

3D Model and Panorama Viewer for Developers

Custom Element

A custom element is used to integrate your 3D model on a web page. You can specify many settings as HTML attributes, style the element (CSS) and listen for events (JavaScript).

```
<hyper-3d id="mymodel" src="model.gltf" ratio="3:2" playing loop ambient-light="1.3">  
</hyper-3d>
```

File Types

Supported file types are .gltf/.glb .fbx .obj and .stl

Support for Draco-compression

Environment Map

For material reflectivity, as background or 360° panorama.

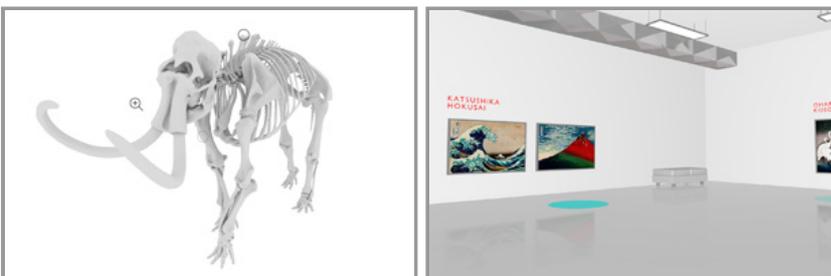


Auto Rotation

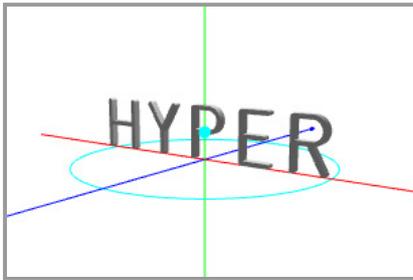
Options for speed, direction, delay after an interruption and a vertical target angle. The view oscillates if the horizontal viewing angle is constrained.

View

- Orbit View: Rotate around a specified point
- First Person View: Look around at a specified point



The viewing angles are specified in human-readable degrees, can be inspected in debug mode and can be constrained.



- The view can be rotated by the user (mouse/touch)
- The view can be zoomed by the user (mousewheel/gesture)
- There is no user-controlled panning.

Animation

You can play the main timeline animation with options for speed, loop and a plackback range. You can change the time even if the animation is not playing. For example, to advance the animation depending on the scroll position of the page.

Effects

You can specify ground reflections and shadows with options for strength, softness and color.



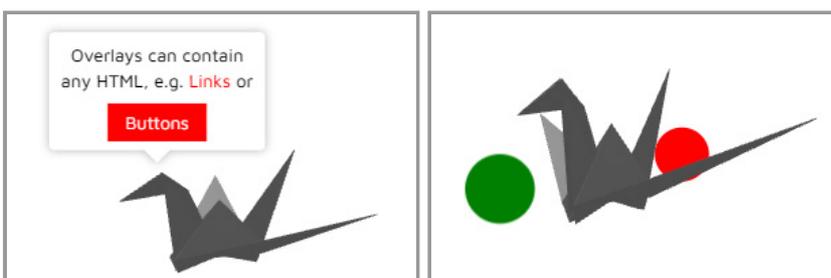
Options for wireframe mode and material color.

Lights

In addition to lights that you include in the model, you can define a point light, camera light and ambient light.

Overlays

The overlay feature moves HTML elements into 3D space. This is especially useful for adding hotspots and labels.



- The overlays are z-ordered and can be on top or behind the 3D rendering.
- The overlays can be scaled based on the current zoom, the position and the element size.
- The overlays can be easily styled with CSS based on view angles, occlusion and the 2D position in the element.

Events

In addition to native events (click, touch, etc.) you can listen to events such as:

- load complete
- grab / release
- view change
- animation complete

Methods

You get powerful and easy-to-use methods to:

- Find the 3D object at the mouse/touch position to make it hoverable/clickable
- Change the view with a transition
- Animate every property
- Get or change objects, positions and materials

Supported 3D content features

You can expect basic 3D features to work:

- Named, hierarchically arranged objects
- Triangulated Geometry, Smooth/Flat Shading, Sharp Edges
- Textures, UV Mapping, Transparency
- PBR materials (GLTF format)
- Main Timeline Animation

Many other features can be baked to textures by your 3D software (such as Ambient Occlusion or Global Illumination) or must be set up with JavaScript/THREE.js (such as Morph Targets, Skeletons or Animation Actions).

Documentation

The documentation comes with many examples and the possibility to run live code.

Table of Contents:

- Getting Started
- **Attributes/Properties**
- **Events**
- **Methods**
- **Guides:**
 - ES6 module import
 - Element size
 - Layout Shifts and Flashing
 - Fallback content
 - Using a data: URI
 - Different models based on @media
 - Supported file types
 - Draco-compressed GLTF files
 - Supported 3D content features
 - Export recommendations
 - Positions and Units
 - Control Element Buttons
 - Accessibility / Keyboard controls
 - HTML elements on top of the <hyper-3d> element
 - HTML elements in 3D space
 - Clicking/Hovering 3D objects
 - Navigation in a 3D scene
 - Observing the view
 - Hiding objects
 - Replacing materials
 - Altering materials of single objects
 - Billboards, a performant alternative to overlays
 - Working with THREE.js
- Troubleshooting