**B00165586 – Challenge 5 – Whack-A-Food – Document**

A screenshot of a computer

Description automatically generated

The opened-up Challenge 5 – Whack-A-Food from Unity – Learn with Code, which is a Whack-A-Mole styled game, where the player needs to hit various kinds of foods, while also avoiding skulls in the process. However, upon opening the game, it was riddled with problems, which includes the difficulty buttons looking very messy, the food objects are being destroyed too soon, the game score is being replaced with the word “score”, there isn’t a way to restart the game upon losing and the difficulty buttons aren’t changing the difficulty is any way. So, I would need to go into the game and fix these changes up.

A screenshot of a computer

Description automatically generated

A screenshot of the first problem getting fixed, which is fixing the text position for the three difficulty options, which were all out of centre, which was solved by going into the text area of each button and editing through the alignment options to centre the text.

A screenshot of a computer program

Description automatically generated

A screenshot of the second problem getting fixed, where the food in the game is being destroyed way too soon, as it’s getting destroyed when the mouse touches it, instead of the player clicking on the piece of food, so this was fixed by changing “private void OnMouseEnter()” to “private void OnMouseDown()” on TargetX.cs.

A screenshot of a computer program

Description automatically generated

A screenshot of the third problem getting fixed, where the total score in the game is being replaced with just “score”, so this is fixed by going into GameManagerX.cs and editing a line in “public void UpdateScore(int scoreToAdd)” with “scoreText text = “Score: “ + score;”.

A screenshot of a computer program

Description automatically generated

A screenshot of the fourth problem getting fixed, where there wasn’t a way to restart the game after getting a game over, so this was fixed by going into GameManagerX.cs and going into “public void GameOver()” and changing the “false” in “restartButton.gameObject.setActive()” to “true”.

A computer screen shot of a program code

Description automatically generatedA computer screen shot of a program

Description automatically generated

A screenshot of the fifth problem getting fixed, was where regardless if the player clicked on the Easy, Medium or Hard button, the game’s difficulty wouldn’t change, so this was fixed by going into GameManagerX.cs and updating “public void StartGame()” with “spawnRate /= difficulty” and going into DifficultyButtonX.cs and updating “void SetDifficulty()” with “gameManager.StartGame(difficulty);”

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generatedA screen shot of a computer code

Description automatically generatedA computer screen shot of a program

Description automatically generated

A selection of photos of the bonus problem being solved, which was the game going on endlessly and not having a way to stop unless the player got a game over for either touching a skull or not clicking on a piece of food fast enough, so I was able to solve this problem, with all of these code inputs, mainly on GameManagerX.cs.