V.R.O.O.M. User Manual

Contents

1 Introduction

This is the user manual for the maze puzzle game: **V.R.O.O.M.**, which is a game in which players control the Maze Navigator through a real world maze with hidden walls. The game works by using software to create virtual walls that are unknown to the player. The player will use the game controller or microphone to control the Maze Navigator through the maze. As the player navigates the maze, they will encounter walls that block their path. It is hugely beneficial for the player to remember the locations of the walls. Once the end of the maze is reached, the player wins!

2 System Requirements

- Computer with Internet access.
- Python version 3.1+ installed.
- Conda installed
- One USB-C port for the provided wireless microphone.

3 Setup and Installation Instructions

3.1 First Time Setup

3.1.1 Setup using executable

1. Download Tailscale and create an account:
   (a) Add Controller: Tailscale Controller Invite
   (b) Add Navigator: Tailscale Navigator Invite
   (c) Once both devices are added, all three programs work together as long as they are all connected to any network (*eduroam* is set up by default on both the Maze Navigator and Controller).
2. Download and run executable:
   - **Mac/Linux (M1/ARM)**: Maze Program Download (ARM)
   - go to the path of the downloaded `maze` file and run the following:

     ```
     chmod +x ./maze
     ```
   - **Windows (x86):** ADD CORRECT DOWNLOAD LINK
   - The downloaded file should already be an executable.
   - **Mac/Linux (Intel)**: ADD CORRECT DOWNLOAD LINK
   - Go to the path of the downloaded `maze` file and run the following:

     ```
     chmod +x ./maze
     ```

3.1.2 Setup Manually

1. Download Tailscale and create an account:
   (a) Add Controller: Tailscale Controller Invite
   (b) Add Navigator: Tailscale Navigator Invite
   (c) Once both devices are added, all three programs work together as long as they are all connected to any network (*eduroam* is set up by default on both the Maze Navigator and Controller).
2. GitHub clone repository:

   ```
   git clone https://github.com/180D-FW-2024/Team5
   ```

3. Setup Virtual Environment:
   (a) `conda create -n maze_env python=3.10`
   (b) `conda activate maze_env`
4. Navigate to /maze-program folder inside of Team5

   ```
   cd /*replace with path*/Team5/maze-program/
   ```

5. Install dependencies found in `requirements.txt`:

   ```
   pip install -r requirements.txt
   ```

6. **(If you didn't get any errors in step 5, you can skip this step)**. The following system-wide dependencies might be required: `portaudio` and `ffmpeg` There are many ways to download and install them. We recommend using HomeBrew. Once HomeBrew is installed, you can use the following commands to install the dependencies:

   ```
   brew install portaudio

   brew install ffmpeg
   ```

   After installing, repeat step 5.

3.2 Setup Each Time the Game is Played

1. **Maze Navigator:**
   (a) Plug the power bank into Raspberry Pi 4 using a USB-C to USB-A cable. Plug the USB-C into the RPi and the USB-A into the power bank.

(b) Turn the Maze Navigator over and make sure that all 4 batteries are installed underneath.

2. **Controller:**

   (a) Plug Raspberry Pi 4 into power bank.

3. **Voice Commands:** Plug the microphone dongle into a USB-C port on your computer and hold down the power button on the back of the microphone until the light on the front starts flashing.

4. **Give both the Maze Navigator and Controller at least 1 minute to start up and run their respective programs.**

5. **Run the Program**:

   (a) If you setup the program using the executable found in section 3.1.1:

   ```
   run the executable maze.exe or use ./maze in the terminal
   ```

   (b) If you manually setup the program using section 3.1.2, run the following in the terminal:

   ```
   python /*replace with path*/Team5/maze-program/maze.py
   ```

4 User Interface Overview

The user has two main ways to play the game:

4.1 Controller

- The three button controller is provided to the user.
- Each button has a movement function for the Maze Navigator:
  - **Left (L):** Rotates the Navigator 90 degrees to the left.
  - **Forward (F):** Moves the Navigator forward one square.
  - **Right (R):** Rotates the Navigator 90 degrees to the right.

4.2 Voice Commands

- The microphone must be connected to the user's computer **before** running the program.
- Voice commands are activated by clicking the **Enable Voice Commands** button in the GUI of the maze program.
- To move the Maze Navigator, speak a sentence into the microphone containing one of the following keywords:

  - "Left", "Forward", or "Right".

- The user can speak in sentences instead of just the keyword. For example, `"Please move forward."` is a completely acceptable input and will move the Maze Navigator forward.
- If the user speaks a sentence containing more than one keyword, priority is given to the keyword spoken first. For example, if the user says, `"Move forward, right."`, the Maze Navigator will move forward and not right.
- Voice commands can be disabled at any time by clicking **Disable Voice Commands** in the GUI of the maze program.

5 Specific Features

5.1 GUI Buttons

1. The **Regenerate Maze** button can be used to generate a new maze layout.
2. The **Arrow Keys** can be used to rotate and move the car.
3. The **Enable/Disable Voice Commands** button enables or disables voice commands.

6 Known Issues and Troubleshooting

- If either/both the Maze Navigator and Controller fail to properly connect, the terminal where you ran the main program will output:

  ```
  Maze Navigator/Controller connection timed out.
  ```

  To resolve this issue:

  1. Make sure that Tailscale is correctly setup and that both the **maze-navigator** and **maze-controller** are visible in the Tailscale app.
  2. Ensure that both the Maze Navigator and Controller are running **before** the main program is started.
  3. If problems persist, please restart all three devices and follow the steps found in Section 3 again.

- If you are having trouble getting the voice commands to pick up input: make sure that their aren't any loud/irregular noises that happen when you start the program. The microphone is calibrated for ambient noise when the program is launched, and a loud noise during that process could cause the program to filter out your voice.