VIRTUAL HUMAN ACTIVITY RECOGNITION SIMULATION SYSTEM

- By Veena Suresh

GOPHER
INDUSTRIES

GUARDIAN MONITOR

T2 2024

Contents

Application Details

The Human (Hospital) Activity Simulator system was built as a web application using a combination of backend and frontend technologies. Here's an overview of how the system was constructed:

Backend (Server-side):

- Python with Flask framework
- Handles routing, data processing, and serving the web pages
- Manages the simulation parameters and room/patient types

Frontend (Client-side):

- HTML, CSS for structure and styling
- JavaScript for interactivity and 3D rendering
- Three.js library for 3D graphics

Main Components:

1. Setup Page (start.html):
   - Allows users to select patient type and room type
   - Sends this data to the server to initialize the simulation
2. Simulation Page (simulator.html):
   - Displays the 3D hospital room and patient
   - Contains controls for different activities
3. 3D Rendering (simulator.js):
   - Uses Three.js to create and render the 3D scene
   - Implements the room layout, patient model, and animations
4. Activity Simulation:
   - Handles different patient activities (walk, sit, lie down, etc.)
   - Applies appropriate animations and position changes
5. CSI Data Generation:
   - Simulates WiFi Channel State Information (CSI) data based on activities
   - Accounts for patient type (e.g., Parkinson's tremors) in data generation
6. Data Export:
   - Allows exporting of simulated CSI data as a CSV file

Key Features:

- Customizable patient types (adult, child, elderly, Parkinson's)
- Different room layouts (standard, ICU, operating room)
- Realistic 3D visualization of activities
- Simulated CSI data generation
- Data export for further analysis

Challenges and Solutions:

- Collision Detection: Implemented to prevent the patient from walking through objects
- Performance: Optimized 3D rendering for smooth animations
- Realism: Refined CSI data generation to reflect real-world patterns

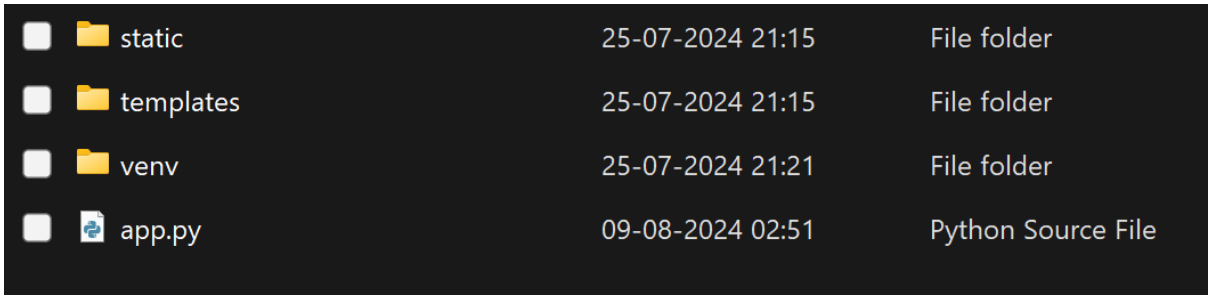
Setting Up

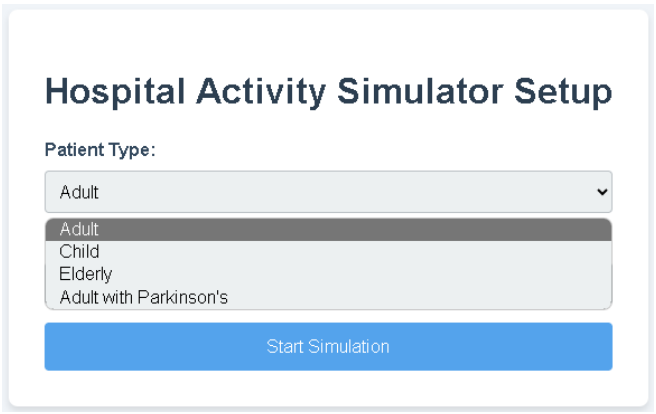
Prerequisites

- Python 3.8 or higher
- pip (Python package installer)
- Git (optional, for cloning the repository)

