Test Report

Overview

- We employed 2 methods of testing in order to fully test our project
  - Backend testing: JUnit: This included testing all methods in the classes that are classified as the model
  - Front end testing: manual user interaction, (We manually tested each and every functionality of of the game through the User interface)

Backend Testing

CoordinatePoint class

Total number of tests = 3

1. testGetRow() - tests to see if the row is returned correctly
2. testGetCol() - tests to see if the column is returned correctly
3. testEquals() - tests to see if the equals method works correctly

GameBoard Class

Total number of tests = 17

1. testFillBoard(): tests fillBoard method. we do this test to see if the board is filled with the correct colors
2. testMakeMove(): tests makeMove method. we do this test to see if the move is made correctly
3. testIsWon(): tests isWon method. we do this test to see if the game is won correctly
4. testIsNotWon(): tests isWon method. we do this test to see if the game is won correctly
5. testNoMovesLeft(): tests noMovesLeft method, getMaxMoves method, getNumMovesLeft method. we do this test to see if the moves mechanism is working correctly
6. testGetNumMovesLeft(): tests getNumMovesLeft method, resetMoves method. we do this test to see if the moves mechanism is working correctly
7. testAddObserver(): tests addObserver method. we do this test to see if the observer is added.
8. testRemoveObserver(): tests removeObserver method. we do this test to see if the observer is removed.
9. testFloodFill(): tests floodFill method, getCellColor method. we do this test to see if the floodFill method is working correctly
10. testGetSelectedColor(): tests getSelectedColor method. we do this test to see if the selected color is correct
11. testGetHint(): tests getHint method. we do this test to see if the hint is correct

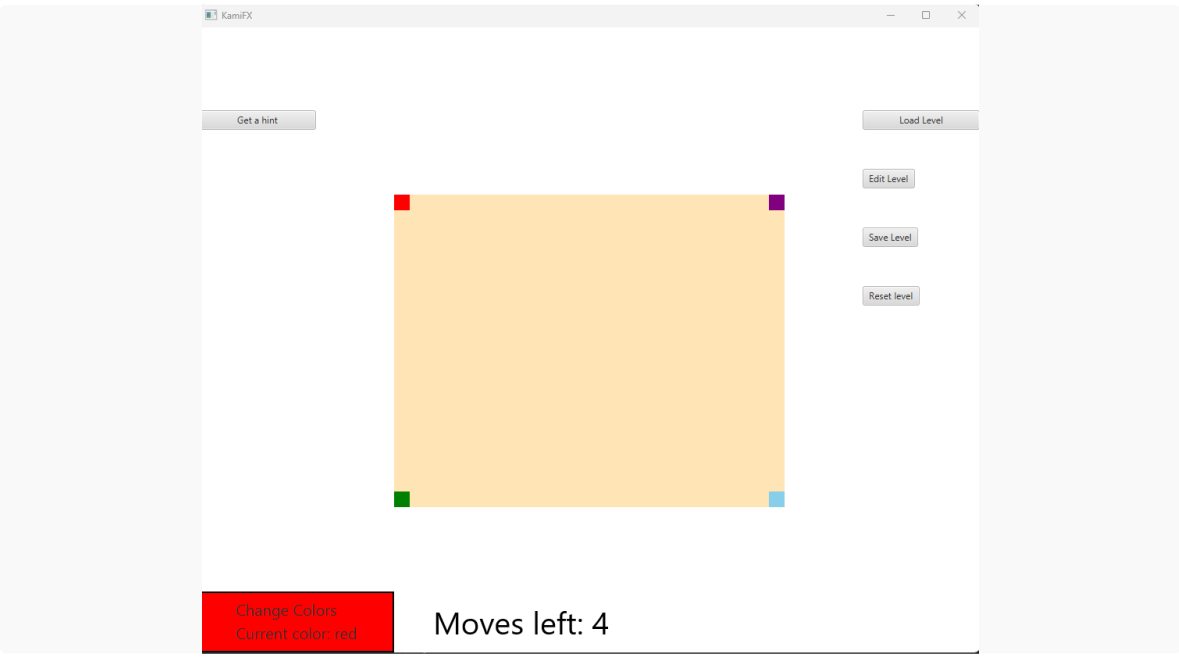
12. testGetLevelName(): tests getLevelName method. we do this test to see if the level name is correct
13. testGetRowsColumns(): tests getRows method, getColumns method. we do this test to see if the rows and columns are correct
14. testGetPalette(): tests getPalette method. we do this test to see if the palette is correct
15. testGetCellColor(): tests getCellColor method. we do this test to see if the cell color is correct
    16: testResetBoard(): tests resetBoard method. we do this test to see if the board is reset correctly
17. testGetAllPiecesInBoard(): tests getAllPiecesInBoard method. we do this test to see if the pieces are correct

```
C:\ManavData\college\Courses\CSC260\csc260-project2-groupd>gradle build

> Configure project :
Project : => no module-info.java found

BUILD SUCCESSFUL in 605ms
8 actionable tasks: 8 up-to-date
C:\ManavData\college\Courses\CSC260\csc260-project2-groupd>
```

