CS 40800

Team 10: Design Inspection, Code Inspection, and Unit Testing

Project Title: LogicAI

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

Design Inspection

Product: LogicAI
Date: 9/19/2019
Author: Development Team

Defect #	Description	Severity	How Corrected
1	When play button was clicked, it would not take you to the new screen when it was supposed to navigate you to the game screen	Critical	Updated the gameState in the previous module because the gameState was not being updated when the player clicked play game, therefore we did not move into the next screen
2	No back button on the game screen, this will bring the screen back to the start screen so the player can select different settings/maps	Workaround	Will be adding the code for a back button on the game screen that will allow the user to navigate them to the start screen to start up a new game
3	Could not view text box for the user to put their username when user was to enter the name but the value of the user name is sent over to the game screen once the game has started	Important	Moved the text box and put a border around it so the user could click and enter their username
4	Scores for users was hard to read due to too many	Workaround	Added code to truncate the number of

	decimal places in the number that was shown on the game screen as it received the information from the scoring module		decimal places in the module for display for the scores so that the scores could be easily seen
5	Highlighting for captured squares was not shown, and it was based on the moves by the user that increases the score and will eventually be the value sent to the database	Important	Wrote code that would check each box every time a new move was played and the check was done in the module for calculating values and then passed into the display. If a box had all 4 sides taken, the box will get highlighted
6	Once the game has ended, the player should be brought to the score screen with their score and username passed in	Important	Will write code to navigate back to the score screen when the game is over to show the final results in another screen of the game
7	Quick play button is not present on the start screen, when the quick play button is clicked, it should bring the player straight to the game screen with a random map and default difficulty	Workaround	Will write code that adds this quick play feature to the start screen. This will allow the user to go straight into the game without having to worry about selecting options
8	Player name not	Workaround	Will write code

	displayed on the game screen and has Player as the name of the player when it is being passed in by the start screen		that will show the player’s name on the game screen, rather than the default “Player” as the name on the screen
9	Map data is handled differently depending if it's in the frontend, backend, or AI code	Critical	Added backend code that could translate the data between the three sections of the project
10	Selecting difficulty and board game not highlighted on the start screen but information would still be sent to the game screen	Important	Wrote code that highlights the selected difficulty for the game board player is interested in before sending the information to the game screen

(Product Design https://docs.google.com/document/d/1hB48TlBX1aKLdBTNbxCo1AP1THCSt6Ez5m1-QE41xWU/edit)

Code Inspection

Product: LogicAI
Date: 9/19/2019
Author: Development Team

Defect #	Description	Severity	How Corrected
1	Sending back map data to the frontend got crazy when the json got more and more complicated for the board game states	Important	Read the whole json into a struct and then simply sending the struct over to the frontend so that it would be in a nicer format for access
2	SQL drivers needed to show state values to pick actions for the AI in Go sometimes did not show that the object from the dependency was already there	Critical	Imported the correct driver for the SQL
3	Mapping with vertices and edges had some of the numberings mislabeled	Important	Re-evaluated the mapping function and determined the correct mappings from vertices to edges with the use of the boards
4	Unused variables in the class with	Workaround	Removed the updates to them,

	the smaller environments that did not need to be updated		and commented out the variables in case they may be needed to brought back in the logic of the game
5	Button navigation to game board page placement is placed on the boards	Workaround	Added code to move the navigation lower so that it is easier to press for the user
6	Sending square data for a single squares got confusing when trying to parse it on the frontend	Workaround	Read the whole json into a struct and then simply sending the struct over to the frontend
7	Database debugging call when obtaining the hash value and state by hash code still appears, which made it hard to see the results of the call and when it ended	Workaround	Removed print statements with the database calls during the process of the calls so that it is easier to see the hash at the end of the calls
8	A user could enter a name that was greater than the max buffer size of a []char. This would cause the server to segfault	Critical	We now check to see what the length of the username is before sending it to the backend. The backend also double checks the size to ensure it will not crash.
9	The order in which edge and square captured	Workaround	We now convert all edges from the 4x4 board to the

	mappings were done was hard to decipher/debug/ maintain		2x2 board, and vice versa to make mappings clearer and easier to maintain.
10	When the frontend tried to post Http requests the backend would not receive them	Critical	We had to change the server permissions to allow incoming connections through a specific IP:PORT
11	When the backend tried to query the database it was getting a panic because it could not open a connection	Critical	We had to go into the mysql server and make a new user that would have all the permissions to take requests from incoming traffic
12	The sketch.js screen could not create a variable of type gameScreen or startscreen when loading up	Important	Had to put the files inside the index.html so other files could read from them
13	The way the backend handed edges + vertices + boxes was different from how the frontend handled them	Important	Had to create different mappings from Frontend to backend. The Frontend only connected vertices and had no concept of edges, the backend had to change these connected vertices into edge numbers.

