



# **Design Document**

## **Wonderland Software Framework**

Date: Aug 31, 2022

## **Table of Contents**

<b>System Overview .....</b>	<b>3</b>
<b>System Design .....</b>	<b>3</b>
<b>Subsystem Design .....</b>	<b>4</b>
<i>Director Subsystem.....</i>	<i>4</i>
<i>Robot Thespian Subsystem.....</i>	<i>5</i>
<b>Workflows .....</b>	<b>5</b>
<b>Demo Run .....</b>	<b>6</b>
<b>Next Steps.....</b>	<b>Error! Bookmark not defined.</b>

# System Overview

For the Wonderland project you will be using the software framework outlined in this document. Each team will construct a single robot thespian. The class will collaboratively write a version of *Alice in Wonderland*. Each team will write lines for their character and upload those to a server called the Director. Once the play starts the director will send the lines to the robot at the correct time. The robots will need to process/decode and then recite the lines to form a cohesive performance with all the robots.

# System Design

The system includes **the director** and **the robot thespians**. The system input is a script that is uploaded to the director that contains commands and cue times for each robot thespian. The output of the system will include the transferring of commands from the director to robots at the correct cue time. The robots will be responsible for processing/decoding the commands to trigger playback their scripts.

Teams are responsible for determining the commands that will be sent to their specific robot. Then uploading the commands to script file. Once all students have included their commands and cue times in the script file it will be uploaded to the director. The software framework can be found on this GitHub page: <https://github.gatech.edu/tbrothers3/ECE-3872>. Teams must use the software provided and change only the parts of the code dictated by the comments.

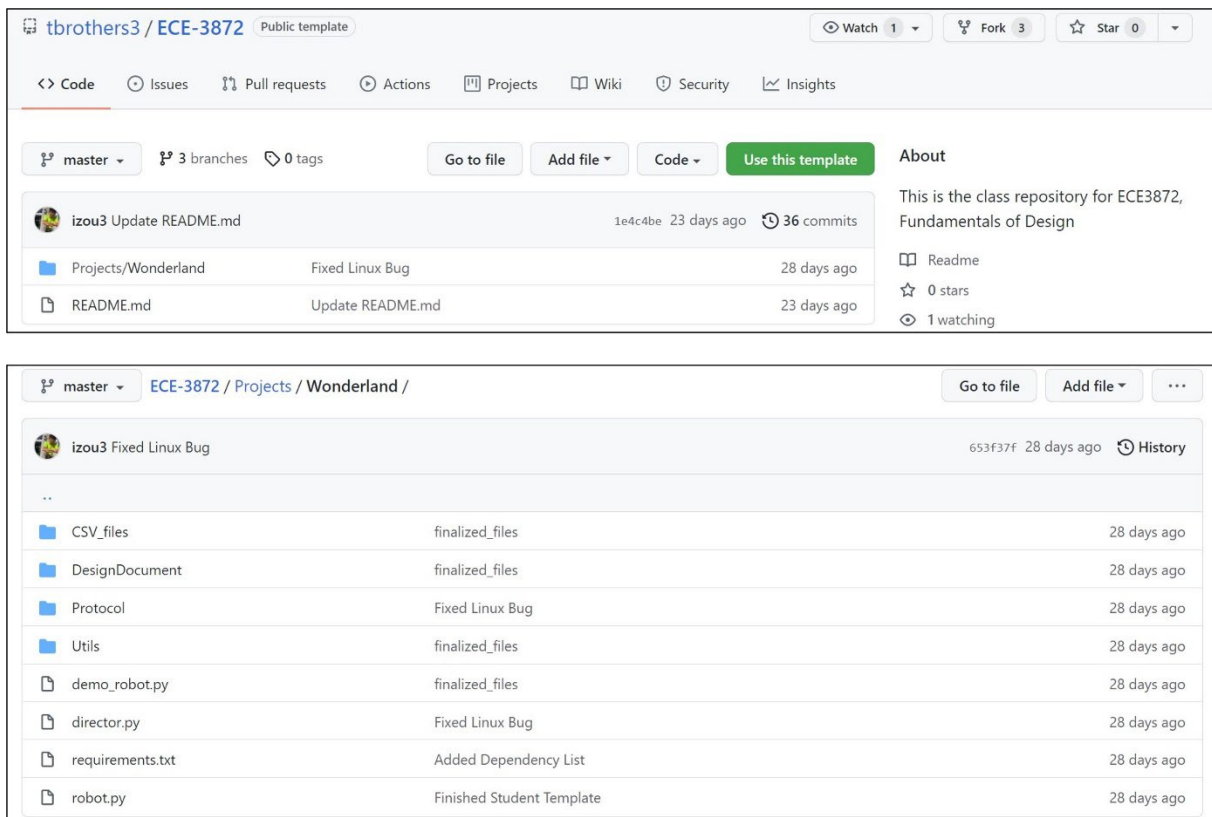


Figure 1. GitHub repo containing software (top). Sub-folders in the repo with files to run/edit (bottom).

