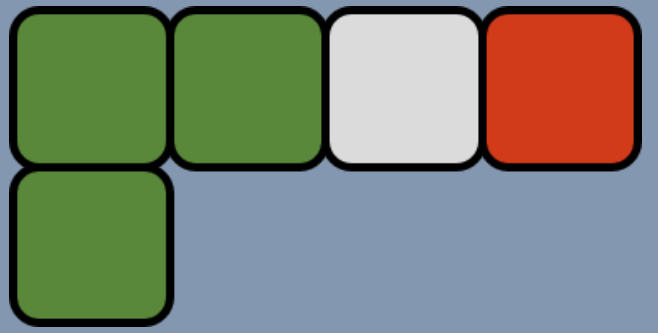
Zhongxuan Liu

Computing

Task 2: Whether a placement is well-formed

Following Criterias of determining valid piece placement

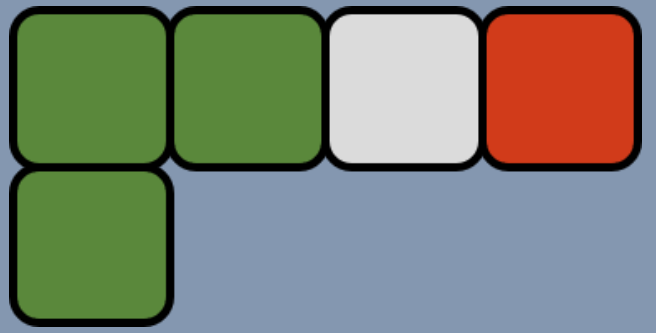
- A piece placement consists of 4 characters
- The first character is in the range of a…j (shape)
- The second character is in the range 0… 8 (column)
- The third character is in the range 0… 4(row)
- The fourth character is in the range 0… 3(orientation)



Task 2: Whether a placement is well-formed

Our solution:

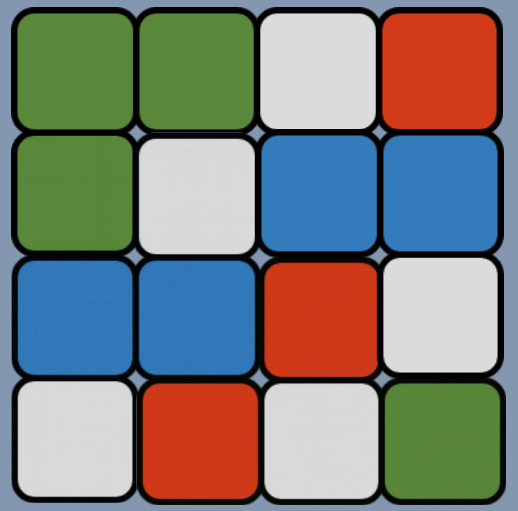
- Create a method which returns false if any criteria is not met.
- isPlacementWellFormed()



Task 3: Whether a placement string is well-formed

A placement string is the list of string consists of all pieces on board

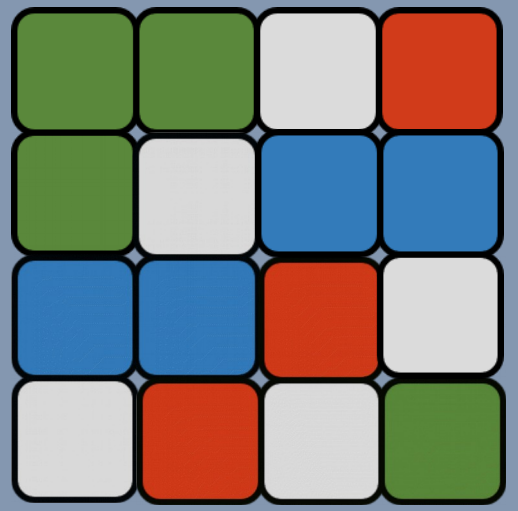
- A placement string is valid when each string on board is valid and the order of the strings is valid.
- Also, no shape appears more than once in the placement.



Task 3: Whether a placement string is well-formed

Our Solution

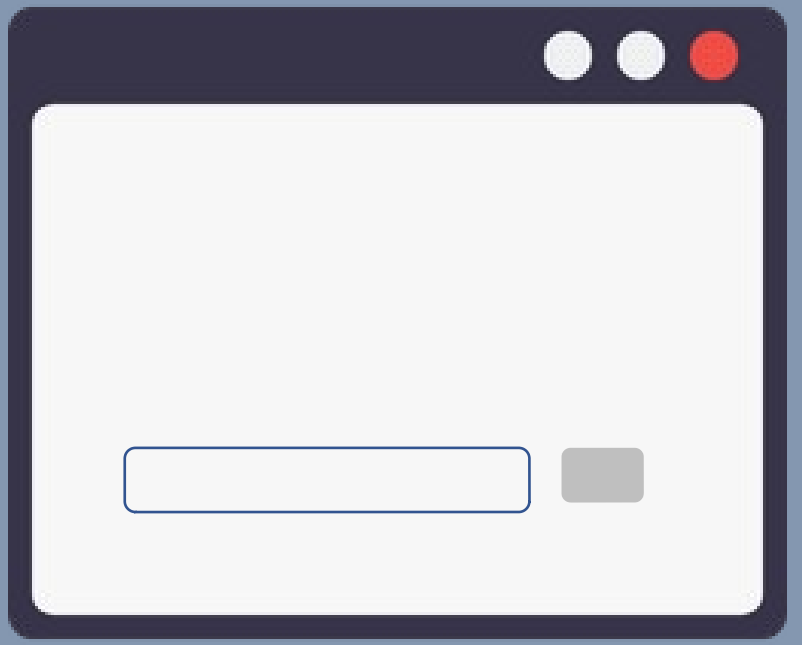
- Part 1: Integrate isPiecePlacementWellFormed() method
- Part 2: HashMap



Task 4: Implement a simple placement viewer

Main Problem: Key in a piece placement string, the piece is shown in javaFX window

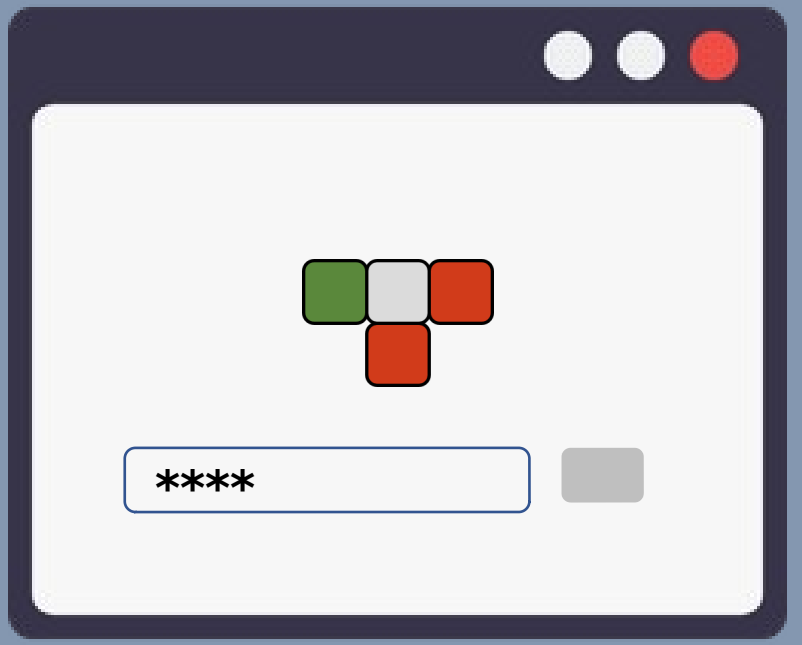
- To get the correct orientation
- At the same time, get the correct size and position