14	LogicAI.go was not able to route the calls to the correct functions	Critical	Had to set up the correct Go folder structure to be able to import packages inside Logic.AI

Unit Testing

Module: Game Startup Module

Product: LogicAI
Date: 9/19/2019
Author: Columbus

Defect #	Description	Input	Output	Severity	How Corrected
1	User would get pushed through to the game if it was their first visit and hadn't entered a name	No username, play button pushed	Expected: an error to appear to indicate that no user name was chosen Received: No error, game started	Critical	Added additional checking to verify that a username had been entered regardless of visitation status
2	Selection screens would continue being drawn after navigating to game page	Play button pushed	Expected: Only the game to show up Received: The game screen and selection boxes both appeared	Critical	Added additional functionality to hide the selection boxes upon screen change
3	Selecting a board n would result in board n-1 being displayed	Board n selected	Expected: Board "n" to appear selected Received: Board n-1 appearing to be selected	Important	Corrected the off by one error by fixing displayed board value to be n+1
4	Entering a non-unique	Non-unique username	Expected: An error	Important	Added frontend

	username would work [but not be updated] and push user through to game screen	for the username input	stating to enter unique name Received: User pushed through to game screen		validation of backend query result to ensure game would not start upon uniqueness error
5	Selecting multiple board in quick succession could result in not last map being selected	Quick multiple board selections	Expected: Last board selected being used Received: Random board being used	Important	Changed method of board retrieval and assignment to prevent the issue
6	Not selecting a board/difficulty upon first visit and pressing play would crash	No board or difficulty	Expected: Game to default to board 1 upon nonentry Received: Game crashed upon nonentry	Critical	Changed to default board and difficulty to Very Easy / Board One if neither had been selected
7	User could choose arbitrarily long names, which would not be stored	Arbitrarily long name beyond 30 characters	Expected: Error to show up to alert user Received: User would get pushed forward to game with no error	Important	Added name validation for size and an error to present user
8	Invalid usernames	Invalid usernames	Expected: Error to	Important	Awaited name

	would be stored on browser despite backend failure		appear and browser to not store the name. Received: Browser stored name rather		validation from backend before caching username on browser

Module: Game Board Module

Product: LogicAI
Date: 9/19/2019
Author: Steve

Defect #	**Description**	**Input**	**Output**	**Severity**	**How Corrected**
1	Area was being added up to more than 1 (100%)	“Map1” as the map for the game board	Expected: area to always add up to 1. Received: values that added above 1 (100%)	Important	Adjusted algorithm to change make sure it added up to 1.
2	Area was not being returned correctly for the game board	“Map1” as the map for the game board	Expected: The area to be sent in the same format as the frontend. Received: an array of values that were sent in the wrong	Important	I was sending the areas of the squares back in a different order than what was expected, so I fixed the order in which they were

			format		sending
3	When a square was taken, the values of the next squares would get messed up	“0 1“ , “0 5“ , “5 6“ , “1 6“ as the vertex pairs for the edges	Expected: once the player/AI takes a square, the other squares would keep the same area. Received: the correct value of the first taken square, but then mess up all the other square that were taken after	Important	Instead of splicing the square from the array I just set all the values to -1 so I would know that it was taken, but would not mess up indexes
4	When the AI took one of the edges of the square, and the other edges were taken, the square would not fill	Player: “0 1“ , “0 5“ AI: “5 6“ , “1 6“ as the vertex pairs for the edges	Expected: The square to be taken if all 4 sides are taken, regardless of player. Received: When the AI took one or more of the sides of the square it would never be filled in.	Important	Had to add the AI edges to the array that checks for squares
5	When the AI or the Player took two squares at once, only	“0 1“ , “0 5“ “1 2“ “2 7“ “6 7 “5 6“ , “1 6“	Expected: Both of the square to be taken by whoever took the	Important	Instead of returning right when I find a square full, I have to

	one of the squares would be taken		edge that completed both squares. Received: One of the squares would be taken by whoever took the edge, but not the other		return an array of squares that were taken
6	When the last edge was clicked, it would not fill in the square	Input all the edges. “0 1” “0 2” ... “2 3 2 4”	Expected: When the last edge is taken in the game it would complete the last square not taken. Received: When the user/AI took the last edge, the square would not fill and area would not be added to the square	Critical	I had to check for squares being filled even when the last move was played
7	Total area was somehow over 100%	Have one person take all the squares of the game	Expected: Either the player or AI to go over 100. I also expect that at the end of the game, the	Important	I was casting floats to ints. I just have to make sure the last square taken will

