Anthony Ma

EECS 1021 Object-Oriented Programming from Sensors to Actuators

2023/04/02

Major Project

INTRODUCTION

Are you currently at your computer staring into the screen with the lights off and blasting music at full volume? Even though you know it is unhealthy, you lack the motivation to fix it. This is where Desk Buddy comes in.

CONTEXT

Desk Buddy is a health monitor that utilizes Arduino hardware and object-oriented program to operate. It specifically checks for high levels of sound or light and a lack of light and warns you so that you make sure to take care of your health. This project takes on the 7th grand Engineering challenge of Advance Health Informatics as it seeks to provide direct information to you about your health through technology. Why is this important? Everyone in the modern day is consistently stuck to their desks and working on a computer without caring for their physical health, specifically their eyes and ears. Continuing to ignore the risks of long-term exposure to high levels of blue light and sound will harm your health in the long run, so changing that now is very important.

TECHNICAL REQUIREMENTS / SPECIFICATIONS

(CLO 2)

- The system should operate on an Arduino device running Firmata.
- The system should be programmed in a Java IDE such as IntelliJ.
- The system should read values from the Light sensor and Sound sensor.
- The system should update the OLED display with values, warnings, and the current time.
- The system should flash the LED whenever there is a warning.
- The system should store data in an ArrayList to be graphed later.
- The system should allow the user to turn off or turn on the display.

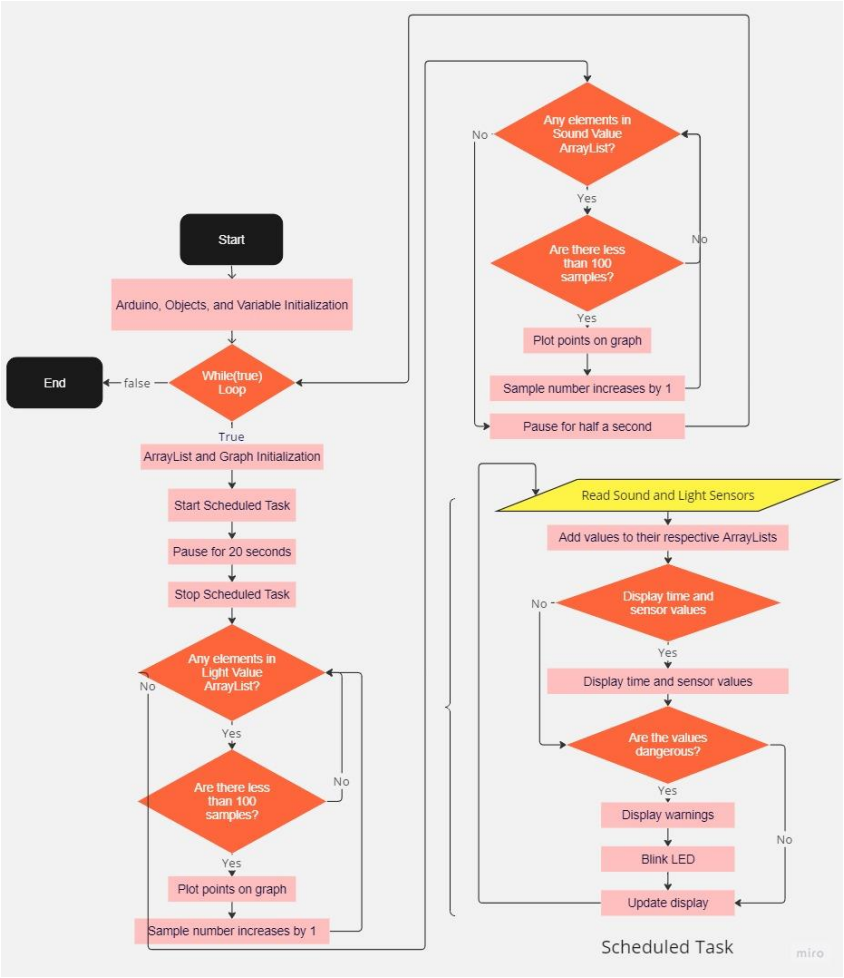

Figure 1 Flow Chart
https://miro.com/app/board/uXjVMV6tp6Q=/?share_link_id=478422965979

(CLO 4)

I planned on dealing with events in this program using the OLED display and LED. When either the light values are too high or low or sound values are too high, there is a warning sent to the OLED display along with the LED being told to blink. When the Potentiometer is to the left the screen will turn on, and when it is turned to the right, the display is turned off.

(CLO 5)

Using ArrayLists, I gathered data from both sensors before graphing and called upon them later using a method from the ScreenTask which was my TimerTask class.

COMPONENTS LIST

- Grove Arduino Kit
  - Utilizes Firmata to operate in IntelliJ.
- OLED Display
  - Connected to I2C 0x78.
- Potentiometer
  - Connected to A0.
- Light Sensor
  - Connected to A6.
- Sound Sensor
  - Connected to A2.