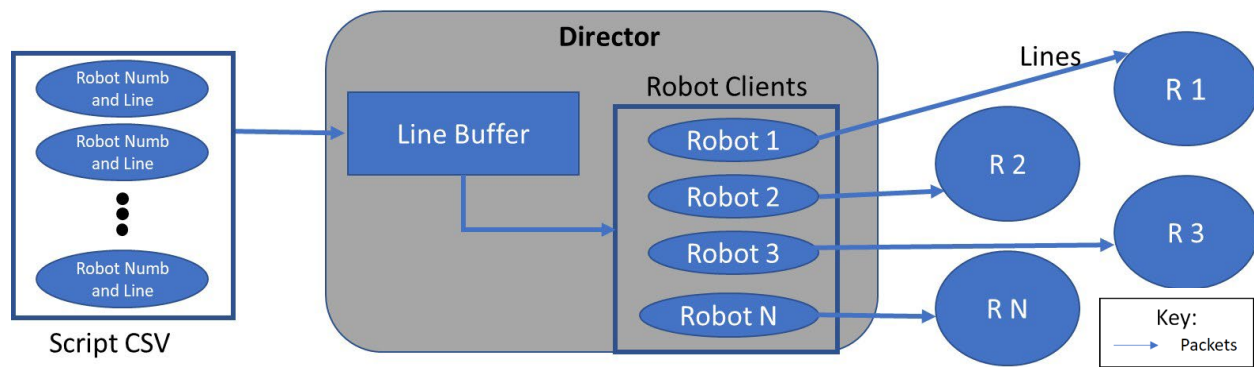


Figure 2. System Block Diagram

### Specifications:

<i>Machine</i>	Raspberry Pi 3 B (Raspberry Pi 3, 4 and Zero W will also work)
<i>OS</i>	Raspberry Pi OS (Linux Distribution)
<i>Python Version</i>	Python 3.7 and newer
<i>Dependencies</i>	Wi-Fi Hotspot (Tested on Windows Laptop Hotspot)

## Subsystem Design

The system's design incorporates two software subsystems: the Director and the Robot Thespian. Each subsystem is written in Python3 and will rely on a private network with limited capacity and bandwidth.

### Director Subsystem

- Clone the repo into the desired working directory on the Linux machine:

```
git clone https://github.gatech.edu/tbrothers3/ECE-3872.git
cd Projects/Wonderland
```

- Install the necessary dependencies:

```
sudo python3 install -r requirements.txt
sudo python3 install keyboard
```

- Create or edit an appropriate CSV file to define the robot order, latency, and command. Then use the following to run the director.

```
sudo python3 director.py <path to csv file>
```

## Robot Thespian Subsystem

- Clone the repo into the desired working directory on the Linux machine:

```
git clone https://github.gatech.edu/tbrothers3/ECE-3872.git
cd Projects/Wonderland
```

- Edit the `robot.py` file to code in your robot's corresponding logic. Please follow the comments for guidance.
- For deployment into actual environment, obtain the director IP and edit the script. Then on bootup, you must run the `robot.py` script.

## Workflows

The Director and Robot thespian subsystems each have their own workflows that run in tandem. The Director establishes individual connections with each of the robots. It then sorts these robots based on the order of the lines each is supposed to recite and then sends out commands to every robot at the correct time. Each of the robot thespians have a common basic workflow. Upon bootup, the thespian connects to the Director and listens for its commands. It then carries out the required action corresponding to the decoded message.

