Haley Rose Flores
CSCI 1600: Real-Time and Embedded Software
December 19, 2023
Professor Milda Zizyte

Capstone Abstract: Crafting a Personalized Visual Experience

Short Description

This mobile application lets users wirelessly personalize their brain wave visualization by altering the colors. Users will first need to select the brain wave they want to change the color of. Users can then adjust three sliders to change the red, green, and blue values to create their color. A square below the sliders will change in real time so users know the color they are creating. The values from 0-255 for red, green, and blue will be displayed below. Users will need to press a button to send the new changes to the server so the visualization can change accordingly.

Summary of Design Challenges

My initial design task involved determining the most efficient method for users to generate their preferred colors. I aimed to ensure that the application was both user-friendly and intuitive. After careful consideration, I concluded that sliders, coupled with a real-time visualization of the selected color, were the most efficient option. Subsequently, my next design challenge centered around determining the optimal approach for transmitting outputs from the mobile application to the Arduino's serial monitor. Recognizing that Processing captures input from the Arduino serial monitor, I understood the necessity of a translation process for this to function seamlessly. Given the requirement for wireless communication, I conducted research and decided to build a server to facilitate this connection.

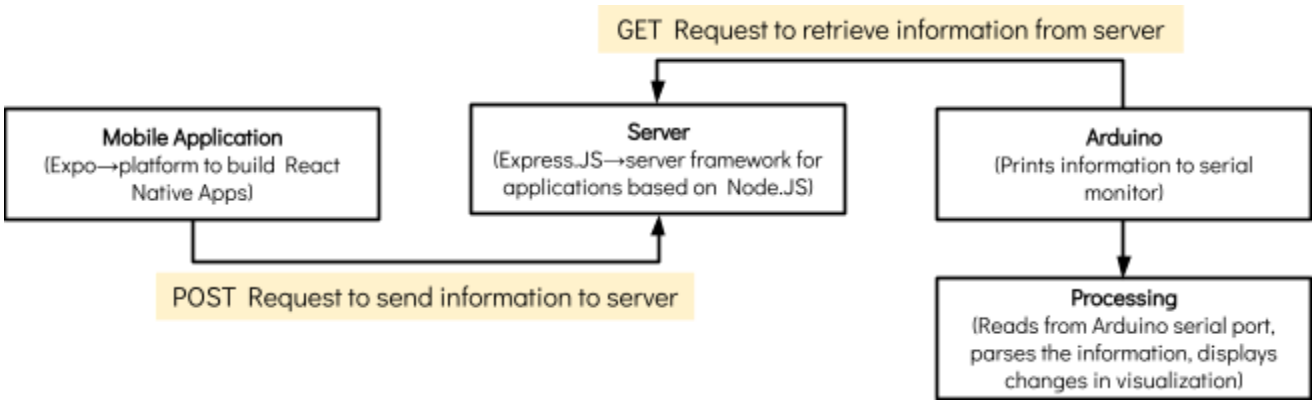
Why Interesting Problem

In the development of our final project, a primary focal point was ensuring accessibility for all users. We acknowledge that our project generated only a visual representation of the brain wave data, presenting potential challenges for users with visual impairments, specifically those affected by color blindness. With that being said, this mobile application empowers users to personalize their visualizations, enabling them to select colors that are most discernible to them.

Approach Taken

I started by developing a mobile application utilizing Expo for quick deployment and swift testing. I then built a server using Express.JS to facilitate communication between the Arduino. When the “Send To Server” button is pressed on the app, a POST request is made to the serve, sending the red, green, blue, and brain wave type values. Those values are stored on the server. The next pass of information occurs on the Arduino. Once the headset is connected and receiving brain wave input, the Arduino consistently makes GET requests to the server to retrieve the information and print it out in the serial monitor. The final steps involved adding a new if-statement for the serial input parser so Processing would recognize this as information coming from the application and process it accordingly. Once it recognizes it, a new function that changes the color of the visualization is called, which passes in the new color values from the serial monitor. The visualization is then altered.