			player + AI will equal 100. Received: The player + AI would go up to 102.		go to 100. This will cause ~3-5% error in the actual area calculation, but since the areas are floating point numbers this is ok.
8	If the screen was a little smaller than my screen, the dotted lines would not connect correctly	Open up the game with a smaller screen	Expected: Dotted and connected lines would connect straight to vertices. Received: Line that would not go to any vertices, and vertices would have no lines depending on how big/small the screen was.	Important	Went back and changed how the line were connected. Now they are all connected relative to the game screen size
9	If the screen was too small, you would not be able to see half the game	Open the game in a small window (~300-400 pixels)	Expected: Gameboard to be scaled with the players screen. Received: When the gameboard was too small the	Critical	Went back and placed vertices relative to screen size.

			screen would not show some of the gameboard for the user		
10	If screen was different sizes, ur mouse would have to hover over different places to select lines	Open the game in bigger and smaller windows	Expected: Position of your mouse to always highlight a edge if it is over it. Received: Depending on the screen size, sometimes you would have to hover over weird spots to highlight and select a line	Important	Went back and checked mouse relative to screen size.

Module: AI States and Values

Product: LogicAI
Date: 9/19/2019
Author: Indhu

Defect #	Description	Input	Output	Severity	How Corrected
1	Maximum depth for the recursion level for enumerating environme	Areas for the 2x2 board to enumerate the states for are provided to create the	Expected: The table for the state and values must be provided for each of	Critical	Changed the recursion to an iterative approach so that the states

	nts	enumeration of the states	the hashes Received: The table was unable to be calculated due to the high depth of the recursion		could be enumerated without a recursion depth overflow error
2	Incorrect states being calculated for the next state on the 2x2 board	A provided state to the game board that is 2x2 with its current taken edges that the next states must be found for	Expected: The next states are all calculated correctly with the possible edges that are available Received: A couple of the next states were incorrect due to the mapping	Critical	Changed the dictionary mapping for the edges to the states in order to ensure that the next states could be calculated correctly based on the existing edges and state by having the correct next unfilled edges
3	Runtime of enumerating all the states was longer than a minute	Areas for the squares on the board on which the enumeration must be performed on to find the possible	Expected: The time to enumerate all the states for training must be below 1 minute Received: The total time to	Workaround	Added in conditional state enumeration in order to discard the impossible states, such as including a check that

		states for the training	enumerate all the states for training took more than 3 minutes		if all the edges of a square are marked as taken, the owner of the square is also there, and if the owner of the square is 0, then it means that no player has that square when one actually does, meaning that the state is impossible
4	State to value conversion was being miscalculated for a state that has already that has a square captured by a player	The current state to calculate the values for is provided for the player	Expected: The value should take into account the square that already has the area for Received: The value does not take into account a square already filled by a player	Important	Add the value of the already taken squares to the player's value function of that state
5	Hashing of	The state	Expected:	Important	Change the

	the state arrays into integers was doing the hashing in the incorrect order for ternary	to which the hashing should be done for to find the hashed value	The hashed value should convert list of states (which is all filled with 0, 1, and 2) and convert it to a decimal number, such that if there is a state like [2, ..., 0, 0, 0], the hash will be (2)(3 ^ 15) Received: The hash was done in the backwards order such that we had (2)(3 ^ 0)		exponent of the hashing such that it goes from 0 for the least significant digit and 15 for the most significant digit

Module: AI Agent and Actions

Product: LogicAI
Date: 9/19/2019
Author: Richard

Defect #	Description	Input	Output	Severity	How Corrected
1	Unfilled edges function	Agentling calling to get unfilled	Expected: list of all the unfilled	Important	Updated unfilled edges

	incorrectly implemented. Does not run safely.	edges	edges of the given environment state Received: variable not found error		function to correctly access class variables and iterate through said variables
2	Take action function incorrectly referencing the player variable. Does not run safely.	AgentX86 attempting to take an action on the environment	Expected: List of the current big board environment state Received: Variable not found error	Critical	Take_action now access the class variable correctly.
3	Unfilled edges was sending the inverse of it’s expected output	Agentling calling to get unfilled edges	Expected: list of the unfilled edges of the given environment state, not the list of filled edges Received: list of the filled edges in the current environment state	Important	Now correctly returns the correct variable
4	Get state value was incorrectly handling the edges	Given a current state and the current move, try	Expected: value of the hashed state should be	Critical	The edges and squares list are now treated in

	and squares list. Treating the edges list as if it contained information for both	and calculate the value of the state if the current one is updated with the move	returned Received: database error due to an incorrect state being hashed		the correct manner, preventing errors when calling the database