

Testing code

Code

```
while True:
    # Use EasyGui's buttonbox to create a main menu for the user.
    # This function returns the text of the button that was clicked.
    choice = eg.buttonbox(
        "What would you like to do?",
        "Main Menu",
        choices=["Add Player", "Show All Players", "Exit"]
    )
```

What it does

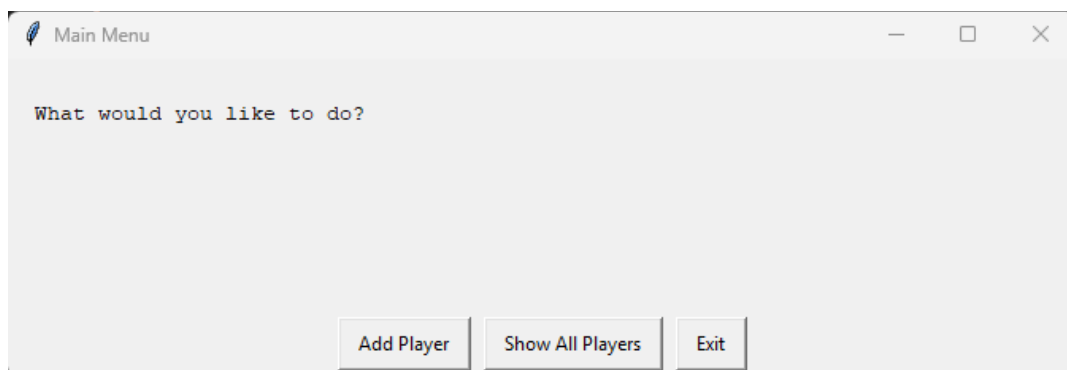
This block of code creates a loop that sets the user into a constant loop which can't be escaped unless you press the exit button or the x at the top corner, and it uses easygui to create a clean, tidy look.

Intended outcome

The intended outcome is for a box to appear with three options, each one should take you to a different box, and if pressed exit, it should exit the menu.

Actual outcome/evidence

In the screenshot below we can clearly see that a box has appeared with three options to choose from.



Code

Prompts the user for a new contact's details and inserts them into the database.

```
"""  
  
msg = "Enter information about your new player"  
title = "Add Player"  
fieldNames = ["Player_name", "Position", "Player_number", "Goals",  
"Assists"]  
  
fieldValues = eg.multenterbox(msg, title, fieldNames)
```

What it does

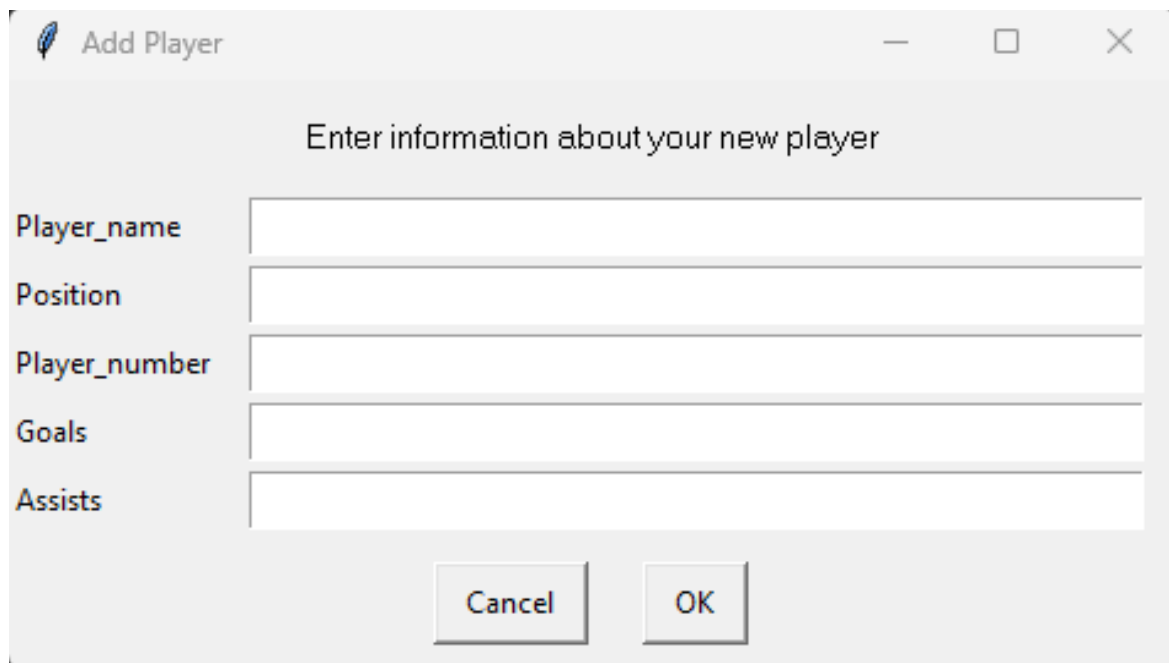
This block of code asks the user to enter the details of the player such as player name, position, player number, goals and their assists.

