Boston University
**Electrical & Computer**
Engineering

# Boston University
# Electrical and Computer Engineering
## EC463 Senior Design Project

# Second Prototype Testing Plan and Report

ICHI

By

Team 10

ICHI

Team Members

Ande Chen achen965@bu.edu
Grace Kim vzgrace@bu.edu
Henry Galindo hgalindo@bu.edu
Jeanette Villanueva jivillanu@bu.edu
Ronald Huang rhuang@bu.edu

## Required Materials

Hardware:

- Raspberry Pi Zero 2W
- Joystick
- 3 push buttons
- 2 switches
- MCP3204 Analog to Digital Converter
- USB Microphone
- Tobii Eye Tracker 5
- Pisugar 1200mAh Battery Module

Software:

- Unity
    - Gaze Point with Tobii Eye Tracker 5
    - Recognize mouse clicks
- Tobii Experience App
    - Recognize user’s eyes

## Hardware Pinout

| Raspberry Pi Pin # | Usage/Description |
|---|---|
| GPIO4 | Right push button input |
| GPIO22 | Left push button input |
| GPIO10 | Serial Data In for Analog-to-Digital Converter (ADC) for Joystick |
| GPIO9 | Serial Data Out for ADC |
| GPIO11 | Clock for ADC |
| GPIO6 | Push to Talk button input |
| GPIO8 | Chip Select/ Shut Down for ADC |
| 5V | 5V connection to battery pack |
| GND | Grounding connections for all components |
| 3V3 | 3.3V power for all components |

**Setup:**

The Raspberry Pi Zero 2W should be plugged into a power source. The hardware inputs (buttons, joystick) as well as the external microphone connected to the Raspberry Pi Zero 2W will be wirelessly transmitted over Bluetooth. The Tobii Eye Tracker 5 should be plugged into the PC and attached to the monitor. The user will then open the Tobii Experience App and follow the steps in connecting their eyes with the tracker. After that, the user will run the Unity program which will test how accurate the Tobii Eye Tracker 5 is along with detecting left and right mouse clicks.

**Pre-testing Set up Procedure:**

Software Side:

Bluetooth Communication

1. Run Python script

Unity

1. Open Unity and make sure the program has no errors prior to running

Hardware Side:

1. Make sure Raspberry Pi Zero 2W is plugged into PC
2. Plug in Tobii Eye Tracker 5 and mount on monitor

**Testing Procedure:**

1. Connect Raspberry Pi Zero 2W to computer, compile and upload Python program
2. Test each component (buttons, joystick, microphone)
3. Run the Unity program

**Measurable Criteria**

The criteria for successful running and output is as follows:

I. The Raspberry Pi Zero 2W should successfully detect hardware inputs:
   A. Push button inputs will be printed to the Tera Term terminal as “Left Click” and “Right Click:
   B. Joystick inputs will be printed to the  as X and Y coordinates representing joystick movement

II. Push to talk button should mute microphone input unless pressed, and switches should keep power from buttons and components
   A. Microphone inputs will be printed to as the minimum, maximum, and delta values when push to talk button is pressed

III. Unity program should successfully change the color of the sphere depending on what type of click it is (left or right)

A. Spheres are initially gray and will turn blue when gaze point is on the sphere
B. When the user looks at the sphere and right clicks, the sphere will turn black. If left clicked, it will turn pink.
    1. Sphere should remain unchanged if not looked at, but clicked on

**<u>Degree of end-to-end completion achieved and demonstrated by this testing</u>**

Demonstrated by this testing, we have completed half of the entire scope of our project. On the software side, we were able to build the appropriate game environment that is compatible with the Tobii SDK and successfully integrate the eye tracker to interact with objects in the game. Our game demonstrates the use of the eye tracker as well as mouse clicks that are registered in combination with the user's eye gaze. We were also able to build a script within the raspberry pi environment that registers the button clicks as well as joystick inputs. On the hardware side, we were able to set up the Raspberry Pi Pico 2W with the external battery, joystick, analog to digital converter and push buttons. We have also set up the means to use the USB microphone for speech to text detection. The remaining steps include moving the hardware design to a more permanent set up and 3D printing the shell and buttons of the controller. In addition, we have to refine the eye-tracking component of the product and build the communication between the hand-held controller and computer actions.

**<u>Discuss conclusions based on test data</u>**

Based on our results, we were able to successfully register the button clicks and joystick movements. Additionally, we were able to demonstrate the microphone picking up noise only when the push-to-speak button is pressed. Further, the speech-to-text function accurately transcribed audio input from a defined microphone. Thus, we can conclude that we have several features working appropriately, but separately. We now need to work on integrating all of our functions together as well as building the eye-tracking component of the project. Our next task is to have the Raspberry Pi communicate with the computer that the eye tracking program runs on. It is still uncertain if the button clicks on the Pi can be registered as actual mouse clicks. In the Unity program, the objects on the screen change color when the gaze point is focused on it and when the mouse is clicked. Theoretically, we want both Unity and the Raspberry Pi to run at the same time and the button clicks would replace the mouse clicks.