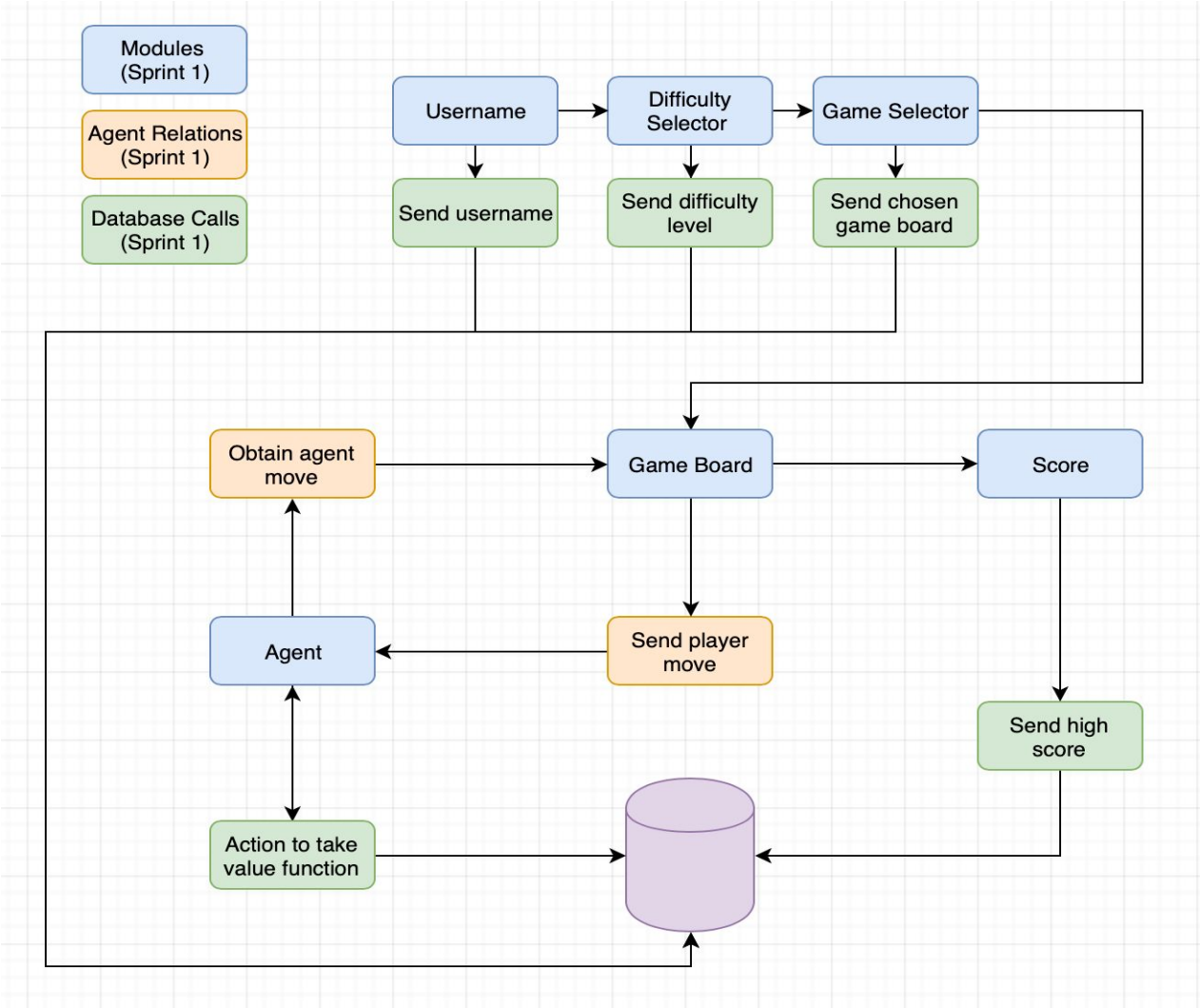
CS 40800
Team 10: Incremental Testing and Regression Testing
Project Title: LogicAI