Task 4: Implement a simple placement viewer

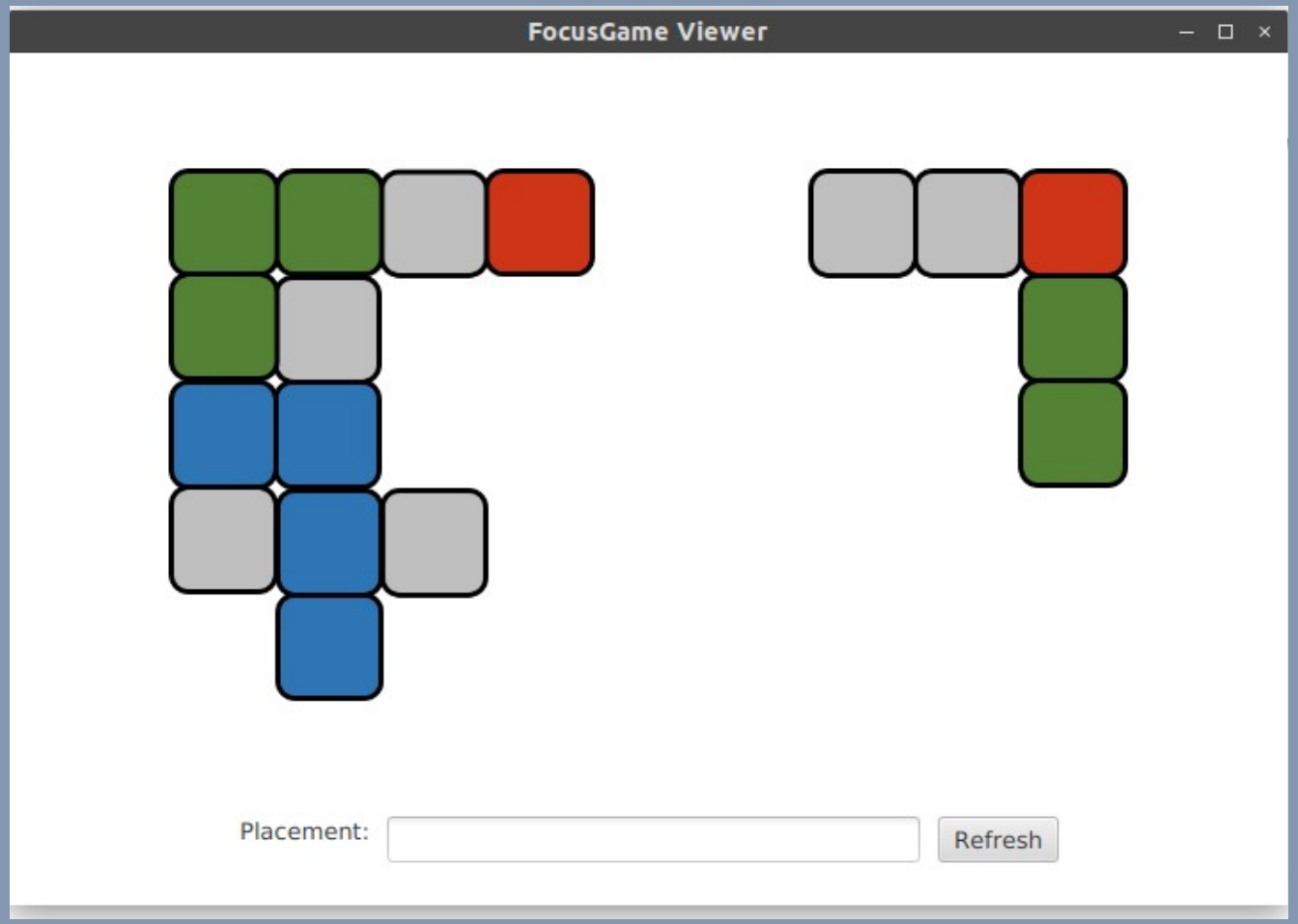
Our Solution:

- Create a method to show the string
- Use a if loop to get the cases of piece placements
- Create a textfield to allow user key in the string of pieces



Task 4: Implement a simple placement viewer

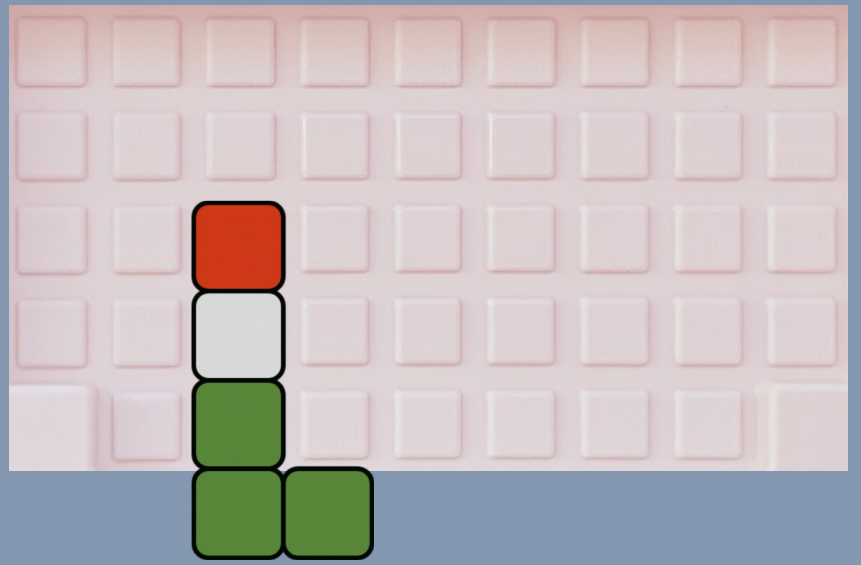
Our Solution:



