Jeopardy Controller

Level: Intermediate

Last Used: Python - 2019-03-25

Author: Jim Burnham - TopClown@STEAMClown.org

Authored: March 25, 2019, Last Revised: 2019-03-25

What You Will Know & Be Able To Do:

- Have a greater level of understanding on how and when to use conditional statements including `if/else`, `for` and `while` loops
- Understand how to nest loops
- Be able to describe a random number and how to make your code check inputs in a random order

Prior Knowledge:

- Doing Math, and checking the results with conditional `if/else`
- Understand conditional loops, including `for` and `while` loops
- Understand how to nest loops
- Labs you may want to do first:
  - Xxx Links

Resources & Materials Needed:

- PC, Laptop or Raspberry Pi
- Link to GutHub for Source - Python Template to start from
- Link to online C++ or Python Compiler
  - Python 3 On-Line Interpreter - Tutorials Point
  - Python 3 Interpreter - Online GDB

How You Will Be Measured:

- Programming Lab Rubric link (coming Soon)
- You will turn in to the Google classroom… Check the Stream or the Programing category for C++ or Python

Scenario & Lab Instructions:

Overview, Introduction and Objective:

This is a lab where you will create a Jeopardy controller. You will get the input from 4-8 buttons. The key to this assignment is to make sure you get the inputs in a random order, so as to make it fair.

Typically when you read inputs, you read them in the same order. In the case of Jeopardy buttons, the first button scanned has an advantage over all the other buttons. In the case of a tie, the first button scanned will always will. Your task is to make it fair, and scan the buttons in a random order each time your check for inputs.

I will provide the hardware to plug into… and I will tell you the input and out pins to use. Tell me if you think you need additional pins. Here is the Jeopardy.py pin assignments

Your code should have the following sections:

1) Setup, Initialize, and reset players buttons
2) Have a “reset” button trigger a reset of all players, check to see that all buttons are reset, then light the “ready” LED
3) Start round timer. If player buzz in time, then they get to answer, else all players lose the round
4) Scan the 4-8 inputs to detect the first one pressed, and lock out any other inputs, and lock all players if time has expired. If the time has expired before a player “buzzes “ in flash all the player LEDs
5) Light the corresponding Player LED, and then go and wait for a “reset” input
6) Extra Credit if using a Raspberry Pi, if your Raspberry Pi plays the Jeopardy Time out song

Copy, Edit & Execute Code

Instructions for accessing any example code on Github. Update the template with your header comments and your code.

- You can start by copying Python Template as a starting point.
- Here is the Jeopardy.py pin assignments

Expected Output & What to Turn In:

1) Turn in - Sudo-Code / Flowchart of programme flow - This can be hand drawn, but electronic copy is needed for “A” work
2) Turn in - Code that implements the Flowchart

Print output:

```
Setting-Up and Initializing the Jeopardy game
```

```
      Resetting Players
      Waiting For Player Button Push . . . . . . . . . . . . . . . . ← dot for time out timer
      Player One is the winner
      Resetting Players
      Waiting For Player Button Push . . . . . . . .
      Player Two is the winner
```

Hint & Dig Deeper:
Pullup / down resistors - https://learn.sparkfun.com/tutorials/pull-up-resistors/what-is-a-pull-up-resistor

Appendix

Please maintain this License and Attribution information with any changes you make.

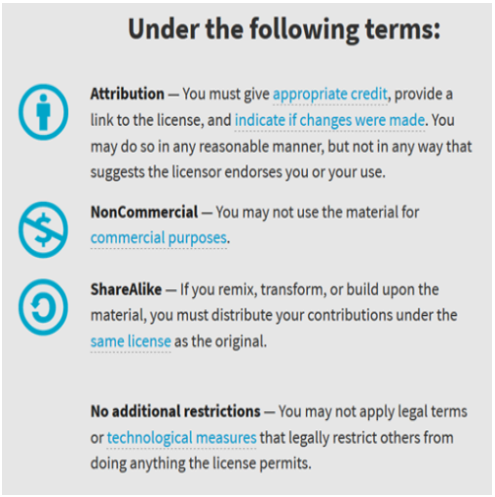
Where to get more information about this lab and the presentation that may go with it? Please visit STEAMClown.org or jim.The.STEAM.Clown's Google Site

Appendix A: License & Attribution

- This interpretation is primarily the Intellectual Property of Jim Burnham, Top STEAM Clown, at STEAMClown.org
- This presentation and content is distributed under the Creative Commons License CC-BY-NC-SA 4.0
- My best attempt to properly attribute, or reference any other sources or work I have used are listed in Appendix C

Appendix B: Coding License & Attribution

- This interpretation is primarily the Intellectual Property of Jim Burnham, Top STEAM Clown, at STEAMClown.org
- The programming code found in this presentation or linked to on my Github site is distributed under the:
  - GNU General Public License v3.0
  - European Union Public Licence EUPL 1.2 or later
- My best attempt to properly attribute, or reference any other sources or work I have used are listed in Appendix C



Appendix C: Primary Sources & Attribution for Material Used

- Doctor Rob, The Math Forum http://mathforum.org/dr.math/