Sprint Planning Document (Sprint 2)

February 18 - March 25 2025

Olivia Monteiro, Alicia Reed, Danny Steuer, Archer Taylor, Eli Weber

High-level Project Overview

Project Mission:

- Provide a client-server architecture to develop educational systems to support STEM interest in K-12 students

Problems We Are Solving:

- There is limited previous work in the field of quadruped research in K-12 students
- Development of interesting, child-friendly educational resources
- Develop STEM interest in young students

Project Overview (High-Level Features):

- **Cloud Server**
  - Cloud connection to link the microphone & voice files with the PuppyPi
  - Support for quadruped control, sending and receiving commands, sending and receiving sensor data
- **Voice Recognition Using LLM**
  - Using a large language model to recognize & process voice data into words to transmit as commands to PuppyPi
- **ROS Programming**
  - Using action groups to take the PuppyPi through a series of preprogrammed motions
    - Sit, lay down, moonwalk, shake

Sprint 2 Planning

Sprint 2 Goals:

1. **Link voice, cloud, and ROS**
2. **Be able to use USB microphone in docker**
3. **Finish reverse engineering WonderPi app**
4. **Work on chaining commands, length of commands, command interruption, and other aspects of app - color tracking, face detection, following mode, etc. (continue through Sprint 3)**
5. **Work on continuous listening (continue through Sprint 3)**
6. **Improve latency & noise (continue through Sprint 3)**
7. **Test commands/command interpretation (continue through Sprint 3)**

Sprint 2 Deliverables:

- **Link voice, cloud, and ROS**
    - **Assigned:** All team members
    - Allow voice programs to run ROS commands
- **Be able to use USB microphone in docker**
    - **Assigned:** Eli Weber, Archer Taylor
    - Check if the USB microphone is usable in docker container
- **Finish reverse engineering WonderPi app**
    - **Assigned:** Alicia Reed
    - Look at app provided by HiWonder and see if there is anything helpful in the code for our project
- **Chaining commands, length of commands, command interruption, other app modes**
    - **Assigned:** Olivia Monteiro, Eli Weber
    - Be able to chain multiple commands together, set commands to be run for a certain time, design control flow for if command get interrupted with other commands, add functionality from other app modes
- **Continuous listening**
    - **Assigned:** Archer Taylor, Danny Steuer, Olivia Monteiro
    - Run loop program at startup to register wake word & commands
- **Improve latency & noise**

    - **Assigned:** Archer Taylor, Danny Steuer
    - Improving latency & noise with voice control
- **Test commands/command interpretation**
    - **Assigned:** All team members
    - Testing various ways of calling the same commands to be sure they run appropriately