Joshua Latvatalo

Huy Nguyen

JuAune Burgess

Design Document: Reversi Online Game

**Purpose:**

For this project, our main goal is to create a fully functioning online Reversi game. In this case fully functioning will entail a server that will allow a player go up against a low, medium, or hard level AI as well as let any variation of the aforementioned AIs compete against each other. The game itself is an 8x8 uncheckered board and initially the center four spaces will have to white and two black discs, each diagonal from their respective match. Each player will take turns making legal moves and in Reversi a legal move is defined by making a straight (horizontal, vertical, or diagonal) line between one disc of your color and the intended move space with any number of the opponent’s discs in between. When a move is made all of the opponent’s discs will be turned over and will take on the color of the other player’s discs. If a player cannot make a valid move their turn is forfeited. The game will end either when the entire grid is filled or when neither player can make a valid move. At this point the player with the most discs of their color on the board wins.

For client based interactions our client will be provided a GUI where they can request a connection to our server. In addition to this they will be provided options as far as the difficulty they would like to play and after that what move they would like to make. The board will be updated after each move is made, inform the player of how many discs each player currently has on the board, and eventually inform the client of a win or loss.

**High level Entities:**

Firstly, the client mentioned here is not actually coded into this project but is actually how we are defining the input that the GUI will be handling to properly create the requested game type and moves.

AI ENGINE

GAME MECHANICS

SERVER

GUI

Next, the GUI is what the client interacts with for creating a connection to the server, sending moves, and updating the game board.

The server connects to the client and hosts the game which relies on the AI Engine and the Game Mechanics to send moves and information to and from the client.

The AI Engine is a class that, according to the difficulty level selected, will determine what moves will be made by the computer against its opponent.

Lastly, the Game Mechanics section is what holds things like the game board, legal moves, updating the board, etc. This is called upon by the AI Engine and the server when either sends a move.