

# SWE4203

Deliverable 2

March 1, 2024

Ben Irvine, Shahriar Kariman, Kisenge Mbaga

# Testing Plan

## Adaptive change:

Organize project repository. -Simplifying folder structure.

Given the project repository, when developers navigate through the project, then there should not be redundant folders that are either empty or incapsulate only one folder.

## Corrective-change:

Cannot place opponent in middle square in certain situations.

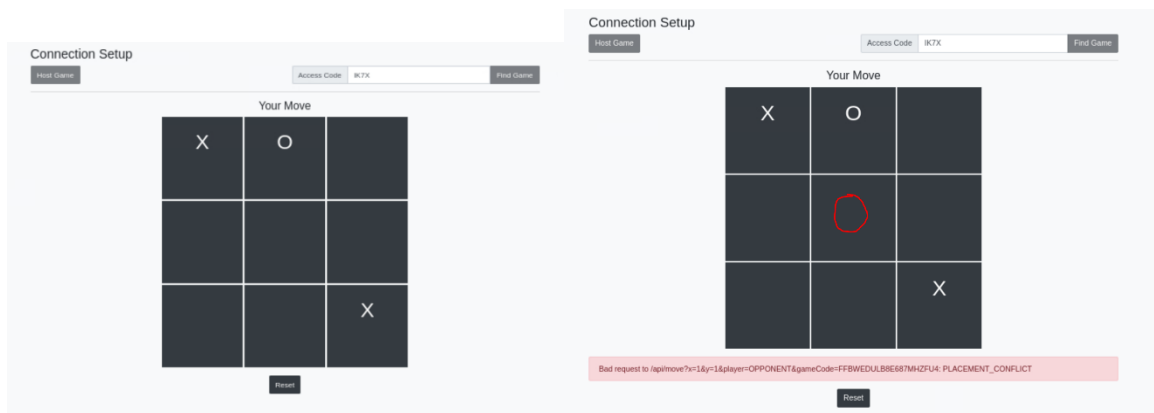


Figure: Scenario where error raised when opponent is placed in the centre

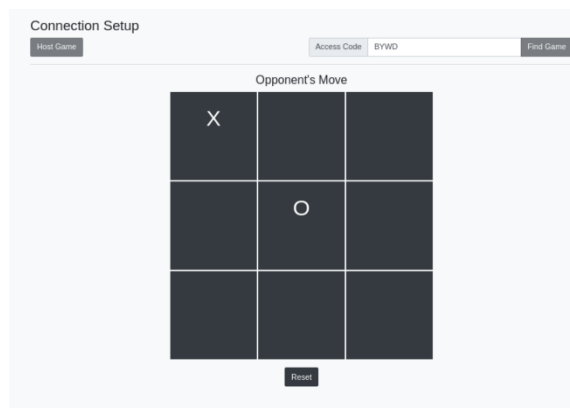


Figure: Scenario where error is not raised when opponent is placed in the centre

Given the game is setup and it is the opponents turn, when the opponent attempts to place their 'o' character in the empty centre square of the game board, then the game should place their 'o' without a placement conflict, as it is a valid move.

**Perfective-change:**

Given two players finish a match, when the players go to press the reset button, then there should be no error messages displayed.

Change the reset functionality such that the existing error messages are removed.

## Candidate Impact Set

Adaptive change: Organize project repository. -Simplifying folder structure.

- Root directory

**Corrective-change:**

Cannot place opponent in middle square in certain situations.

- The entities predicted to be directly impacted by the corrective change is *Game.java*. The source code modified is all located in *Game.java*. *GameManager.java* which calls *Game.java* and other higher classes are not expected to be impacted, as there are no changes in function or class parameters.

Perfective-change: Fix the reset functionality such that the existing error messages are removed after reset.

- The component that is most likely the cause of this bug is what stores the error in the app which would be the *GameManager*.

