Documentation of the prototype 1

The following documentation will describe in detail about the prototype

Setup

1. **Prerequisites**:
    - A modern web browser with WebGL support.
    - Webcam access for pose and gesture detection.

2. **Dependencies**:
    - **TensorFlow.js**: For pose estimation and hand gesture detection.
    - **PoseNet** and **HandPose models**: For detecting pose and hand landmarks.
    - **Three.js**: For 3D rendering.

3. **Running the Application**:
    - Open the index.html file in a web browser.
    - Allow webcam access when prompted.

Navigation

- **Open Palm**: Hold your hand open to navigate to a random page after 3 seconds.
- **Closed Palm**: Keep your hand closed to navigate to index.html after 10 seconds.
- **No Hands**: If no hands are detected, the application logs "No Hands Detected."

==

On start up – menu.js

This script creates a **3D galaxy animation** using **Three.js**, with additional features for **body tracking** and **hand gesture detection** using **TensorFlow.js** models (**HandPose** and **MoveNet**). Users can control the galaxy's rotation using their body movements or mouse, and navigate between pages using hand gestures.

Features

- **3D Galaxy Animation**: A particle system creates a galaxy with spiral arms and color gradients.
- **Body Tracking**: The user's body movements control the galaxy's rotation and camera tilt.
- **Hand Gesture Detection**: Uses HandPose to detect open or closed palms.
- **Navigation**: Navigates to random pages based on hand gestures.
- **Fallback to Mouse Control**: If body tracking or webcam access is unavailable, the user can control the galaxy using the mouse.

Smoke (prototype) – script6.js

This script combines **PoseNet** for pose estimation, **HandPose** for gesture detection, and **particle effects** for interactive visuals. It captures the user's webcam feed, detects their pose and hand gestures, and renders visual effects based on the detected keypoints.

Features

- **Pose Estimation**: Uses PoseNet to detect the user's pose and keypoints.
- **Hand Gesture Detection**: Uses HandPose to detect open or closed palms.
- **Particle Effects**: Particles are created at the positions of detected keypoints.
- **Navigation**: Navigates to random pages based on hand gestures.

==

Psychedelic wave (prototype) – script4-alt.js

This script creates an interactive web application that combines **webcam feed processing**, **particle effects**, and **hand gesture detection**. The particles form wave-like patterns influenced by the brightness of the webcam feed, and hand gestures control navigation between pages.

Features

- **Wave-Like Particle Effects**: Particles form smooth, wave-like patterns.
- **Color Transitions**: Particles transition between two predefined colors.
- **Hand Gesture Detection**: Uses HandPose to detect open or closed palms.
- **Navigation**: Navigates to random pages based on hand gestures.

==

Vortex (prototype) – script3.js

This script is an advanced version of **script1.js**, leveraging **Three.js** to create a 3D particle system that interacts with the user's webcam feed. It integrates the **HandPose model** for gesture detection and creates a visually dynamic experience.

Features

- **3D Particle System**: Particles form a vortex-like effect influenced by the webcam feed.
- **Hand Gesture Detection**: Uses the HandPose model to detect open or closed palms.
- **Navigation**: Navigates to random pages based on hand gestures.

==

Planet (prototype) – script1.js

This script creates an interactive web application that uses the user's webcam feed to detect hand gestures and control particle animations on a canvas. The application uses the **HandPose model** for gesture detection and navigates between pages based on hand gestures.

Features

- **Hand Gesture Detection**: Uses the HandPose model to detect open or closed palms.
- **Particle Effects**: Particles move based on the brightness of the webcam feed.
- **Navigation**: Navigates to random pages based on hand gestures.

Credits

This project use:

- **Tensowflow.js (Pose detection):** https://blog.tensorflow.org/2018/05/real-time-human-pose-estimation-in.html

- **Handpose**: https://github.com/tensorflow/tfjs-models/tree/master/handpose

- **Three.js (Any 3D or particle related visual):** https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene

- **MediaDevice (Webcam):** https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia

- **Canvas API:** https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API