# GAME MECHANICS-I

## What is our GOAL for this CLASS?

In this class, we have written code to display the joining player , display the position of the player by colors .

## What did we ACHIEVE in the class TODAY?

- Wrote code to check the color position of the players
- Wrote functions to define the player moves.

## 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 create multiplayer games using a database where all the records of players were kept in the database.

Today, you added the logic for the movement of player pieces:-

1. Create a **checkColorPosition()** function to get the position of the player piece.

```
def checkColorPosition(boxes, color):
    for box in boxes:
        boxColor = box.cget("bg")
        if(boxColor == color):
            return boxes.index(box)
    return False
```

2. Write a **playerMove1()** to define the movement of the player piece on the player 1 board.

```
def movePlayer1(steps):
    global leftBoxes

    boxPosition = checkColorPosition(leftBoxes[1:],"red")

    if(boxPosition):
        diceValue = steps
        coloredBoxIndex = boxPosition
        totalSteps = 10
        remainingSteps = totalSteps - coloredBoxIndex

        if(steps == remainingSteps):
            for box in leftBoxes[1:]:
                box.configure(bg='white')

            global finishingBox

            finishingBox.configure(bg='red')

            global SERVER
            global playerName

            greetMessage = f'Red wins the game.'
            SERVER.send(greetMessage.encode())

        elif(steps < remainingSteps):
            for box in leftBoxes[1:]:
                box.configure(bg='white')

            nextStep = (coloredBoxIndex + 1 ) + diceValue
            leftBoxes[nextStep].configure(bg='red')
        else:
            print("Move False")
    else:
        # first step
        leftBoxes[steps].configure(bg='red')
```

3. Similarly write **movePlayer2()** function to define movements of player 2 on the board.

```
def movePlayer2(steps):
    global rightBoxes

    # Moving to reverse order
    tempBoxes = rightBoxes[-2::-1]

    boxPosition = checkColorPosition(tempBoxes,"yellow")

    if(boxPosition):
        diceValue = steps
        coloredBoxIndex = boxPosition
        totalSteps = 10
        remainingSteps = totalSteps - coloredBoxIndex

        if(diceValue == remainingSteps):
            for box in rightBoxes[-2::-1]:
                box.configure(bg='white')

            global finishingBox

            finishingBox.configure(bg='yellow', fg="black")

            global SERVER
            global playerName

            greetMessage = f'Yellow wins the game.'
            SERVER.send(greetMessage.encode())

        elif(diceValue < remainingSteps):
            for box in rightBoxes[-2::-1]:
                box.configure(bg='white')

            nextStep = (coloredBoxIndex + 1 ) + diceValue
            rightBoxes[::-1][nextStep].configure(bg='yellow')
        else:
            print("Move False")
    else:
        # first step
        rightBoxes[len(rightBoxes) - (steps+1)].configure(bg='yellow')
```

## What's NEXT?

In the next class,  we will be working on finishing the game in the next class.

## Expand Your Knowledge:

Explore more about the creating  GUI using Tkinter  through this link :
https://realpython.com/python-gui-tkinter/