Let's Make a Deal (Monty Hall) — MATLAB Simulation Report

Author: Habib Hammam Kurniawan
Teacher: Muhammad Qomaruz Zaman, S.T., M.T., Ph.D.
Class name: Algoritma dan Komputasi

Project Title

Let's Make a Deal (Monty Hall) Simulator with MATLAB

Project Description

This project implements a Monte Carlo simulator for the Monty Hall problem ("Let's Make a Deal") and generalizes it from the classic 3-door setting to N doors. For each $N \in \{3, \dots, 15\}$ and a fixed number of trials (default: 50,000), the program estimates the empirical winning probabilities when the player **stays** with the initial choice versus when the player **switches** after the host opens all but one non-prize door. Results are printed in a formatted console table and visualized in a plot comparing simulation vs. theory.

Problem Statement

Given N closed doors with one hidden prize, the player selects an initial door. The host (who knows the prize location) opens $N-2$ non-prize doors, leaving exactly two closed doors: the player's original pick and one other. The player must decide whether to *stay* or *switch*. We study how the winning probabilities evolve as N increases.

Theoretical Background

In the N-door Monty Hall variant described above:

$$\Pr(\textbf{stay wins}) = \frac{1}{N}, \qquad \Pr(\textbf{switch wins}) = 1 - \frac{1}{N} = \frac{N-1}{N}.$$

Intuition: the initial pick hits the prize with probability $1/N$. If the initial pick is wrong (probability $(N-1)/N$), the host's action concentrates that entire probability mass on the single other remaining closed door, hence switching wins with $(N-1)/N$.

Methodology & Algorithm

Simulation Logic

For each trial:

1. Randomly place the prize behind one of N doors.
2. Player picks an initial door uniformly at random.
3. The host opens $N-2$ doors that are *neither* the prize door nor the player's current door.
4. Two doors remain closed: (a) the original choice (for **stay**); (b) the other unopened door (for **switch**).
5. Record wins for each strategy.

Repeat for many trials and compute empirical rates.

Implementation Notes (Mapping to Code)

- `simulate_monty_hall`: runs *n_trials* for a given *n_doors*, returning empirical win rates for **stay** & **switch**.
- `simulate_host_actions`: exact host behavior — opens $N-2$ doors from the set {not prize, not initial}.
- `calculate_theoretical`: returns $1/N$ and $(N-1)/N$ for comparison.
- `create_plot`: produces two subplots (win probabilities & switch-advantage curve).

Parameters

- Door range: $N \in [3, 15]$ (configurable via `min_doors`, `max_doors`).
- Trials per N: default `trials = 50000` (can be increased for lower variance).

User Manual

1. Open MATLAB, place the script file (e.g., `monty_hall_simulator.m`) in the current folder or on the path.
2. Run the simulation by calling `monty_hall_simulator;` in the Command Window.
3. Observe the console table showing (*Stay (Actual/Theory)* vs *Switch (Actual/Theory)*) for each N, and the generated figure with two panels:
   - Win Probability (%) vs. Number of Doors (simulation vs. theory),
   - Switch Advantage = (Switch / Stay).

Expected Theoretical Values (Selected N)

N	Stay Theory (%)	Switch Theory (%)	Switch Advantage (×)
3	33.33	66.67	2.00
5	20.00	80.00	4.00
10	10.00	90.00	9.00
15	6.67	93.33	14.00

Sample Console Output (Format)

```
Doors | Stay (Act/Thy) | Switch (Act/Thy)  | Advantage
------|----------------|------------------|----------
  3   | 0.333/0.333    | 0.667/0.667      | 2.0x
  4   | 0.251/0.250    | 0.749/0.750      | 3.0x
 ...
 15   | 0.067/0.067    | 0.933/0.933      | 14.0x
```

Results & Discussion

Empirical vs. Theory. With 50,000 trials per N, the simulated probabilities closely track the theoretical lines $1/N$ and $(N-1)/N$. As N grows, the *stay* probability decays to zero while the *switch* probability approaches one.

Key Insights.

- Switching is always better for $N \geq 3$, with advantage $(N-1):1$.
- The larger the door count, the larger the benefit of switching (e.g., $\approx 14\times$ at $N = 15$).
- The Law of Large Numbers ensures simulation curves converge to theoretical values as trials increase.

Limitations & Extensions.

