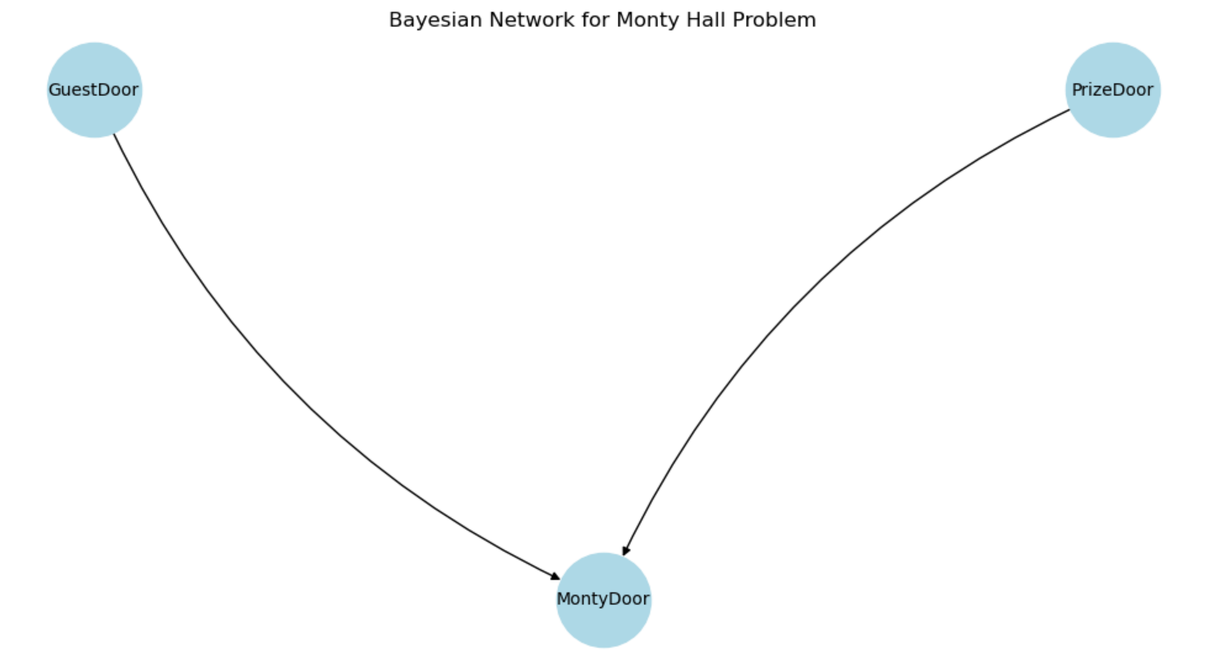
In [ ]:

```
import networkx as nx
import matplotlib.pyplot as plt

# Create a directed graph
G = nx.DiGraph()

# Add nodes (variables in Bayesian Network)
nodes = ["GuestDoor", "PrizeDoor", "MontyDoor"]
G.add_nodes_from(nodes)

# Add edges (dependencies in Bayesian Network)
edges = [("GuestDoor", "MontyDoor"),
         ("PrizeDoor", "MontyDoor")]
G.add_edges_from(edges)


# draw the graph
plt.figure(figsize=(10, 5))
nx.draw(G, pos, with_labels=True, node_color="lightblue", edge_color="bla
        node_size=3000, font_size=10, arrows=True, connectionstyle="arc3,

# Add a title
plt.title("Bayesian Network for Monty Hall Problem")

# Show the graph
plt.show()
```



In [7]:

```
# The Monty Hall Problem
import random

def monty_hall_round(switch: bool):
    "simulate one round of the Monty Hall problem."
    # randomly assign the prize behind one of the three doors
    prize_door = random.randint(0, 2)

    # sontestant makes 1 random choice
    chosen_door = random.randint(0, 2)

    # rules, game leader opens a door that is neither the chosen door nor
    for door in range(3):
        if door != chosen_door and door != prize_door:
            door_opened = door
            break

    # if the contestant switches, they pick the remaining closed door
    if switch:
        chosen_door = 3 - chosen_door - door_opened

    # true + win, false + lose
    return chosen_door == prize_door

def monty_hall_simulation_function(switch: bool, trials: int = 1000):
    "simulate multiple rounds of the Monty Hall problem."
    wins = sum(monty_hall_round(switch) for _ in range(trials))
    return wins / trials  # Compute the win probability

stay_win_probability = monty_hall_simulation_function(switch=False)
switch_win_probability = monty_hall_simulation_function(switch=True)

print(f"Win probability when staying: {stay_win_probability:.4f}")
print(f"Win probability when switching: {switch_win_probability:.4f}")
```

```
Win probability when staying: 0.3450
Win probability when switching: 0.6630
```

```
In [ ]:
```