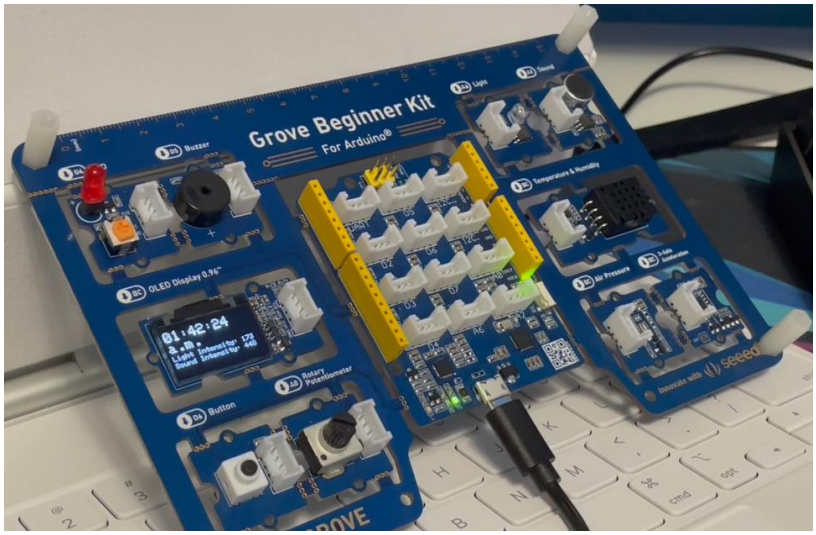

Figure 2 Setup

PROCEDURE

When I first started this project, I referred to the given Engineering challenges and sample ideas. While I had many ideas such as a smart home system, or auto-lock system, I ended up deciding on a Desk Buddy that would monitor your exposure to light and sound surrounding you. The coding process was straightforward as I knew right away that I wanted to use TimerTask to update the OLED screen constantly and feed the information to a dynamic graph. Where I found trouble was implementing a way for the graph to be dynamic, yet also store the data in a collection to be fed later on. My solution, in the end was to use a while loop that iterated endlessly so that the timer could be cancelled and initiated every loop as my problem before this was that the timer.cancel() method would completely halt data collection in its entirety. This problem mainly was a result of my using ideas from the minor project as I used the StdDraw class from the StdLib to create the graph and display values after a pause. Luckily the while loop solved my problem and only left me with a minor issue which was the OLED screen glitching out after each loop. To remedy this, I ended up adding a delay which would give the OLED screen time to reset and not be overwhelmed. An idea that I had was to display the current time on the OLED screen so that it could act similarly to a digital clock too. At first, I tried using System.CurrentTimeMillis() but realized it was not what I wanted, but luckily after a quick search, I found a class with a method that stored the current time in the format of a digital clock as a string which was called LocalTime**[2]**. One tricky challenge I had was trying to find the sweet spots for the sensors where it would register as a warning or not register at all. This is mainly because the sound intensity sensors take more sporadically compared to the light sensor which took some time to accommodate for. Another challenge that I faced was clearing the graph after hitting the maximum number of samples. This is mainly because the clear() method would remove all points for the first for-each loop but would still add points for the next for-each loop which was not what I wanted. I fixed this issue by adding the condition that after both for-each loop, once the sample number was larger than 100, it would clear the graph only then which allowed both sets of data to be consistent with each other. **(CLO 5)** In the end, my project changed slightly throughout the process of coding but overall my goals stayed the same and I was able to adjust accordingly to any challenges I faced.

TEST
(CLO 1)

I tested that this system worked by utilizing System.out.println() statements throughout the main method to keep track of values. This was especially helpful for the number of samples as when I was having trouble with the clear() method stated above, I thought the issue at first was that the sample numbers were increasing at different rates when theoretically they should be increasing at the same rate. Luckily I was correct and the sample number was not the issue after checking it over.