Intended outcome

After the user has pressed the add_button, another box should appear that asks the user for user_input for the details of the player they want to add.

Actual outcome/evidence

After pressing the add player button, I was met with another box that asked for the details of the player I want to add.



The screenshot shows a standard Windows-style dialog box titled "Add Player". Inside the dialog, there is a subtitle "Enter information about your new player". Below this, there are five text input fields, each preceded by a label: "Player_name", "Position", "Player_number", "Goals", and "Assists". At the bottom of the dialog, there are two buttons: "Cancel" and "OK". The dialog box has a light gray background and a title bar with a feather icon, a minus sign, a maximize button, and a close button (X).

Code

try:

```
# Execute a SELECT query to get the Player name and Details of all contacts.
```

```
cursor.execute("SELECT Name, Position, Player_number, Goals, Assists FROM Football_Database")
```

```
# Fetch all the results from the query as a list of tuples.
```

```
rows = cursor.fetchall()
```

What it does

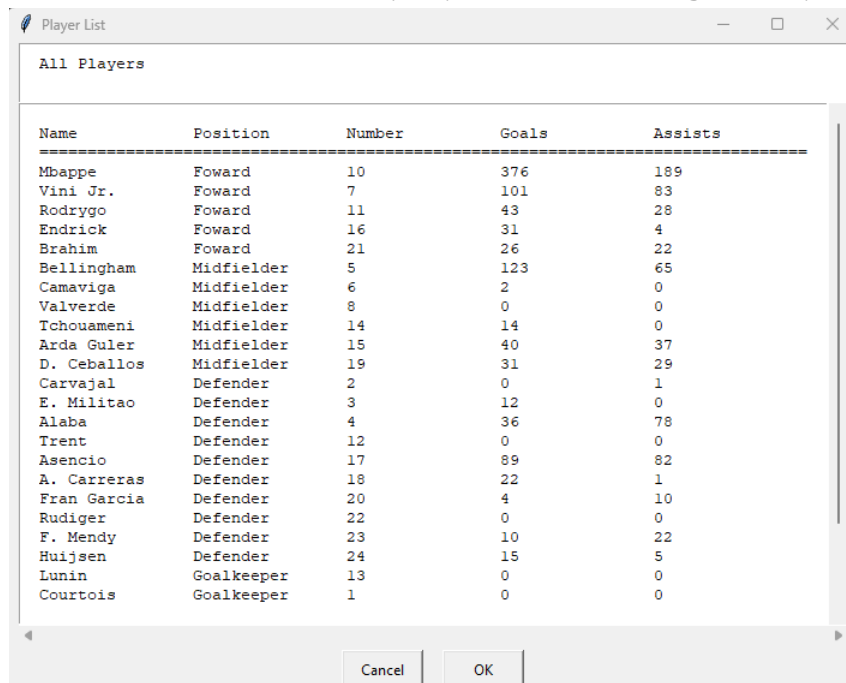
This chunk of code tells the database to have player name, position, player number, goals and assists. It also carries the information from the SQL database Football_Database and stores it into a list.

Intended outcome

The aim is to have the name of all the Real Madrid players, and their details in a row to see how many goals, and assists they have scored.

Actual outcome/evidence

When pressed to show all players, another box appears showing the details of all Real Madrid players, including newly added ones.



The screenshot shows a window titled "Player List" with a table of Real Madrid players. The table has five columns: Name, Position, Number, Goals, and Assists. The data is as follows:

Name	Position	Number	Goals	Assists
Mbappe	Foward	10	376	189
Vini Jr.	Foward	7	101	83
Rodrygo	Foward	11	43	28
Endrick	Foward	16	31	4
Brahim	Foward	21	26	22
Bellingham	Midfielder	5	123	65
Camaviga	Midfielder	6	2	0
Valverde	Midfielder	8	0	0
Tchouameni	Midfielder	14	14	0
Arda Guler	Midfielder	15	40	37
D. Ceballos	Midfielder	19	31	29
Carvajal	Defender	2	0	1
E. Militao	Defender	3	12	0
Alaba	Defender	4	36	78
Trent	Defender	12	0	0
Asencio	Defender	17	89	82
A. Carreras	Defender	18	22	1
Fran Garcia	Defender	20	4	10
Rudiger	Defender	22	0	0
F. Mendy	Defender	23	10	22
Huijsen	Defender	24	15	5
Lunin	Goalkeeper	13	0	0
Courtois	Goalkeeper	1	0	0

Code

```
elif choice == "Exit" or choice is None:  
    # The loop breaks if the user clicks 'Exit' or closes the window  
    ('None').  
    break  
  
# Close the database connection when the program's main loop ends.  
conn.close()  
eg.msgbox("Goodbye!", "Exiting Program")
```

What it does

When the user presses exit the, it stops the loop, closes the data connection, and then shows a pop up message saying goodbye.

Intended outcome

The intended outcome for this code is for another box to appear when the user clicks, exist. Another message should appear saying goodbye closing the program.

Actual outcome/evidence

When I pressed exist, another message appeared for the last time, saying good bye. Then I pressed ok and the program closed.

