

3D Mirror Configurator - Documentation

Project Overview

The 3D Mirror Configurator is a web-based application that allows users to adjust the dimensions, tilt, and sensor position of a mirror. This application enables users to configure the mirror in a virtual environment and responds to user interactions through 3D visualization.

Technologies and Libraries Used

- **HTML5:** Defines the structure of the page, including the forms and 3D view area.
- **CSS3:** Defines the style and layout of the page, including basic styles and layout adjustments for responsive design.
- **JavaScript:** Manages dynamic content and interactions.
- **Three.js:** A JavaScript library for creating 3D graphics. It provides 3D modeling for the mirror and sensor.

Installation and Configuration

1. HTML

The HTML file creates the user interface, including:

- Mirror dimensions (width and height)
- Mirror tilt
- Option to add a sensor
- Selection of switch position

2. CSS

The CSS file styles the page, including:

- General layout and typography
- Form elements and labels
- Styles for the 3D view area

3. JavaScript

The JavaScript file manages 3D visualization and user interactions:

- Creates a 3D scene using Three.js.
- Configures the 3D model of the mirror and sensor.
- Listens for user interactions and updates the scene.
- Implements performance optimizations.

Performance Improvements

- **Memoization:** Uses `useMemo` and `useCallback` in React applications to prevent unnecessary re-renders of components.
- **Component Splitting:** Divides components into smaller parts for better management and performance.
- **Debouncing:** Applies debouncing to user interactions (e.g., slider movements) to enhance performance.
- **Lights Optimization:** Uses multiple types of lights in the Three.js scene to improve visual quality. Different light types (ambient, directional, point, spot) highlight details in the scene.

User Interactions

- **Mirror Dimensions:** Width and height inputs allow users to change the mirror dimensions. Changes scale the mirror model accordingly.
- **Mirror Tilt:** The slider adjusts the tilt angle, affecting the 3D appearance of the mirror.
- **Sensor Addition:** Users can add a sensor and select its position. The sensor is displayed on the mirror based on the selected position.
- **Switch Position:** When a sensor is added, the switch position is selected by the user and shown on the mirror.

Code Explanations

- **HTML:** Contains form elements and the 3D view area. It also loads the Three.js library and the custom JavaScript file.
- **JavaScript:** Creates the 3D scene with Three.js, configures the mirror and sensor, and updates the scene based on user interactions.
 - `updateMirrorDimensions`: Updates the mirror dimensions and recalculates the sensor position.
 - `handleTiltInput`: Updates the mirror's tilt.
 - `handleSensorVisibility`: Adjusts the sensor's visibility and position.
 - `updateSensorPosition`: Calculates and sets the sensor's position on the mirror.
-