Project #3: The Game of Pig

Overview

Write a program that will simulate the dice game *Pig* for 2 players.

Background: Game Rules

The rules to the (dice) game of Pig:

- You will need 2 dice.
- To Play:
    a. The players each take turns rolling two die.
    b. A player scores the sum of the two dice thrown (unless the roll contains a 1): If a single number 1 is thrown on either die, the score for that whole turn is lost (referred to as "Pigged Out"). A 1 on both dice is scored as 25.
    c. During a single turn, a player may roll the dice as many times as they desire. The score for a single turn is the sum of the individual scores for each dice roll.
    d. The first player to reach the goal score wins unless a player scores higher subsequently in the same round. Therefore, everyone in the game must have the same number of turns.

Execution and User Input

This program is quite interactive with the user(s) and will take in the following information; please review the sample input / output sessions for details; we describe them again here emphasizing input.

The program will prompt for the number of points the game will be played to (1-100).

Once a valid goal score has been entered the game will "roll" the dice for the first player by randomly selecting the valid values 1 thru 6 for each of the two dice. The score will be displayed and the player will be prompted to continue their turn by rolling the die again (unless they have "pigged out" by rolling a single 1).

The player will have the option of continuing or ending the turn by entering 'y' for Yes.

When Player 1 completes their turn, Player 2 will engage the exact same way.

At the conclusion of both player's turns the total score for each will be compared to the goal score to determine if the game has ended. If the goal score has *not* been reached, then the game continues with both players taking their turn in sequence. Once one or both of the players reaches the goal score the game ends and the results are displayed. The game may end in a tie.

At the conclusion of a game, the program will prompt whether a new game shall be played: 'y' for Yes.

Requirements

- The name of the Java Project must be `Project3`.
- Generally, user input must not be case sensitive.
- The project must consist of the following classes.
    - Class `EntryPoint`: contains the `main` method and instantiation of the `GameController` class; this class has been provided.
    - Class `GameController`: Manages the flow of 2-player game play.

        - The players must have the option to play as many games as desired.
        - The user must supply a valid maximum score.
        - After the goal score is reached by at least one player, the winner and winning score must be displayed.
        - Will show total scores after each round of rolls.
    - Class `PigDice`: Holds the state of a set of dice for a player.
        - Scoring must be correct for all combinations of dice.
    - Class `Die`: Represents a standard, fair, 6-sided die.
        - Simulates a die being rolled.
        - The face value of the die can be observed without rolling.

Specifications

`Die` class

- Provides all functionality for random number generation and access of the rolled face value.
- Must provide an overloaded constructor that, for testing purposes, takes a random number seed.
- *For testing and submission, use **5** and **10** as seeds to the random number generator*.

`EntryPoint` class

- Contains the `main` method. It declares and allocates a `GameController` object and invokes the `play` method.

`GameController` class

- Method `play` contains the central game control logic.
- Use of additional private methods *must* be used for modularization.
    - For example, since the play method controls the game logic, a private method `takeTurn` (which takes the keyboard `Scanner` and a `PigDice` object) should call a method that takes a turn for a single player.
    - A second private method would maintain functionality to acquire the valid 'goal' score.

`PigDice` Class

This class (1) maintains the Player Pig score for the round, (2) maintains and saves the total Pig score for a player, and (3) provides functionality related to rolling and scoring die. The following methods discuss that functionality in more detail.

- Method `currentTotal` returns the current total integer score for a game of Pig.
- Method `currentRound` returns the current integer score for a round. Please see input / output ***Session 5*** for an example differentiating round versus total scores.
- Method `rollDice` simulates rolling two die.
- Method `evaluate` computes the rolled score and adds to the current round score.
- Method `lastRoll` acquires a formatted string (consistent with the sample session output) describing the values of the last dice roll.
- Method `save` will (1) add the current round score to the total score and (2) clear the current round score, and (3) return the complete round score.

- Method `piggedOut` returns true / false whether the last roll contained a 1 and ended the roll.
- Additional private methods used for modularization:
    - Method `singleOneRolled` returns true / false whether *one* of the two die resulted in a 1.
    - Method `doubleOnesRolled` returns true / false whether *both* die rolled a 1.

Recommendations

- Review the code that has been provided in the skeleton.
- Before adding any coding, create outlines of the source code including method prototypes (return types, names, and parameters).
- Ensure you can mimic exactly the sample sessions provided in this document.
- Be thorough in your testing because it is an instructor's job to try and break your code.

Sample Input / Output Sessions

User input is stated in green below.