Task 5: Whether a placement string is valid

Main Problem: Is placement string valid

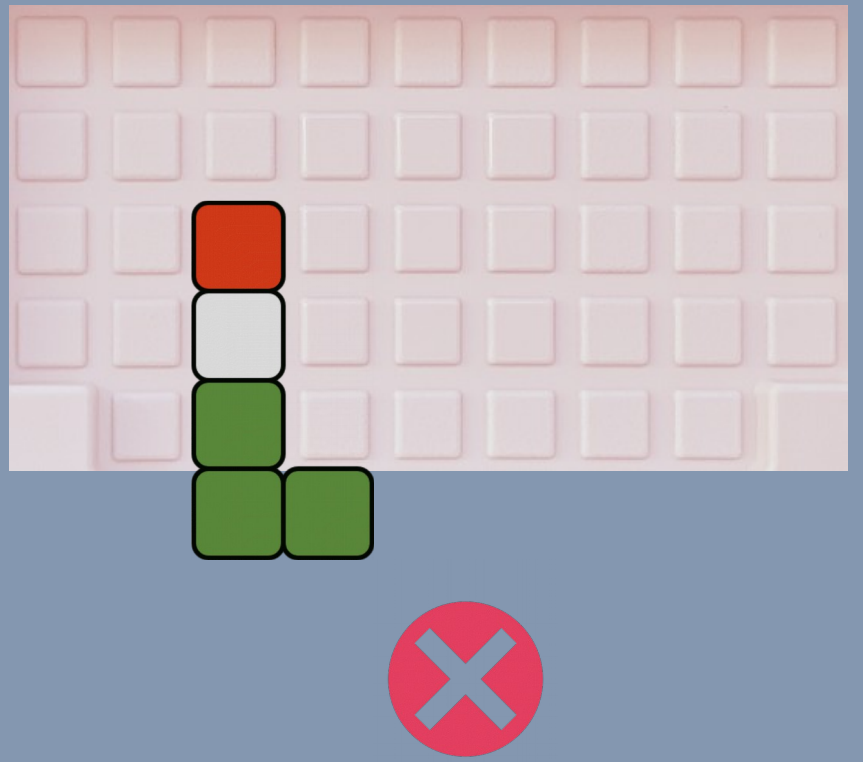
- To be valid, the placement string must be:
- well-formed, and
- each piece placement must be a valid placement according to the rules of the game: 1. pieces must be entirely on the board
  2. pieces must not overlap each other



Task 5: Whether a placement string is valid

Our Solution:

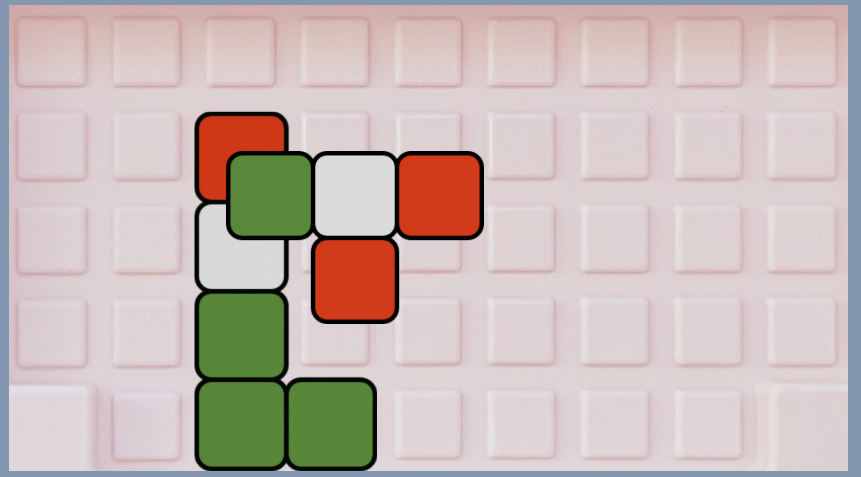
- Firstly, get a empty board. Then put the pieces onto the empty board.
- Use a for loop to check the x and y coordinate of the pieces in the string list, if the piece goes out of the boundary set, then the piece placement string is not valid, the method returns false.



Task 5: Whether a placement string is valid

Our Solution:

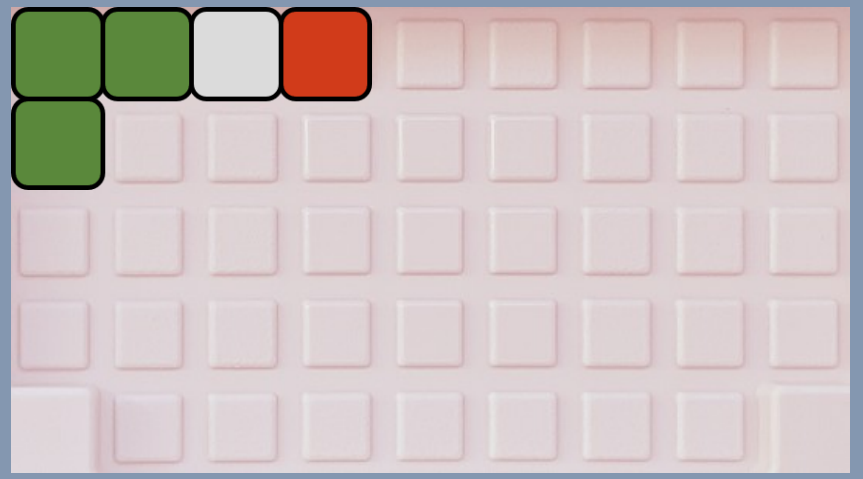
- Secondly, create a new list that contains the position information of the pieces tested, and the check if subsequent pieces have same coordinate range with the list created before. If it overlaps, return false.



Task 6: determine the set of all viable piece placements given existing placements and a challenge

Main Problem: Where are the places that I can put a new piece?

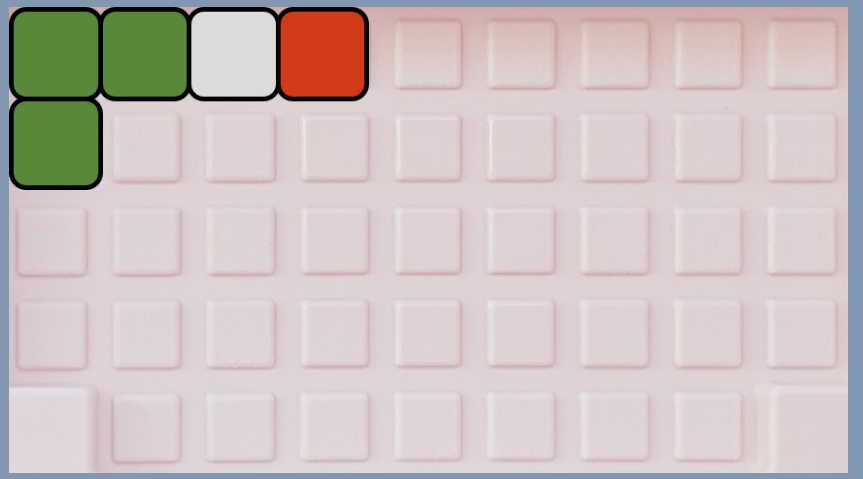
- This problem is vital to solve as the result given in this part will be used to determine if a piece is placed in available space as well as to find the hint. Hence we need this part to solve later tasks.



