CptS 111 Introduction to Algorithmic Problem Solving (http://piazza.com/wsu/fall2016/cpts111/home)

Washington State University (https://wsu.edu)

Gina Sprint (http://eecs.wsu.edu/~gsprint/)

PA4 Conditionals (100 pts)

Due Monday, October 24 at Midnight.

Learner Objectives

At the conclusion of this programming assignment, participants should be able to:

- Implement `while` loops

Prerequisites

Before starting this programming assignment, participants should be able to:

- Get input from the user
- Display output to the user
- Define and call functions
- Implement `if` statements

Acknowledgments

Content used in this assignment is based upon information in the following sources:

- Andy O'Fallon's (http://www.eecs.wsu.edu/~aofallon/) CptS 121 Craps Assignment

Overview and Requirements

Note: This assignment is adapted from a programming assignment by Andy O'Fallon.

Write a program to play the dice game "Craps" (craps.py). Craps is one of the most popular games of chance and is played in casinos and back alleys throughout the world. The rules of the game are straightforward:

A player rolls two dice. Each die has six faces. These faces contain 1, 2, 3, 4, 5, and 6 spots. After the dice have come to rest, the sum of the spots on the two upward faces is calculated. If the sum is 7 or 11 on the first throw, the player wins. If the sum is 2, 3, or 12 on the first throw (called "craps"), the player loses (i.e. the "house" wins). If the sum is 4, 5, 6, 8, 9, or 10 on the first throw, then the sum becomes the player's "point." To win, you must continue rolling the dice until you "make your point." The player loses by rolling a 7 before making the point.

Program Details

Write a program that implements craps according to the above rules.

Wagering

Your program should allow for wagering. This means that you need to prompt the user for an initial bank balance from which wagers will be added or subtracted. Before each game (i.e. before the first roll) prompt the user for a wager. Once a game is lost or won, the bank balance should be adjusted. **Wagers must be less than or equal to the available bank balance.**

Note: You only need to prompt for a wager before each game (i.e. the first roll, not subsequent "point" rolls). You do not need to prompt for a wager before each roll when the player is trying to make their point. This is bonus (see below)!

Main Game Loop

When a game ends and the bank balance is adjusted, prompt the user for whether they would like to keep playing craps or quit. If the user runs out of money but would like to keep playing, allow the user to add more money to their bank balance.

Functions to Define

At a minimum, your solution must contain the functions below. These functions will help you get started! You are encouraged to define more than the ones listed as you wish.

1. (5 pts) `display_game_rules()`: Prints out the rules of the game of "craps".
2. (5 pts) `get_bank_balance()`: Prompts the player for an initial bank balance from which wagering will be added or subtracted. The player entered bank balance (in dollars, i.e. $100.00) is returned.
3. (5 pts) `get_wager_amount()`: Prompts the player for a wager on a particular roll. The wager is returned.
4. (5 pts) `is_valid_wager_amount(wager, balance)`: Checks to see if the wager is within the limits of the player's available balance. This function is a *predicate* function:
   - If the wager exceeds the player's available balance, then `False` is returned; otherwise `True` is returned.
5. (5 pts) `roll_die()`: Rolls one die. This function should randomly generate a value between 1 and 6, inclusively. Returns the value of the die.
6. (5 pts) `calculate_sum_dice(die1_value, die2_value)`: Sums together the values of the two dice and returns the result.
   - Note: this result may become the player's point in future rolls.
7. (10 pts) `is_win_loss_or_point(sum_dice)`: Determines the result of the first dice roll.
   - If the sum is 7 or 11 on the roll, the player wins and "win" is returned.
   - If the sum is 2, 3, or 12 on the first throw (called "craps"), the player loses (i.e. the "house" wins) and "loss" is returned.
   - If the sum is 4, 5, 6, 8, 9, or 10 on the first throw, then the sum becomes the player's "point" and "point" is returned.

8. (10 pts) `is_point_loss_or_neither(sum_dice, point_value)`: Determines the result of any successive roll after the first roll.
    - If `sum_dice` is the `point_value`, then "point" is returned.
    - If `sum_dice` is a 7, then "loss" is returned
    - Otherwise, "neither" is returned.
9. (30 pts) `main()`: Runs the main game loop and makes use of the above functions in order to play the game of craps as explained above.

Note: you will most likely define more functions than the ones listed above.

Chatter Messages

As the game progresses, print various "chatter" messages based on the number of rolls taken so far by the player, the current balance, and whether or not the player just won his roll such as, "Sorry, you busted!", or "Oh, you're going for broke, huh?", or "Aw c'mon, take a chance!", or "You're up big, now is the time to cash in your chips!"

Use function(s) to do this!

Bonus (5 pts)

Prompt for a wager before each roll, even when the player is trying to make their point. Remember, **wagers must be less than or equal to the available bank balance.** The available bank balance is based on the accumulation of the previous wagers this game.

Submitting Assignments

1. Use the Blackboard tool https://learn.wsu.edu (https://learn.wsu.edu) to submit your assignment to your TA. You will submit your code to the corresponding programming assignment under the "Content" tab. You must upload your solutions as `<your last name>_pa4.zip` by the due date and time.
2. Your .zip file should contain your .py file.

Grading Guidelines

This assignment is worth 100 points + 5 points bonus. Your assignment will be evaluated based on a successful compilation and adherence to the program requirements. We will grade according to the following criteria:

- 80 pts for adherence to functional decomposition stated above (see the individual points above).
- 10 pts for defining function(s) to display chatter messages based on the current state of the game.
- 10 pts for adherence to proper programming style established for the class and comments.