Author: Fredrik Ryberg 2019-06-28

Software documentation MontyHall

Development of console application according to the following specification of requirements and problem description:

Problem description:
Assume that you are attending a TV show where you can win money by picking the right box. The game show host shows you three boxes explaining that the money is in one of the boxes. He asks you to pick one of them without opening it. After you have picked one of the boxes he opens one of the other two boxes which is empty. Now he turns to you and asks, do you want to change your mind, picking the remaining box?

Task:
Write a readable and maintainable program, randomly simulating this event over and over again in the quest of answering following question. Do I stand a better chance to win if I change my mind? The only assumption allowed is that three boxes exist.
You are free to choose the type of application (Web, desktop, console, …). Your solution should last for a long time and as always, the conditions might change over time.

Take your time and focus on maintainability, clean code, documentation and how you deliver your solution.

Target framework: .Net Core

Version: 1.1

Git: https://github.com/fredrikryberg/MontyHall.git

Type: Console application

Author: Fredrik Ryberg 2019-06-28

Usage:

Enter number of simulations and the program will run all simulations on the different scenarios to finally present the results. Please see entry point of application in Program, Main method for examples when instancing and simulating the monty hall problem with different parameters, or in this case configuration, for simplicity.

Solution:

This software runs simulations with different configurable scenarios for the Monty Hall game:

- Contestant never switches boxes.
- Contestant switches boxes at the end of the round.
- The host, unknowingly and randomly, reveals a box with the risk of ending the round in a loss for the contestant, if the host picks the price.
- The host knowingly reveals an empty box of the remaining two boxes.

Author: Fredrik Ryberg 2019-06-28

Class design:

MontyHallGame constructor:

- Number of simulations
- Host
- Contestant

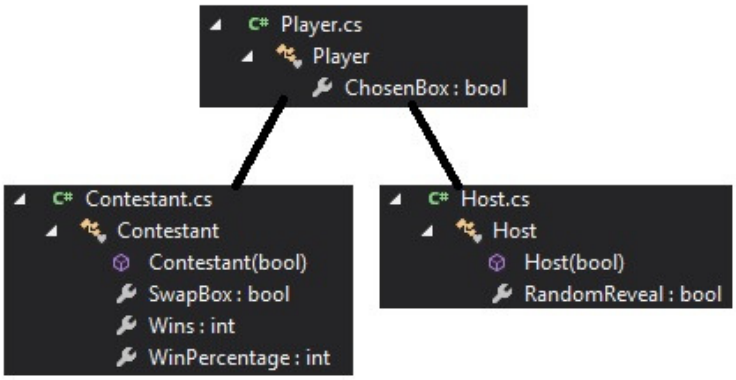
Abstract Player (Base class)

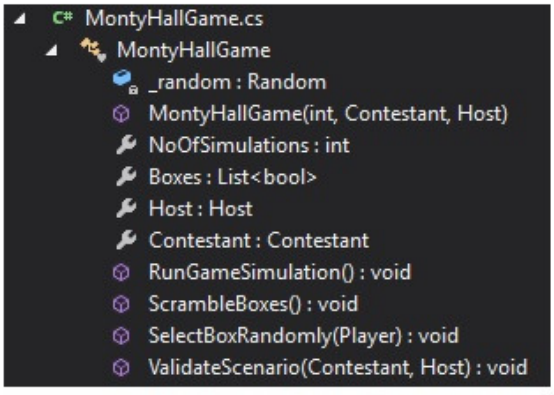
Contestant (Derived from Player class) constructor:

- Swap box or not

Host (Derived from Player class) constructor:

- Randomly reveal boxes or not





Author: Fredrik Ryberg 2019-06-28

Flow chart: