User Manual Monte Carlo Program

By: Jacob Attia

## **Introduction**

This program is a simulation of the Monty Hall problem, a well-known probability puzzle based on a game show scenario. The program plays the game 10,000 times with and without switching doors to analyze the probability of winning the car under both strategies.

## **Program Components**

The program consists of three main classes:

1. GameTester - The entry point that starts the simulation.
2. Game - Implements the Monty Hall logic for a single game round.
3. Door - Represents a door that may or may not contain a car.

## **How the Program Works**

### 1. GameTester (Main Execution Point)

* The GameTester class initializes the GameSimulation and calls the run() method.
* This starts two sets of 10,000 simulations:

- One where the contestant does not switch doors.

- One where the contestant switches doors.

* The results are printed to the console, showing win percentages for both cases.

### 2. GameSimulation (Runs Multiple Simulations)

* Calls simulate() twice:
* First with switchDoor = false (contestant sticks to the initial choice).
* Then with switchDoor = true (contestant switches after a goat is revealed).
* Each simulate() method runs the Game.playGame() function 10,000 times.
* The number of wins is counted and converted into a winning percentage.
* The results are printed to display how often switching vs. not switching wins the car.

### 3. Game (Monty Hall Logic)

* Implements a single round of the Monty Hall game.
* Steps of the game:
  1. A car is randomly placed behind one of three doors.
  2. The contestant picks a door randomly.
  3. The host reveals a door that does not contain the car and is not the contestant's choice.
  4. If switchDoor == true, the contestant switches to the remaining unopened door.
  5. The method returns true if the contestant's final choice has the car.

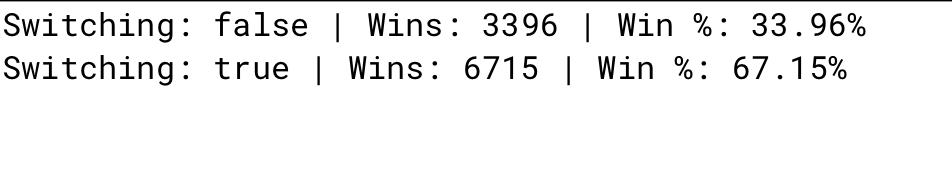
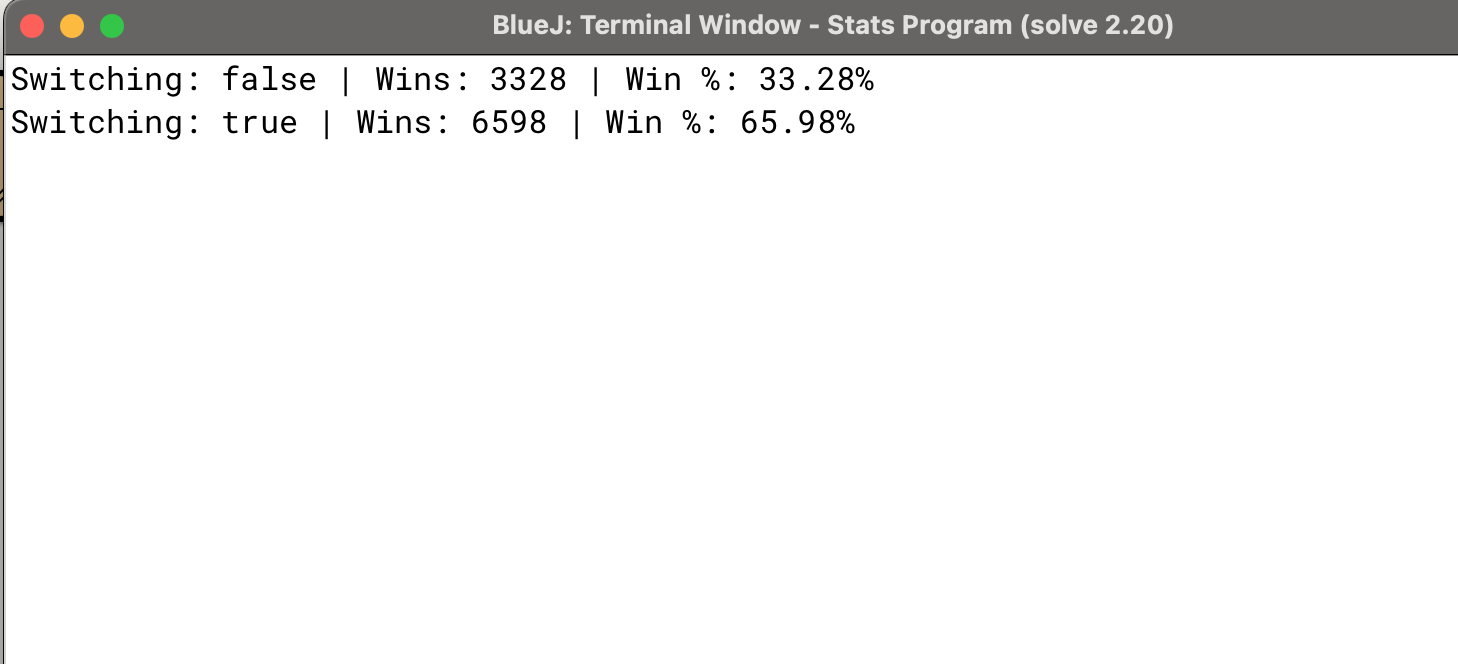
### 4. Door Class (Represents Individual Doors)

* Each Door object has a name and a hasCar flag.
* The setCar() method places a car behind a door.
* The hasCar() method checks if the door contains a car.

## 

## 

## **Expected Output**



## 

## **Conclusion**

This program effectively models the Monty Hall problem and verifies the mathematical probability that switching doors gives a higher probability of winning. The simulation confirms that the best strategy is to always switch doors after the host reveals a goat.