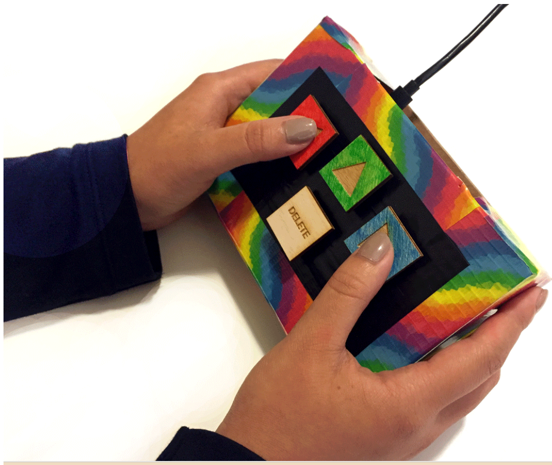
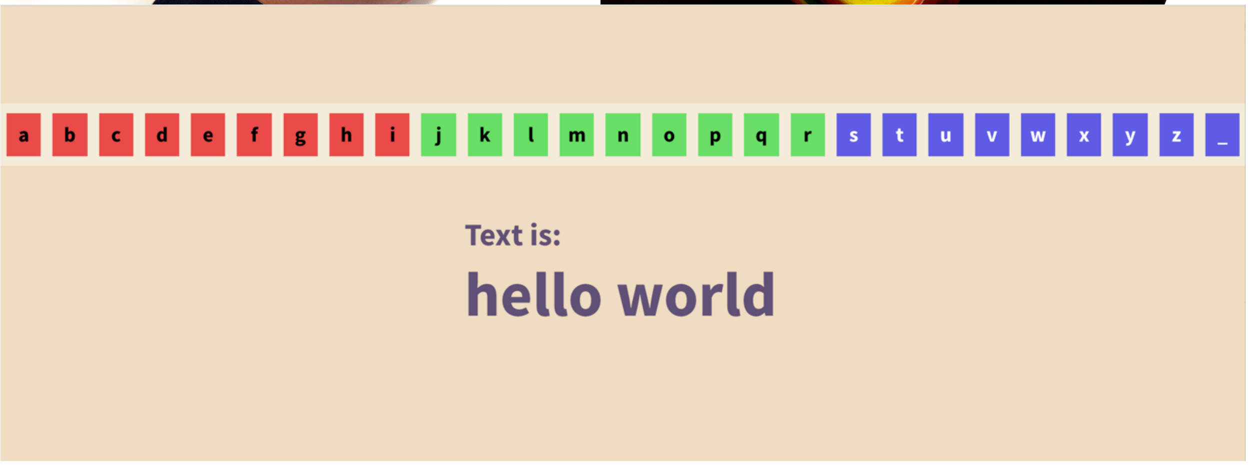
Serena Chang
IDD Text-Entry Project
HW 2

Github repo link: https://github.com/sychang/hw2_sychang_ternary

Video demo link: https://youtu.be/9Idxmy4s4xc





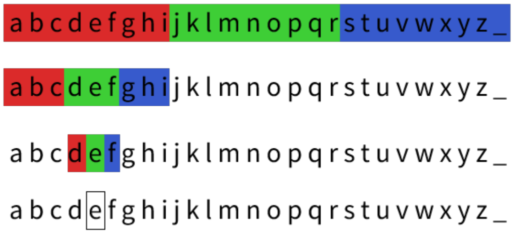
a b c d e f g h i j k l m n o p q r s t u v w x y z _

