CSE 4345 Requirements and Design Document

Nathan Moore, Gustavo Castillo, Peter DeNicola

Non-Functional Requirements:

Requirement 1

Description: The product shall be called ‘Halma 3.0’.

Requirement 2

Description: The product shall be a client/server web-service.

Requirement 3

Description: The product’s server-side code shall be written in Python 2.7.

Requirement 4

Description: The product will be able to accept a Json parameter as input and interpret it.

Requirement 5

Description: The product shall output a formatted Json output to be read by an AI program.

Requirement 6

Description: The product shall be able to accept user input to change the AI program

Requirement 7

Description: The product shall use HTML5 in the GUI in the front end

Requirement 8

Description: The product shall use Javascript for the front end logic

Requirement 9

Description: The product shall connect multiple AIs through HTTP POST requests

Requirement 10

Description: The product's client-side code shall be written in HTML and JavaScript.

Functional Requirements:

Requirement 11

Description: The product shall display an 18 cell by 18 cell playing board for the user.

Use Case: The user of the product will want to be able to see the progress of the two, competing AIs in order to verify their effectiveness and correct operation. Therefore, the game engine of the product shall display a visual representation of the game board. In this case, the board shall have 18 cells in each row and 18 cells in each column. The user merely needs to see the moves the AIs make.

Requirement 12

Description: The product shall display 12 circular pieces for each user's AI.

Use Case: The user of the product will want to be able to see the progress of the two, competing AIs in order to verify their effectiveness and correct operation. Therefore, the game engine of the product shall display a visual representation of the Halma game pieces. Each piece shall be represented as a circular object inside of one of the cells on the game board. Each piece shall be placed in different cells based on the moves made by its controlling AI. The user merely needs to see the moves the AIs make.

Requirement 13

Description: The product shall link two AIs to a UI that will be their interface to the game engine and compete against each other.

Use Case: The game will be played by two AIs instead of two human users. The two AIs will interact with each other through an interface that will also connect them to the primary game engine. The AIs shall connect as clients to the game engine server. The AIs must be properly programmed in order to the play the Halma game according to the rules established by the game engine, otherwise, errors will result during gameplay. Moreover, the AIs must be able to send and receive data via JSON in order to interface with the UI.

Requirement 14

Description: The product shall implement a timer to control when the AIs take their turns.

Use Case: The AIs will interface with each other and the game engine via a UI. The game engine shall implement a timer that it will use to request an AI to take its next turn at regular intervals. This is because the AIs must take turns and play the game in an orderly fashion. The user should be able to keep track of the AIs' moves using the GUI, and, therefore, the AIs will have to wait a few seconds between turns for the sake of the user. The AIs must not take turns on their own but only when the game engine requests their next turn, otherwise, there will be errors.

Requirement 15

Description: The product shall display moves as they occur.

Use Case: As the AI's make their turns a spectator will be able to see the specific jumps taking place by looking into the console. Otherwise, the board will be updated visually with each move.

Requirement 16

Description: The product shall display visual notification when an AI's piece has entered the correct goal destination. The same will be true when a piece enters an opponent's destination.

Use Case: When an AI's piece reaches the goal destination, a visual cue such as a change in color will alert the user to this occurrence. If a piece leaves the destination, the color will be removed to indicate this as well. When a piece enters an opponent's destination, it will glow a different color to indicate the occurrence.

Requirement 17

Description: The product shall enter a Game Over mode when one AI has completed their goal. This mode will allow the option to view the move list or start a new game.

Use Case: User will be able to interact with several buttons that are added to a Game Over screen allowing the user to repick AI's, restart game, or view the move list. This should only be available after the game has been completed by one AI.

Requirement 18

Description: The product shall implement error handling to handle improper responses from the AIs to prevent crashes.

Use Case: The software will attempt to handle and deal with any errors it detects in response from the AI's so that whatever response is received will not crash the overall program.

Requirement 19

Description: The product shall have 3 second intervals in between moves

Use Case: The software should have a reasonable interval in between moves. Users should not wait too long or too short for the program to make a move through a players AI.

Requirement 20

Description: The product shall validate moves to ensure that they are legal moves being made, if an illegal move is attempted then the game will display an illegal move message.

Use Case: The software should check a move received to ensure that it is a legal move as to maintain the integrity of the game. Sometimes an AI will return an invalid move and the software needs to detect that to ensure that it does not make the move and displays an error message.

Requirement 21

Description: The product shall ensure that if a piece attempts an illegal move, it will penalize the piece with a loss of move during that turn

Use Case: When an illegal move is attempted, there should be a penalty. Because every piece is given a chance to move, and all pieces are equally important, we believe that not allowing the piece to complete the turn intended is penalty enough.