### Director workflow:

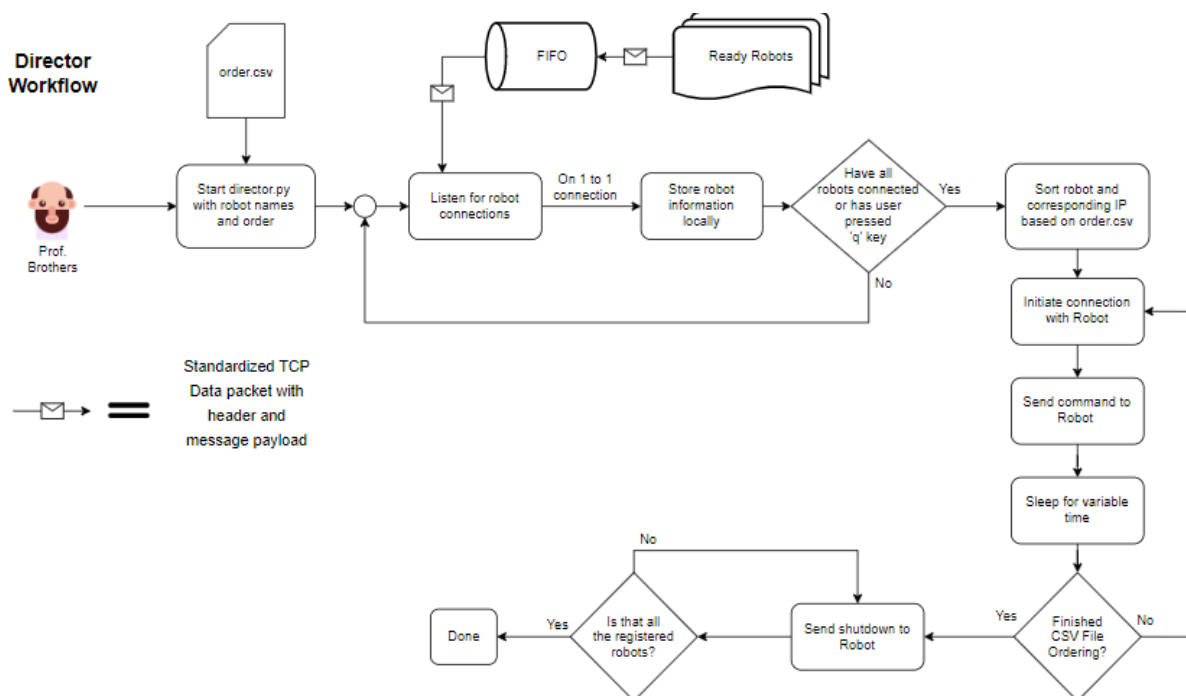


Figure 3. Director workflow

## Robot workflow:

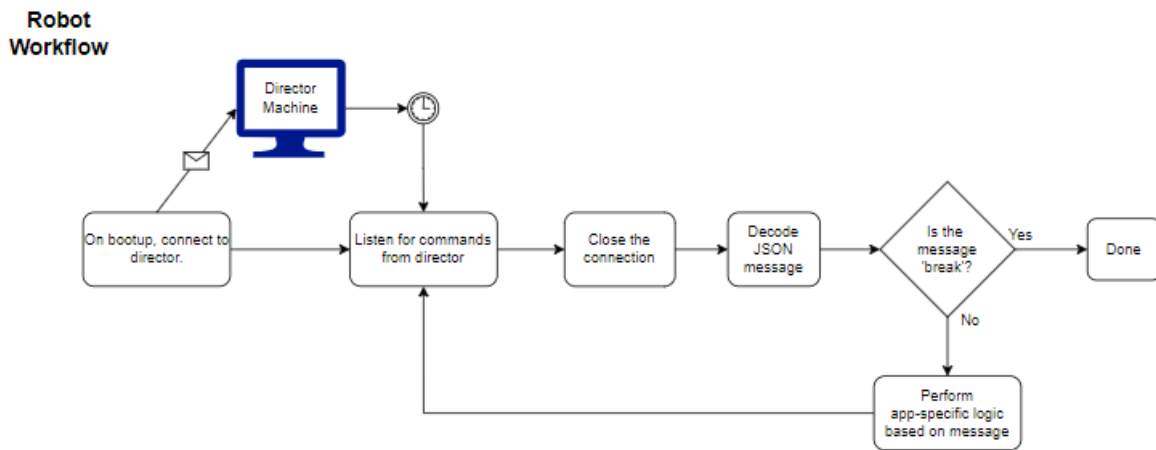


Figure 4. Robot workflow

## Demo Run

- Clone the repo into the desired working directory on the Linux machine:

```
git clone https://github.gatech.edu/tbrothers3/ECE-3872.git
cd Projects/Wonderland
```

- Install the necessary dependencies:

```
sudo python3 install -r requirements.txt
```

You can either get two ssh or shell sessions on the Pi and run the commands for the director and robot thespian mentioned [earlier](#).

If not, do the following:

- Create separate sessions for the robot:

```
sudo apt update
sudo apt install screen
```

- Run the director in one session:

```
screen -d -m sudo python3 director.py <path to csv file>
```

- Run the robot in a separate terminal:

```
screen -d -m sudo python3 demo_robot.py
```

- View all the screen sessions:

```
screen -ls
```

- Attach to the director screen session to enter q to continue the demo:

```
screen -r <director screen session name>
```