Haley Rose Flores
CSCI 1600: Real-Time and Embedded Software
December 19, 2023
Professor Milda Zizyte



Challenges Encountered

Embarking on the development of a mobile application posed a challenge for me, as it was all new to me. To overcome this hurdle, I conducted research and utilized videos to familiarize myself with the intricacies of mobile app development. However, the most challenging obstacle arose when I grappled with the task of transferring information from the application to Processing to reflect changes in the visualization.

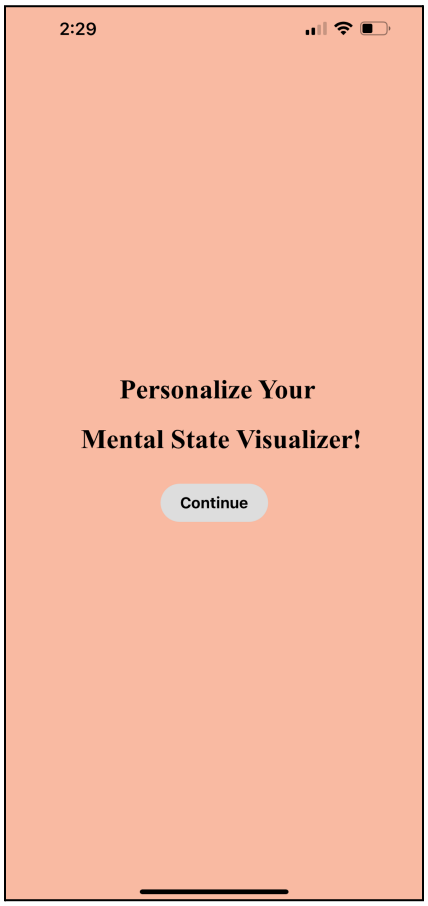
After researching, I concluded that constructing a server would offer the most effective solution. Initially, my approach involved making a POST request to the server to transmit information from the application, which would then be written to the Arduino port. This method, however, introduced the complication where Processing could not read from the same port that the server was writing to in order to pass the information. To resolve this issue, I experimented with techniques to open and close the port during color transmissions from the application, but I could not discover a solution. Consequently, I pivoted to an alternative strategy. I decided to initiate a GET request from the Arduino to retrieve information, allowing it to be printed in the Arduino serial monitor. This modification enabled Processing to read from the Arduino port seamlessly, ultimately presenting the accurate visualization of the data from the application.

Haley Rose Flores
CSCI 1600: Real-Time and Embedded Software

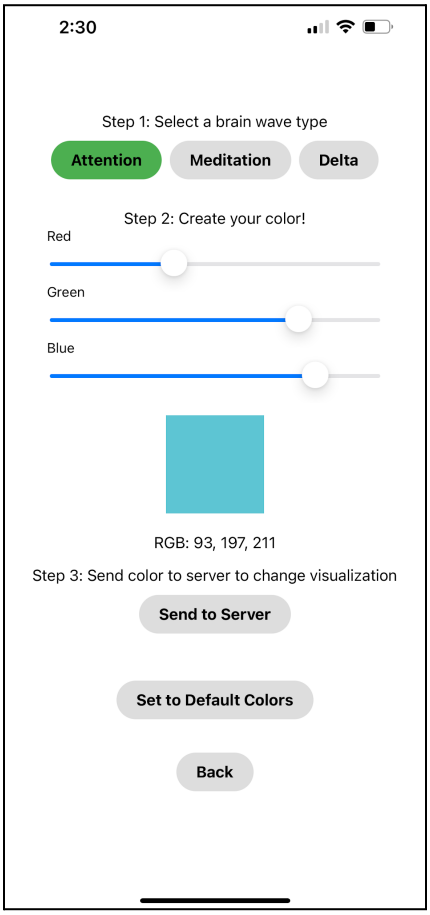
December 19, 2023
Professor Milda Zizyte

Photo

Home Screen



Color-changing Screen



Steps to Run Application:

1. Enter the project folder in the terminal
2. Upload the Arduino code onto the Arduino and keep the serial monitor closed
3. In a terminal window, start the Expo Go application by running the command '*npx expo start*' (Expo Go must be downloaded on mobile phone, then you can scan QR code from terminal)
4. In another terminal window, start up the server by running the command '*nodes server.js*'
5. Open processing and observe the changes being made to the visualization

GitHub Link: https://github.com/hflores02/CapstoneProject/tree/main