

GAME MECHANICS-II



What is our GOAL for this CLASS?

In this class, we have written code to display the turn of the player also display the winning message when a player wins

What did we ACHIEVE in the class TODAY?

- Wrote code to display the player turn.
- Wrote code to display the name and the score of the player.
- Wrote code to display the winning message and add the reset functionality

Which CONCEPTS/ CODING BLOCKS did we cover today?

Add a bulleted list of new coding concepts that were covered in the class.

- GUI using tkinter.
- Functions

How did we DO the activities?

In earlier classes we have learned to write logic for the player movements.

Today, you completed the ludo ladder game by:-

1. Write a **receivedMsg()** function to get the received message from the client and perform activities such as displaying the dice value , player turn on the screen.

```
def receivedMsg():  
  
    global SERVER  
  
    global playerType  
  
    global playerTurn  
  
    global rollButton  
  
    global screen_width  
  
    global screen_height  
  
    global canvas2  
  
    global dice  
  
    global gameWindow  
  
    global player1Name  
  
    global player2Name  
  
    global player1Label  
  
    global player2Label  
  
    global winingFunctionCall  
  
  
    while True:
```

```
message = SERVER.recv(2048).decode()

if('player_type' in message):

    recvMsg = eval(message)

    playerType = recvMsg['player_type']

    playerTurn = recvMsg['turn']

elif('player_names' in message):

    players = eval(message)

    players = players["player_names"]

    for p in players:

        if(p["type"] == 'player1'):

            player1Name = p['name']

        if(p['type'] == 'player2'):

            player2Name = p['name']

elif('❎' in message):

    # Dice with value 1

    canvas2.itemconfigure(dice, text='\u2680')

elif('❏' in message):

    # Dice with value 2

    canvas2.itemconfigure(dice, text='\u2681')

elif('❐' in message):

    # Dice with value 3
```

```
        canvas2.itemconfigure(dice, text='\u2682')

    elif('4' in message):

        # Dice with value 4

        canvas2.itemconfigure(dice, text='\u2683')

    elif('5' in message):

        # Dice with value 5

        canvas2.itemconfigure(dice, text='\u2684')

    elif('6' in message):

        # Dice with value 6

        canvas2.itemconfigure(dice, text='\u2685')

    elif('wins the game.' in message and winingFunctionCall ==
0):

        winingFunctionCall +=1

        handleWin(message)
```

2. Define a **handleWin()** function to remove the roll dice button from the player screen, display the player win message and place the reset button .

```
def handleWin(message):
    global playerType
    global rollButton
    global canvas2
    global winingMessage
    global screen_width
    global screen_height
    global resetButton

    #destroying button
    if('Red' in message):
        if(playerType == 'player2'):
            rollButton.destroy()

    if('Yellow' in message):
        if(playerType == 'player1'):
            rollButton.destroy()

    # Adding Wining Message
    message = message.split(".")[0] + "."
    canvas2.itemconfigure(winingMessage, text = message)

    #Placing Reset Button
    resetButton.place(x=screen_width / 2 - 80, y=screen_height - 220)
```

3. Write a if condition to check if the message contains “player1Turn” and “player2Turn” in the message. If “player2Turn” in message then call the **movePlayer1()** function by passing diceValue as it's parameter.
4. If “player1Turn” in message then call the **movePlayer2()** function by passing diceValue as it's parameter.

```
if('player1Turn' in message or 'player2Turn' in message):
    diceChoices=['❶', '❷', '❸', '❹', '❺', '❻']
    diceValue = diceChoices.index(message[0]) + 1

    if('player2Turn' in message):
        movePlayer1(diceValue)

    if('player1Turn' in message):
        movePlayer2(diceValue)
```

5. Write an IF condition to check if the message contains the playerTurn and the player type.
If the message contains **player1Turn** and playerType as player1 then display the roll button for player 1.

If the message contains **player2Turn** and playerType as player2 then display the roll button for player 2.

```
if('player1Turn' in message and playerType == 'player1'):
    playerTurn = True
    rollButton = Button(gameWindow, text="Roll Dice",
fg='black', font=("Chalkboard SE", 15), bg="grey", command=rollDice,
width=20, height=5)
    rollButton.place(x=screen_width / 2 - 80,
y=screen_height/2 + 250)

    elif('player2Turn' in message and playerType == 'player2'):
```

```
playerTurn = True
rollButton = Button(gameWindow, text="Roll Dice",
fg='black', font=("Chalkboard SE", 15), bg="grey", command=rollDice,
width=20, height=5)
rollButton.place(x=screen_width / 2 - 80,
y=screen_height/2 + 260)
```

What's NEXT?

In the next class, we will be working on file transfer protocol.

Expand Your Knowledge:

Explore more about the creating GUI using Tkinter through this link :

<https://realpython.com/python-gui-tkinter/>