Task 6: determine the set of all viable piece placements given existing placements and a challenge

Our Solution:

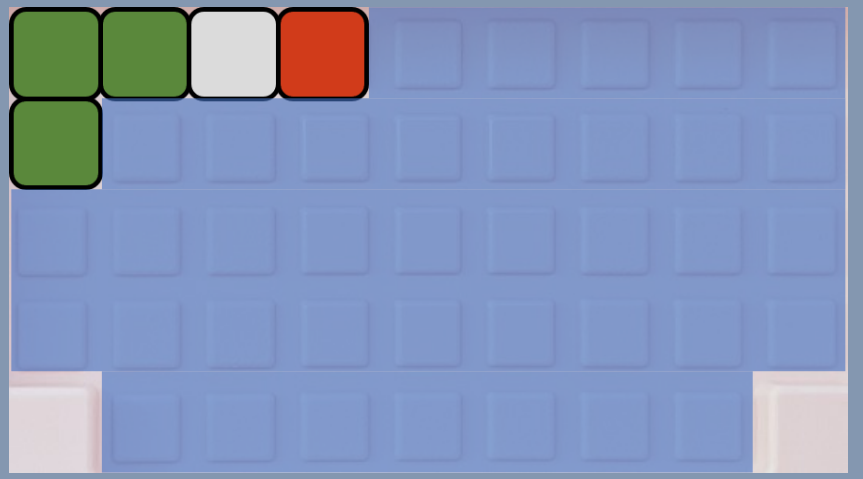
- First, each slot on the board is designated with an integer as its position information.
- The next step is to calculate and store the position of each tile of the pieces existing on the board in a list.
- The position of piece tiles are checked with the position of challenge tiles
  - If piece position overlaps with challenge position, then the color of tile is checked



Task 6: determine the set of all viable piece placements given existing placements and a challenge

Our Solution:

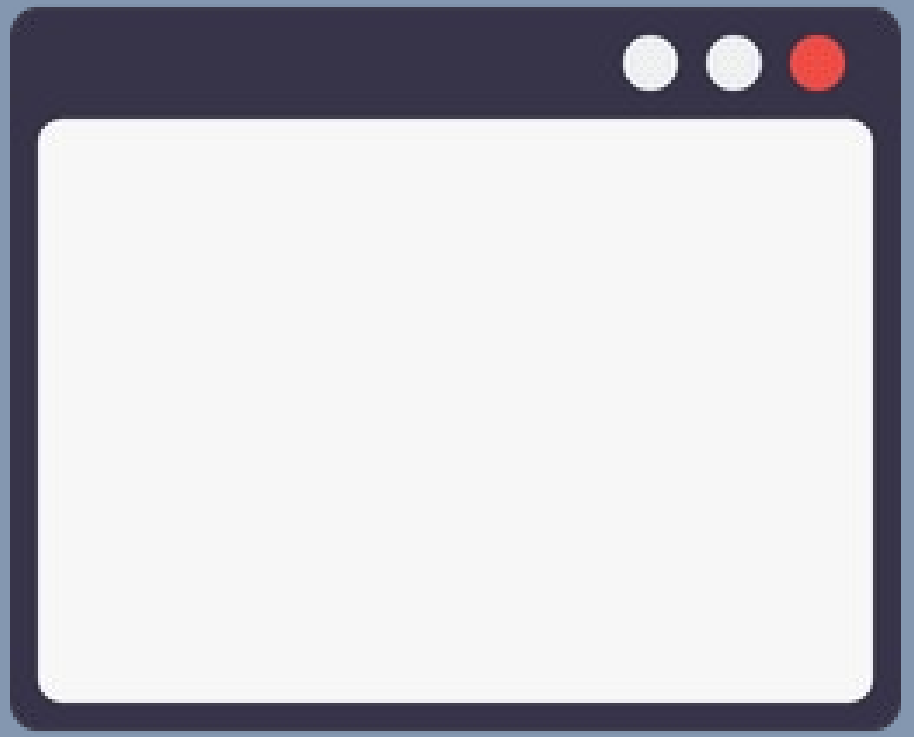
- Then check the pieces that are not on the board, find the valid placements.
- This is done by checking one tile on the piece with one position on the board. If the tile overlaps with challenge, check the color.
- If the first tile placement is valid, calculate the position of the other tiles and check them one by one.
- If the tile color does not match the challenge, change tile
- If the tile goes out of the boundary of board, change orientation of tile and check again
- Traverse with every piece and every orientation.



Task 7: A basic playable Focus Game

Main Problem: Implement a basic playable Focus Game in JavaFX that only allows pieces to be placed in valid places

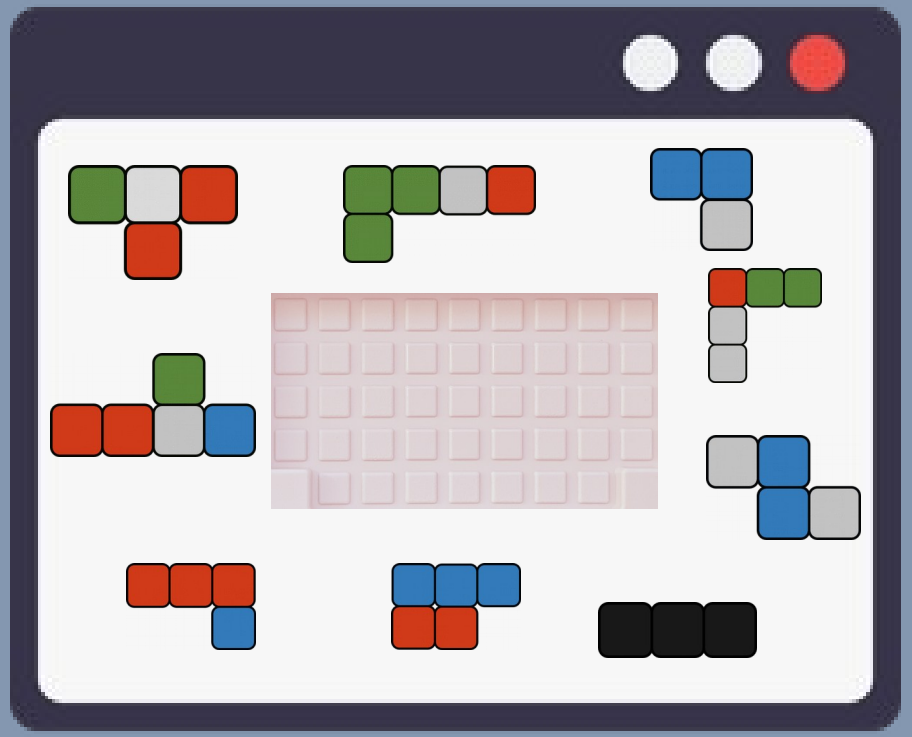
- This can be achieved by using the results of task 6.
- Usage of javaFX is required.