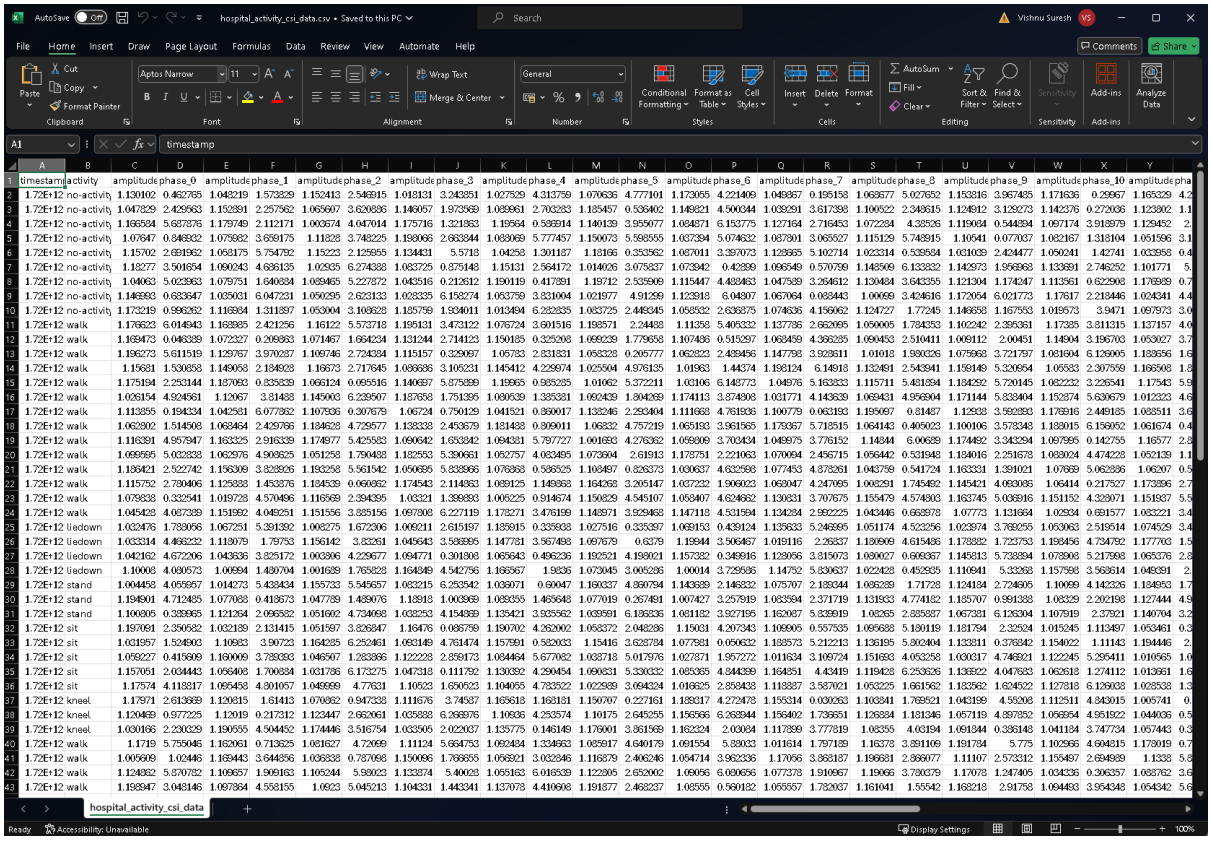
To use this simulation:

1. Save all the provided files in their respective locations within your project directory.
2. Ensure you have Flask installed (*pip install flask*).
3. Run the Flask application (*python app.py*).
4. Open a web browser and navigate to *http://localhost:5000*.
5. Select a patient type and room type on the start page.
6. Use the buttons on the simulation page to perform various activities.
7. Click the "*Export Data*" button to download the collected CSI data as a CSV file.