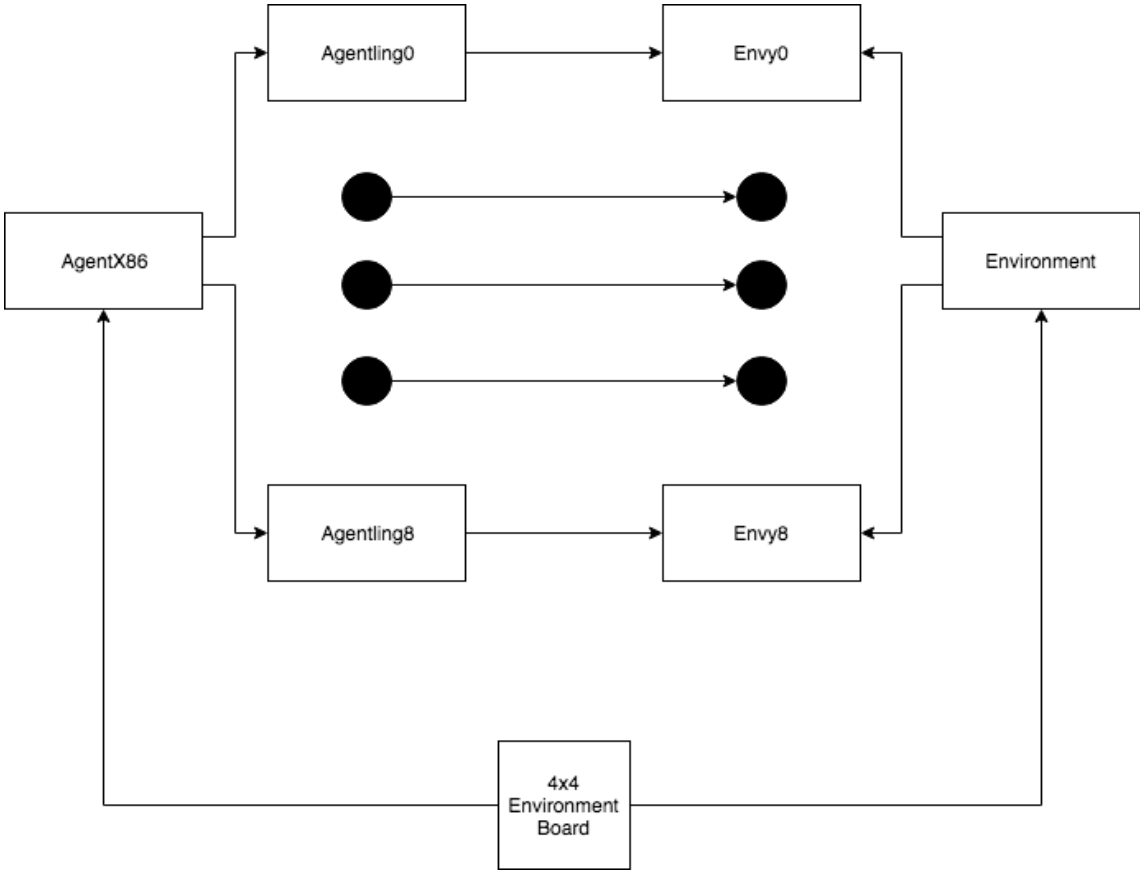
Team Members: Indhu Meena Ramanathan, Richard Hansen, Steven Dellamore, Columbus Holt

Classification of Components

Overall Design:



Agent Design:



Modules:

Game Selector Module:

On the start screen, we will have a module to allow the user to select the game board that they would like to play on from the variety of options. Each game board will have a trained agent that it is associated with the choice of the game board.

Input: Selection of game board
Output: Preview of game board
Parent Dependency: None
Child Dependency: Game board with correctly selected board

Difficulty Selector Module:

On the start screen, we will have various difficulty levels and game boards. The user will be able to choose a difficulty level, and they would also be able to choose the game board. If the user does not select either the game board or the difficulty, it will chosen by default. The choice of the difficulty will impact the agent's choices.

