**Games.py – part 2**

The second game we are going to create is rock paper scissors. This version is similar to the version we made several weeks ago with a few changes and sophistications.

First let’s make a new file called simply ‘rps.py’. There are 4 functions to build in this file so let’s lay out the skeleton and I’ll explain each one in turn.

import random

def choice\_menu\_ui():

def choice():

def cpu\_choice():

def rock\_paper\_scissors():

We import the ‘random’ module since we need the computer to make a random choice between the three option. Then we have ‘choice\_menu\_ui()’ dedicated to outputting the choices available to the player, much like ‘difficulty\_menu\_ui()’ in the previous game. ‘choice()’ handles the logic for getting the player’s choice and ‘cpu\_choice()’ does the same for the computer. In ‘rock\_paper\_scissors()’ we compare the two choices and determine if the player won, lost or drew against the computer before sending the player back to the top menu, which we will build in part 3.

We’ll get ‘choice\_menu\_ui()’ out of the way first. Again we’ll use a string literal to print out the menu.

return '''

----------------------------------

Please select an option

Enter a number to select an option

----------------------------------

1 - Rock

2 - Paper

3 - Scissors

----------------------------------

'''

‘choice()’ is actually very similar to ‘difficulty()’ in the last example. Once we have printed out the menu we’ll enter an infinite loop that will only end once the correct input is entered. We’ll also use a ‘try except’ block to handle incorrect data types.

print(choice\_menu\_ui())

while(True):

# Handle invalid input type

try:

user\_input = int(input("-> "))

except:

print("Invalid input, please try again")

continue

We want the player’s input to be a number between 1 and 3, exactly like ‘difficulty()’. In fact the only difference between the two functions is that here we are returning a string and not a number. While you might be tempted to copy and paste I’d still recommend typing this out all over again to get the code under your fingers.

# If input is not 1-3 display error

if((user\_input < 1) or (user\_input > 3)):

print("Invalid input, please try again")

else:

if(user\_input == 1):

print("Rock selected\n")

return "rock"

elif(user\_input == 2):

print("Paper selected\n")

return "paper"

else:

print("Scissors selected\n")

return "scissors"

Getting the computer’s choice is relatively simple, but we also need to assign the random number generated to one of the three possible choices. First we need to get our random number, so in ‘cpu\_choice()’ type the following.

computer\_number = random.randint(0, 2)

Here we use the ‘randint’ function from the ‘random’ module to get a number from 0 to 2. With this we can assign a choice to the generated number.

# Assign a choice depending on the number

if(computer\_number == 0):

return "rock"

elif(computer\_number == 1):

return "paper"

else:

return "scissors"

This is much like the ‘if’ statement presented above but we are only returning a string.

Now we can complete our game by determining the winner. First we need to use the functions we just built to get each player’s choices, so in ‘rock\_paper\_scissors()’ type the following.

player\_choice = choice()

computer\_choice = cpu\_choice()

Now we have each player’s choice assigned to their own variables. Now before we print out the winner it would be a good idea to print out what the choices were.

# Output each player's choices before we show the winner

print("\nPlayer: {}\nComputer: {}".format(player\_choice, computer\_choice))

Now it’s a case of going through each condition. First we handle a draw.

# If both players choose the same, it's a draw

if(player\_choice == computer\_choice):

print("The result is a draw")

The next case is if the player chose a rock.

# If the player chose rock

elif(player\_choice == "rock"):

if(computer\_choice == "paper"):

print("The computer wins!")

else:

print("The player wins!")

If the player chose a rock, we enter another ‘if’ statement asking what the computer chose. Since we’ve already taken care of the draw condition then there are two possibilities for the computer’s choice: paper or scissors, and a winner can be determined accordingly. We’ll do the same thing for the other two conditions.

# If the player chose paper

elif(player\_choice == "paper"):

if(computer\_choice == "scissors"):

print("The computer wins!")

else:

print("The player wins!")

# If the player chose scissors

elif(player\_choice == "scissors"):

if(computer\_choice == "rock"):

print("The computer wins!")

else:

print("The player wins!")

And at the very end we add the following to take us back to the top menu which we will build next.

return 0

You can test this game by adding a function call to ‘rock\_paper\_scissors()’ at the very bottom. Remember to remove all indentation otherwise python will think it’s inside another function.

We are now ready to complete the application in part 3.