Session 1: Valid Input of Max Score

```
What score would you like to play to? (100 max)
-1
What score would you like to play to? (100 max)
101
What score would you like to play to? (100 max)
500
What score would you like to play to? (100 max)
100

PLAYER 1
D1 (2), D2 (1) scored 0 points.
You pigged out this turn.

PLAYER 2
D1 (3), D2 (2) scored 5 points.
Your current roll is 5 points. Keep rolling? Respond (Y/N) only.
```

Session 2: Complete Session with Player 2 Winning

```
What score would you like to play to? (100 max)
65

PLAYER 1
D1 (1), D2 (3) scored 0 points.
You pigged out this turn.

PLAYER 2
D1 (6), D2 (2) scored 8 points.
Your current roll is 8 points. Keep rolling? Respond (Y/N) only.
y
D1 (3), D2 (3) scored 6 points.
Your current roll is 14 points. Keep rolling? Respond (Y/N) only.
y
D1 (3), D2 (4) scored 7 points.
Your current roll is 21 points. Keep rolling? Respond (Y/N) only.
y
D1 (4), D2 (5) scored 9 points.
```

```
Your current roll is 30 points. Keep rolling? Respond (Y/N) only.
y
D1 (6), D2 (5) scored 11 points.
Your current roll is 41 points. Keep rolling? Respond (Y/N) only.
y
D1 (4), D2 (4) scored 8 points.
Your current roll is 49 points. Keep rolling? Respond (Y/N) only.
y
D1 (5), D2 (6) scored 11 points.
Your current roll is 60 points. Keep rolling? Respond (Y/N) only.
y
D1 (2), D2 (3) scored 5 points.
Your current roll is 65 points. Keep rolling? Respond (Y/N) only.
n
Your total for the round was 65 and your total score is 65.

Player 1: 0 -- Player 2: 65

Player 2 wins!

would you like to play again? Respond (Y/N) only.
n
Goodbye
```

Session 3: Complete Session with Player 1 Winning

```
What score would you like to play to? (100 max)
54

PLAYER 1
D1 (4), D2 (3) scored 7 points.
Your current roll is 7 points. Keep rolling? Respond (Y/N) only.
y
D1 (3), D2 (6) scored 9 points.
Your current roll is 16 points. Keep rolling? Respond (Y/N) only.
y
D1 (6), D2 (4) scored 10 points.
Your current roll is 26 points. Keep rolling? Respond (Y/N) only.
y
D1 (1), D2 (1) scored 25 points.
Your current roll is 51 points. Keep rolling? Respond (Y/N) only.
y
D1 (1), D2 (1) scored 25 points.
Your current roll is 76 points. Keep rolling? Respond (Y/N) only.
n
Your total for the round was 76 and your total score is 76.

PLAYER 2
D1 (2), D2 (6) scored 8 points.
Your current roll is 8 points. Keep rolling? Respond (Y/N) only.
y
D1 (6), D2 (1) scored 0 points.
You pigged out this turn.

Player 1: 76 -- Player 2: 0

Player 1 wins!

Player 1: 76 -- Player 2: 0

would you like to play again? Respond (Y/N) only.
n
Goodbye
```

Session 4: Complete Session with a Tie

```
What score would you like to play to? (100 max)
7

PLAYER 1
D1 (4), D2 (5) scored 9 points.
Your current roll is 9 points. Keep rolling? Respond (Y/N) only.
n
Your total for the round was 9 and your total score is 9.

PLAYER 2
D1 (6), D2 (3) scored 9 points.
Your current roll is 9 points. Keep rolling? Respond (Y/N) only.
n
Your total for the round was 9 and your total score is 9.

Player 1: 9 -- Player 2: 9

The result is a tie.
would you like to play again? Respond (Y/N) only.
n
Goodbye
```

Session 5: Multi-turn Incomplete Run

```
What score would you like to play to? (100 max)
100

PLAYER 1
D1 (3), D2 (5) scored 8 points.
Your current roll is 8 points. Keep rolling? Respond (Y/N) only.
n
Your total for the round was 8 and your total score is 8.

PLAYER 2
D1 (3), D2 (5) scored 8 points.
Your current roll is 8 points. Keep rolling? Respond (Y/N) only.
n
Your total for the round was 8 and your total score is 8.

Player 1: 8 -- Player 2: 8


PLAYER 1
D1 (1), D2 (4) scored 0 points.
You pigged out this turn.

PLAYER 2
D1 (1), D2 (4) scored 0 points.
You pigged out this turn.

Player 1: 8 -- Player 2: 8


PLAYER 1
D1 (2), D2 (5) scored 7 points.
Your current roll is 7 points. Keep rolling? Respond (Y/N) only.
n
Your total for the round was 7 and your total score is 15.

PLAYER 2
D1 (2), D2 (5) scored 7 points.
Your current roll is 7 points. Keep rolling? Respond (Y/N) only.
n
Your total for the round was 7 and your total score is 15.

Player 1: 15 -- Player 2: 15
```

```
PLAYER 1
D1 (1), D2 (4) scored 0 points.
You pigged out this turn.

PLAYER 2
D1 (1), D2 (4) scored 0 points.
You pigged out this turn.

Player 1: 15 -- Player 2: 15


PLAYER 1
D1 (1), D2 (4) scored 0 points.
You pigged out this turn.

PLAYER 2
D1 (1), D2 (4) scored 0 points.
You pigged out this turn.

Player 1: 15 -- Player 2: 15


PLAYER 1
D1 (4), D2 (2) scored 6 points.
Your current roll is 6 points. Keep rolling? Respond (Y/N) only.
n
Your total for the round was 6 and your total score is 21.

PLAYER 2
D1 (4), D2 (2) scored 6 points.
Your current roll is 6 points. Keep rolling? Respond (Y/N) only.
n
Your total for the round was 6 and your total score is 21.

Player 1: 21 -- Player 2: 21
```