Task 7: A basic playable Focus Game

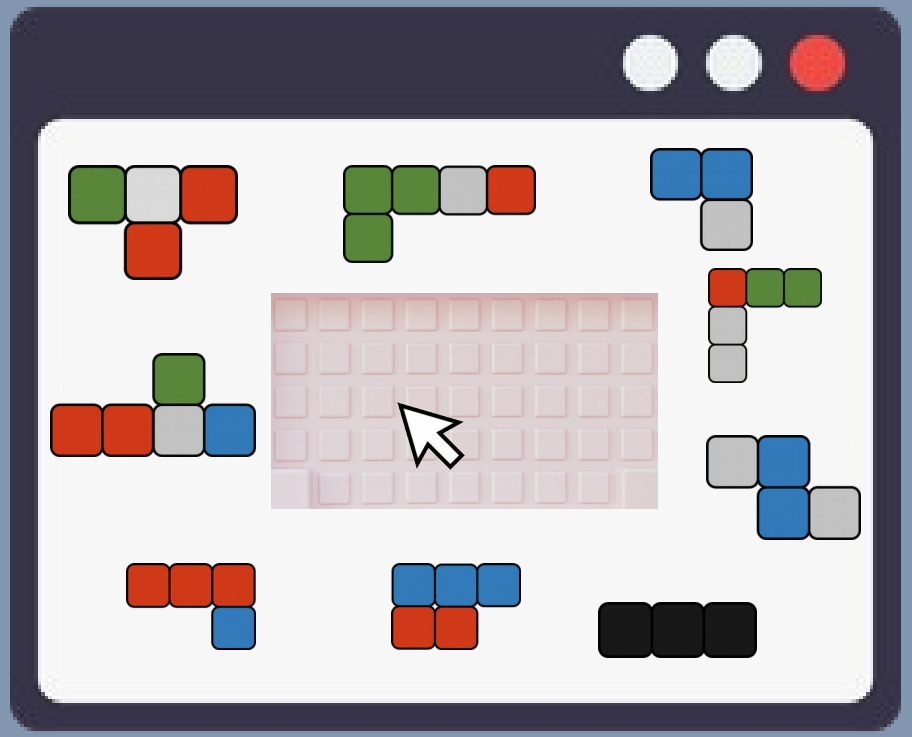
Our Solution:

- There are 2 aspects to solve the problem
- One is to make a visible window that displays a board and pieces.
- Board is placed into the scene using makeBoard method, and the dimensions of board is declared.
- The image of board is cropped into a fitter shape.
- The pieces are placed by creating a class Pieceview, which puts the pieces into their default positions(original position).



Our Solution:

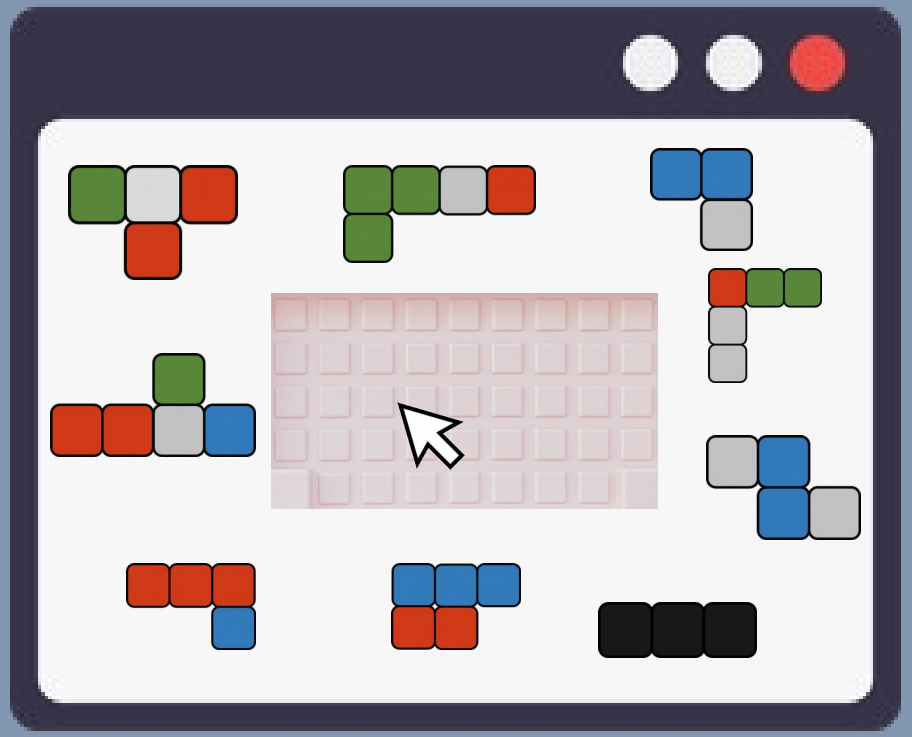
- The other aspect is to create ways of interacting with the pieces. This is achieved by using the events methods in javaFX library.
- The methods used to move the pieces are created in the PieceView class.



Task 7: A basic playable Focus Game

PieceView:

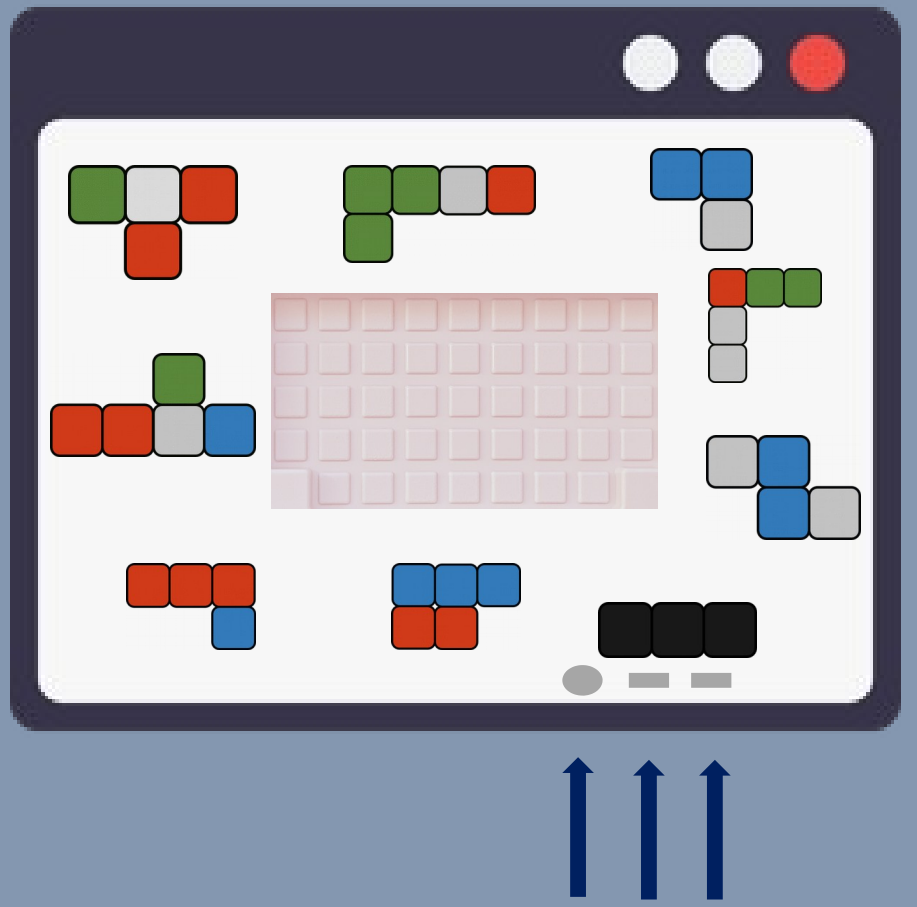
- PieceView class created by ZhongXuan is contains main methods that give the pieces interactable features.
- The makePiece method is used to set the correct size of the pieces, as well as to put them in the correct intial positions.
- The event methods are used to drag the pieces.
- TestSnap is an interesting method that can put the piece into nearby empty spaces when released.



