Assignment 1: Yatzy!

**Scope:** 1hp

**Subjects to Cover:** Syntax, built-in types, built-in data structures, functions, list comprehension, lambdas, iterators, generators

**Estimated preparation time:**15 hours (Lectures, labs, exercises, reading)

**Estimated implementation time:**10 hours

Description

Yatzy is a popular dice game that can be played either alone or with friends. The game is played over 15 turns, each with different rules, or categories, for scoring. Each turn is played in the same way, starting with the first player rolling the five available dice. The player then chose any number of dice to keep in such a way that they maximize their chance for a high score given this turns scoring rules, and re-roll the rest. This process is repeated once for up to a total of three rolls and two keeps. Once the player either keep all the dice or has rolled three times their score for this turn is noted down on the score pad and the next player play their turn. Should the player fail to fulfill the combination of dice needed for the current category, a score of 0 is achieved for that category. When all players have played the turn, the first player starts the next turn. This is done until all turns has been played, after which each players' total score is calculated. The player with the highest total score win.

Opportunity to test out Yatzy!

During the lab session Thursday 5/9, there will opportunity to play Yatzy in H320. This opportunity is provided to give those students interested a session to learn and understand the rule set of Yatzy!. It is strongly recommended to use this session to get an understanding of the game and how it can be divided into sub tasks that can later be used when you sit down to program the assignment. Students that wish to test Yatzy! will be divided into groups and play together. Take this opportunity to discuss and help each other understand the problem space.

For those wishing to use the lab session for programming, H321 will be open for the ordinary labs.

Categories

The fifteen categories are, in order:

**Ones:** Try to get as many ones as possible. The points given is equal to the number of dice showing one.  
**Twos:** Try to get as many twos as possible. The points given is equal to 2 \* the number of dice showing two.  
**Threes:** Try to get as many threes as possible. The points given is equal to 3 \* the number of dice showing three.  
**Fours:** Try to get as many fours as possible. The points given is equal to 4 \* the number of dice showing four.  
**Fives:** Try to get as many fives as possible. The points given is equal to 5 \* the number of dice showing five.  
**Sixes:** Try to get as many sixes as possible. The points given is equal to 6 \* the number of dice showing six.  
**Pair:** Try to get a pair of any number. The points given is equal to the two dice added together.  
**Two Pairs:** Try to get two distinct pairs. The score is equal to the four dice added together.  
**Three of a kind:** Try to get three dice showing the same number . The score is equal to the three dice added together.  
**Four of a kind:** Try to get four dice showing the same number . The score is equal to the four dice added together.  
**Small straight:** Try to get the sequence 1, 2, 3, 4, 5 across the dice. Awards 15 points.  
**Large straight:** Try to get the sequence 2, 3, 4 ,5, 6 across the dice. Awards 20 points.  
**Full house:** Try to get one Three of a kind and one Pair. The score is equal to all dice added together.  
**Chance:** Try to get the dice to show as high of a total as possible. Awards the total as points.  
**Yatzy:** Try to get all dice showing the same number. Awards 50 points.

Implementation Details

* The scoring pad is to be implemented with a dictionary containing each category and a score associated to the category.
* The die is to be implemented using a generator.
* Each category is to be implemented using a separate function (one function/category)
* The player is to be implemented using a named tuple containing the player’s name, scoring pad, and the five dice they are to use.
* The game is to be implemented according to the following flow:
  1. Players are added through input by the user until a user input ‘start’
  2. The order of the players is shuffled
  3. Until all categories on the scoring pad are filled:
     + Present the current category (as ordered above)
     + Roll the player’s dice
     + Let the player chose what dice to keep
     + Let the player reroll if not all dice are kept
     + Let the player chose what dice to keep
     + Let the player reroll if not all dice are kept
     + Calculate the score according to the rules of the category
       - If more than one choice can be made, automatically chose the highest scoring choice
       - If no choice can be made, put 0 as score
     + Repeat for each player
  4. Calculate the total score for all players
  5. Sort the players in descending order, presenting the winner first and the player with least point last