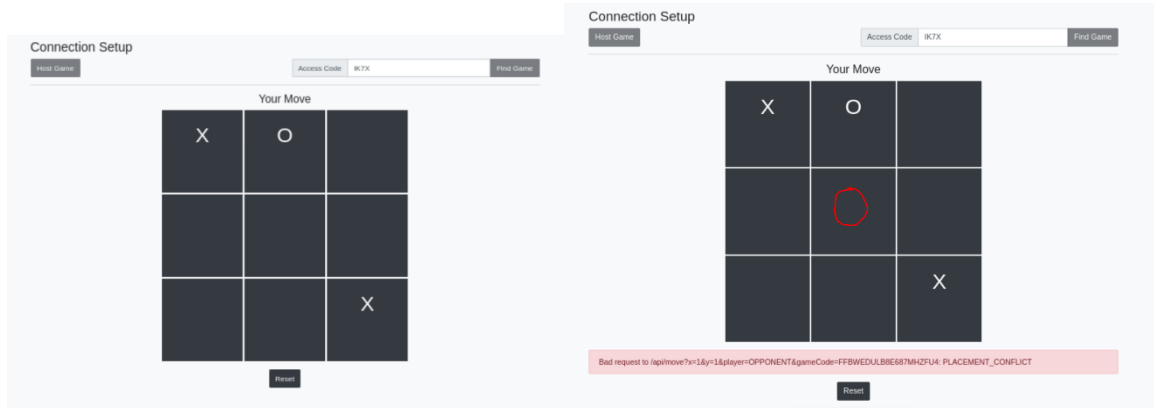
## Actual Impact Set

Adaptive change: Organize project repository. -Simplifying folder structure.

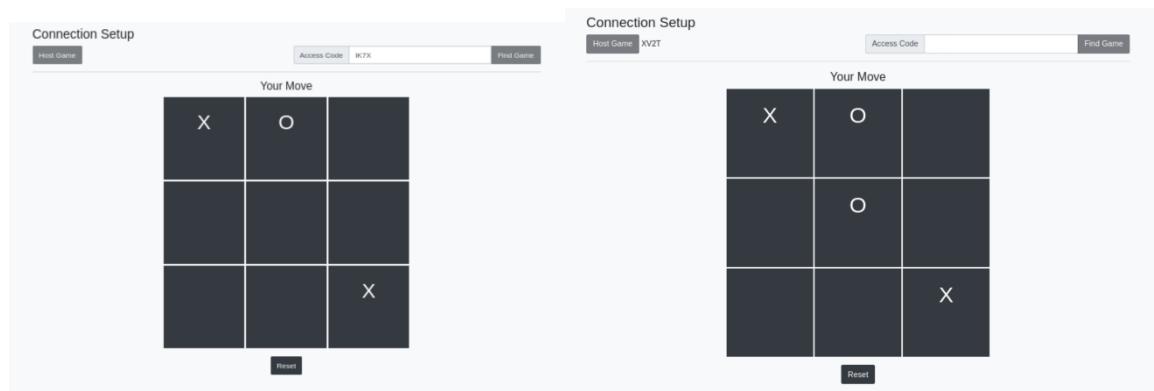
- Root directory

**Corrective-change:** Cannot place opponent in middle sometimes

- The actual entity predicted to be directly impacted by the corrective change is *Game.java*. The source code modified is all located in *Game.java*. No other effects were not noticed for the gameplay, and no other entities were missed.



*Figure: Scenario where error raised when opponent is placed in the centre*



*Figure: Same scenario but no error or effects after the fix*

**Perfective-change:** Fix the reset functionality such that the existing error messages are removed after rest.

- We were right in the sense that the component storing the error message is responsible however that component was not the GameManager it was the resetGame function in index.js.

# Reflection of Process

In the frontend of the application the X and the O's are displayed on a div with an absolute position that limits the styling option for the X and O's for example centering the content would be exceedingly difficult that way hence why we did not make that change for this milestone even though we tried at first.

We could re-do the frontend entirely to have more styling options this includes the text fields, the error messages and the boxes with the X and O's.

- Pros: The app would look more appealing without significant changes to the rest of the system.
- Cons: would take lots of time.

Also, the game ends before the message for the losing side is updated so at the end while the winners display message is "You won!" the other player still has "Your move". This change would affect both the frontend and the GameManager and Game which would cause a larger ripple effect than anticipated.

And separating the classes by hierarchical functionality and limiting the number of parameters passed between them helped to limit the number of modifications across different files, for a single purposed correction.