Task 7: A basic playable Focus Game

Board.class

- makeBoard method is created to show the board in the window.
- makeControls method creates the clickable buttons on the user interface.
- There are 3 buttons placed at right bottom corner of the window:
  - restart : To set the game to its initial stage
  - Hint : To get the next viable placement of tile
  - ChangeChallenge ： change the challenge shown in the center of board.



Task 8: Implement Challenges

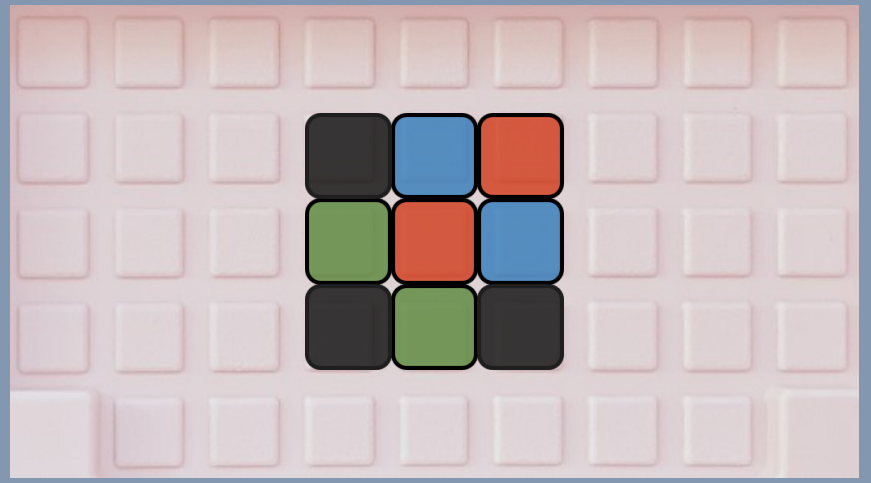
Main Problem: show the challenge on the right section of board

- Two issues to solve: get a section on the board and put challenge in to the section.

Task 8: Implement Challenges

Our Solution:

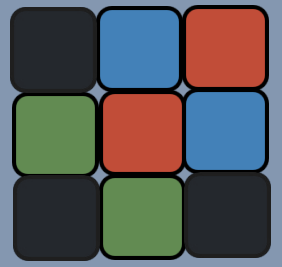
- Get the coordinate of squares in the central 9X9 lattice block, make the coordinates into a list called positions.
- Make each lattice in the block as a imageView object.
- Then use a for loop to put pictures of each square into the lattices.
- Since we defined the size of a square in the start of Board.class, the squares will fit in the board perfectly.



Task 9: Find the solution of game

Main Problem: To get the solution of game given a challenge.

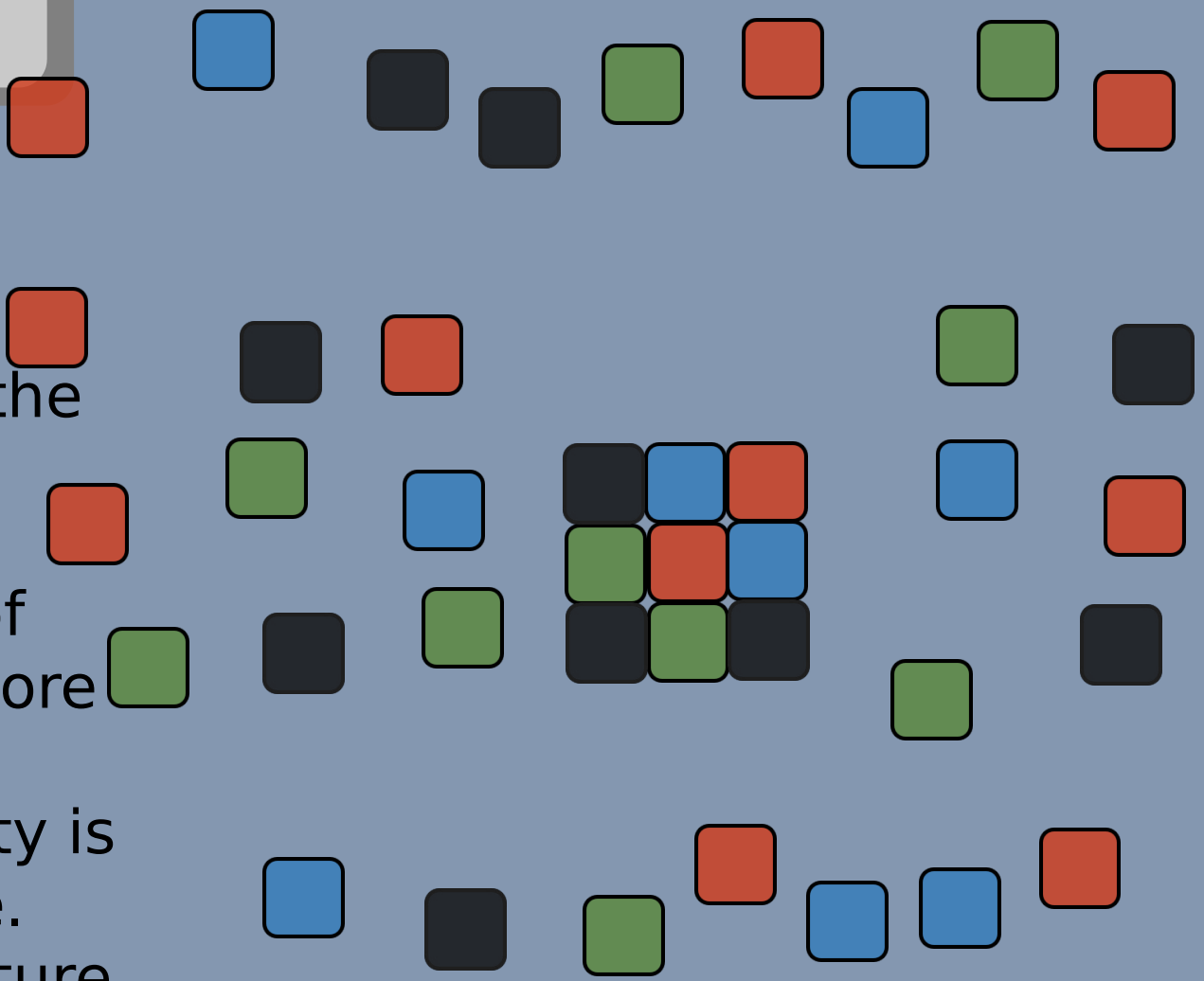
- In this part, the solution of game is determined by computer when a challenge is given.
- This is crucial as this tells the game when to terminate.



Task 9: Find the solution of game

Our Solution:

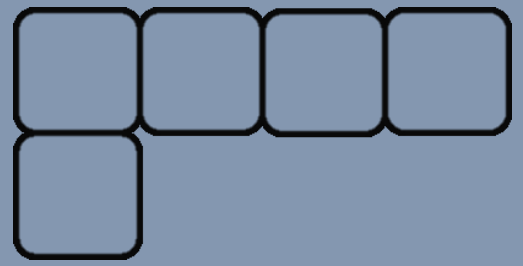
- From piece A, place the pieces on the board one by one with all the orientations.
- At first, there are over thousands of possibilities of placements. With more pieces put in, the possibilities are reduced and the eventual possibility is one complete solution to the game.
- As code executes, we used a structure that merged possible placements of every piece.



