ENGI1020 – Lab 5 Logbook

|  |  |  |
| --- | --- | --- |
| **Name** | **Lab & Date** | **Input & Output Devices** |
| Anton Guaman | Lab 5  November 8, 2020 | Input: Button and touch sensor  Other Components: LCD screen  Output: Console |

* 0 Preparation
  + - Summary of lab task in your own words
    - Summary of preparation (no need to repeat entirely)

In the lab preparation I selected the speed work project. For the design part I did a description of how the code should work using variable names that could be used in the lab. For the last part of the preparation the testing matrix. I did three test scenarios with their purpose on why they are being tested and their expected outcomes.

The lab is a game that involves two inputs: button and touch sensor. These are the buttons for the players. The game will use the LCD screen to let the players know when the game begins and game ends. The result of which player won will be outputted in the console. The game will begin with the LCD screen on white it will wait a random amount of time(the random module will be imported as well as the time module) before the LCD screen turns red showing the players the game is on and when there is a winner the LCD screen will turn green.

* 1 Implementation
  + 1.1 Converting Design to Implementation
    - Notes about how you took your design and implemented it
    - DO NOT just insert your code (though if it is useful to include lines of code to explain, you can)

In the script given the modules that we will use were already imported. They are necessary to make the code work.

﻿from time import \*

from random import \*

The first step was two use commands, so the user knows when the game is about the begin, it has begun and it has ended. We use the commands lcd\_rgb() to change the colour of the LCD screen to the ones needed. To select a random integer we use the command randint(,) and store it in a varible named ﻿waitTime. Then we place waitTime in the command sleep(waitTime) so it will wait a certain time before the game begins.

﻿lcd\_rgb(225, 225, 225) ---white colour

waitTime = randint(1, 6)

sleep(waitTime)

lcd\_rgb(255,0,0) ---- red colour

In the second step we define gametime that holds the parameters startTime and units. Inside the function we use if else statements to convert the time if needed from seconds to milliseconds or to minutes. The variable that will store the conversion is named duration. Finally, we use the return command for duration.

duration \*= 1000 --- for milliseconds

duration /= 60 for --- for minutes

In the third step we use a while loop that will determine if b1 or b2 are the winners(b1 and b2 are preset values that store the port number of the inputs) b1 = 2 and b =4. To access these values, we need to access the function ﻿checkDigitalInputs(b1, b2). We store this in the variable name winner so we can use it in the while loop. We also use sleep inside the function for a waiting time.

winner = checkDigitalInputs(b1, b2)--- outside loop

while winner == 0:

sleep(0.05)

(This last line is to update the winner)winner = checkDigitalInputs(b1, b2) ---inside loop

Finally, in the last step we print our results. In this lab we were asked to print the duration of the game and who is the winner. We will do this using the printing command.

* + To print the duration we will access the gametime function and print it using the parametrs start and ‘s’. If we would like the duration in milliseconds or minutes we would change the parameter ‘s’ to ‘m’ or ‘min’.

﻿ print(gameTime(start, 's'))

* + To print the winner, we just print what the variable winner stored.

print(winner)

* + 1.2. Errors Encountered While Implementing
    - Notes about errors encountered in Python interpreter or syntax problems
    - Include notes about solution
    - If useful, include names of errors

No errors encountered will implementing. Although, something important was to understand how the functions given in the script worked. This was key to understand how to address the implementation.

* + 1.3 Details and Facts from Implementation to Remember
    - Details about concepts that you learned from implementation
    - Technical details about interpreter, Arduino equipment, engi1020 module related to implementation
    - Anything else you think will demonstrate your learning and will be useful to remember for future

A key concept to know is what we have learned in class about functions. To understand a function read the contract. This was really helpful for this lab. Then understand what a global and local variable. These concepts were applied, and I understand them more after the implementation.

* 2 Testing
  + 2.1. Test Plan and Results
    - Should be in table format!!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Input Manipulation** | **Output**  **Expected** | **Output**  **Observed** | **Investigation** |
| **Test 1** | **How did you**  **change the**  **input?**  I pressed the input device(button) stored in port 2.  \*testing the output of winner | **What did you**  **expect to**  **observe?**  I expect the console to print 2 which stands for the input device(button) pressed. | **What did you**  **observe?**  I observed the number 2 in the console. | **Did they match?**  **If not, what did**  **you do (can**  **reference**  **Section 2.2)**  Yes, they did match! |
| **Test 2** | I pressed the input device(touch sensor) stored in port 4.  \*testing the output of winner | I expect the console to print 4 which stands for the input device(touchsensor) pressed. | I observed the number 4 in the console. | Yes, they did match! |
| **Test 3** | In the print section for the duration of game I set it like this:  ﻿ print(gameTime(start, 's'))  \*testing the duration | I expect the console to print a value like x.xxxxx for example: 5.098993.  \*time will vary since we are using a random int for the time interval until the game begins. | I observed the value: 4.82342423…. | Yes, they did match! |
| **Test 4** | In the print section for the duration of game I set it like this:  ﻿ print(gameTime(start, 'm'))  \*testing the duration | I expect the console to print a value like xxxx.xxxxx for example: 5000.098993.  \*time will vary since we are using a random int for the time interval until the game begins | I observed the value: 2268.041578 | Yes, they did match! |
| **Test 5** | In the print section for the duration of game I set it like this:  ﻿ print(gameTime(start, 'min'))  \*testing the duration | I expect the console to print a value like x.xxxxx for example: 0.083333.  \*time will vary since we are using a random int for the time interval until the game begins | I observed the value: 0.0635673…. | Yes, they did match! |

* 2.2. Errors Encountered while Testing
  + - Details about semantics or syntax errors discovered through testing
    - Expanding on “Investigation Column” above

No errors encountered while testing.

* + 2.3. Details and Facts from Testing to Remember
    - Details about concepts that you learned from testing
    - Technical details about interpreter, Arduino equipment, engi1020 module related to testing
    - Anything else you think will demonstrate your learning and will be useful to remember for future

I believe the key concepts of functions are necessary to remember. Like identifying the function contract and knowing how to interpret it. Also, identifying the local and global variables of the script is key.

* 3. Reflection and Conclusion
  + 3.1. Reflection Question Answers
    - Describe the **challenges** and **highlights** of working on a less 'constrained' project like in Lab 5.

I enjoyed working in a project like lab 5. This project was rewarding although it was challenging at intimidating at a first glance. This all went away when understanding the functions through their contracts, I learned how they work. This gave me confidence on how to address the implementation and testing. Being able to understand local and global variables was important as well. There were some decisions we had to make like the random integer part where it was more reasonable for the values to be close for the user to not wait that much. I learned this thanks to the explanation in the video. I think the major highlight of the lab is to see how it all came together. It felt great to see how the expected values appeared in the screen when testing.

* + 3.2 Additional conclusion Notes