

3 in 1 Game Design Inspection

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

Design Inspection:

Product	3 in 1 Game		
Date	2/11/18		
Author	Development Team		
Defect #	Description	Severity	How Corrected
1	Tic Tac Toe game takes input using stdin not buttons	3	Change all occurrences of stdin in Tic Tac Toe to take input from buttons
2	Connect 4 game takes input using stdin not buttons	3	Change all occurrences of stdin in Connect 4 to take input from buttons
3	Hangman game takes input using stdin not buttons	3	Change all occurrences of stdin in Hangman to take input from Android text field
4	Tic Tac Toe game displays the board using stdout	3	Display the board on the android screen instead of the terminal
5	Connect 4 game displays the board using stdout	3	Display the board on the android screen instead of the terminal

6	Hangman game displays the game by printing out a string to the terminal instead of showing it graphically on the app	3	Display the board on the android screen instead of the terminal
7	When exiting out into the main menu and then re-entering, Tic Tac Toe doesn't restart game and shows the old board	3	Added code for restarting the game (which resets the board) on relaunch of activity
8	When exiting out into the main menu and then re-entering, Connect 4 doesn't restart game and shows the old board	3	Added code for restarting the game (which resets the board) on relaunch of activity
9	When exiting out into the main menu and then re-entering, Hangman doesn't restart game and shows the old board	3	Added code for restarting the game (which resets the game) on relaunch of activity

Code Inspection:

Product	3 in 1 Game		
Date	2/11/18		
Author	Development Team		
Defect #	Description	Severity	How Corrected
1	Global variables are declared but not used	1	Remove all unnecessary global variables
2	Constant variables are declared and not used	1	Remove all unnecessary constant variables

3	Certain variables were represented as coordinates in Tic Tac Toe which were hard to read and understand	1	Strings are now used to store variables and needs to be parsed to ints. This is easier to read.
4	If statements in Tic Tac Toe were very long, hard to read, and contained small errors because the code was hard to read	2	We converted many if statements to switch statements to improve readability.
5	Certain lines of code in connect 4 mixed with if a player was a 'O' or an 'X'	2	Fixed all occurrences of errors which mixed up the player's piece.
6	In connect 4 one for loop in the code accessed an array out of bounds	2	Have an if statement which makes sure that accessing the array is in bounds.

Unit Testing:

Tic-Tac-Toe Module

Product	3 in 1 Game		
Date	2/11/18		
Author	Development Team		
Defect #	Description	Severity	How Corrected
1	Verifying that the game was over by checking to see if a user won was not working correctly	1	One of the indices checked was i instead of 1, syntax error that was easily corrected
2	The gameOver() method was not	1	The method was only checking to see if a user had won. Added additional

	recognizin that the game was over if a tie occured		functionality to the method to check if there were no more moves possible by either player
3	Game would return an arrayIndexOutOfBounds error once the user reached their last turn	3	The computer would still attempt to make a move after the board was full. The array storing the moves it made was still being populated even though the board wasn't updating. This was because the AI wasn't recognizing that the game was completed. Added a check to verify the game was still taking place before making a move.
4	The AI was making multiple moves in one turn through the computerTurn() method	2	There were missing break statements in the switch statement within the computerTurn() method, causing the CPU to make multiplte turns. Break statements were added.
5	The AI never loses. Either wins or draws	1	Dumbed down AI by not having it make two good selections before guessing randomly. Now doesn't make mistakes, but doesn't make good decisions either.
6	The player turn was able to select -1 as a row or column value, leading to an arrayIndexOutOfBounds error	2	The code was checking < -1 instead of < 0, allowing the user to select rows that did not exist. Changed the code to < 0
7	Clicking on a certain button in the Tic Tac Toe game did nothing	3	The code had the wrong button id associated with the button and thus caused the button to do nothing. This was fixed by connecting the button to the correct id.

Connect 4 Module

Product	3 in 1 Game		
Date	2/11/18		
Author	Development Team		
Defect #	Description	Severity	How Corrected

1	AI doesn't always recognize that player is about to win.	1	AI should make checks for if the player is about to win.
2	CheckWinner() method doesn't check for the case where every spot on the board has been occupied to determine that the game ended in a tie.	3	Create a method that checks to make sure there are open spots on the board.
3	CheckWinner() method has trouble recognizing corner diagonal victories.	3	Fix CheckWinner() to check for those edges cases.
4	Game would immediately determine that either 'X' or 'O' was the winner	3	Board was being initialized at times with a player instead of blank, so changed constructor to fix this error.
5	CheckWinner() method wouldn't recognize horizontal wins that started on either the left corner or right corner.	3	Added statements in the CheckWinner() code that made sure edge cases were being checked for.
6	Out of bounds inputs were being accepted. Char wouldn't be placed on board but players turn would be skipped	3	Began checking for valid input and made sure player was prompted until a valid input was placed.

Hangman Module

Product	3 in 1 Game		
Date	2/11/18		
Author	Development Team		
Defect #	Description	Severity	How Corrected

1	Program doesn't recognize the same letter being guessed twice	2	Check the array of guessed characters before scanning the mystery word
2	Program sometimes reads twice from System.in	2	Flush the buffer after receiving user input
3	Program fails to end after 6 incorrect guesses if the incorrect guesses are not in order	2	Update a counter after each guess and check for lose condition after each guess
4	Program accepts invalid characters like parentheses	1	Check input for only valid alphabet characters
5	Program counts an invalid character as an incorrect guess	1	Added a condition to check for invalid characters, then break out of incrementing the invalid guess count.

