

**CS 40800**

**Team 10: Product Backlog**

**Project Title: LogicAI**

**Team Members:** Indhu Meena Ramanathan, Richard Hansen, Steven Dellamore,  
Columbus Holt

## Problem Statement

In the world of game-playing, there are a myriad of approaches that could be taken in order to play the game, either with other players or with an agent based on artificial intelligence. These agents are able to play against any number of different players, all with different strategies and approaches, and yet, the agent is able to play a competitive match. Our goal for this project is to develop an artificial intelligence agent to play a fractured line-matching game. The goal of the game is to enclose a shape by selecting the edges of the shape on the board, with the player and AI alternating turns, and winning points when completing a shape. This will provide players with a challenging competition against a well-trained AI, that they are free to access whenever they want to. This gives players the ability to improve their competitive and problem-solving skills through an interactive challenge game. The uniqueness lies in the creation of an agent that will be playing with various levels of difficulty in the game.

## Background Information

In our project, we will be creating a game called LogicAI. In this game, we will allow a user to play against a reinforcement learning agent. It can sometimes be challenging to find an area to experience the vastness of the artificial intelligence field. There is so much news and information about artificial intelligence being spread in the media, yet there is little interaction that people are able to have with these new and exciting algorithms and processes. Our game provides a simple and entertaining method for people to experience the intelligence and compete against an agent that is using artificial intelligence. We will also allow the users to experience playing against an agent by providing multiple levels of difficulty.

## Environment

Our frontend will be written in JavaScript, specifically with the p5.js library. This library will allow us to intuitively integrate real-time graphics rendering with the functionality necessary for our application. Our backend will be written using Golang, connected to a MySQL database. Golang allows us to easily write maintainable code, and the associated tests. Our artificial intelligence agent will be trained with the use of reinforcement learning and we will be using Python to train the agent. Our server will be hosted on a DigitalOcean VPS running Debian. Hosting our

server on a VPS allows us more flexibility with how we implement our artificial intelligence agent alongside our backend.

## Functional Requirements

Backlog ID	Functional Requirement	Hours	Status
1	As a user, I would like to be able to enter my name upon my first time playing the game	2	Sprint 1: Ready
2	As a user, I would like to be able to select the difficulty of the AI	2	Sprint 1: Ready
3	As a user, I would like to be able to press a button to start the game	3	Sprint 1: Ready
4	As a user, I would like to be remembered [by the browser] so that information is stored between sessions	4	Sprint 1: Ready
5	As a user, I would like to be able to select a game board to play on	2	Sprint 1: Ready
6	As a user, I would like to be able to navigate to the game board page so that I would be able to go to the game	3	Sprint 1: Ready
7	As a user, I would like to be given 20 different game boards so that I can play different versions of the game	16	Sprint 2: Ready
8	As a user, I would like to have dotted lines connecting vertices so that I know all my available moves on the game board	8	Sprint 1: Ready
9	As a user, I would like to be able to select lines and have them highlighted, so I know what moves I have made on the game board	8	Sprint 1: Ready
10	As a user, I would like to be able to capture shapes, by selecting enclosing lines, so I can capture shapes to win the game	8	Sprint 1: Ready
11	As a user, I would like to be able to see my opponent's moves highlighted, allowing me to see what moves they have made on the game board	4	Sprint 1: Ready
12	As a user, I would like to be able to see the	4	Sprint 2:

	score of the match so that I know if I am winning or not		Ready
13	As a user, I would like to be able to see what shapes my opponent has captured, so I can view the current board state	3	Sprint 1: Ready
14	As a user, I would like to be able to capture multiple squares, and have my score update accordingly, so I can see my current performance	2	Sprint 1: Ready
15	As a user, I would like the game to end once all available moves have been taken, and a final score being calculated	4	Sprint 1: Ready
16	As a user, I would like to see a preview of the game board that I can choose from so that I can select one that I am interested in	4	Sprint 1: Ready
17	As a user, the game board is able to distinguish between shapes captured by me and my opponent	3	Sprint 1: Ready
18	As a user, I would like to only be able to choose a unique name	1	Sprint 1: Ready
19	As a user, I would like to view my previous game score at the end of the game so I can see how well I competed against the AI	4	Sprint 2: Ready
20	As a user, I would like to have a quick-play option so I can immediately get into the game without having to make a username	3	Sprint 2: Ready
21	As a user, I would like to view my previous score ranked against the previous user scores, so I know how well my previous game compares	3	Sprint 2: Ready
22	As a user, I would like to be able to scroll through multiple past high scores, to see past scores	3	Sprint 2: Ready
23	As a user, I would like to navigate back to the home page when I complete a game	2	Sprint 2: Ready