Task 10: Implement Hint

Main Problem: Hint is the next valid placement of a tile.

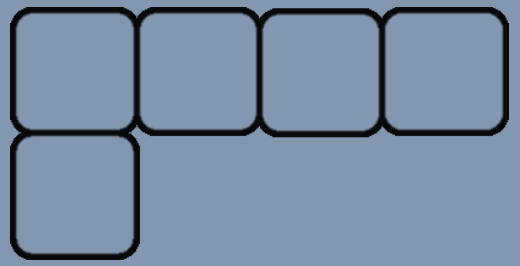
- Hint is shown when the mouse is placed over the ‘Hint’ button.



Task 10: Implement Hint

Our Solution:

- In this case, we used a rather simple solution.
- First we get the empty slots on the board using the current status defined in previous code.
- Then a random piece is picked using random method predefined in java.
- After that, use a loop to set the correct orientation of the piece, and check if the piece fits in the empty space.
- If not fit, change another piece. Try in the loop again.

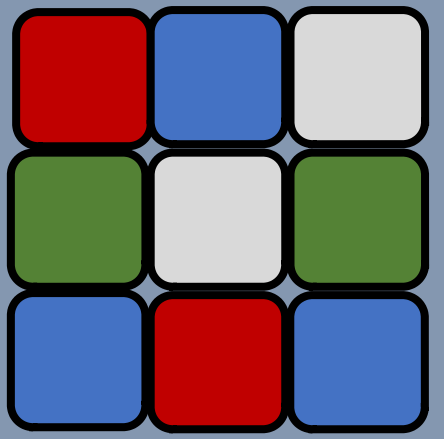


Task 11: Generating Challenge

Main Problem: Get interesting new challenges

Our Solution:

- Use the results of task 9(generating solution of game)
- Use getSolution method to search the solution of the challenge with 9 red tiles.
- Then change the color of one tile in the challenge, and continue to search for the solution of challenge.
- If not found, change another color and search.

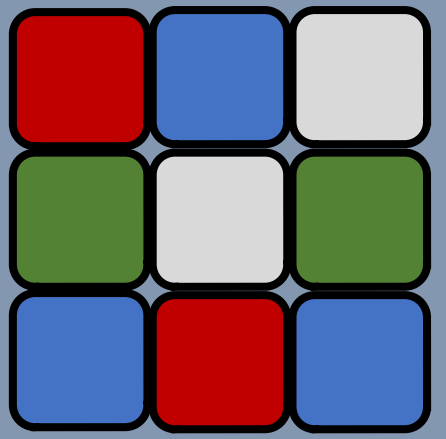


Task 11: Generating Challenge

Main Problem: Get interesting new challenges

Our Solution:

- If the challenge solution is still not found, there will be 2 tiles color changed and continue searches the solution
- The loop goes on until one solution is found and that challenge is the new challenge!

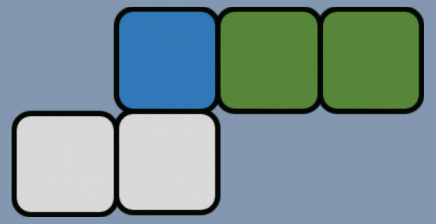


User Interface Improvements

It is crucial to have a more user friendly interface for a game, as this would make user spend more time with the game.

Our Solution:

- 1. Redraw the pieces
- 2. Cut the picture of the board and make it more fit in the window
- 3. Put the pieces at better places where they would not overlap with board
- 4. Make the challenge more transparent so the player will not mix the challenge and the pieces already put on board

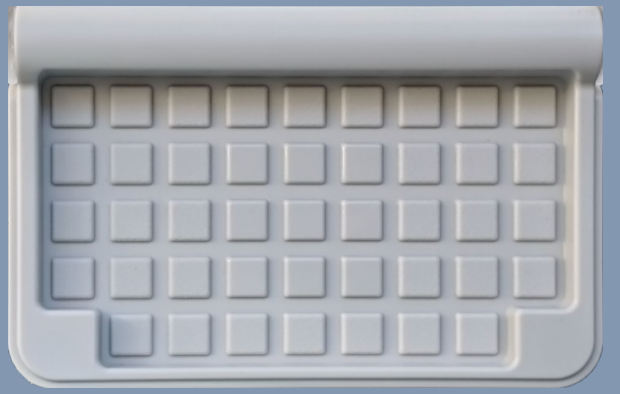


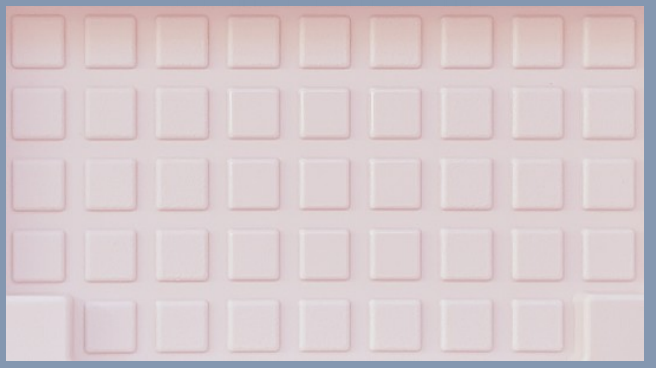
User Interface Improvements

It is crucial to have a more user friendly interface for a game, as this would make user spend more time with the game.

Our Solution:

- 1. Redraw the pieces
- 2. Cut the picture of the board and make it more fit in the window
- 3. Put the pieces at better places where they would not overlap with board
- 4. Make the challenge more transparent so the player will not mix the challenge and the pieces already put on board





User Interface Improvements

It is crucial to have a more user friendly interface for a game, as this would make user spend more time with the game.

Our Solution:

- 1. Redraw the pieces
- 2. Cut the picture of the board and make it more fit in the window
- 3. Put the pieces at better places where they would not overlap with board
- 4. Make the challenge more transparent so the player will not mix the challenge and the pieces already put on board

