

CS1400 Assignment 2: The Game of Pig

You should pass each part of this assignment off to a lab assistant before the end of the day on each due date.

In addition, each part must be submitted electronically before 11:55 pm on the due date. Use the assignment submission system discussed in class to submit your homework.

If necessary, your instructor will give you more detailed instructions for passing off and submitting your work.

Objectives

- Create a program in a development environment
- Use conditional control structures
- Use looping control structures
- Use variables to track information
- Define and use functions

Assignment

In the Game of Pig, two opposing players attempt win by being the player with the most *total points* after at least one player exceeds 100 total points.

At the beginning of the game, both players have 0 *total points*. In each round, player 1 takes a turn, and player 2 takes a turn. The game ends if at the end of a round, at least one of the players has 100 or more points. The winner is the player with the most *total points*.

The player begins a turn with 0 *turn points*. The player rolls a six sided die. If the die roll produces a 1, then the player's *turn points* are removed, and their turn is over. If any other number is rolled (2-6), the roll is added to the player's *turn points*. The player now chooses whether to quit the turn, or continue the turn. If the player quits the turn, their *turn points* are added to their *total points*. If the player continues, then the player rolls the die again, and follows the above outline based on the value rolled.

In this assignment, you will create a computer program that will allow two users to play the Game of Pig against each other.

If you have questions, discuss them with your instructor.

Part 1

We will break the assignment into two parts. Each part will involve the creation of one or two functions. When working together, the functions will create the full behavior of the game. Breaking a problem into smaller pieces to solve a bit at a time is call *decomposition*.

Rolling the die

To create a game that uses a die, you will need some method of creating random numbers, to simulate a die roll. Call your function `roll_die`. It should receive an integer as a formal-parameter which tells it the number of sides on the die. It should generate a random number from 1 to the number of sides. For example, a six-sided die can only produce the integers 1 through 6, inclusive. Then it should return the randomly generated number.

Random Numbers

To generate random numbers in a program, we use mathematical techniques to generate numbers that appear to be random. Lucky for you, the mathematical techniques have already been written into a function.

To use the function, you need to add an extra line at the top of your program file:

```
import random
```

Then, in any part of your program you can use the following code to generate a random number from 1 to 6.

```
x = random.randrange(1,7)
```

Yes, we put 1, 7 as the parameters to create numbers from 1 to 6 inclusive. You should try this function out with various parameters to observe its behavior. Remember that each call only returns one value. But the value is randomly chosen from the specified range.

Finally, your code should look like this after implementing the `roll_die` function:

```
import random

def roll_die(sides):
    r = random.randrange(1, sides+1)
    return r
```

For more on random numbers, see section 9.2 (pages 268-270) of the text book. You can also read the [online python documentation](#).

After you create your `roll_die` function, test it to make sure it is working correctly. For example, if you roll the die many times, do all of the 6 numbers show up? How would you test it?

Taking a turn

If necessary, re-read the discription of a single player's turn from above.

Create a function `take_turn` to implement ability of a single player to take a single turn. The function should receive an integer for a formal-parameter that identifies the player number. It should return the number of *turn points* aquired by the player.

Below is a pseudo-code description of the algorithm you should use to implement the `take_turn` function.

- set turn points to 0
- set keep rolling flag to 1
- print beginning of turn message for player
- wait for the enter key to be pressed
- while the user can and wants to keep rolling
 - roll the six-side die
 - tell the user the value of the roll
 - if the roll was a 1
 - set turn points to 0
 - set keep rolling flag to 0
 - else (the roll was not a 1)
 - add the value of the roll to turn points
 - tell the user the current number of turn points
 - ask the user if they want to continue
 - if yes
 - set keep rolling flag to 1
 - otherwise (no)
 - set keep rolling flag to 0
- print end of turn message
- return the number of turn points

As you create your `take_turn` function, you should test it to be sure it behaves correctly. Track the turn points manually to be sure that the function is added up points correctly.

How will you test your function?

Below is an example output from the `take_turn` function for player 1.

```
=====
Player 1  press enter to begin your turn.

You rolled a 4
Your turn points are now 4 .
Continue rolling (0=no,1=yes)? 1

You rolled a 3
Your turn points are now 7 .
Continue rolling (0=no,1=yes)? 1

You rolled a 1
Turn over
```

```
=====
```

This is an example for player 2.

```
=====
Player 2  press enter to begin your turn.

You rolled a 3
Your turn points are now 3 .
Continue rolling (0=no,1=yes)? 1

You rolled a 3
Your turn points are now 6 .
Continue rolling (0=no,1=yes)? 1

You rolled a 5
Your turn points are now 11 .
Continue rolling (0=no,1=yes)? 0

Turn over
=====
```

In the file that contains the definitions of your `roll_die` and `take_turn` functions, add the following code:

```
for i in range(10):
    print roll_die(4)

print

take_turn(1)
take_turn(2)
```

Run this program to make sure it behaves as expected. Fix any problems.