24	As a user, I would like to see a timer for how long I've been playing the current game	4	Sprint 2: Ready
25	As a user, I would like to be able to have a trained agent that is able to pick edges on the game board so that I would be able to have an opponent	16	Sprint 1: Ready
26	As a user, I would like to be able to have a trained agent that is able to play with various levels of difficulty against me so that I can decide how I would like to play	16	Sprint 1: Ready
27	As a user, I would like to have an agent that is trained against multiple models so that I would be able to have a variety of different boards to play with	16	Sprint 2: Ready
28	As a user, I would like to have a way to go back once the game started so that if I clicked the wrong board or difficulty I can easily change back	3	Sprint 1: Ready

# Non-Functional Requirements

## Architecture

The frontend of this application will be developed with the use of p5.js and JavaScript. This will allow us to easily use our application to any browser, such as Chrome and Safari. We will be using the p5.js client-side library because it will allow us to create an interactive game, where the user will be able to select any of the edges to create shapes and have filled in shapes to gain points. Then, for the back-end, we will be using Golang, which has a myriad of test frameworks. The routes will be using Golang and we know that it will also be able to be tested with many frameworks. The backend will also have a connection to the database, which will be using MySQL. The backend and the database will be running on our Digital Ocean server. Then, we will also have a reinforcement learning agent that is created and trained in Python. This agent will be using reinforcement learning mechanisms to learn more about the structures and the moves that would need to be done.

## Security

To keep our user's information secure, we will avoid storing any personal information relating to them as we are interested in allowing the player to play a game. Users will be identified by cookies stored in their browser. This means that any breach related to our backend database will incur no penalty to our users.

## Usability

We hope to have a very clean and user-friendly interface. This will include the use of clean design and plain language. Our application will allow the user to choose from a variety of agents that the user would be able to play against. This will allow the user to have more customization with the type of agent that they would be competing against with the appropriate level of difficulty for the game play.

## Performance

Our application will provide users with quick response times, allowing them to enjoy our game with no headache and little wait time. By preparing boards before our users play the game, users don't have to wait for an AI to be trained for each specific board, while allowing the variety of multiple boards. We are also using p5.js, which is a library that is based for creating interactive experiences. This will allow the user to have a clean gameplay experience.

# Use Cases

## Case 1

Enter name upon opening site

Action	System Response
1. Site is opened for the first time	2. Display "Enter Name" text field
3. Enter Name	4. Send request to the backend to save name in the database
	5. If successful, generate and store a cookie for the user and navigate user to main menu

## Case 2

Select difficulty of AI agent

Action	System Response
1. Select difficulty from a group of options for the difficulty of the AI	2. Send request to backend to set proper difficulty for the AI agent upon next game load

## Case 3

Start the game

Action	System Response
1. "Play" button is pressed on the home page	2. Send request to backend to retrieve selected board and the selected difficulty for that board
	3. If successful, navigate user to the game board screen



#### Case 4

Be remembered by the browser

Action	System Response
1. Start game in browser	2. Application creates a cookie for the user and information about the cookie is saved in the database
3. Close the game and browser	
4. Open the game again in the browser	5. Application checks for cookie stored in browser
	6. If cookie exists, load main menu with returning user's data

#### Case 5

Select game board to play with

Action	System Response
1. Select a game board from the list of options for the game board	2. Send map request to receive map metadata from the backend
	3. Parse the metadata into a readable format for the game

#### Case 6

Navigate to game board page

Action	System Response
1. Select the appropriate game start button	2. Send request to retrieve the correct game board and model
	3. Display the correct game board
	4. Interaction options enabled for the player

### Case 7

Have 20 trained boards that I can choose from

Action	System Response
1. User has navigated to the main menu	2. Request is sent to backend to obtain information about the game boards
	3. Application displays all 20 of the current game boards available to the user that have been trained by the reinforcement learning agent

### Case 8

Know all my available moves on the board

Action	System Response
1. Be in play game state	2. The gameboard is loaded following these specifications
	3. All lines will be loaded between dots as dotted lines

