3 in 1 Game Design Inspection

Chris Coraggio, Muhammad Hamza Farrukh, Brad Howell, Rizwanulla Mohammed, Syed Ahad Sagheer

Severity Key:

- 1 is most severe
- 2 is median severity
- 3 is least severe

Design Inspection:

Product	3 in 1 Game		
Date	3/4/18		
Author	Development Team		
Defect #	Description	Severity	How Corrected
1	Tic Tac Toe board didn't get updated for opponent until a minute after the player made a move	1	Decreased the polling time from the app to the server to a few seconds
2	Connect 4 board didn't get updated for opponent until a minute after the player made a move	1	Decreased the polling time from the app to the server to a few seconds
3	Both Hangman players got different words	1	Instead of randomly choosing the words locally on the app, the words are retrieved from the server for both the players
4	In Connect 4 when selecting to play either multiplayer or single player (with AI) both	2	Found in the code the location where we checked the whether the player wanted to play single player or multiplayer and made sure the correct selection was made.

	instances led to the player playing a multiplayer version with the game.		
5	In Tic-tac-toe when selecting to play either multiplayer or single player (with AI) both instances led to the player playing a multiplayer version with the game.	2	Found in the code the location where we checked the whether the player wanted to play single player or multiplayer and made sure the correct selection was made.
6	In the hangman module, selecting either the single player or multiplayer selection both resulted in no activity being performed	1	We added intents to the correct activities
7	When connecting to a friend in the connect 4 multiplayer game, the nickname was not able to be fetched for connection.	1	Stored nickname on the server so that it could be fetched, and you could select a valid opponent to face.
8	When connecting to a friend in the Tic-Tac-Toe multiplayer game, the nickname was not able to be fetched for connection.	1	Stored nickname on the server so that it could be fetched, and you could select a valid opponent to face.
9	When connecting to a friend in the hangman multiplayer game, the nickname was not able to be fetched for connection.	1	Stored nickname on the server so that it could be fetched, and you could select a valid opponent to face.

Code Inspection:

Product	3 in 1 Game		
Date	3/4/18		
Author	Development Team		
Defect #	Description	Severity	How Corrected
1	Global variables are declared but not used	3	Remove all unnecessary global variables
2	Constant variables are declared and not used	3	Remove all unnecessary constant variables
3	Code in hangman activity class was unoptimized and was slow	2	Update certain UI pictures in constant time by individually updating picture instead of using loops.
4	If statements in hangman were very long, hard to read, and contained small errors because the code was hard to read	3	We broke up if statements and removed unnecessary ones.
5	Not all comments correspond to the code underneath	3	Remove/reorganized comments
6	UI buttons on the connect 4 and tic-tac-toe games when selected and didn't behave in the way that they were expected to behave.	1	The buttons were being mapped to the wrong functions when UI was implemented to code. Fixed it by fixing the mapping of functions.

Unit Testing:
The system relies on the following three modules:

- Tic Tac Toe Module
- Connect 4 Module
- Hangman Module

Tic-Tac-Toe Module

This module is responsible for playing the game Tic Tac Toe. It encompases code for the AI and multiplayer components of the game. This module was tested through blackbox testing. We did not use automated testing because automating tests with a server is difficult.

Product	3 in 1 Game		
Date	3/4/18		
Author	Development Team		
Defect #	Description	Severity	How Corrected
1	Game board was not being updated after first user played turn and lead to app crashing	1	Fixed bug in server code which stopped sending the board pieces constantly
2	Game would end too soon	1	Fixed bug in server code which ended game too early
3	Incorrect player would win	1	Fixed by correctly putting code in server which told the clients who won the game
4	Game would crash when server sent incorrect board pieces	1	Added checks in the server code which ensures the pieces sent are valid

Connect 4 Module

This module is responsible for playing the game Connect 4. It encompases code for the AI and multiplayer components of the game. This module was tested through blackbox testing. We did not use automated testing because automating tests with a server is difficult.

Product	3 in 1 Game		
Date	3/4/18		
Author	Development Team		
Defect #	Description	Severity	How Corrected
1	Game board was not	1	Found and fixed a bug in the server code

	being updated after first user played turn		which stopped sending the board pieces constantly
2	Game would end as soon as one player won without sending that information to the other player. So player 1 would win and the game would also end for player 2 but player 2 wouldn't see the final move or know why the game ended.	1	Made sure that the server would send the last bit of information and would end the game after all information was sent to both players.
3	Server would never update the rightmost column for either player	1	The dimensions of the board on the server were changed to match those on the app
4	The game wouldn't restart if all the possible locations for pieces were filled. This would mean that if the game ended in a tie then the game would end.	2	We added a check so that if the game ended in a tie it would restart the game and the players can play again.

Hangman Module

This module is responsible for playing the game Hangman. It encompases code for the AI and multiplayer components of the game. This module was tested through blackbox testing. We did not use automated testing because automating tests with a server is difficult.

Product	3 in 1 Game		
Date	3/4/18		
Author	Development Team		
Defect #	Description	Severity	How Corrected
1	Program crashed when out of words in multiplayer array	1	Create a method to grab more random words if the user guesses more than 10 words in a given round.

2	Program accepted previous words as correct answers	2	Rewrote checking method for correct answers to only consider what the user is currently guessing.
3	Program printed words in different order for different clients	2	Populated the server with the array instead of having the client fill in the order
4	Program assigned the score to the wrong user	1	When posting to the server, pass an object with the username and the score for the round