When this is working, pass off this part by showing your code to the lab assistant, and demonstrating the running program. Finally, submit this file according to your instructor's instructions.

Congratulations! In part 2 you will finish your program.

Part 2

Giving instructions

Create a function called `show_instructions` that prints out a description of how to play the game for the users. This function does not receive any parameters. It does not return any values.

This is an example of the instructions that may be printed by your function:

```
Welcome to the Game of Pig.  To win, be the
player with the most points at the end of the
game.  The game ends at the end of a round where
at least one player has 100 or more points.

On each turn, you may roll the die as many times
as you like to obtain more points.  However, if
you roll a 1, your turn is over, and you do not
obtain any points that turn.
```

The main game control

Create a function called `main` that controls the whole game. This function receives no parameters and returns no values.

Below is a pseudo-code description of the algorithm you should use to implement the `main` function.

- show the instructions
- set player 1's total points to 0
- set player 2's total points to 0
- while player 1's total points are less than 100 and player 2's total points are less than 100
 - show the total points for each player

- have player 1 take a turn
- add player 1's turn points to their total points
- show the total points for each player
- have player 2 take a turn
- add player 2's turn points to their total points
- display the end of game message
- show the total points for each player
- if player 1 has more total points than player 2
 - show that player 1 is the winner
- else if player 2 has more total points than player 1
 - show that player 2 is the winner
- otherwise
 - show that the game is a tie

See the sample output for a full game at the end of this document.

Finally, add a call to `main` at the end of your program file to have your program run.

Run the program to make sure it behaves as expected. Fix any problems.

When this is working, pass off the assignment by showing your code to the lab assistant, and demonstrating the running program. Finally, submit this file according to your instructor's instructions.

Congratulations! You're done.

Sample output of a full game:

```

Welcome to the Game of Pig.  To win, be the
player with the most points at the end of the
game.  The game ends at the end of a round where
at least one player has 100 or more points.

On each turn, you may roll the die as many times
as you like to obtain more points.  However, if
you roll a 1, your turn is over, and you do not
obtain any points that turn.

Player 1 has 0 points.
Player 2 has 0 points.
=====
Player 1  press enter to begin your turn.

You rolled a 5
Your turn points are now 5 .
Continue rolling (0=no,1=yes)? 1

You rolled a 5
Your turn points are now 10 .
Continue rolling (0=no,1=yes)? 1

You rolled a 6
Your turn points are now 16 .
Continue rolling (0=no,1=yes)? 1

You rolled a 3
Your turn points are now 19 .
Continue rolling (0=no,1=yes)? 2

Turn over
=====

Player 1 has 19 points.
Player 2 has 0 points.
=====
Player 2  press enter to begin your turn.

You rolled a 3
Your turn points are now 3 .
Continue rolling (0=no,1=yes)? 1

```

You rolled a 1
Turn over
=====

Player 1 has 19 points.
Player 2 has 0 points.
=====

Player 1 press enter to begin your turn.

You rolled a 6
Your turn points are now 6 .
Continue rolling (0=no,1=yes)? 1

You rolled a 4
Your turn points are now 10 .
Continue rolling (0=no,1=yes)? 1

You rolled a 4
Your turn points are now 14 .
Continue rolling (0=no,1=yes)? 1

You rolled a 5
Your turn points are now 19 .
Continue rolling (0=no,1=yes)? 2

Turn over
=====

Player 1 has 38 points.
Player 2 has 0 points.
=====

Player 2 press enter to begin your turn.

You rolled a 2
Your turn points are now 2 .
Continue rolling (0=no,1=yes)? 1

You rolled a 3
Your turn points are now 5 .
Continue rolling (0=no,1=yes)? 1

You rolled a 2
Your turn points are now 7 .
Continue rolling (0=no,1=yes)? 1

You rolled a 6
Your turn points are now 13 .
Continue rolling (0=no,1=yes)? 1

You rolled a 4
Your turn points are now 17 .
Continue rolling (0=no,1=yes)? 1

You rolled a 1
Turn over
=====

Player 1 has 38 points.
Player 2 has 0 points.
=====

Player 1 press enter to begin your turn.