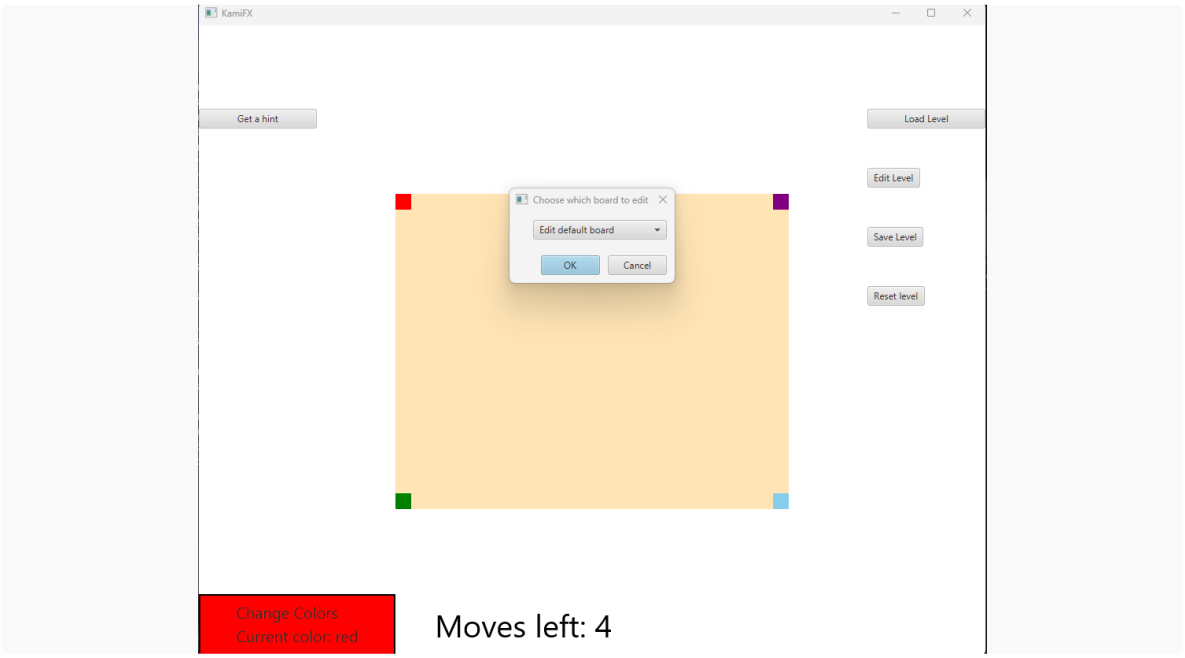
Front end testing:

1: Check if edit and saving a level works:

