

Test Case ID	How I fixed the issue	Evidence (Key Words)
Example	<i>"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."</i>	command = function()
1	At first when I assigned the button to a different background colour it did not properly change the colour of the button. I noticed that the current theme was overriding the changes that I wanted to make. Therefore I fixed this by assigning a new style that is more flexible for style changes.	style = ttk.Style() style.theme_use("clam")
2	At first when I assigned the new font to the button, it had no change. I noticed that this character could only be affected by some fonts in particular. I therefore fixed this by changing the font to a more appropriate one	font=('Segoe UI Symbol', 50),
4	At first when I tried to use the button widget, it had an offset everytime it was pressed. I noticed that this was an unchangable feature of the tk button widget and therefore fixed this by changing the widget to a ttk.button modifying attributes via tt.styles	button = ttk.button
7	At first when I tried to use the button to show the state of what it would look like during the turn of 'x's it wouldn't show the new colour. I noticed that I needed to change the colour of the button in the deactivated state. Therefore I added a custom state background during its deactivation.	style.map("Crosses.TButton", background=[("disabled", "#F8CECC")])
9	At first I wanted to change the circle's style colour of the button when activated by the player to be blue however was red. I noticed that it inherited the same style that was being used for the 'x's. I fixed this by making a completely seperate style for it.	style.configure('Circles.TButton', foreground='Black', font=('Segoe UI Symbol', 50), padding=yippe, relief="flat") style.map("Circles.TButton", background=[("disabled", "#DAE8FC")])
13	At first when I was trying to pause the game when the player had achieved the win condition, it failed to show all move that were made by players since it wouldnt load the last move made. I noticed that I needed to put a wait function before the game was recreated and therefore show that the players on how they won.	if len(set(x_list).intersection(set(win_scenario))) == 3: root.after(2000, score_add("cross"))
13	At first when I won the game using 'x's it did not showcase the green indicator and went straight to resetting the game. Therefore I temporarily deactivated the function that was later called in the after function to show the style.	if len(set(x_list).intersection(set(win_scenario))) == 3: for button in set(x_list).intersection(set(win_scenario)): self.record_list[button].config(style="XWin.TButton") #root.after(2000, score_add("cross"))

14	At first when I won the game, the game did not give time for the animation of winning to play. I noticed that I was calling the animation during the after state rather than waiting for it to be called after 2000ms. I fixed this by adding the command function rather than calling the function itself	<code>root.after(2000, lambda: score_add("cross"))</code>
16	At first when I tried to play the original game programme it did not load. I noticed that the referencing of all the widgets had been unable to converse with one another making it difficult for the programme to find the necessary defined variables too early or too late. Therefore I fixed this issue via making a seperate classes to organise the code and make it more manageable to refer in other pieces of code in my programme.	(Too Large to Paste)
23	At first when I opened the scoreboard popup widget, the leaderboard of player names in the sample did not inherit the style that I had created. I had noticed that the reference I was using was incorrect. Therefore I fixed this by changing the reference of the treeview widget to allow it to inherit the style.	<code>"Custom.Treeview"</code> <code>"Treeview"</code>
25	At first when I tried initiating a game with the AI, it decided to place its markers on spaces that were already claimed by either itself or myself. I noticed that I did not give constraints on what the ai was able to choose from in the 3x3 grid. Therefore I made a list that will act as an ever changing list to give reference on what the ai can choose.	<code>possible_moves = [1,2,3,4,5,6,7,8,9]</code>
26	At first when I left the game with no available moves to make, the programme did not automatically restart. I noticed that the ai did not update correctly on when the game has restarted. Therefore I fixed this by using a variable to detect how many moves have been made to therefore reset the game when it meets the requirement.	<code>if disable_count == 9:</code> <code> print("huh")</code> <code> self.parent.after(2000, lambda: self.parent.score_add("tie"))</code>
29	At first when I played the game against the ai, the ai would make another move even though I had won the game. I noticed that the game did not detect if the player had one or not and only cared about available spcaes. Therefore I made a variable to detect when the player has won against the ai and vice versa allowing to make a move or not.	<code>if disable_count == 9 and game_won == False:</code> <code> print("huh")</code> <code> self.parent.after(2000, lambda: self.parent.score_add("tie"))</code>