Requirement 22

Description: The product shall allow an AI to move all of its pieces once per turn.

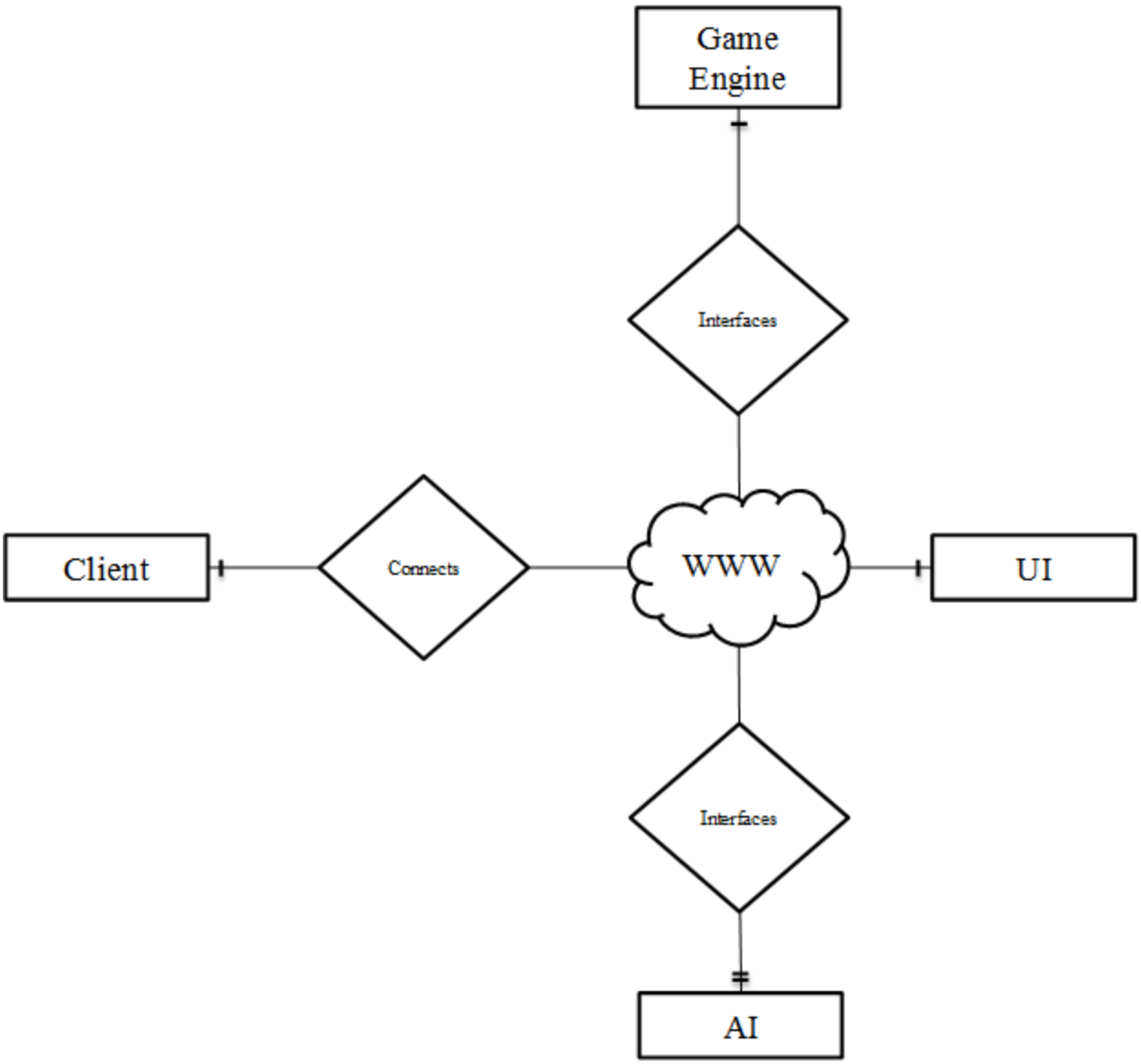
Use Case: AI games can be drawn out; thus, by allowing all pieces to be moved in a single turn, the game will speed up.

Requirement 23

Description: The product shall implement all remaining features of Halma 2.0 Multiplayer.

Use Case: The user can expect to be able to make moves, including jumps, and make use of all the other features already developed for the Halma game.

Software Architecture Diagram:



Source Code Control System

We will be using GitHub: https://github.com/nathandmoore/Halma_3_0