Submitting

Header Comments

Your program must use the following standard comment at the top of *each source code file*. Copy and paste this comment and modify the parenthesized values accordingly.

```
/*
* @author (Student Name)
* <p> (File Name)
* <p> (Assignment)
* <p> (Describe, in general, the code contained.)
*/
```

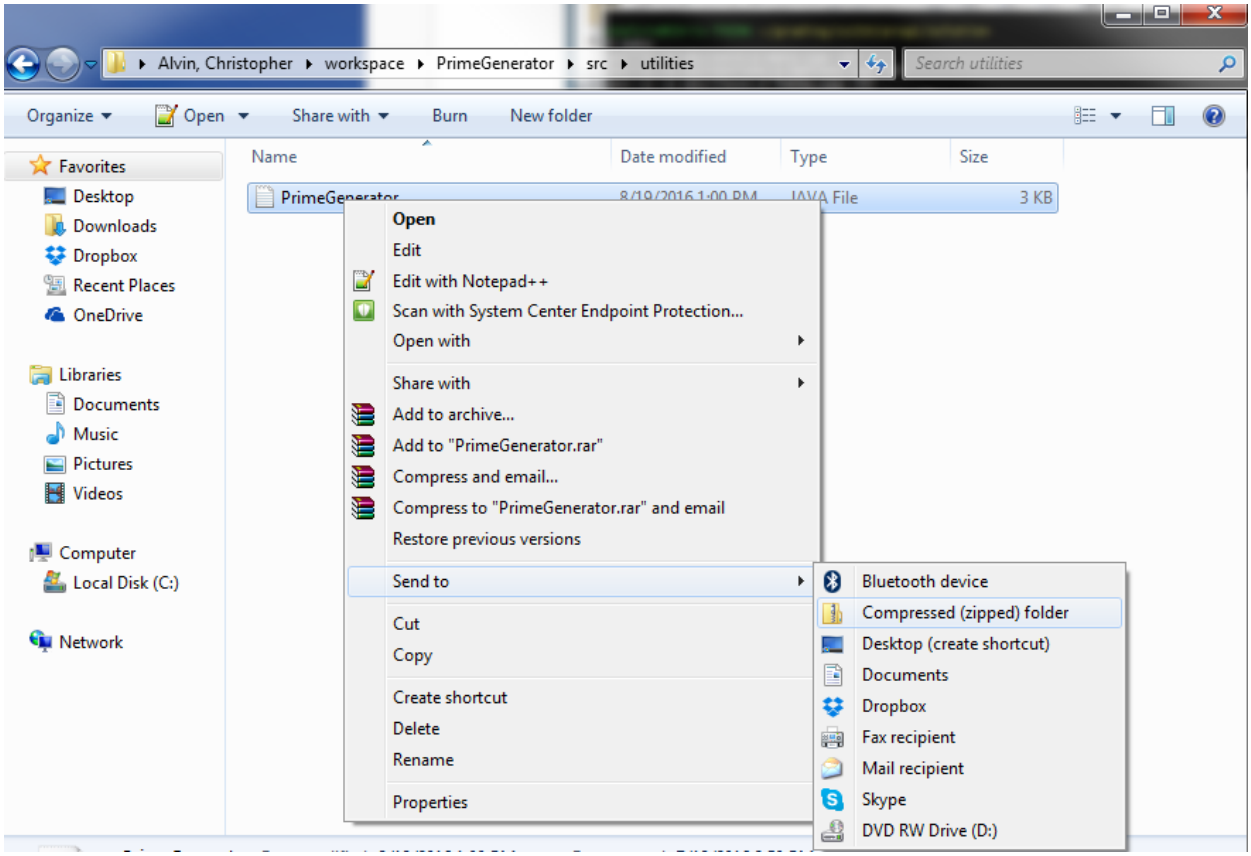
Inline Comments

Please comment your code with a reasonable amount of comments throughout the program. Each block of code (3-4 or more lines in sequence) in a method should be commented.

Although it is an issue of style and preference, please avoid long comments to the right of lines of source code. Long, ubiquitous comments to the right of code will result in a deduction.

Final Submission File

Create a zip file (`proj3.zip`) containing only the source code files (`EntryPoint.java`, `GameController.java`, `PigDice.java`, and `Die.java`). In Windows, (1) select the root project folder, (2) right-click, and (3) Send to > Compressed (zipped) folder:



Submit your zip as directed by your instructor. Be sure to review the university policy on academic dishonesty: this is an individual project.