"""  
@ Clever Programmer  
Rock paper scissors  
"""  
import random  
import simplegui  
  
  
# Global variables that all functions know about.  
# DO NOT EDIT THESE GLOBAL VARIABLES  
# OR YOUR GAME WILL BREAK.  
COMPUTER\_SCORE = 0  
HUMAN\_SCORE = 0  
human\_choice = ""  
computer\_choice = ""  
  
  
def choice\_to\_number(choice):  
 """Convert choice to number."""  
 # If choice is 'rock', give me 0  
 # If choice is 'paper', give me 1  
 # If choice is 'scissors', give me 2  
 rps\_dic = {'rock': 0, 'paper': 1, 'scissors': 2}  
 return rps\_dic [choice]  
  
  
def number\_to\_choice(number):  
 """Convert number to choice."""  
 # If number is 0, give me 'rock'  
 # If number is 1, give me 'paper'  
 # If number is 2, give me 'scissors'  
 rps\_dic = {0: 'rock', 1: 'paper', 2: 'scissors'}  
 return rps\_dic[number]  
  
  
def random\_computer\_choice():  
 """Choose randomly for computer."""  
   
 # lookup random.choice()  
 return random.choice(['rock', 'paper', 'scissors'])  
  
def choice\_result(human\_choice, computer\_choice):  
 """Return the result of who wins."""  
   
 # DO NOT REMOVE THESE GLOBAL VARIABLE LINES.  
 global COMPUTER\_SCORE  
 global HUMAN\_SCORE  
   
 # based on the given human\_choice and computer\_choice  
 # determine who won and increment their score by 1.  
 # if tie, then don't increment anyone's score.  
   
 # example code  
 # if human\_choice == 'rock' and computer\_choice == 'paper':  
 # COMPUTER\_SCORE = COMPUTER\_SCORE + 1  
human\_number = choice\_to\_number(human\_choice)  
computer\_number = choice\_to\_number(computer\_choice)  
if (human\_number - computer\_number) % 3 == 1:  
 COMPUTER\_SCORE += 1  
elif human\_number == computer\_number:  
 print('Tie')  
else:  
 HUMAN\_SCORE += 1  
   
# DO NOT REMOVE THESE TEST FUNCTIONS.  
# They will test your code.  
def test\_choice\_to\_number():  
 assert choice\_to\_number('rock') == 0  
 assert choice\_to\_number('paper') == 1  
 assert choice\_to\_number('scissors') == 2  
   
def test\_number\_to\_choice():  
 assert number\_to\_choice(0) == 'rock'  
 assert number\_to\_choice(1) == 'paper'  
 assert number\_to\_choice(2) == 'scissors'  
   
def test\_all():  
 test\_choice\_to\_number()  
 test\_number\_to\_choice()  
  
# Uncomment to test your functions.  
test\_all()  
  
  
# Handler for mouse click on rock button.  
# This code is for the GUI part of the game.  
def rock():  
 global human\_choice, computer\_choice  
 global HUMAN\_SCORE, COMPUTER\_SCORE  
   
 human\_choice = 'rock'  
 computer\_choice = random\_computer\_choice()  
 choice\_result(computer\_choice, human\_choice)  
  
def paper():  
 global human\_choice, computer\_choice  
 global HUMAN\_SCORE, COMPUTER\_SCORE  
   
 human\_choice = 'paper'  
 computer\_choice = random\_computer\_choice()  
 choice\_result(computer\_choice, human\_choice)  
   
# Handler for mouse click on paper button.  
def scissors():  
 global human\_choice, computer\_choice  
 global HUMAN\_SCORE, COMPUTER\_SCORE  
   
 human\_choice = 'scissors'  
 computer\_choice = random\_computer\_choice()  
 choice\_result(computer\_choice, human\_choice)  
  
# Handler to draw on canvas  
def draw(canvas):  
   
 try:  
 # Draw choices  
 canvas.draw\_text("You: " + human\_choice, [10,40], 48, "Green")  
 canvas.draw\_text("Comp: " + computer\_choice, [10,80], 48, "Red")  
   
 # Draw scores  
 canvas.draw\_text("Human Score: " + str(HUMAN\_SCORE), [10,150], 30, "Green")  
 canvas.draw\_text("Comp Score: " + str(COMPUTER\_SCORE), [10,190], 30, "Red")  
   
 except TypeError:  
 pass  
   
  
# Create a frame and assign callbacks to event handlers  
def play\_rps():  
 frame = simplegui.create\_frame("Home", 300, 200)  
 frame.add\_button("Rock", rock)  
 frame.add\_button("Paper", paper)  
 frame.add\_button("Scissors", scissors)  
 frame.set\_draw\_handler(draw)  
  
 # Start the frame animation  
 frame.start()  
   
play\_rps()  
  
"""   
[0, 1, 2]  
[r, p, s]  
  
rock vs scissors  
(rock - scissors) % 3 == 1  
(0 - 2) % 3 == 1  
(-2) % 3 == 1  
1 == 1  
rock wins  
  
paper vs scissors  
(paper - scissors) % 3 == 1  
(1 - 2) % 3 == 1  
(-1) % 3 == 1  
2 == 1  
scissor wins  
  
paper vs rocks  
(paper - rocks) % 3 == 1  
(1 - 0) % 3 == 1  
1 % 3 == 1  
1 == 1  
paper wins  
"""