

```

import random
'''
Author: Lukas Ottenhof Date: Class: AUCSC 235

This clas is for a dice game, it creates the dice object. This class
contains all parts of the game relating to the dice attributes,
randomizing value of the dice, creating the face of the
dice, and a method randomizing the dice attributes to simulate the dice being
rolled. This class is used as the object for the game class.
'''
class TwoDice:
    def __init__(self):
        self.value1 = 0 #value of the dice face
        self.value2 = 0
        self.diceFace1 = [] #this will store the dice fice visual
        self.diceFace2 = []
        self.doubles = False
        self.singleOne = False

        #getting random ints for the dice values simulating dice roll
        self.value1 = random.randint(1, 6)
        self.value2 = random.randint(1, 6)

        #setting initial values
        self.diceFace1 = self.diceRoller(self.value1)
        self.diceFace2 = self.diceRoller(self.value2)
        self.doubles = self.checkDoubles()
        self.singleOne = self.checkSingleOne()

    '''
    this method is called upon during the user and computer turns. It is used
    to randomize our value and create a new dice face to simulate the dice
    being rolled
    '''
    def roll(self): #creates new values for the items below
        self.value1 = random.randint(1,6)
        self.value2 = random.randint(1,6)
        self.diceFace1 = self.diceRoller(self.value1)
        self.diceFace2 = self.diceRoller(self.value2)
        self.doubles = self.checkDoubles()
        self.singleOne = self.checkSingleOne()
        self.print2dDice(self.diceFace1, self.diceFace2)

    '''
    this method is called upon during the user and computer turns. It is used
    to randomize our value and create a new dice face to simulate the dice
    being rolled
    '''
    def diceRoller(self, value):

        #initial array, a blank dice before it is given value
        dice = [
            [" ", "-", "-", "-", "-", "-", "-", "-", "-"],
            ["|", " ", " ", " ", " ", " ", " ", " ", "|"],
            ["|", " ", " ", " ", " ", " ", " ", " ", "|"],
            ["|", " ", " ", " ", " ", " ", " ", " ", "|"],
            ["|", " ", " ", " ", " ", " ", " ", " ", "|"],
            [" ", "-", "-", "-", "-", "-", "-", "-", "-"]
        ]

        match value: #giving the dice marks to show the user the value
            case 1: #each case represents a number on a 6 sided dice
                dice[2][4] = "o"
            case 2:
                dice[1][2] = "o"
                dice[3][6] = "o"
            case 3:

```

```

        dice[1][2] = "o"
        dice[2][4] = "o"
        dice[3][6] = "o"
    case 4:
        dice[1][2] = "o"
        dice[1][6] = "o"
        dice[3][2] = "o"
        dice[3][6] = "o"
    case 5:
        dice[1][2] = "o"
        dice[1][6] = "o"
        dice[2][4] = "o"
        dice[3][2] = "o"
        dice[3][6] = "o"
    case 6:
        dice[1][2] = "o"
        dice[1][6] = "o"
        dice[2][2] = "o"
        dice[2][6] = "o"
        dice[3][2] = "o"
        dice[3][6] = "o"
    case _:
        print("Error!")
    return dice
'''
checking to see if the dice are doubles
'''
def checkDoubles(self):
    return self.value1 == self.value2

'''
checking to see if only one of the dice rolled is a 1
'''
def checkSingleOne(self):
    return self.value2 == 1 and self.value1 != 1 or self.value1 == 1 and self.value2 != 1

'''
this method is used to print the dice side by side. it counts the number
of rows the dice have, then prints the first character in each dice then
moves onto the next row and each character in that row will be printed
'''
def print2dDice(self, dice1, dice2):
    cols = 9
    rows = 4
    while rows >= 0:
        for i in range(cols):
            print(dice1[rows][i], end = " ")

        print(" ", end=" ")

        for i in range(cols):
            print(dice2[rows][i], end = " ")
        print()
        rows -= 1

class DiceGame:
    def __init__(self):
        self.userPoints = 0
        self.computerPoints = 0

    # Call the welcome method on the instance of DiceGame

