EVALUATION

I created a title in order to welcome the user to the naughts and crosses game. The time module is used here so the user can clearly read and understand any outputs. I have created an artistic title for aesthetics which will attract the user to my program.

Validation was added within this menu, when the user chooses option 1(START THE GAME) or option 2(QUIT). This Validation is in the form of a while loop, it stops the user from proceeding until they input valid data. The valid data is in the form of the user choosing either option 1(START THE GAME) or option 2(QUIT). This has been done to ensure that the program runs smoothly allowing, the user to enjoy it more. A way to improve this menu is by changing the font, size and colour of the text displayed to the user. However I am unable to do this due to restriction put on me by the program.

After the user chooses option 1 (START THE GAME), they are shown the rules of the game. These rules dictate how the game should be played thereby, reducing the number of times the user enters erroneous data. This allows a constant flow of gameplay. The time module was also re-used here to enable the user to focus on the rules. This is good for the user because they will be more likely to enjoy playing the game. To improve this, I could ask the user if they understood the rules, and if they didn’t, I could have created an alternative set of rules. Another improvement could just be to elongate the time between the program displaying the rules and asking the user for their next input. This is an improvement because it gives the user even more time to fully understand the mechanics of the game.

In order to get information about what icon the user will represent, I have asked the user to input their preference giving them two options of “X” or “O”. While getting the user to do this, I have added validation to each section, this acts as a safeguard against invalid data that the user tries to enter. If the user inputs invalid data, the validation process will repeatedly prompt the user until valid data is entered. An example of this, is when the user enters their preferred icon they want on the grid. The user will be continuously prompted for valid data if, the value entered is outside the range of 1 and 2. A way to improve this part of the program would be to create a system wherein, the users’ icon is pre-determined. This is an improvement because it stops the user from worrying about having to make choices and instead, allows them to get on with playing the game.

The procedure “maingame” handles the users’ icon placement on the grid. Within this procedure, previously named variables were called in order to obtain the original stored inputs the user made. For example, “Xwin”, which is updated after every complete game. This is good for the user because they would be able to keep track of the games they have had. Which allows the game to be more competitive.

The procedure “x\_place” handles the users’ “X” icon placement on the grid. Within this procedure, the variable “x” is created here, this variable is one of the two user inputs during the main portion of gameplay. Previously named variables were called in order to obtain the original stored inputs the user made. For example, “maingame”, which is updated after every complete game. This is good for the user because it allows them to see where they have placed their icon and how that effects their next move. Which allows the game to become a lot more tactical.

The procedure “o\_place” handles the users’ “O” icon placement on the grid. Within this procedure, the variable “o” is created here, this variable is one of the two user inputs during the main portion of gameplay. Previously named variables were called in order to obtain the original stored inputs the user made. For example, “maingame”, which is updated after every complete game. This is good for the user because it allows them to see where they have placed their icon and how that effects their next move. Which allows the game to become a lot more tactical.

In order for the user to have more than one turn, iteration has been used to loop the game while constantly checking for a winner, this has been done by using the while, if and elif loops.