### Case 9

View my selected lines to see my moves

Action	System Response
1. Select a line on the game board	2. Send request to select the edge of the shape for the user
	3. On success, dotted lines are replaced with a solid highlighted line where the user clicked

### Case 10

#### Capture shapes on the board

Action	System Response
1. Select the 4th edge of a shape to enclose the shape	2. Send request to select the edge
	3. On success, shape is filled with the user's color
	4. Score is added to the user's score

### Case 11

#### See my opponent's moves

Action	System Response
1. Make a move to trigger the opponent's move	2. User's move is recorded
	3. User's move is shown with a highlighting on the game board
	4. AI makes a move
	5. AI's move is represented by replacing a dotted line with the AI's color

### Case 12

#### View scores of match

Action	System Response
1. Start the game	2. Scores are displayed on the screen
3. Capture a shape	4. Request is sent to update the scores
	5. Scores on the screen are updated based on captured shape's points

### Case 13

#### View opponent's captured shapes

Action	System Response
1. Make a move	2. User's move is recorded
	3. AI makes a move, if this move captures a shape, the shape will be filled with the AI's colors

### Case 14

#### Capture multiple shapes and update score

Action	System Response
1. Select an edge that captures multiple shapes	2. Edge data is sent to the backend
	3. All square scores are summed
	4. New game state and score is displayed

### Case 15

#### Have a game that ends and view final score

Action	System Response
1. Take final turn	2. User edge data is sent to backend
	3. AI makes final move
	4. There are no more moves available, the game ends and scores are final
	5. Final scores are shown to the user

#### Case 16

##### Preview game boards

Action	System Response
1. Navigate to home page	2. Request is sent to obtain preview information about the game boards
	3. Previews are shown for the game boards

#### Case 17

##### View differences between my shapes and my opponent's shapes

Action	System Response
1. Make a move on the board by selecting an edge that completes a shape	2. Request is sent and my move is displayed and shape is captured
	3. AI makes a move and will potentially capture a shape and the shape will have a different fill

#### Case 18

##### Choose unique name

Action	System Response
1. Type a name into the select name box	2. Request is sent and database is checked for the name
	3. If the name is already taken, a system message will be displayed accordingly

#### Case 19

##### View all my game scores

Action	System Response
1. Complete any game against the AI	2. Score is saved in the database
	3. Navigate to screen with scores

	4. Previous game scores shown
--	-------------------------------

#### Case 20

Have quick play option

Action	System Response
1. Click on "Quick Play" on homepage	2. Unique username is generated, random game board and difficulty are chosen
	3. Navigates to play game page

#### Case 21

View my scores ranked against other players' scores

Action	System Response
1. Complete a game to receive a score	2. High scores are all displayed, ranked in descending order
	3. The user's score will be highlighted and ordered amongst the scores shown

#### Case 22

Scroll through multiple past scores

Action	System Response
1. Scroll through scores list containing the information about the users and their scores	2. Display all past scores
3. View the past scores that have been scrolled	

#### Case 23

As a user, I would like to navigate back to the home page when I complete a game

Action	System Response
1. Finish the game	2. Present with the option to navigate back to the home page

3. Click on navigate back to home page	4. Navigate the user to the home page
--	---------------------------------------

#### Case 24

##### View timer

Action	System Response
1. Be in state of playing game	2. Obtain timer and display timer

#### Case 25

##### Have an agent that is able to pick edges on the board

Action	System Response
1. Be in state of playing the game where you have just played a move	2. Request is sent for agent to make move
	3. Agent makes a move and chooses an edge
	4. Chosen edge is displayed

#### Case 26

##### Have an agent that is able to play with various levels of difficulty

Action	System Response
1. Be in the state of playing the game where you have just played a move	2. Request is sent for agent to make move
	3. Based on the level of difficulty, the agent chooses the action to make with the use of probability
	4. The edge is chosen and displayed

### Case 27

Have an agent that is trained against multiple models

Action	System Response
	1. Train models for various game board types
2. Click on a game with a specific level of difficulty to play game	3. Finds the information for the model from the back-end and obtains the information for the model when the agent makes a move

### Case 28

Be able to navigate back if I choose wrong board

Action	System Response
1. Click the back button when on the game screen	2. Request is sent to navigate back
	3. Send user back to the home screen
4. Change preferences	5. Navigate back to game screen