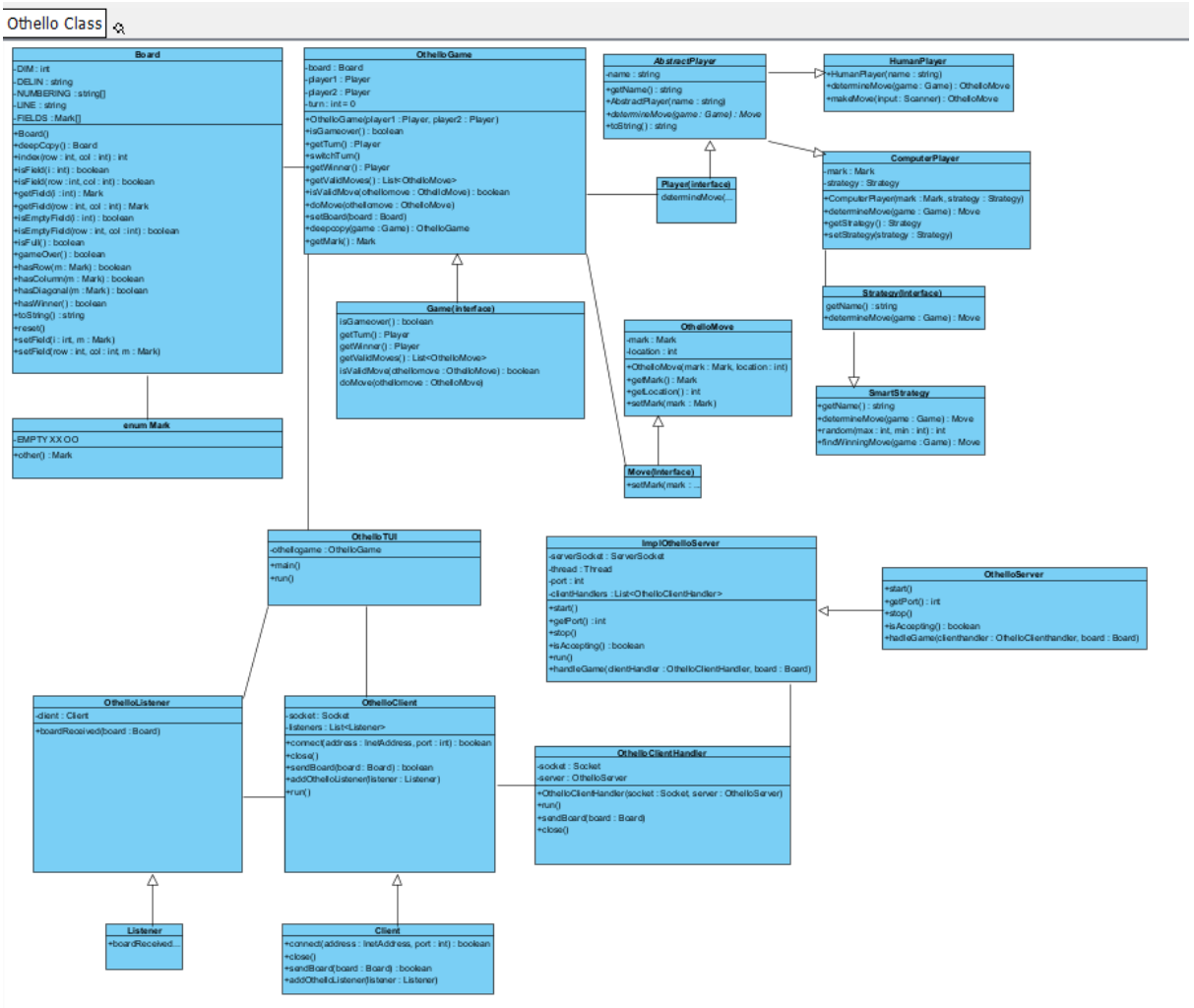
Othello Project Final Report

Red 14

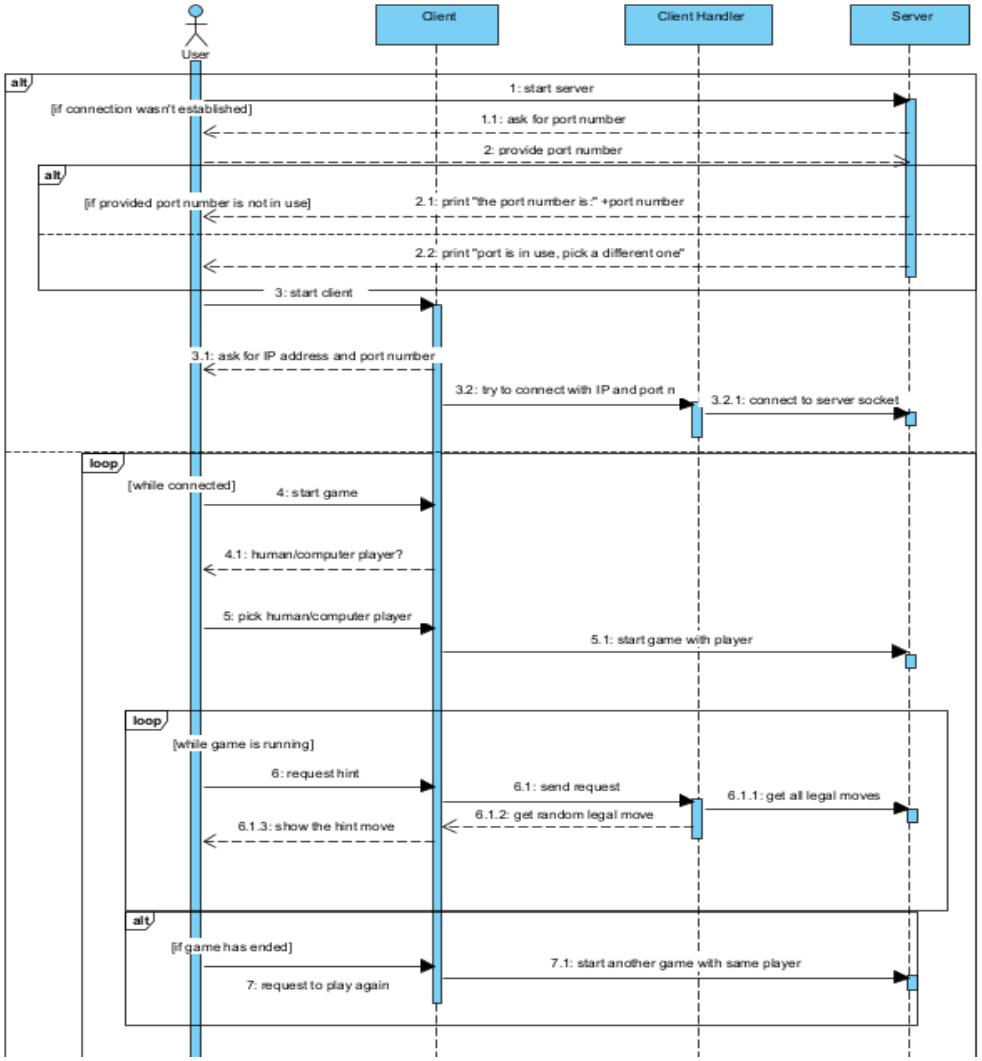
Shun Nishijima s2977923
Mateusz Bartnicki s3006891

Explanation of realised design

Class Diagram



Sequence Diagram



Description of each classes

Exception flow

Concurrency mechanism

Reflection on design

Initial Design Red 14

Initial Design

Final Design

System tests

Testing Report for FR 5
FR 5: *The name or description of Functional Requirement 5*
Expected behaviour: *A detailed description of how the software should behave, that is, the relationship between inputs and outputs, how it should interact with the environment or with other parts of the system, etc.*
Testing result • *Step 1* • *Step 2* • *Step 3* *Some screenshots, etc*

Test Strategy:

a) Scope of testing:

- Simple Othello game logic program that is able to act as a virtual board and game master at the same time, playing the game with inputs from players.
- An “AI” player, acting as a player, with two difficulty settings.
- An easy to use and clear game client that receives all the sent information and shows the user all the necessary visuals to play the game.
- A server that connects two clients to a game. The game is executed on the server itself and is played with the input from the connected clients.
- Fluid server-client infrastructure, with insurance that there will be close to none data loss or crashes between the two.

b) Risks

- An individual could breach the server and either cheat by altering the board in illegal ways or simply break the game.
- A lot of traffic could overload the connection/server and cause lag.
- Inefficient code could slow down the execution of the game making the experience slow.
- Badly implemented “AI” player could make facing it either too easy or too hard.
- Server code that does not respect thread safety could lead to data being altered in unintended ways making the game unplayable.

c) Test levels
    - Singular player: the perspective of the game for a regular player joining from their machine for a game.
    - Server: everything that takes place directly on the server, meaning any “backend” processes happening exclusively on it.
