

## ПАРАДОКС МОНТИ ХОЛЛА

Паныш И., Дроздов В. С.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Розум Г. А. – магистр техники и технологии,  
ассист. каф ИПиЭ

Цель работы – разработка программы для проведения исследования парадокса Монти Холла.

Участнику игры предлагался выбор из 3 дверей. За двумя из них не было призов а за одной Приз. Ведущий знает где находится приз. Участник выбирает одну дверь, далее ведущий открывает одну дверь за которой нету приза, и участнику предлагается изменить свое решение или остаться при своем. Парадокс заключается в том, что вероятность того, что при смене своего решения шанс выиграть выше. Разработанная программа "играет" в эту игру, и результаты действительно таковы. При смене решения шанс выигрыша в среднем около 66%, а если игрок не меняет своего решения, то шанс выигрыша около 33%.

Смысл в том, что, делая первый выбор, участник разделяет двери на А (выбранную), В и С (оставшиеся). Шансы (P) на то, что машина стоит за дверью А, равны 1/3, а на то, что она за дверьми В и С равны 2/3. И шансы на успех при выборе дверей В и С вычисляются так:  $P(B) = 2/3 * 1/2 = 1/3$  и  $P(C) = 2/3 * 1/2 = 1/3$

Где 1/2 является условной вероятностью того, что машина находится именно за этой дверью, при условии, что машина не за той дверью, что выбрал игрок.

Разработанная программа способствует проведению исследования парадокса Монти Холла. Программа написана на языке С#. Приведен код программы, каждый желающий может повторить и проверить результат:

```

namespace MontiHoleParadox
{
    class Program
    {
        static void Main(string[] args)
        {
            rand = new Random(DateTime.UtcNow.Millisecond);
            while(true)
            {
                int wins = 0, loose = 0;
                List<Game> gamesListWithChange = new List<Game>();
                List<Game> gamesListWithNotChange = new List<Game>();
                for (int i = 0; i < 1000; i++)
                {
                    gamesListWithChange.Add(new Game());
                }
                for (int i = 0; i < 1000; i++)
                {
                    gamesListWithNotChange.Add(new Game());
                }
                for (int i = 0; i < gamesListWithChange.Count; i++)
                {
                    Gamer gamer = new Gamer(gamesListWithChange[i]);
                    if (gamer.gameWithNotChangeDoor())
                        wins++;
                    else
                        loose++;
                }
                Console.WriteLine(String.Format("Кстаемсѧ при своем\nWINS: {0}\nLOOS: {1}\n-----", wins,
                loose));

                wins = loose = 0;
                for (int i = 0; i < gamesListWithNotChange.Count; i++)
                {
                    Gamer gamer = new Gamer(gamesListWithNotChange[i]);
                    if (gamer.gameWithChagerDoor())
                        wins++;
                    else
                        loose++;
                }
                Console.WriteLine(String.Format("Пенѧем выбор\nWINS: {0}\nLOOS: {1}\n-----", wins, loose));
                Console.ReadLine();
            }
        }
        static Random rand;
        public static Random getRandom()
        {
            return rand;
        }
        public static Int32 GetPercent(Int32 b, Int32 a)
        {
            if (b == 0) return 0; return (Int32)(a / (b / 100M));
        }
    }
}
public class Game
{
    public doorState[] doors;
    public int selectedDoor;
}

```

```

public Game()
{
    doors = new doorState[3];
    int r = Program.getRandom().Next(3, 11);
    r /= 3;
    for (int i = 0; i < doors.Length; i++)
    {
        if (i == r-1)
            doors[i] = doorState.Prize;
        else
            doors[i] = doorState.Empty;
    }
    public override string ToString()
    {
        int winDor = 1;
        for (int i = 0; i < doors.Length; i++)
        {
            if (doors[i] == doorState.Prize)
                winDor = i + 1;
        }
        return winDor.ToString();
    }
    public enum doorState
    {
        Open,
        Prize,
        Empty
    }
    public class Gamer
    {
        Game currentGame;
        public Gamer(Game game)
        {
            game.selectedDoor = Monty.generateRandom()-1;
            currentGame = game;
        }
        public bool gameWithNotChageDoor()
        {
            Monty.OpenDoor(currentGame);
            if (currentGame.doors[currentGame.selectedDoor] == doorState.Prize)
                return true;
            else
                return false;
        }
        public bool gameWithChagerDoor()
        {
            Monty.OpenDoor(currentGame);
            for (int i = 0; i < currentGame.doors.Length; i++)
            {
                if (i != currentGame.selectedDoor && currentGame.doors[i] != doorState.Open)
                {
                    currentGame.selectedDoor = i;
                    break;
                }
            }
            if (currentGame.doors[currentGame.selectedDoor] == doorState.Prize)
                return true;
            else
                return false;
        }
    }
    public class Monty
    {
        public static int generateRandom()
        {
            int r = Program.getRandom().Next(3, 11);
            r /= 3;
            return r;
        }
        public static void OpenDoor(Game game)
    {
        int r = Program.getRandom().Next(1, 10) / 5;
        if (r == 0)
            for (int i = 0; i < game.doors.Length; i++)
            {
                if (i != game.selectedDoor && game.doors[i] != doorState.Prize)
                {
                    game.doors[i] = doorState.Open;
                    break;
                }
            }
        else
            for (int i = game.doors.Length-1; i >= 0; i--)
            {
                if (i != game.selectedDoor && game.doors[i] != doorState.Prize)
                {
                    game.doors[i] = doorState.Open;
                    break;
                }
            }
    }
}
}

```

Ведущий, открывая заведомо проигрышную дверь из двух оставшихся, сообщает игроку 1 бит информации и изменяет тем самым условные вероятности для дверей B и C на значения 1 и 0. Теперь шансы на успех будут вычисляться так:  $P(B) = 2/3 \cdot 1 = 2/3$  и  $P(C) = 2/3 \cdot 0 = 0$

Разработанная программа подтвердила, что шанс выиграть при смене решения становится около 98%. В отличии от классического случая вероятность попадания с ходу в цель 1/50 или 2% вместо 1/3, в то время как вероятность выбора двери с Призом составляет 98%.

#### Список использованных источников:

1. Flanagan, D. JavaScript: The Definitive Guide: Activate Your Web Pages (Definitive Guides) / D.Flanagan // O'Reilly Media – 2011. – P. 1096.
2. Node.js [Электронный ресурс]. – Режим доступа: <https://nodejs.org>.

```

Остаемся при своем
Выигрыш: 19
Проигрыш: 981
-----
Меняем выбор
Выигрыш: 35
Проигрыш: 965)
-----
Остаемся при своем
Выигрыш: 12
Проигрыш: 988
-----
Меняем выбор
Выигрыш: 29
Проигрыш: 971)
-----
Остаемся при своем
Выигрыш: 20
Проигрыш: 980
-----
Меняем выбор
Выигрыш: 34
Проигрыш: 966)
-----

```

```

Остаемся при своем
Выигрыш: 339
Проигрыш: 661
-----
Меняем выбор
Выигрыш: 662
Проигрыш: 338)
-----
Остаемся при своем
Выигрыш: 340
Проигрыш: 660
-----
Меняем выбор
Выигрыш: 668
Проигрыш: 332)
-----
Остаемся при своем
Выигрыш: 369
Проигрыш: 631
-----
Меняем выбор
Выигрыш: 644
Проигрыш: 356)
-----

```