Exercise 9: Implementation of uncertain methods for an application

Uncertain method:

Aim:

To implement uncertain method for solving Monty Hall problem.

Algorithm:

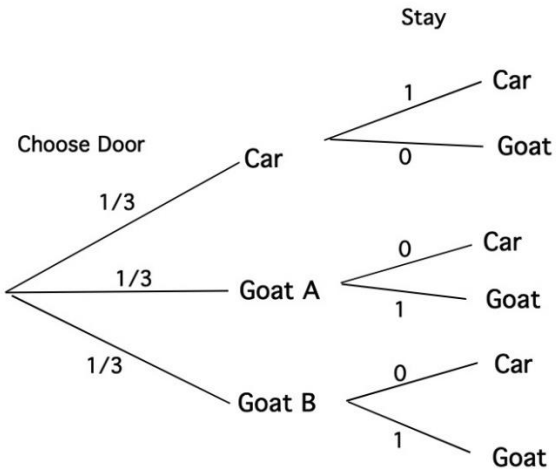
The Monty Hall problem is a brain teaser, in the form of a probability puzzle named after the host of the TV series, 'Let's Make A Deal'. The game involves three doors, given that behind one of these doors is a car and the remaining two have goats behind them. So you start by picking a random door, say #2. On the other hand, the host knows where the car is hidden and he opens another door, say #1 (behind which there is a goat). You are now given a choice, the host will ask you if you want to pick door #3 instead of your first choice i.e. #2.

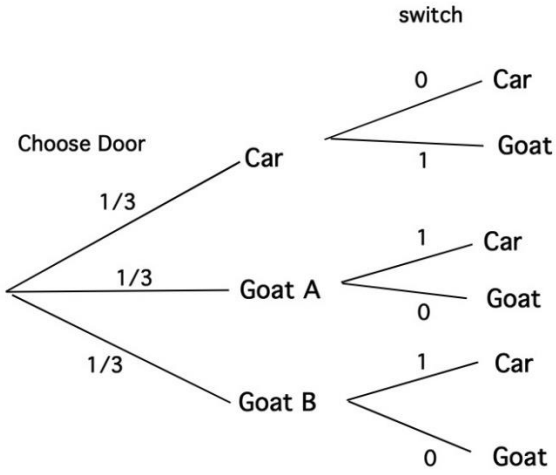
The graph has three nodes, each representing the door chosen by:

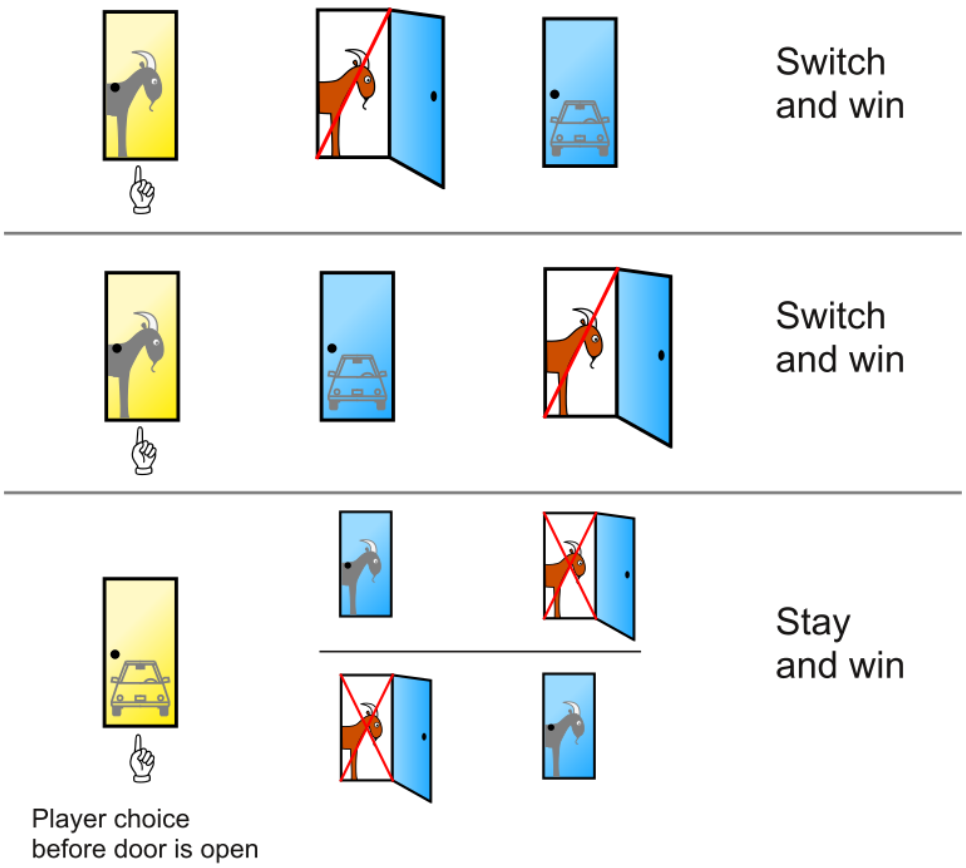
1. The door selected by the Guest.

