Coding Project Python Group #28

Assignment Overview

In this assignment we created the classic tic-tac-toe game using the Python program and Tkinter as an interface for rendering graphics. With this assignment we learn how to create buttons and define text on buttons. Furthermore, we learn how to use lambda functions in combination with callback functions linked to events such as mouse clicks. The output is a simple and easy to use game of Tic-Tac-Toe.

Assignment Specifications

Tic-Tac-Toe is a game for two competitors. Competitor1 plays X and competitor plays O. They both take turns placing their marks on a grid of 3x3 cells. If a given competitor gets 3 marks in a row horizontally, vertically, or diagonally, then that competitor wins the game. The game will be tied if no one gets 3 in a row by the time all the boxes are marked.

Assignment Deliverables

The deliverable for this assignment is the following file:

Group project #28.py – the source code for your Python program

Assignment Notes

This program consists mainly of 5 steps, which we will explain below.

1) **Import Modules**:

   Firstly, we need to import tkinter, which is a standard graphical interface for Python used to render graphics and can be used to create numerous other games such as Rock Paper Scissors or Snake. To display message boxes in our application we also import the module tkMessageBox. If you would like to be creative you can for example also import font from tkinter to change the fonts of the fields.

2) **Initialise window**:

   With Tk() we initialise the window and name it using title(). Here we named it game. We also need the method mainloop() at the end of the code to run Tkinter in Python, so it reacts to the clicks on the boxes.

3) **Define a function to check the result**:

   This function needs to check, which of the two competitors gets 3 in a row up, down, across or diagonally. So, we create a function (win) and determine its conditions respectively all possible ways to win the game.

4) **Define a function to check the winner**:

   Here we need to check who is the winner, if there even is one (a tie needs to be a possibility as well). The competitors take turn to click on a box of their choice. There are a total of 9 boxes and accordingly maximum 9 possible moves per game. We are counting the number of clicks. The count starts at 0. Whenever a competitor makes a move there will be one less option by increasing the count. Each time the count is even competitor1 makes a move else it is competitor2's turn.

   With config(), the fields are marked with the character or text specified in advance. Here we use X and O. The symbols can be easily replaced with whatever one would like to use. The method showinfo() is used to the competitor relevant information, basically when competitor1 or competitor2 wins or when there is a tie. We then use destroy() to exit the mainloop() once the game is over (win or tie).

5) **Define labels and buttons**:

   In the last step we need to define the boxes (Buttons in Tkinter). We differentiate between label(), which the competitors are unable to alter and button(). Game is the window we initialised before in step 2. Using the text widget we are able to store the value we want to display on the label and adjust its font and size. With the command binding we associate a callback function with the mouse clicks on the boxes and use the lambda() to send the specific data to the callback function. With Tkinter and config() one can then change the colours of the symbols in the boxes as well as their background to whatever colour one likes.

Group members:

Severin Bollhalder

Andrina Castelberg

Andrea Dell'Anna

Dimitrij Pantic