Input: Selection of difficulty
Output: None
Parent Dependency: None
Child Dependency: Board with correctly selected game board difficulty

Username Module:

On the start screen, we will allow the user to enter their username. This username is used to keep track of the score in the database and then show them the results after the game is completed.

Input: Username with alphanumeric characters
Output: None
Parent Dependency: None
Child Dependency: Game board with user's name

Game Board Module:

The game play screen consists of the game board and the results. The results show the scores during the game from both the agent and the player. When an edge is taken, the score will be changed depending on shape completion. When the user hovers over an edge, that edge will be highlighted to show that possible move. When an edge is taken, the dotted lines will be changed to solid lines to denote that the edge is taken and will

also show information about which player has taken that edge. When a square has been completed, the board will have the shape filled in.
Input: Player select edges
Output: Display of game board
Parent Dependency: Game Selector, Difficulty Selector, Username, Agent
Child Dependency: Agent, Score

Score Module:
The score module will be increased for both the Player and the AI. The values will directly represent area captured on the game board. These values will help the player know how many points they have and how many points are left on the board. This is shown on the game screen while the player is playing the game.
Input: Game board filled
Output: Percent of board taken
Parent Dependency: Game board
Child Dependency: None

Agent Module:
The agent class structure shows how an agent is merged with an environment to create an intelligent opponent for our users to play against. On the left and right sides of the below design, you can see that we have an AgentX86 Class and an Environment Class. These both represent high-level members of our game. Because the game cannot be handled as a whole (processing/memory restrictions) the game needs to be broken down into more manageable chunks. AgentX86 and Environment are both representations of the original 4x4 game board, and merge logic from the Agentling and Envy Classes. The Agentling is an agent that is trained against its own Envy, which is a 2x2 chunk of the original 4x4 board. We have 9 Agentlings, and 9 Envys. Each Agentling is abstracted from the idea of the larger 4x4 game board, and only attempts to play optimally on the 2x2 board. The decisions made by the Agentlings are sent up to an AgentX86 object, which then weights and combines the decisions from all 9 Agentlings to make a singular final decision for the 4x4 board.
Input: Player moves
Output: Agent moves
Parent Dependency: Game board
Child Dependency: Game board

Incremental Testing Approach:

We will be using top-down incremental testing for this project. We will be using top-down testing because we would like to test the top modules first, such as selecting the difficulty, game board, and the username. Then, with this information we will be moving onto testing the information for the game board. With the game board and all the top-level modules, we will be going down to the agent and the scores. This will allow us to ensure that all parts of the project work together in the general workflow. We have dependencies from the top level of the project to the bottom levels of the project and doing testing this way will allow us to take care of those dependencies as we will be going through the flow of the game. When a new module or component is added, it will be able to be added in the correct order.

Incremental Testing and Regression Testing:

Module	**Game Selector Module**		
Incremental Testing			
Defect #	**Description**	**Severity**	**How to correct**
1	When user clicked on a map, selection not indicated on the screen when choosing from the options	Important	Change the layout of the organization on the screen to incorporate the different color for the highlight so that the selection could be easily identified
Regression Testing			

Defect #	Description	Severity	How to correct
1	When user selected a map the displayed map would be one higher than the selected map, meaning that the incorrect map would be chosen	Critical	Fixed off by one error to set correct board to be displayed upon board selection after identifying the location

Module	Difficulty Selector Module		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	Incorrect difficulty selection was resulting because it was getting passed a string with the difficulty from the frontend, but backend required an integer	Important	Convert the string difficulty to an integer difficulty so the backend could interpret it and set the correct level for the difficulty
Regression Testing			
Defect #	Description	Severity	How to correct

1	Http requests did not include a difficulty key with an integer value	Important	Create a dictionary of difficulty to integer value that can easily convert between the two if needed

Module	Username Module		
Incremental Testing			
Defect #	**Description**	**Severity**	**How to correct**
1	User could enter non alphanumeric characters as part of username	Critical	Added additional verification to ensure usernames do not contain non-alphanumeric characters
Regression Testing			
Defect #	**Description**	**Severity**	**How to correct**
No defect			