    - Game: perspective of two clients/players and their thread on the server all taking part in one game instance.
    - Whole system: all the players/clients in their games, all taking place on the server.

d) Test types
    - Functional: tests that make sure everything works in the technical sense, meaning there are no bugs, glitches or unintended side effects that would affect the service negatively
    - Security: test if all the exchanges are properly encrypted and if confidential data cannot be reached from the outside by unauthorized parties.
    - Performance: see if the game runs smoothly, without too many stutters or lag while also putting as little strain on the users machine as possible.
    - User Interaction: test with users, see if the UI feels intuitive to navigate, if the learning curve of using it isn’t too hard and the process isn’t annoying.

Test Plan

a) Schedule :
    - Game logic and tests for it should be completed before the Midway Submission deadline.
    - Testing of any feature should be completed within 2 days of its first implementation or any future adjustments of it.
    - The whole testing process and any final checks should be completed before the 30th of January.

b) Deliverables:
    - Before testing: test schedule and plan document
    - During testing: data required for testing, testing tools
    - After testing: test results, logs, bug reports, summary of the process

c) Objectives:
    - The main objective of the testing is to ensure a service that is functional and secure, meaning there are slight to none chances of bugs and crashes as well as no possibility of data loss.
    - The other objective is a product that is easy and satisfying to use, which doesn’t strain the user to find any function/data they might want to access and makes the navigation process intuitive and as a result satisfying as a whole.
    - The testing units should have an average of 90% coverage of the project.

Unit Tests

- TestOthelloGame

System Tests:

- Board
- ComputerPlayer(Hard and Easy)
- Game
- Server

Least-tested component

Reflection on process

Shun

Mat