

Assignment 11

Due April 16

These two programs are going to take some thought but they are not too difficult.

You have the tools: buttons, labels, etc.

Now it is time to use the tools to create interactive graphical programs.

No program should crash. And you should make some cool GUIs.

1. 123-Nim



Write a program that allows a person to play 123-Nim against the computer. The initial configuration of 123-Nim consists of a pile of 5 to 50 sticks or matches. Each player may take 1, 2, or 3 sticks on his/her turn, hence the name "123-Nim." The player who takes the last stick wins the game. The player should be shown the initial pile of sticks and given the opportunity to go first or second. The computer and player alternate turns until the game is over. When the game is over, a message appears stating who won, and the player may choose to quit or play again. A running total of the number of games played and the number won by the player is kept in some area of the screen.

A perfect strategy for this game has the computer choosing $n \% 4$ sticks, where n is the number of sticks remaining in the game, and $n \% 4$ is not zero. If $n \% 4 = 0$, then the computer randomly chooses 1, 2, or 3 sticks. Use this strategy for the computer.

You design the GUI any way that you like. However, the pile of sticks should probably just be a number on a label...don't display 50 sticks!

Use buttons and/or text fields for input. The choice is yours...but make it nice.

2. Car or Goat?



The Monty Hall Problem, derives its name from a classic TV game show, Let's Make a Deal, starring perennial host Monty Hall. During the show, a contestant is shown three closed doors labeled 1, 2, and 3. Behind one of the doors is a new sports car and behind each of the other doors is a rather handsome goat. Of course, Monty knows which door conceals the car.

After the contestant selects a door (1, 2 or 3), Monty opens one of the other two doors revealing a goat. Two doors now remain closed; one hides a car, the other a goat. The contestant is now given an option: stick with his/her original choice or switch to the other closed door.

Should the contestant switch doors? Keep the original door? Stay or switch, does it make a difference? How often does the contestant go home with the car and how often with a goat? You may be surprised by the answer.

Write an application that simulates this game. Use a random number to choose the door (1,2, or 3) that hides the car.

The GUI should display three doors labeled 1, 2, and 3. A player chooses a door by clicking on the door. After a player chooses a door, one of the other doors is opened revealing the picture of a goat, or perhaps just the word *Goat*. The player now has a choice: click on the original door again or switch doors by clicking on the other closed door. The player clicks on one of the two doors and the door is opened revealing the prize, a car or a goat.

Your GUI should also provide a *Reset* button allowing a contestant to play again. Include labels that show the number of times the game is played, the number of times the contestant switches

doors, the number of times a player wins the car, and the number of times the player chooses the goat. Include an *Exit* button.

Play the game many times always switching doors. Then play a series of games in which you never switch. What have you discovered?

Check this out it should give you a good idea of the game works:

[http://www.shodor.org/interactivate/activities/simplemontyhall/?version=1.6.0_10&browser=Mozilla&vendor=Sun Microsystems Inc.](http://www.shodor.org/interactivate/activities/simplemontyhall/?version=1.6.0_10&browser=Mozilla&vendor=Sun_Microsystems_Inc.)

or see the problem in the movie "21"

<http://www.youtube.com/watch?v=cXqDIFUB7YU&feature=related>

also on YouTube

<http://www.youtube.com/watch?v=mhlc7peGIGg>