**Project Name:** Monty Hall Simulation Problem

**Github Link:** https://github.com/projectsforstudents2022/Monty\_Hall\_Simulation\_Problem.git

**Why was this project created?**

A contestant is shown three doors that are the same. There is a car behind one of them. The remaining two hide goats. One of the doors must be selected by the contestant without being opened. Monty, who is aware of the location of the vehicle, then unlocks one of the other two doors. When he has a choice between more than one door, he always opens the one he knows to be incorrect, and when the contestant's first pick hides the car, he randomly selects which door to open. Monty offers the candidate the opportunity of moving to the other unopened door or staying with their first selection after they have opened the wrong door. The winner is then given whatever is located behind the door they select.

**What problem is it solving?**

We will begin by breaking down the Monty Hall problem in pieces. After the

player selects a door, Monty opens one of the remaining two doors and reveals

what the door is hiding. Monty though, opens his door following this strategy:

1. Monty always opens a door that hides a goat

2. Monty never opens the door the player selects

3. If Monty can open more than one door, following strategies 1 and 2, then he opened the door at random. This strategy happens when the contestant selects the winning door.

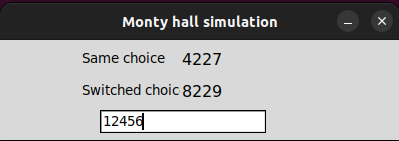
**Entire explanation of project**

* **PROPOSED APPROACH**

The upper limit of the event's relative frequency over a large number of trials is the frequentist definition of probability. This definition will be used to a Python simulation of the Monty Hall game's winning chances. As we just demonstrated, changing doors is the best course of action in the Monty Hall game because there is a 2/3 chance that the player's initial door will not be behind the car. This result is consistent with the simulations we performed before. We must consider the knowledge we got when the host opened the door and exposed the goat. It may seem counterintuitive that with two doors left in the game, the likelihood for either would be anything other than 50-50.

Since P(A|B) = P(A), we can conclude that the occurrence of event B had no impact on the likelihood of event A. P(A) continues to be 1/3, and the probability of A' also stays the same at 2/3. The car is now only concealed behind the last door the player didn't select at the beginning of the game, which is all that remains of event A'. Due to the new knowledge provided by event B, the likelihood that the other door contains the car has increased from 1/3 at the beginning of the game to 2/3.

* **RESULT**



**CONCLUSION**

To make it simple to run numerous tests rapidly, I have wrapped the simulation code in a function. Be advised, however, that the function can take some time to deliver a result for extremely many games.