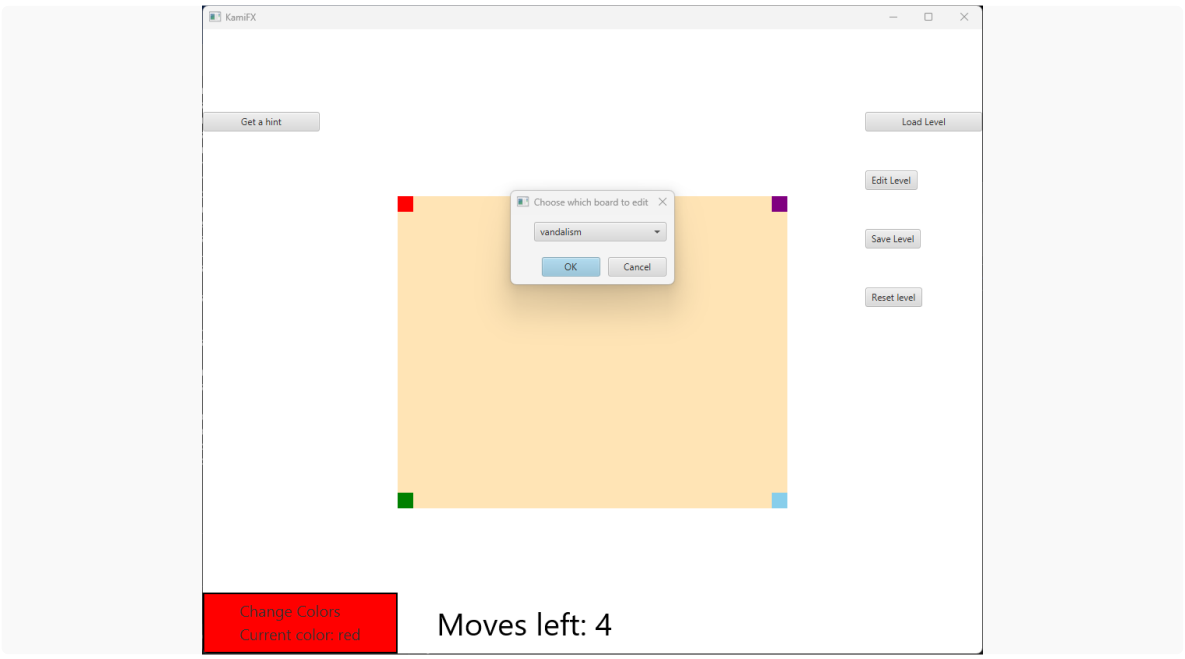
- open a random board



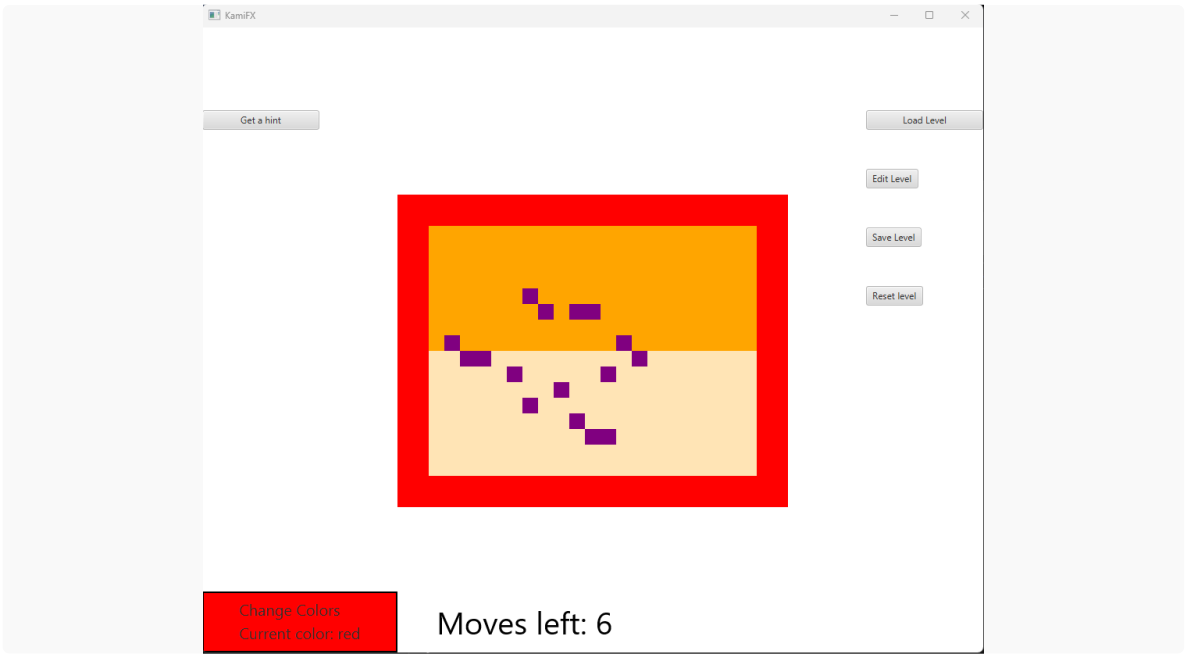
- select edit level



- select the board you want to edit from the dropdown list:



- click ok

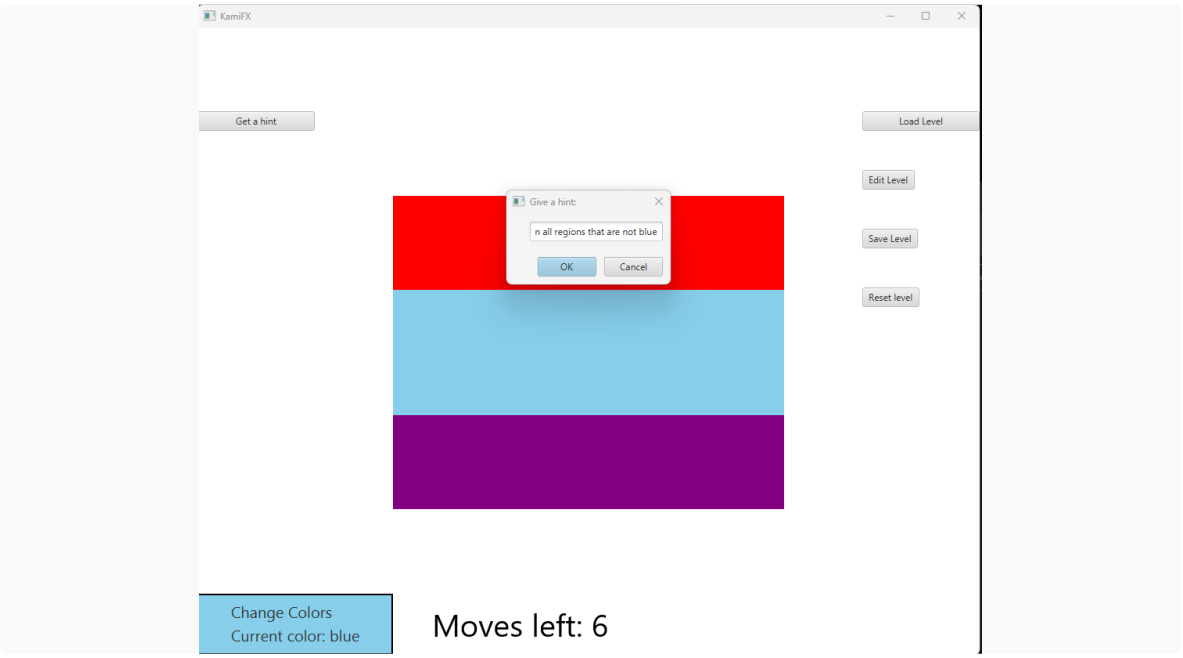


5b: test if all colors are available to choose from on the pallete

- in order to edit the board, select the color from the palette you like and start filling all the boxes