Contributions

Li Anbang

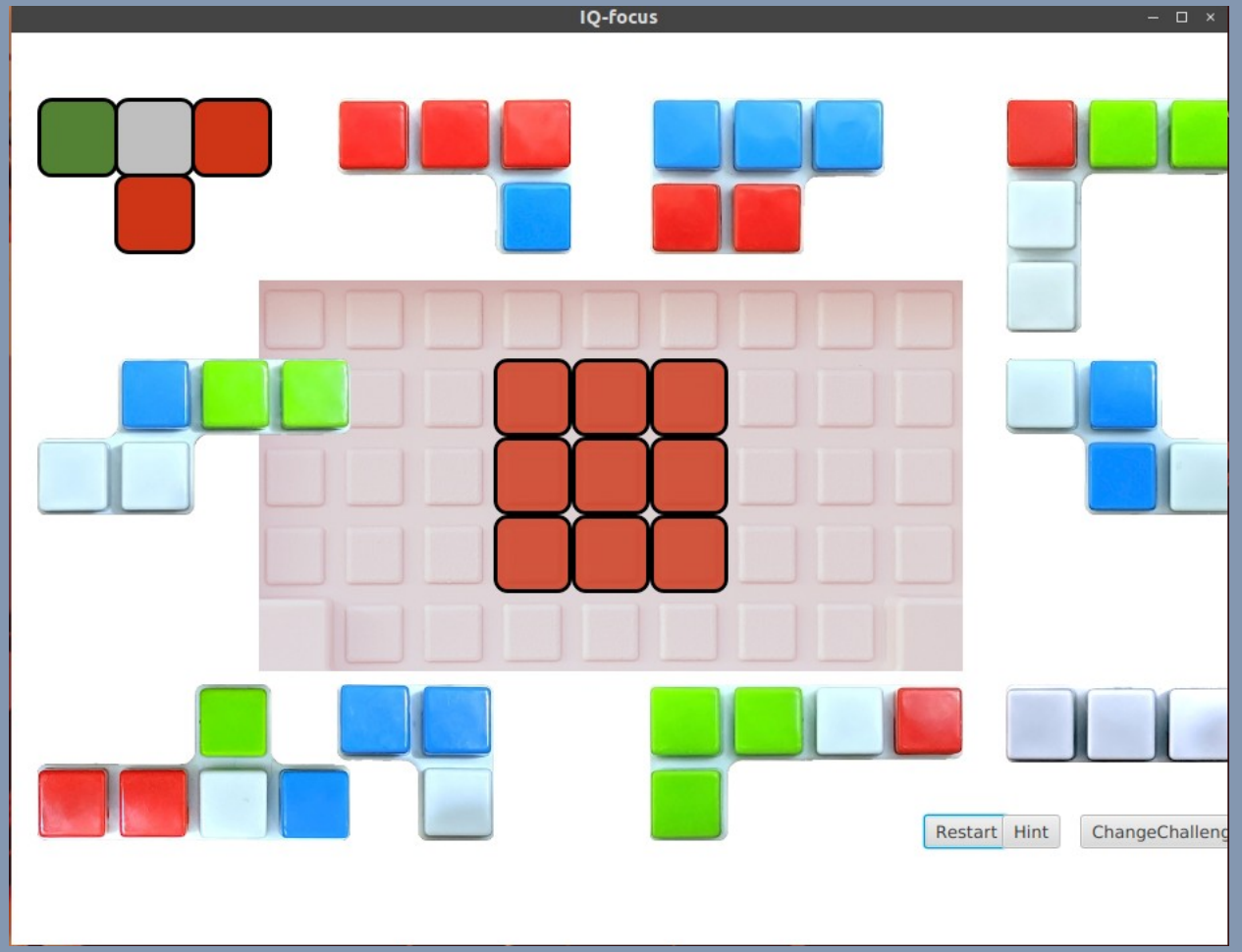
- Early Stage Tasks, Task 2,3,4
- User interface of game
- Presentation

Zhongxuan Liu

- Main frame of the game
- Control of the game
- Hint generating
- Challenges

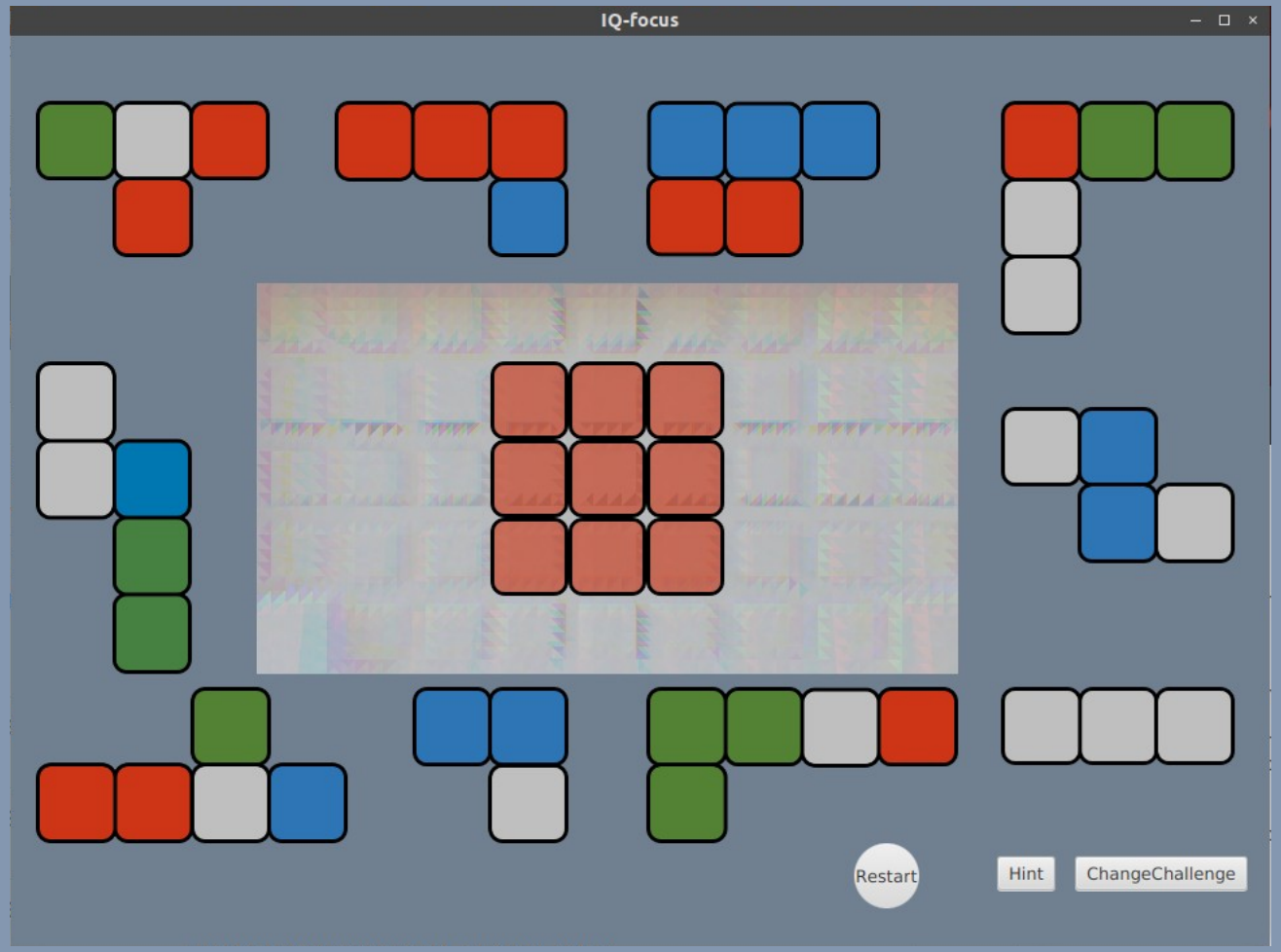
Task 7: A basic playable Focus Game

The initial version:



Screen Shots of Final Version

Start of the game



Screen Shots of Final Version

Once Finished