- This model assumes an *ideal* host: always opens $N-2$ goat doors and never reveals the prize.
- Extensions: biased host behaviors, multiple staged reveals, noisy player beliefs, or cost for switching.

Flowchart

How to Read the Plot

- **Top panel**: Blue (stay) and red (switch) lines show empirical win rates; dashed lines show theoretical curves. Gap widens with N.
- **Bottom panel**: Switch advantage = (switch / stay). This grows approximately linearly with $N - 1$.

Conclusion

For the Monty Hall problem with N doors and an informed host who opens $N - 2$ non-prize doors, switching yields a win probability of $(N - 1)/N$, dominating the stay strategy's $1/N$. Simulations confirm the theory and illustrate that the benefit of switching grows rapidly with the number of doors.

Appendix: File & Function List

- `monty_hall_simulator.m` (main): sets parameters, prints table, calls plotting.
- `simulate_monty_hall`: returns empirical stay/switch win rates for given N and trials.
- `simulate_host_actions`: exact host opening logic, leaving two closed doors.
- `calculate_theoretical`: returns $(1/N, (N - 1)/N)$.
- `create_plot`: renders probability and advantage plots.

Another Documentation

- Repository (if any): `https://github.com/habibhkrnwn/algoritmadankomputasi/tree/main/Letsmakeadeal`
- Demo video (optional): `https://drive.google.com/file/d/1uOG7vqexzjxAWRNZ7UWcfhCvY8sJvkik/view?usp=shar`

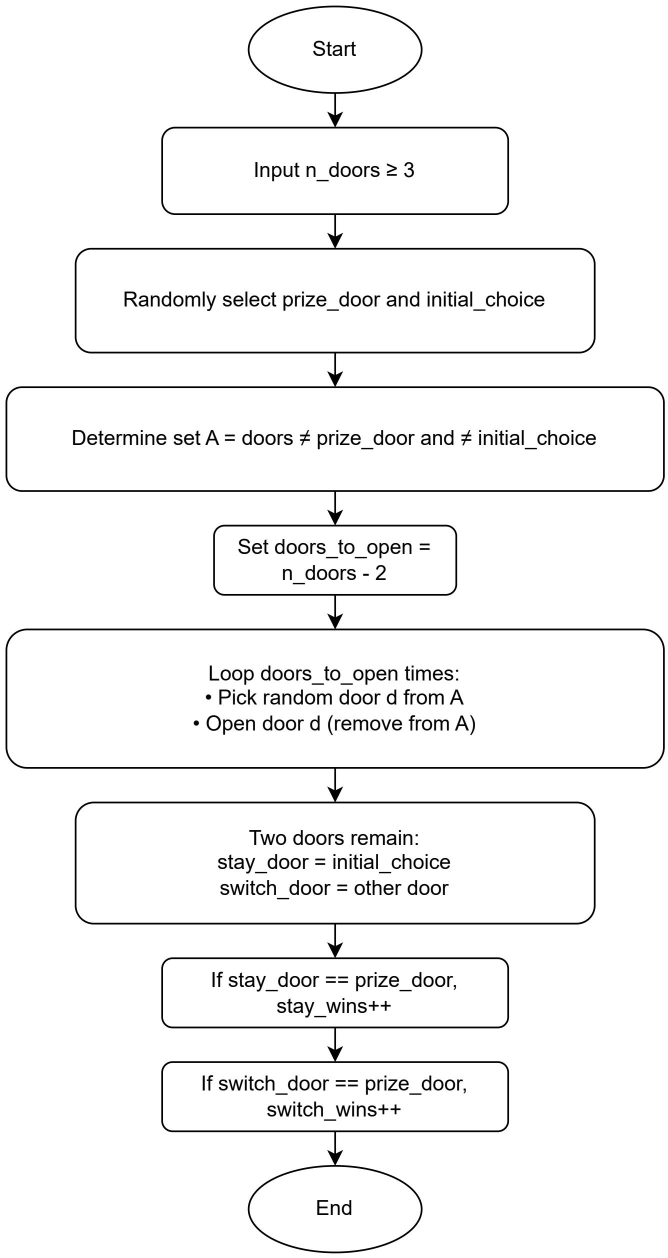

Figure 1: High-level simulation flow: initialize → random prize & pick → host opens $N-2$ doors → evaluate stay/switch.