Test Case ID	How I fixed the issue	Evidence (Key Words)
Example	*"At first, when I clicked the Add button, nothing happened. I realised I forgot to connect the command= in the Button to the function. I fixed this by adding command=add_task."*	***command = function()***
1	At first when I initiated the programme, the supposed txt file had not been created. I realised that the programme was putting the txt file in the folder that I had open being my assessment repository. Therefore I fixed this by opening a new vs code studio to test temporarily the conditions of how txt file saving works.	**(Nil)**
5	At first when I initiated the programme it had only accessed the keys of the dictionary. I noticed that I did not include the values that correspond with the keys in the dictionary inside the txt file. Therefore I fixed this by added the ability to check the dictionary for its values.	**with open(file_path, "r") as file:** **for line in file.readlines():** **print(line.split())** **# Automatically assigns values to the vars in list** **value, key = line.split()** **newerlist[key] = value**
8	At first when I opened the scoreboard menu, it displayed the correct amount of 10 placings available. However I noticed that there was an error regarding the amount of placings in the list available. Therefore I fixed this issue by adding a variable limit counter everytime a placing was displayed.	**for player_val in sorted_player_dir:** **if line_max <= 9:** **player_data = f"{placings[line_max]} {player_val[1]} {player_val[0]}"** **self.score_display.insert("", "end", values=player_data)** **line_max += 1**
9	When I tested the capabilites of the back button during PVE it did not proceed to the popup but rather came up with an error. I noticed that it would be hard to reuse the same function when refering to a button that does another different task. Therefore I created a separate function for a different purpose.	**def exit(self):** **self.grandparent.deiconify()** **self.destroy()**

10	At first when I opened this confirmation popup via the pve section it did not showcase all entry widget. I had noticed that I had repeatidly used the wrong variable when placing a entry widget on the widget frame interface. Therefore I fixed this by changing the vairbale reference.	**self.letter1 = ttk.Entry(self.widget_frame)** **self.letter1.grid(column=0, row=1)** **self.letter2 = ttk.Entry(self.widget_frame)** **self.letter1.grid(column=1, row=1)** **self.letter3 = ttk.Entry(self.widget_frame)** **self.letter1.grid(column=2, row=1)** **self.letter1 = ttk.Entry(self.widget_frame)** **self.letter1.grid(column=0, row=1)** **self.letter2 = ttk.Entry(self.widget_frame)** **self.letter2.grid(column=1, row=1)** **self.letter3 = ttk.Entry(self.widget_frame)** **self.letter3.grid(column=2, row=1)**

16	At first when I was testing the pve gamemode, the ai did not make another decision to place another circle tile. I had noticed that I added a limit to how many times the ai could move. Therefore I fixed this by removing the limit.	**if possible_moves != [] and game_won == False:** **for win_scenario in win_list:** **if len(set(x_list) & set(win_scenario)) >= 2:** **test = (set(x_list) ^ set(win_scenario)).pop()** **if test in possible_moves:** **win_list.remove(win_scenario)** **print(f"WIN LIST:{win_list}")** **self.record_list[test].config(text="o", state=tk.DISABLED, style="Circles.TButton")** **o_list.append(test)** **o_list.sort()** **possible_moves.remove(test)** **looking = False** **disable_count += 1** **self.win_cond()** **break** **else:** **while looking:** **a = random.randrange(1,9)** **for move in possible_moves:** **if a == move:** **self.record_list[a].config(text="o", state=tk.DISABLED, style="Circles.TButton")** **o_list.append(a)** **o_list.sort()** **possible_moves.remove(move)** **looking = False** **disable_count += 1** **self.win_cond()** **else:** **while looking:** **a = random.randrange(1,9)** **for move in possible_moves:** **if a == move:** **self.record_list[a].config(text="o", state=tk.DISABLED, style="Circles.TButton")** **o_list.append(a)** **o_list.sort()** **print(move)** **possible_moves.remove(move)** **looking = False** **disable_count += 1** **self.win_cond()**

17	At first when I had changed the reference of all the global variables the programme did not work. I realised that when I referencing the variables, I was not giving them time to reset after a game had finished. Therefore I created a separate function that will be called to reset these variables	**# Resets all game properties to default state** **def game_set():** **game_properties.win_list = [** **[1, 2, 3],** **[4, 5, 6],** **[7, 8, 9],** **[1, 4, 7],** **[2, 5, 8],** **[3, 6, 9],** **[1, 5, 9],** **[3, 5, 7]** **]** **game_properties.game_won = False** **game_properties.disable_count = 0** **game_properties.crosses = True** **game_properties.x_list = []** **game_properties.o_list = []** **game_properties.possible_moves = [1,2,3,4,5,6,7,8,9]** **game_properties.record_list = {}**