```

```

self.welcome()

print() # For readability
print("Here we go...")
'''

this while loop is the start of the turns, while the user and
computer haven't reached the winning number of points it will
continue until someone reaches the winning number of points
'''

while self.userPoints <= 60 and self.computerPoints <= 60:

    self.userTurn()
    if self.userPoints <= 60: #this if is implemented so that if the user
        #wins the game will end instead of the computer having another turn
        self.computerTurn()

    #prompts so the user knows who won
    print("TaTaTah Drum rolllllllll")
    if self.userPoints > self.computerPoints:
        print("The winner is: You")

    else:
        print("The winner is: The Computer")

'''

These are the initial welcome prompts. They will be printed at the start
of the game.
'''

def welcome(self):
    print("=====")
    print("                The Dice Game")
    print("Roll the dice, accumulating the total to add "
          + "to your score.")
    print("Hit 60 before the computer and you win!")
    print("If you roll doubles, you get double the value! "
          + "And you must roll again.")
    print("If you roll a one - you are done,")
    print("                unless it's snake eyes!")

'''

this is the method for the users turn, the variable keepPlaying is used
to determine whether the user wants to keep playing, it will start as y
so the user will play at least once. If it changes the turn will end.
This method takes the object from main as well as the users and computers
points. The users points will be added to then returned so that the value
of users points will be updated in main. computer points is taken so that
the score of each player can be displayed properly at the end of the turn
'''

def userTurn(self):

    keepPlaying = ['y'];#while this is equal to y the user turn
    #will continue
    userInput = "" #declaring a variable that will store the users input

    print("-----")
    print("Your turn")
    while keepPlaying[0] == 'y':#while user wants to
        #keep playing and hasnt rolled a single 1 loop will continue
        dice = TwoDice()

```

```

print("Rolling...")
dice.roll()#created new values for our object so simulate the
#rolling of dice
print()

if dice.doubles:#if the value of each dice is
    #the same, add the points multiplied by 2 and roll again
    self.userPoints += ((dice.value1 * 2) + (dice.value2 * 2))
    print()#help with output readability
    print("Doubles! Roll again!")

#this else if statement will end the users turn and take away all
#of their points if they roll only a single one

elif dice.singleOne:#if the checker says there is a single 1

    self.lostAll(self.userPoints);#uses lost all passing along
    #the users points
    self.userPoints = 0
    keepPlaying[0] = 'n' #this ends the turn

#if there is no special condition due to the values of the dice
#the users points will be added to their total and they will have
#the option to play again
else:
    self.userPoints += dice.value1 + dice.value2;#adds users
    if self.userPoints > 60:
        return
    else:
        userInput = input(f"Roll Again? (current score is: {self.userPoints})
Enter 'y' for yes 'n' for no:")

    #the following 2 lines makes it so the user can enter any word
    #starting with y and the turn will continue. it does this by
    #making the users input all lowercase and taking the first
    #letter inputed
    userInput = userInput.lower()
    keepPlaying[0] = userInput[0]

    if keepPlaying[0] != 'y':
        print("Staying")

    #if keepPlaying does not equal yes then end

print(f"Score: Player: {self.userPoints}; Computer: {self.computerPoints}")

...

this is the method for the computers turn, the variable keepPlaying is
used to determine whether the computer will keep playing, while it is
equal to 0 or 1 the computer will keep playing and if it hits 2 the turn
will end. This variable is determined by randomly picking an int between
0 and 2, giving is a 1/3 chance at staying. This method takes the object
from main as well as the users and computers points. The computer points
will be added to then returned so that the value of computerPoints will
be updated in main. the users points is taken so that the score of each
player can be displayed properly at the end of the turn
'''

def computerTurn(self):

```

```

rand = random.randint(1,6)#used to pick if the
#computer will stay or roll again
dice = TwoDice()
keepPlaying = 0; #if this int is equal to 1 or 0 it will keep
#rolling, it if is equal to 3 its turn will end
#giving it a 2/3 change of continuing
print("-----")
print("Computer's turn")

while keepPlaying == 0 or keepPlaying == 1:

    #will continue
    print("Rolling...")
    dice.roll()#creating new values for the object
    print()

    if dice.doubles:#if the value of each dice is
        #the same, add the points multiplied by 2 and roll again
        print()
        print("Doubles! Roll again!")
        self.computerPoints += ((dice.value1 * 2) + (dice.value2 * 2))

    #this if statement will end the users turn and take away all of
    #their points if they roll only a single one

    elif dice.singleOne: #if the checker says there is a single 1

        self.lostAll(self.computerPoints)#calls last all passing along
        #the computers points
        self.computerPoints = 0
        keepPlaying = 2#makeing the loops condition false to end the
        #turn

    else:
        #if there is no special condition due to the values of the dice
        #the coumputers points will be added to their total.
        self.computerPoints += dice.value1 + dice.value2
        if self.computerPoints > 60:
            return
        else:
            keepPlaying = random.randint(0, 2)#randomizes keep playing
            #if when keepPlaying2 was randomized and it came out to
            #equal 2 the computer will stay
            if keepPlaying == 2:
                print("Staying")

    #to display scores
    print(f"Score: Player: {self.userPoints}; Computer: {self.computerPoints}")

'''
this is for when a player rolls a single 1. it takes the points of the
player that lost and displays prompts to let the user know what happened
'''

def lostAll(self, pointsLost):
    print()
    print("OH NO...You lost it all!")
    print(f" You lost: {pointsLost}")

# uses the DiceGame class to run the game
DiceGame()

```