2. The door containing the prize (car)

3. The door Monty chooses to open.







Program:

```
import random
def get_non_prize_door(host, num_doors, player_choice):
    i = 1
    while (i == host or i == player_choice):
        i = (i + 1) % (num_doors)

    return i


def switch_function(shown_door, num_doors, player_choice):
    i = 1
    while (i == shown_door or i == player_choice):
        i = (i + 1) % (num_doors)

    return i


def monty_hall_game(switch, num_tests):
    win_switch_cnt = 0
```

```
    win_no_switch_cnt = 0
    lose_switch_cnt = 0
    lose_no_switch_cnt = 0
    doors = [0, 1, 2]  # Get the doors
    num_doors = len(doors)  # Get the number of doors

    for i in range(0, num_tests):
        door_with_prize = random.randint(0, num_doors - 1)
        print('\nRandom Test Case ',i+1,":")
        host = door_with_prize
        player_choice = random.randint(0, num_doors - 1)
        print('The Player Chose Door:', player_choice, '')
        original_player_choice = player_choice
        shown_door = get_non_prize_door(host, num_doors, player_choice)
        if switch == True:
            player_choice = switch_function(shown_door, num_doors, player_choice)

        if player_choice == host and switch == False:
            # Then the player wins from not switching
            print('Player Wins (No switch) - The player chose door: ', player_choice, ' Original choice: ',
                original_player_choice, ', Door with prize:', door_with_prize, ', Shown Door: ',
shown_door,'\n\n')
            win_no_switch_cnt = win_no_switch_cnt + 1
        elif player_choice == host and switch == True:
            # Then the player wins from switching
            print('Player Wins (switch) - The player chose door: ', player_choice, ' Original choice: ',
                original_player_choice, ', Door with prize:', door_with_prize, ', Shown Door: ',
shown_door)
            win_switch_cnt = win_switch_cnt + 1
        elif player_choice != host and switch == False:
            # The player lost from not switching
            print('Player Lost (No switch) - The player chose door: ', player_choice, ' Original choice: ',
                original_player_choice, ', Door with prize:', door_with_prize, ', Shown Door: ',
shown_door)
            lose_no_switch_cnt = lose_no_switch_cnt + 1
        elif player_choice != host and switch == True:
            # The player lost from switching
            print('Player Lost (switch) - The player chose door: ', player_choice, ' Original choice: ',
                original_player_choice, ', Door with prize:', door_with_prize, ', Shown Door: ',
shown_door)
            lose_switch_cnt = lose_switch_cnt + 1
        else:
            print('SOMETHING IS WRONG')

    return win_no_switch_cnt, win_switch_cnt, lose_no_switch_cnt, lose_switch_cnt, num_tests

x = monty_hall_game(True, 10)
print('\n\nWin switch %: ', x[1]/ x[4])
```

```
print('Lose switch %: ', x[3]/ x[4])
print('Win No switch %: ', x[0]/ x[4])
print('Lose No switch %: ', x[2]/ x[4])
```

OUTPUT:

```
Random Test Case  1 :
The Player Chose Door: 0
Player Wins (switch) - The player chose door:  2  Original choice:  0 , Door with prize: 2 , Shown Door:  1

Random Test Case  2 :
The Player Chose Door: 1
Player Wins (switch) - The player chose door:  2  Original choice:  1 , Door with prize: 2 , Shown Door:  0

Random Test Case  3 :
The Player Chose Door: 0
Player Lost (switch) - The player chose door:  2  Original choice:  0 , Door with prize: 0 , Shown Door:  1

Random Test Case  4 :
The Player Chose Door: 0
Player Lost (switch) - The player chose door:  2  Original choice:  0 , Door with prize: 0 , Shown Door:  1

Random Test Case  5 :
The Player Chose Door: 0
Player Wins (switch) - The player chose door:  2  Original choice:  0 , Door with prize: 2 , Shown Door:  1

Random Test Case  6 :
The Player Chose Door: 1
Player Wins (switch) - The player chose door:  0  Original choice:  1 , Door with prize: 0 , Shown Door:  2

Random Test Case  7 :
The Player Chose Door: 1
Player Wins (switch) - The player chose door:  0  Original choice:  1 , Door with prize: 0 , Shown Door:  2

Random Test Case  8 :
The Player Chose Door: 0
Player Wins (switch) - The player chose door:  2  Original choice:  0 , Door with prize: 2 , Shown Door:  1

Random Test Case  9 :
The Player Chose Door: 0
Player Lost (switch) - The player chose door:  2  Original choice:  0 , Door with prize: 0 , Shown Door:  1

Random Test Case  10 :
The Player Chose Door: 1
Player Wins (switch) - The player chose door:  2  Original choice:  1 , Door with prize: 2 , Shown Door:  0


Win switch %:  0.7
Lose switch %:  0.3
Win No switch %:  0.0
Lose No switch %:  0.0


Process exited with code: 0
```

Result:

The given program was successfully created using uncertain method for solving Monty Hall problem and was successfully executed in AWS.