Text is:

hello world

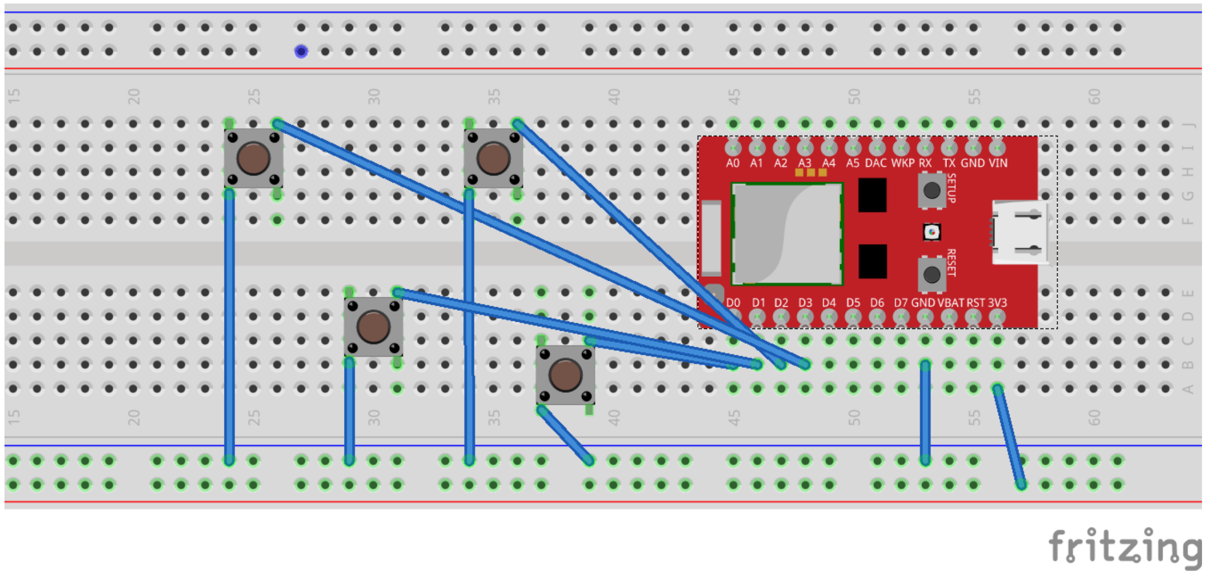
I chose a ternary search-based text selection technique where each of the 27 characters (all lowercase English letters and a space) are selected with a unique combination of three button presses. The text is displayed on a web server and the user first selects whether the character they desire is in the first, second, or third third of the alphabet list (space is at the end of the third third), denoted in red, green, and blue respectively. After the first selection, that third is then divided into thirds, and so on until only one letter remains selected. If they error before the letter selection, it will re-expand the selection so they can restart selecting that letter. Otherwise, the last typed letter will be selected.

For instance, to select "e" one can press the red, green, and green buttons to narrow down the selection, as shown on the right.


a b c d e f g h i j k l m n o p q r s t u v w x y z _
a b c d e f g h i j k l m n o p q r s t u v w x y z _
a b c d e f g h i j k l m n o p q r s t u v w x y z _
a b c d e f g h i j k l m n o p q r s t u v w x y z _

The Arduino code works by using an internal pull-down setup to listen for a button press. The IDE sends a unique serial output for each button pressed. The Python script, key_entry.py reads in the serial input and assigns a value to each character (0 for red, 1 for green, 2 for blue, 3 for not selectable), and updates the state of the data structure, all_info, in which the word and acharacter:value dictionary are stored. Another Python script, server.py simultaneously has threads running a web server while simultaneously reading output from key_entry.py. A jQuery file issues a JSON request to get the data from all_info and updates the CSS classes of each letter and refreshes the server every 300 milliseconds to create a UI that updates with each button press.

The physical device includes wiring 4 buttons to ground and the microcontroller pins. They are arranged so a user can place their middle three fingers ergonomically as the toggle controls and access "DELETE" if necessary, or they can hold it with both hands, similar to a video game controller. I initially wanted to a create a hardware interface with 27 RGB LEDs to change color based on the state of the selection. However, I quickly realized there were not enough pins on the Redbear Duo to support that many LEDs, so I switched focus to a software based user interface and put more effort into the virtual display. Below is the hardware wiring diagram.



From this assignment, I learned to start simple and test functionality as I went. I started by testing simple button functionality before building the software side. The assignment was decent in scope – stressful to think about at first and time-consuming to refine, but the constraints on the switches gave us good guidelines to work with. I learned how to create a web server and program jQuery to respond to external input, so this project pushed me to be more creative and learn new skills.