Code Documentation for MonteCarloDoorGame Project

By Rachel Hussmann

Overview

The purpose of the MonteCarloDoorGame project was to answer the question of "With regards to the Monty Hall problem, is it statistically better to switch your door choice or keep your door choice?" This project has three classes: Door, DoorGame and DoorGameTester, all which work together to give the user the answer to the question.

Background

What is the Monty Hall Problem?

The Monty Hall Problem originated from a game show called "Let's Make a Deal" where the gameshow host would show the contestant three doors, with only one having a big prize like a new car. The contestant would pick their door, and the gameshow host would open up one of the doors and reveal what was behind the door. The contestant would then get the option to keep their original choice or switch to the other door. Eventually, people began to debate over what was the statistically better option: keeping your original choice or switching your choice when given the option. Thus, the Monty Hall problem was created.

How It Works

Door

The Door class is responsible for holding information about a door in the Monty Hall problem, specifically what is behind this instance of the door. This class contains getters and setters for one variable:

- String behindDoor

DoorGame

The DoorGame class is responsible for running the simulations and calculating the probability of winning if the player switched choices or stayed with their original choice. DoorGame has two variables and five major methods:

- int stayWins – The number of wins that occur when the player kept their choice
- int switchWins – The number of wins that occur when the player switched their choice
- generateDoors
    - Parameter: None
    - Functionality – Randomly selected the available doors to create the list of doors for a trial
    - Returns: ArrayList<Door> - The randomized list of doors
- runGame
    - Parameter: int trials – The number of trials to be run for the simulation
    - Functionality – Called the methods to run the trials for when the player kept their choice and for when the player switched their choice
    - Returns: ArrayList<Double> - The percentages of wins when the player kept their choice and when the player switched their choice
- stayTrials
    - Parameter: int trials – The number of trials to be run for the simulation
    - Functionality – Ran the trials where the player kept their choice
    - Returns: None
- switchTrials
    - Parameter: int trials – The number of trials to be run for the simulation
    - Functionality – Ran the trials where the player switched their choice
    - Returns: None
- calculatePercentages
    - Parameter: int stayWins – The number of wins that occurred when the player kept their door choice, int switchWins – The number of wins that occurred when the player switched their door choice
    - Functionality – Calculates the percentage of wins that occurred during both trials
    - Returns: ArrayList<Double> - An ArrayList of doubles that contains the percentages of winning when the player kept their choice and when the player switched their choice

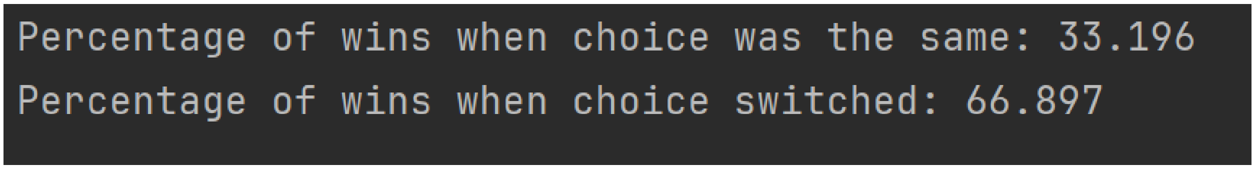
DoorGameTester

The DoorGameTester class contains the main method and runs the program. It contains a tester object of the DoorGame class. It runs the trials and outputs the calculation to the user.

Output

The program outputs the probability of winning when the player kept their original choice and the probability of winning when the player switched their choice.

Screenshot



```
Percentage of wins when choice was the same: 33.196
Percentage of wins when choice switched: 66.897
```