

Pig Statistics

Objective: Enhance our Pig Game to run a Monte Carlo statistical analysis.

Background:

You wrote the **Pig** game which pitted the user against the computer in a game of chance. This time you will make the computer run a simulation of many turns by itself. In each turn, the computer will “hold at 20 points” just like in the prior game. But instead of adding the turn scores together, your program will count how many times the computer’s turn ended with 0 points, 20 points, 21 points, 22 points, etc. The objective is to empirically calculate the probability of getting each one of these scores given a large number of turns.

This is a form of Monte Carlo statistical analysis. In Monte Carlo, the simulation is run many times and each outcome is counted. After thousands or millions of runs, the results are divided by the total number of runs to get an estimated probability. These simulations are used on a variety of statistical problems to help predict outcomes which include financial forecasts, weather forecasts, the decay of isotopes, the flow of auto traffic, the outcomes of games, etc.

The advent of modern computers made the Monte Carlo method practical since computers could run many simulations quickly and accurately. Stanislaw Ulam invented the technique in the 1940’s and, working with John von Neumann along with the ENIAC computer, ran a Monte Carlo simulation to determine radiation shielding of neutrons. It was a secret project at the time, so it required a “secret code” name. Ulam decided to name it after the famous Monte Carlo Casino in Monaco.

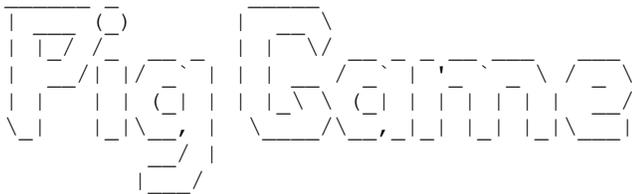
Assignment:

Add functionality to your **PigGame.java** to run a Monte Carlo simulation of the results. The simulation will perform many turns, each turn will “hold at 20 points.”

Prompt the user whether to play the game with the computer, or to run a simulation. If the game is selected, then **PigGame** should perform as before. If a simulation is chosen, then prompt the user for the number of turns to perform. During the simulation count how many times a “hold at 20” turn produces 0 points, 20 points, 21 points, etc. After the simulation print out the probability in a table.

A sample run:

```
% java PigGame
```



```
The Pig Game is human vs computer. Each takes a turn rolling a die and the first to score 100 points wins. A player can either ROLL or HOLD. A turn works this way:
```

```
ROLL: 2 through 6: add points to turn total, player's turn continues
      1: player loses turn
HOLD: turn total is added to player's score, turn goes to other player
```

```
Play game or Statistics (p or s) -> s
```

```
Run statistical analysis - "Hold at 20"
```

```
Number of turns (1000 - 1000000) -> 1000000
```

```
Estimated
Score Probability
```

0	0.62450
20	0.09999
21	0.09502
22	0.07405
23	0.05387
24	0.03524
25	0.01734