3D Human Activity Simulator
Current Activity: no-activity
CSI Data Points: 53
Walk
Kneel
Sit
Stand
Lie Down
Export Data

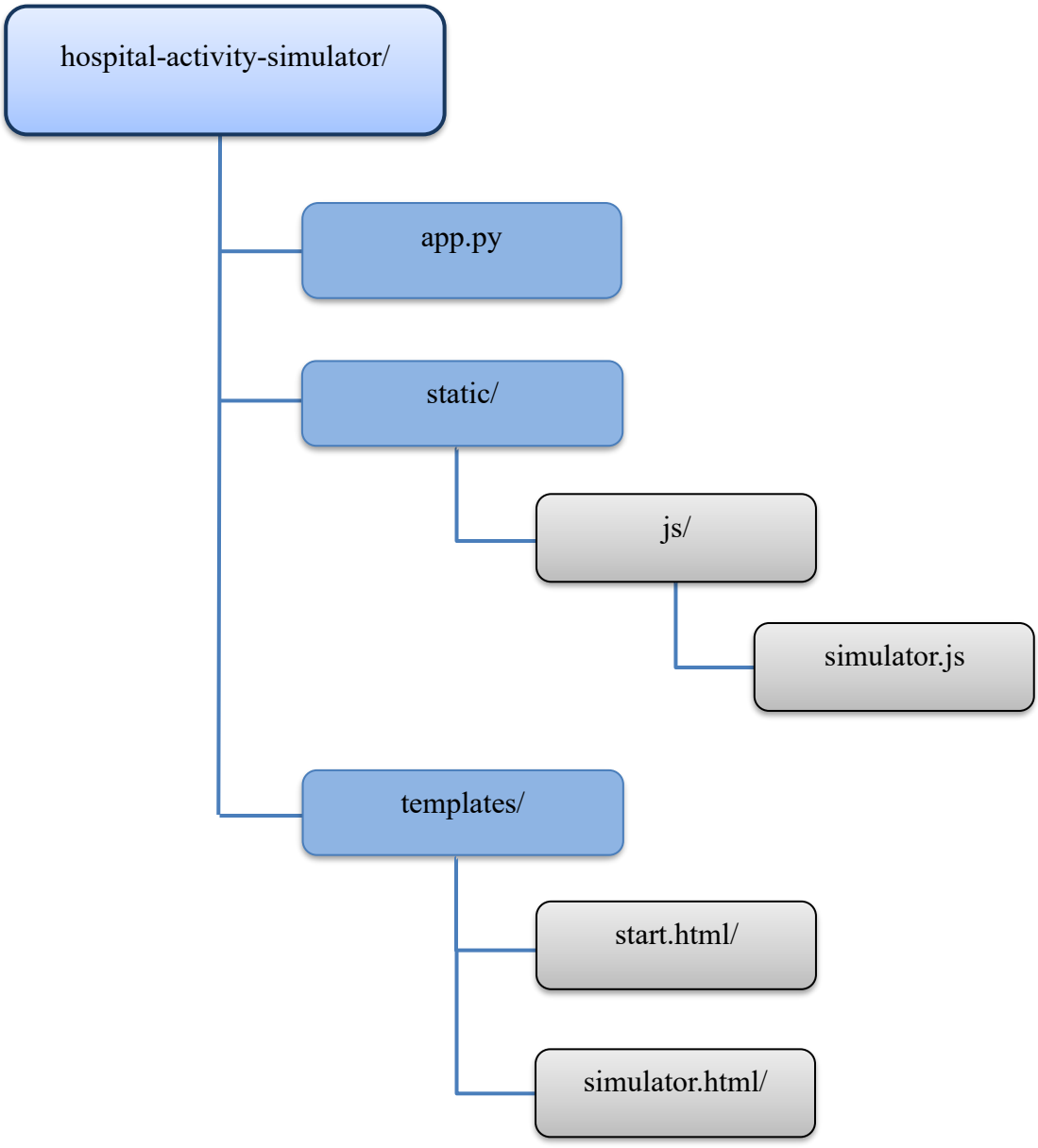

hospital_activity_csi_data.csv • Saved to this PC
Search
Vishnu Suresh
File
Home
Insert
Draw
Page Layout
Formulas
Data
Review
View
Automate
Help
Comments
Share
Paste
Cut
Copy
Format Painter
Clipboard
Aptos Narrow
11
Font
Wrap Text
Merge & Center
Alignment
General
Number
Conditional Formatting
Format as Table
Cell Styles
Styles
Insert
Delete
Format
Cells
AutoSum
Fill
Clear
Sort & Filter
Find & Select
Editing
Sensitivity
Add-ins
Analyze Data
A1
timestamp
timestamp
activity
amplitude_phase_0
amplitude_phase_1
amplitude_phase_2
amplitude_phase_3
amplitude_phase_4
amplitude_phase_5
amplitude_phase_6
amplitude_phase_7
amplitude_phase_8
amplitude_phase_9
amplitude_phase_10
1.72E+12 no-activity
1.72E+12 walk
1.72E+12 kneel
1.72E+12 sit
1.72E+12 stand
1.72E+12 liedown
hospital_activity_csi_data
Ready
Accessibility: Unavailable
Display Settings
100%

Step 1: Set Up the Project Structure

The folder named `*Virtual HAR System*` in Git has the following files:

- `*start.html*`
- `*app.py*`
- `*simulator.html*`
- `*simulator.js*`

Create the following directory structure as given below and add the code files from Git (Guardian/AI Guardian/Virtual HAR System at master · Gopher-Industries/Guardian (github.com)) accordingly to their respective folders as represented in the structure.



Step 2: Run the Application

- Start the Flask application: ` *python app.py*`
- Open a web browser and navigate to *http://127.0.0.1:5000/*

Step 3: Start using the Application

- Select a patient type and room type on the start page.
- Use the buttons on the simulation page to perform various activities.
- Click the "*Export Data*" button to download the collected CSI data as a CSV file.

The Need for a Virtual Human Activity Recognition (HAR) System

The development of a virtual Human Activity Recognition (HAR) system is essential for several reasons, particularly in the context of overcoming practical and logistical challenges in real-world data collection and testing. Here's why a virtual HAR system is needed:

1. Ethical Approval Delays
