Gesture-Controlled Interface for Robotic Simulations

Project Repository

For detailed code, setup instructions, and further documentation, please visit the GitHub repository of this project:

GitHub Repository: https://github.com/jomoslomo/csc-4444-finalproject (https://github.com/jomoslomo/csc-4444-finalproject)

Description

This project creates a gesture-controlled interface using a combination of web technologies and Python for real-time gesture recognition. The interface allows users to control a 3D visualization environment powered by Three.js through hand gestures captured via a webcam. This interaction is facilitated by a Node.js WebSocket server which communicates between the Python-based gesture recognition module and the Three.js frontend.

Components

- **Python Gesture Recognition**: Utilizes OpenCV and MediaPipe to process video input from a webcam, detecting hand gestures to interface with the 3D environment.
- **Node.js WebSocket Server**: Manages real-time communication between the Python script and the web frontend.
- **Three.js Web Application**: Displays a 3D model of a robotic arm which users can interact with using specific gestures.

Installation

Prerequisites

- Node.js and npm
- Python 3.x
- pip (Python package installer)
- A webcam

Setup Instructions

1. **Clone the repository**

```
git clone <repository-url>
cd <project-directory>
```

2. **Setup the Node.js WebSocket Server**

```
cd ws-server-example-nodejs
npm install
cd ..
```

3. **Setup the Python Environment**

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

4. **Run the Web Server for the xArmEX Project**

```
npm install -g http-server
http-server ./xArmEX -p 8000
```

Running the Project

Execute the launch.sh script to start all components of the project:

```
./launch.sh
```

User Interaction Guidelines

Joint Selection with the Left Hand

To interact with the robotic arm, users should position the back of their left hand towards the webcam. This orientation ensures optimal recognition of hand gestures used to select specific joints of the robot:

- One Finger: Selects the first joint of the robotic arm.
- Two Fingers: Selects the second joint, and so forth.

Controlling Movement with the Right Hand

Once a joint is selected, the user can manipulate it using their right hand by targeting specific hit boxes displayed on the screen:

- Top and Left Hit Boxes: Moves the selected joint clockwise.
- Right and Bottom Hit Boxes: Moves the selected joint counterclockwise.

Operating the Robotic Claw

The functionality of the robotic claw is controlled via distinct gestures made with the right hand:

- Peace Symbol (Middle and Index Finger Extended): Opens the claw.
- Fist: Closes the claw.

This gesture-based system provides a user-friendly interface that allows for precise and intuitive control of the robotic arm, making it suitable for environments where contactless interaction is advantageous.