Module	Game Board Module
Incremental Testing	

Defect #	Description	Severity	How to correct
1	Clicking on lines to select moves was not working for the user	Critical	Adjusted position of lines and mouse position so user could select their moves in a nice manner
2	When it was the agent's turn to play, the agent would not always make a move	Critical	Check to see if the hash codes that are associated for each of the players actually send back a response or an empty response rather than erroring out
3	Game board was not being sized correctly for different screen types on different screens for the laptop	Critical	Include relative values for the game board rather than the absolute values so that it would be fitting for the screen type on a web application

4.	When the AI or the Player would take two squares with the same lines, the score would not be updated to include the areas of both those squares	important	Add a check to see once the player took two squares with one move, the square array would be correctly modified and the score would be correctly added once
Regression Testing			
Defect #	**Description**	**Severity**	**How to correct**
1	Situation where AI and player where they could take two squares with the use of one edge	Important	Ensure that both the squares get highlighted and the correct value of the area of square one + square two to whoever took both squares by setting by that situation and seeing that updating the square area on both sides

2	Different size boards (100,100), (100,200),(200,100) not on correct proportions of the screen	Important	Check the locations with respect to the vertices on a xy plane to determine the correct locations and proportions on screen

Module	Score Module		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	When player and AI make moves, the score does not add up to 100 from both players	Workaround	Ensure that the rounding is done correctly for both the players so that it adds up to 100 when the last square is taken based on the rounding between the two
Regression Testing			
Defect #	Description	Severity	How to correct

1	Display of the score was not shown correctly with the appropriate number of decimal places	Workaround	Had to only truncate when displaying the score, not when adding the scores together. Also make sure when the final square is taken you do not truncate because then it would add up to 99.

Module	Agent Module		
Incremental Testing			
Defect #	**Description**	**Severity**	**How to correct**
1	Getting the potential actions that could be completed for moves in the game board was erroring when the hash codes that could be taken by the agent in the game board had no possibilities	Critical	Check the length of the potential actions that could be taken before building the database call statement so that when retrieving the information on the values that correspond to the hash

			codes, the empty dictionary would be returned rather than the error with selecting from an empty list of values
2	Updating the state history for an agentling has an error when that specific agentling does not have a state history to update for that specific game	Important	Check whether the state history for that agentling exists before obtaining the rewards and updating in the database, if it does not have a history, just continue and do not make update to its existing values
3	Updating the state of the player after an action taken on the game board did not consider the squares captured by the agentling and only considered the	Important	When determining the state for choosing an edge, also consider the squares that would be completed by choosing that edge by keeping track

	edges captured by the agentling		of the completion of squares with the newly chosen edge in another array by indicating that the square is taken in the environment and updating the state accordingly
4	The complicated mapping of the 2x2 to 4x4 environments must be maintained between versions of the code	Important	Ensure that the mappings are correct and that the correct number of environments are updated for each action made. Ie, a single edge chosen can update up to 6 environments in the when going from the 4x4 to the 2x2 space
Regression Testing			
Defect #	Description	Severity	How to correct

1	State information was only considering the edges themselves in the squares but did not consider the player who owned the square in some areas and considered the owner of the square in other areas after fixing the defect with the state history update	Important	Modify the state object such that it contains both the edges and the squares in the same list rather than have it as the same list in some places and multiple lists in other places so that the same functions could be used throughout when considering the taken edges and the edges that can be taken
2	When testing that the updates of the value is done correctly, the ternary finder for a hash code of 0 did not return a valid ternary	Workaround	Modify the ternary finding code such that when the hash code of 0 is put in as parameter an array of length 1 with a 0 is returned instead of an integer with 0

3	When attempting to get values back from the database on move, if an empty state history was passed there would be an error	Important	Modify the sql command code to handle an empty list of values, and not query when necessary, returning an empty dictionary for those values
4	When attempting to fill a 2x2 square in the game board, when all the squares were filled, the values were unable to handle a completely filled square	Important	Modify the function to get values by ensuring that the sql call was made when hash codes existed