   - Challenge: Conducting real-time human activity recognition often requires collecting data from human subjects, which necessitates obtaining ethical approval. This process can be time-consuming and may delay the project's progress.
   - Solution: A virtual HAR system allows for the simulation of human activities without the need for direct human involvement, bypassing the need for ethical approval and enabling continuous project development.
2. Controlled Environment for Testing
   - Advantage: A virtual environment provides a controlled setting where variables can be precisely managed. This control is crucial for consistent and repeatable testing of machine learning algorithms, ensuring that the models are tested under standardized conditions.
   - Outcome: The ability to control variables reduces noise and variability in the data, leading to more accurate model evaluations.
3. Rapid Prototyping and Iteration
   - Benefit: A virtual system facilitates rapid prototyping and testing of machine learning models. Developers can quickly implement and test new algorithms, make adjustments, and immediately observe the effects on simulated data.
   - Impact: This accelerates the development cycle, allowing for faster iteration and refinement of algorithms before they are deployed in real-world applications.
4. Scalability and Flexibility
   - Feature: The virtual HAR system can simulate a wide range of activities, environments, and scenarios that might be challenging or impossible to replicate

in real life. For example, different room layouts, varying levels of activity intensity, or specific patient conditions can be modeled.

- Benefit: This flexibility allows for comprehensive testing across multiple scenarios, ensuring that the algorithms developed are robust and versatile.

5. Cost Efficiency
    - Cost-Effective: Developing and maintaining a virtual system is generally more cost-effective than setting up real-world testing environments. It eliminates the need for physical space, equipment, and human participants, reducing overall project costs.
    - Result: Resources can be allocated more efficiently, focusing on development and optimization rather than logistical and operational costs.