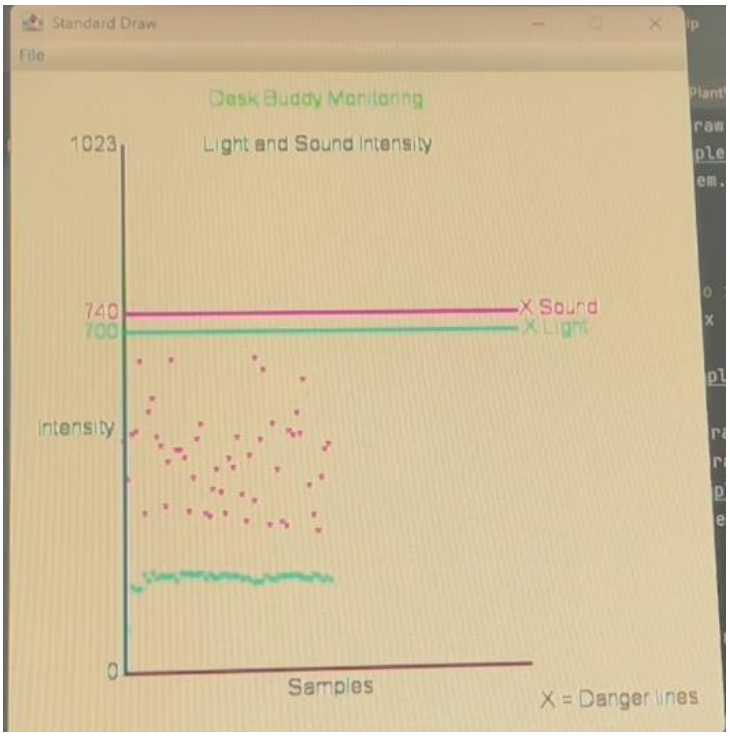

Figure 3 Graph of sensor values

LEARNING OUTCOMES

Yes, my project addresses all of the learning outcomes.

- **(CLO 1)** I demonstrated the ability to test and debug a given program through numerous test runs and System.out.println()s to check if the values are correct.
- **(CLO 2)** I demonstrated my ability to build an application that meets the given requirements from a given problem specification and API, as I used both Firmata4j and StdLib to create my project. These libraries are appropriate for this project as they allow for the use of the Arduino board and the capability to graph respectively.
- **(CLO 3)** I demonstrated the ability to use ready-made collections to solve the problem of this project by storing both light and sound sensor values in an ArrayList each to be plotted later on.
- **(CLO 4)** I demonstrated the ability to build an event-driven application by using the if, else-if and else conditional statements to create a reactive system that performed different actions based on those values such as turning on the OLED screen when the potentiometer is turned to the left or when a warning is sent when the sound intensity is too high.
- **(CLO 5)** I demonstrated the ability to use an object-oriented program to solve engineering problems by using Java in IntelliJ where I utilized libraries such as Firmata4j and StdLib along with classes such as Timer/TimerTask, LocalTime**[2]**, and ArrayList to solve my engineering problem, which was the 7th Grand Challenge of Advance Health Informatics.

CONTINGENCY

Yes, I did have an idea in mind that I mentioned before that I really wanted to try but was not able to in the end. This is because the Idea that I had was a smart home system that required the use of a physical keypad and the Arduino IDE to implement as it was something that I had experience with in the past. I decided not to implement this idea due to time and money constraints along with the fact that the Arduino IDE does not fall under the specifications for this project. Next time, I would try to do more research into finding methods that could allow me to write the same kind of program but without the physical keypad and Arduino IDE such as using the computer keyboard as user input or other kinds of actuators that could serve the same purpose in the program. Looking ahead to ENG 4000, a lesson that I learned that I would want to apply to the capstone project is learning to put more time into the planning stage before testing as it saves far more time and allows for more contingency plans in case one idea does not work the way I want.

ADDITIONAL MATERIAL

I believe that this project is a candidate for “exceeding expectations”. This is due to the real-world applications of the Desk Buddy as many people already utilize some kind of Digital clock on their desk. Although this model may not have fancy features such as being a wireless charger or a pencil holder, I believe that the current state of the product can be implemented into similar applications to make it even more marketable as there are not many digital clocks that can support you as a health monitor on the market. What makes this product even more important in this day and age, is how necessary it is without people realizing it. A majority of us tend to disregard our health when using technology by not noticing how bright a screen is or how loud our music is. So by creating a product that brings the user attention to these issues, it can help prevent people from damaging their eyes and ears when sitting at their computers everyday.

CONCLUSION

In conclusion, I believe that I was able to exceed expectations and achieve all the necessary learning outcomes for this project. Although there were some lows, there were just as many highs that allowed me to create a project I could be happy with.

REFERENCES APA Style

[1] Oracle. (2023, January 9). *DateTimeFormatter (Java Platform SE 8)*. Java™ Platform Standard Ed. 8. https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html

[2] Oracle. (2023, January 9). *LocalTime (Java Platform SE 8)*. Java™ Platform Standard Ed. 8. https://docs.oracle.com/javase/8/docs/api/java/time/LocalTime.html

[3] Seeed Studio, Inc. (n.d.). *Grove Beginner Kit for Arduino: Seeed studio wiki*. Seeed Studio Wiki RSS. https://wiki.seeedstudio.com/Grove-Beginner-Kit-For-Arduino/

[4] Smith, J. A. (2022, February 25). *Easy java + arduino with Firmata (updated)*. York University. https://www.yorku.ca/professor/drsmith/2022/02/25/easy-java-arduino-with-firmata/

[5] The Trustees of Princeton University. (n.d.). *StdDraw*. Princeton University https://introcs.cs.princeton.edu/java/stdlib/javadoc/StdDraw.html