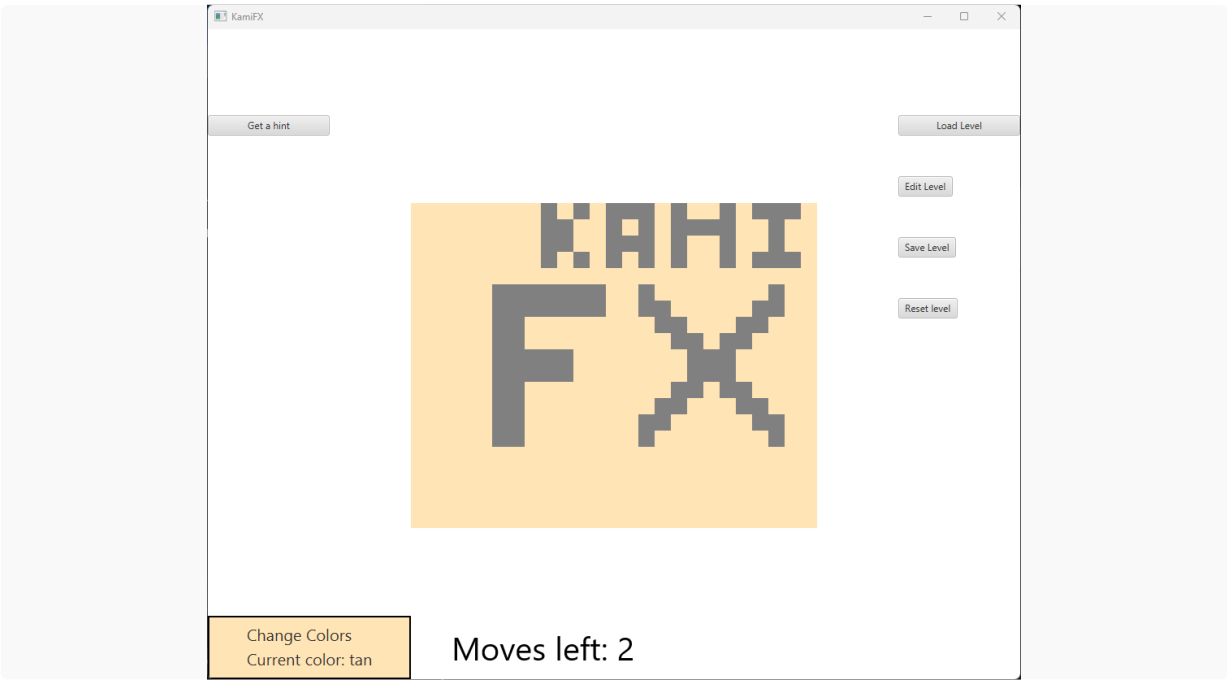
- once the board is complete, click on the save level button. the game will ask you to name the level. i am naming it "level2Test"
- click on ok

- It will then ask you to give a hint in order to solve the level. For simplicity, i will have the hint as "choose the blue color and click on all regions that are not blue"
- click on ok