You rolled a 3
Your turn points are now 3 .
Continue rolling (0=no,1=yes)? 1

You rolled a 4
Your turn points are now 7 .
Continue rolling (0=no,1=yes)? 1

You rolled a 1

Turn over

=====
Player 1 has 38 points.
Player 2 has 0 points.

=====
Player 2 press enter to begin your turn.

You rolled a 3
Your turn points are now 3 .
Continue rolling (0=no,1=yes)? 1

You rolled a 6
Your turn points are now 9 .
Continue rolling (0=no,1=yes)? 1

You rolled a 4
Your turn points are now 13 .
Continue rolling (0=no,1=yes)? 1

You rolled a 2
Your turn points are now 15 .
Continue rolling (0=no,1=yes)? 1

You rolled a 1
Turn over

=====
Player 1 has 38 points.
Player 2 has 0 points.

=====
Player 1 press enter to begin your turn.

You rolled a 3
Your turn points are now 3 .
Continue rolling (0=no,1=yes)? 1

You rolled a 5
Your turn points are now 8 .
Continue rolling (0=no,1=yes)? 1

You rolled a 6
Your turn points are now 14 .
Continue rolling (0=no,1=yes)? 1

You rolled a 6
Your turn points are now 20 .
Continue rolling (0=no,1=yes)? 1

You rolled a 2
Your turn points are now 22 .
Continue rolling (0=no,1=yes)? 1

You rolled a 5
Your turn points are now 27 .
Continue rolling (0=no,1=yes)? 2

Turn over

=====
Player 1 has 65 points.
Player 2 has 0 points.

=====
Player 2 press enter to begin your turn.

You rolled a 6
Your turn points are now 6 .
Continue rolling (0=no,1=yes)? 1

You rolled a 3
Your turn points are now 9 .

Continue rolling (0=no,1=yes)? 1

You rolled a 1
Turn over

=====

Player 1 has 65 points.
Player 2 has 0 points.

=====

Player 1 press enter to begin your turn.
1

You rolled a 3
Your turn points are now 3 .
Continue rolling (0=no,1=yes)? 1

You rolled a 2
Your turn points are now 5 .
Continue rolling (0=no,1=yes)? 1

You rolled a 3
Your turn points are now 8 .
Continue rolling (0=no,1=yes)? 1

You rolled a 4
Your turn points are now 12 .
Continue rolling (0=no,1=yes)? 1

You rolled a 4
Your turn points are now 16 .
Continue rolling (0=no,1=yes)? 2

Turn over

=====

Player 1 has 81 points.
Player 2 has 0 points.

=====

Player 2 press enter to begin your turn.

You rolled a 1
Turn over

=====

Player 1 has 81 points.
Player 2 has 0 points.

=====

Player 1 press enter to begin your turn.
1

You rolled a 3
Your turn points are now 3 .
Continue rolling (0=no,1=yes)? 1

You rolled a 6
Your turn points are now 9 .
Continue rolling (0=no,1=yes)? 1

You rolled a 3
Your turn points are now 12 .
Continue rolling (0=no,1=yes)? 1

You rolled a 3
Your turn points are now 15 .
Continue rolling (0=no,1=yes)? 1

You rolled a 5
Your turn points are now 20 .
Continue rolling (0=no,1=yes)? 1

You rolled a 4
Your turn points are now 24 .
Continue rolling (0=no,1=yes)? 1

You rolled a 4
Your turn points are now 28 .
Continue rolling (0=no,1=yes)? 2

Turn over
=====

Player 1 has 109 points.
Player 2 has 0 points.
=====

Player 2 press enter to begin your turn.

1
You rolled a 3
Your turn points are now 3 .
Continue rolling (0=no,1=yes)? 1

You rolled a 3
Your turn points are now 6 .
Continue rolling (0=no,1=yes)? 1

You rolled a 4
Your turn points are now 10 .
Continue rolling (0=no,1=yes)? 1

You rolled a 2
Your turn points are now 12 .
Continue rolling (0=no,1=yes)? 1

You rolled a 6
Your turn points are now 18 .
Continue rolling (0=no,1=yes)? 1

You rolled a 1
Turn over
=====

The game is over.
Player 1 has 109 points.
Player 2 has 0 points.
Player 1 wins.

[Sample Assignment for High Low Game](#)