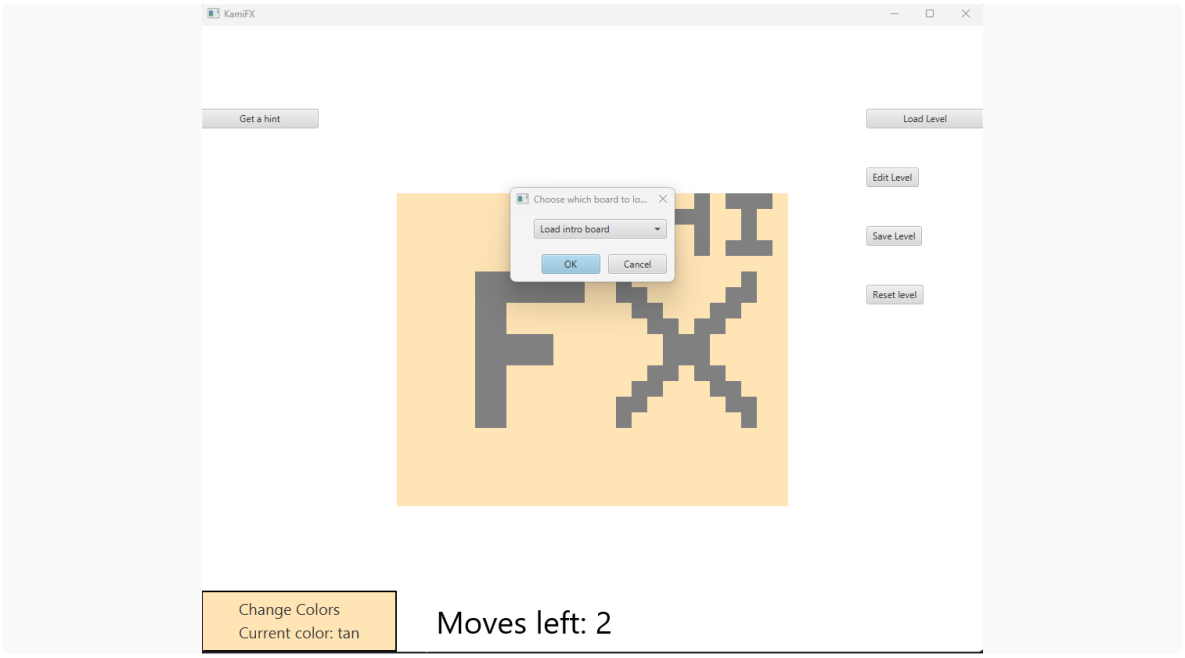


- it will now ask you to give the number of moves allowed to complete this board. For simplicity, I will have the number of moves to be 2.

1. Loading an already existing board (we will test the board we just created above to see if all the functions work):
   - starting point



- click on load level



- Select appropriate level from dropdown
- click okay to see the new board loaded and the color pellet changed

2: Check the color palette for a given board:

- click on the color palette to check if all relevant colors are there:

3 Check the get hint by clicking on the hint: it should display some text in order to solve the level

4: change the board and then reset it to go back to the original board:

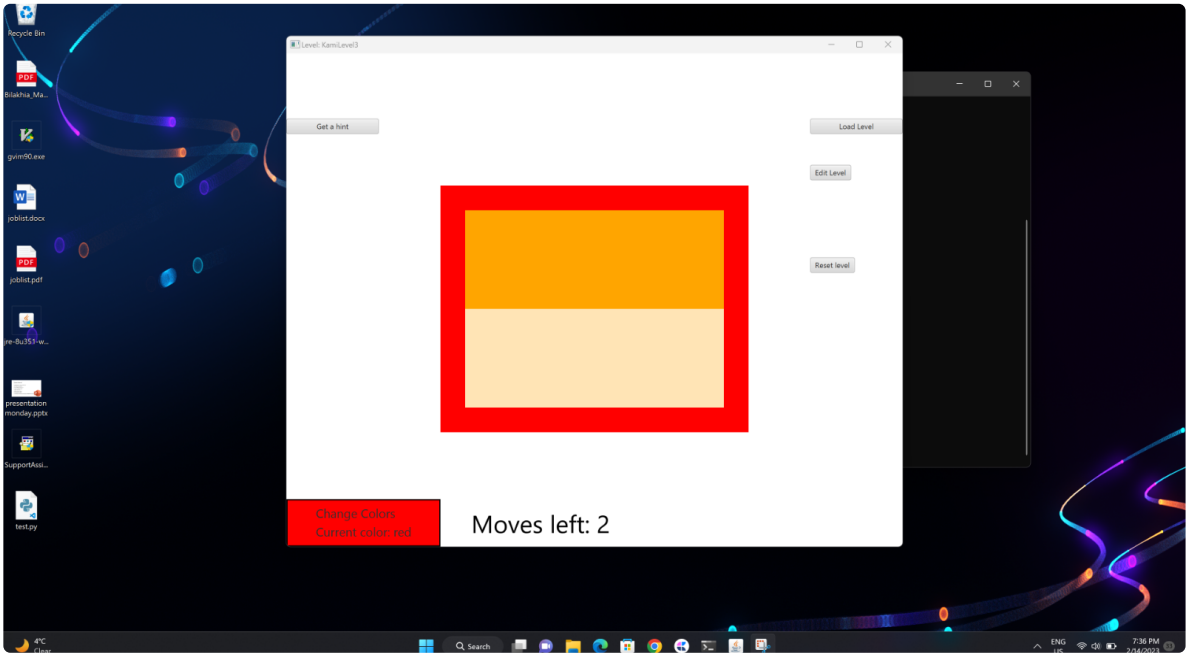
- board changed by selecting the blue area using the color purple:

- board is reset using the reset level button

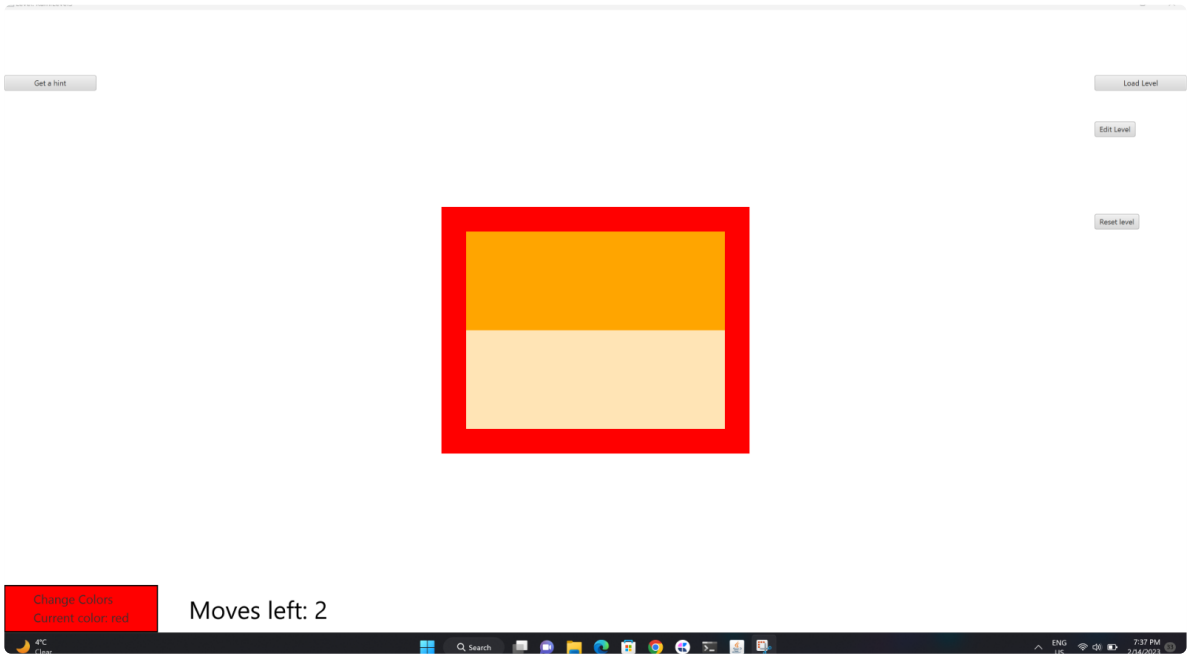
New Improvements after issues were fixed and merged:

1) Full screen mode

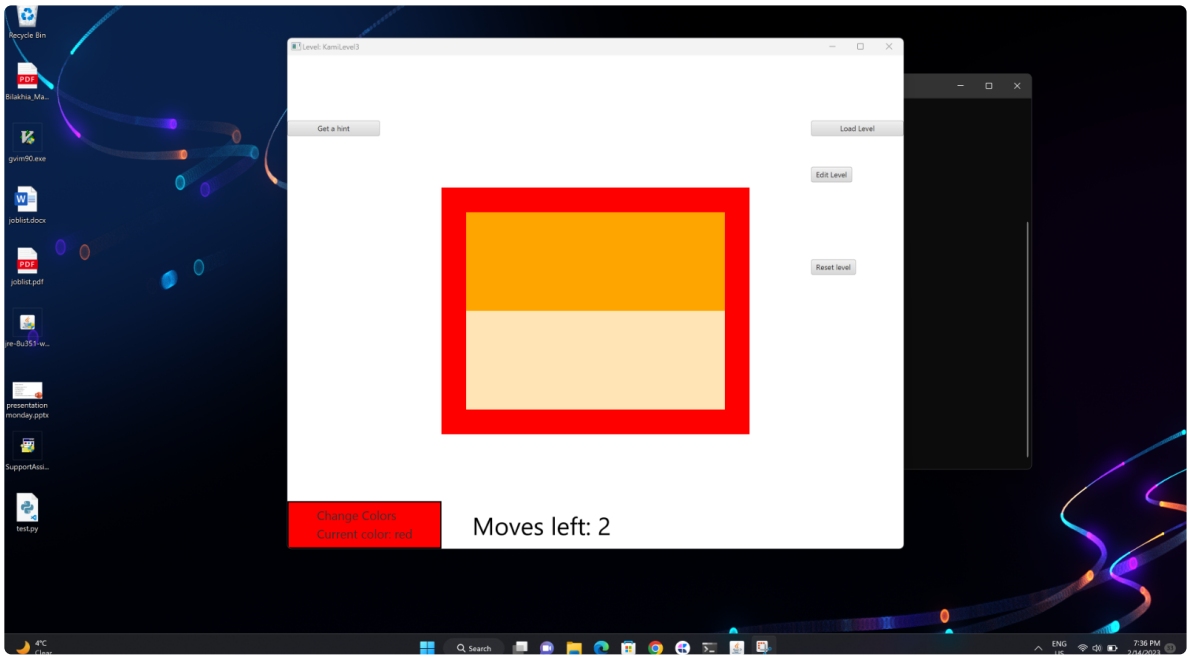
Window at original size:



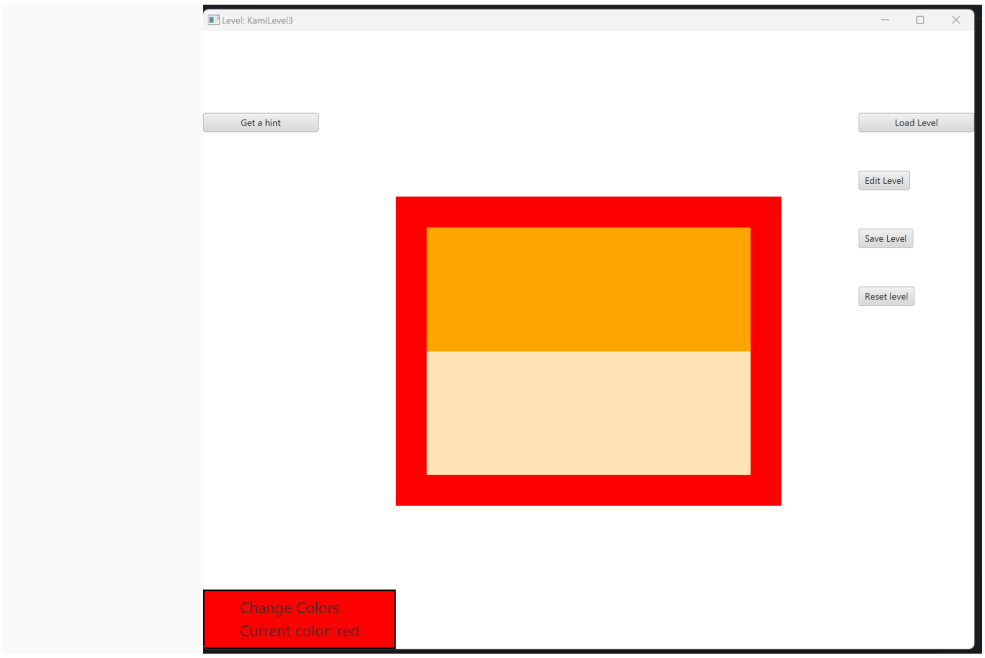
Window in full screen:



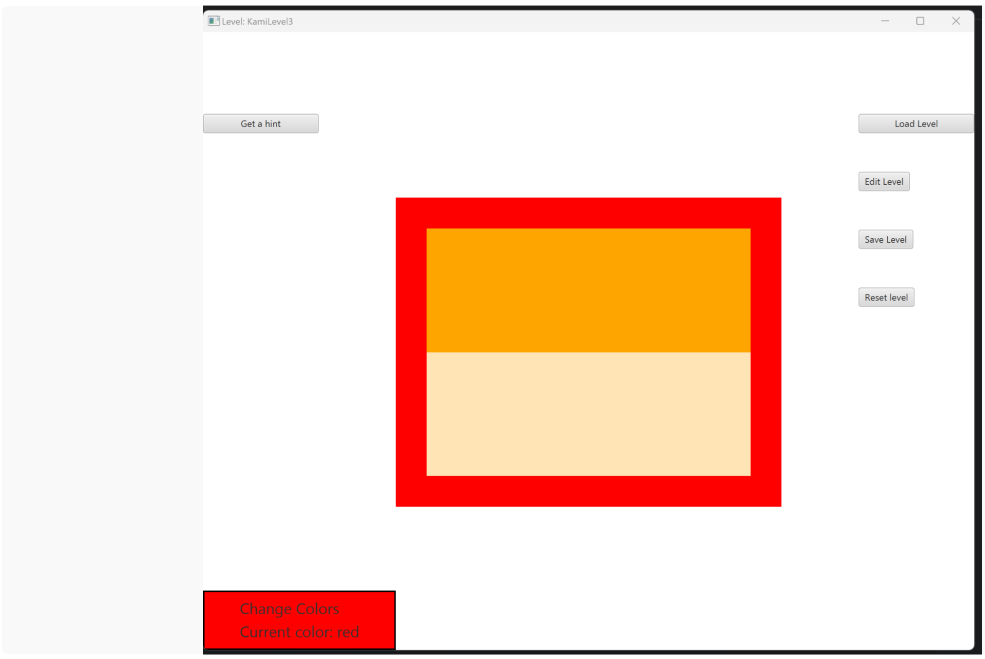
2) GUI elements are now correctly hidden and revealed when a user is playing the board versus when they're editing a board



- now in edit mode:

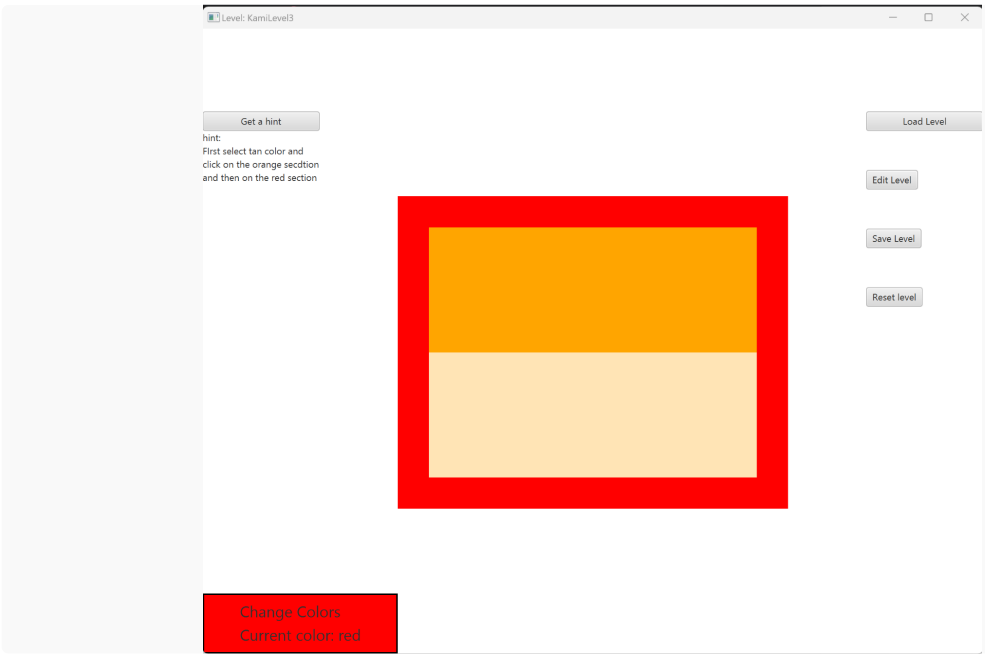


3) Title of the window is now set to the current level name.

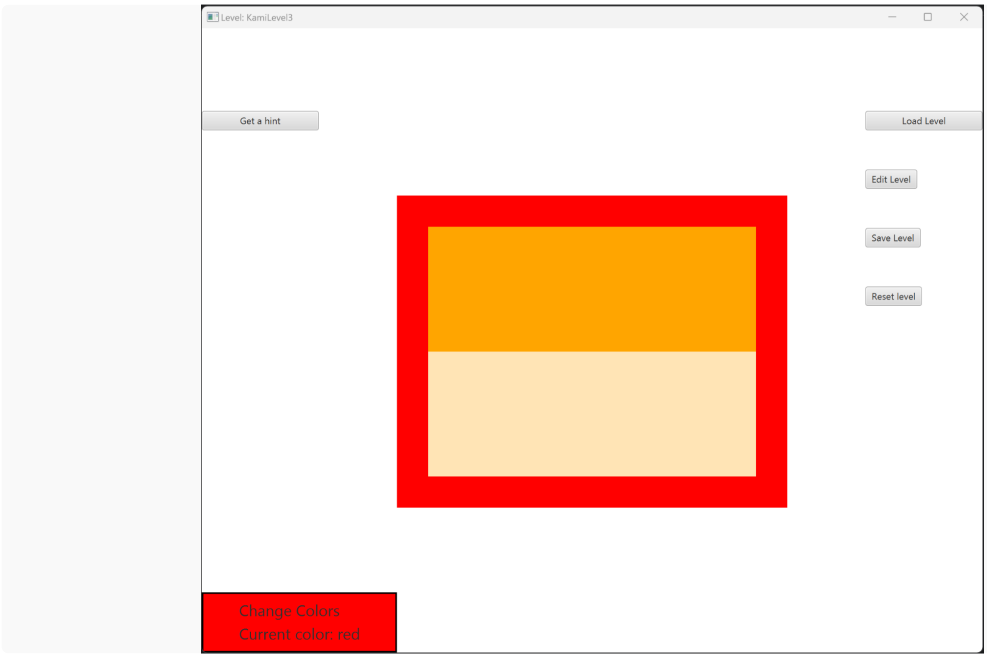


4) Hints should also reset when you reset the board:

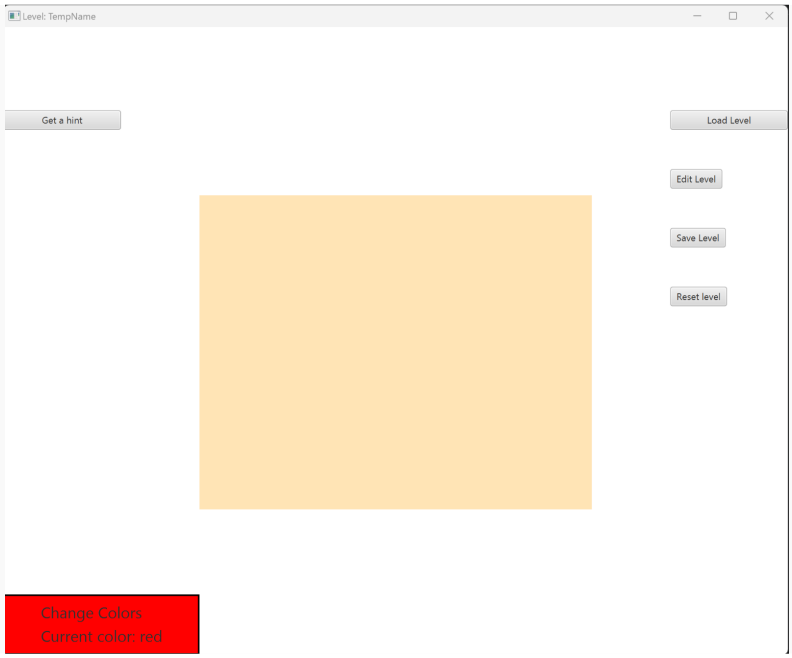
- After clicking the hint button before hitting reset



- After hitting reset



5) resetting the default should not make the board disappear, the default board should stay as it is..


Level: TempName
Get a hint
Load Level
Edit Level
Save Level
Reset level
Change Colors
Current color: red