

# Interface Development and Design

## WebXR

Efrei Paris

2025 - 2026

[daniel.mai@efrei.fr](mailto:daniel.mai@efrei.fr)

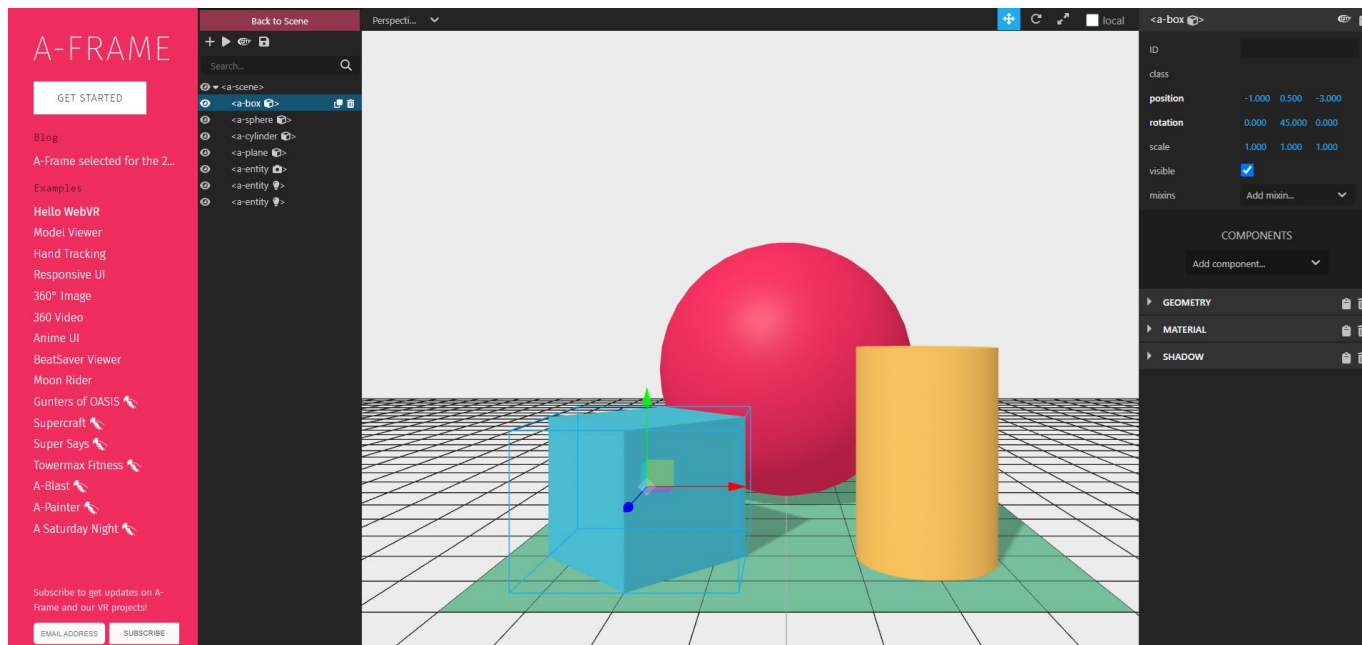


PARIS PANTHÉON-ASSAS UNIVERSITÉ

# Sommaire

- I. Rappel des connaissances dev web
- II. WebAR
- III. Web3D
- IV. WebVR**

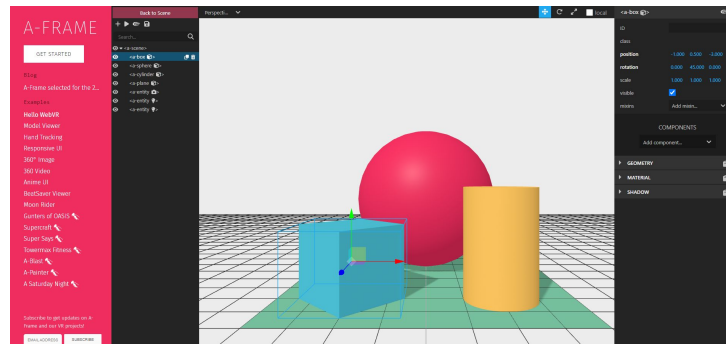
# A-FRAME inspector



<https://aframe.io/examples/showcase/helloworld/>

# A-FRAME inspector

- Vérifier, modifier et optimiser une scène 3D directement dans le navigateur
- **Activation** : **Ctrl + Alt + i**
- **Explorer la structure de la scène** : parcourir toutes les entités (entity) et composants (component) présents dans la scène
- **Modifier les entités** : ajuster les propriétés comme la position, la taille et les attributs des objets 3D directement dans le navigateur
- **Prévisualisation des effets** : observer les modifications en temps réel lorsque vous modifiez les objets, la lumière ou les effets dans la scène



- Accélère le développement du projet
- Facilite la vérification et l'optimisation de la scène sans avoir à recharger la page

# Lumière dans A-Frame

## Directional Light

- Émet des rayons parallèles, semblable à la lumière du soleil
- Convient pour éclairer toute la scène de manière uniforme
- Utilisée pour projeter des ombres détaillées et réalistes

## Point Light

- Émet la lumière à partir d'un point dans toutes les directions, comme une ampoule
- Idéal pour éclairer une zone spécifique autour de la source

## Ambient Light

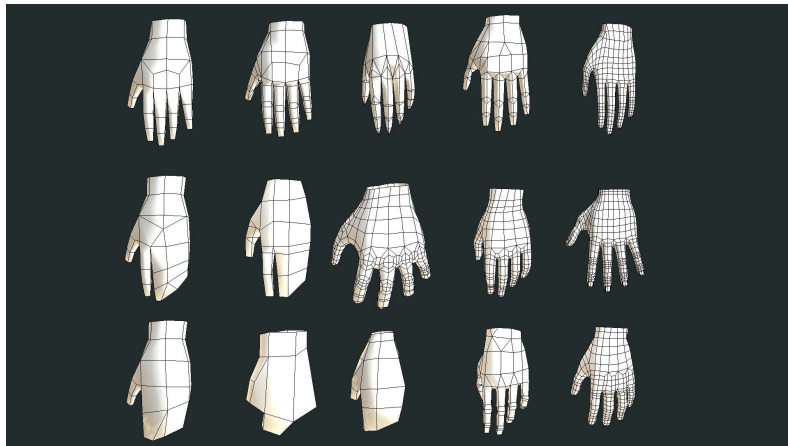
- Fournit un éclairage doux et uniforme dans toute la scène
- Utilisée pour simuler l'éclairage global, sans ombres marquées

## Activation des ombres

- Les objets 3D peuvent recevoir et projeter des ombres en activant l'attribut `shadow` sur les lumières et les entités :
  - **Projeter des ombres** : l'objet doit avoir l'attribut **`castShadow`**
  - **Recevoir des ombres** : l'objet doit avoir l'attribut **`receiveShadow`**

# Caméra et contrôleurs VR dans A-Frame

- **Caméra** : permet à l'utilisateur de voir à travers la scène en suivant ses mouvements de tête
- **Contrôleurs** : les mains de l'utilisateur sont représentées en VR
  - Peuvent être configurées avec des rayons laser pour sélectionner des objets ou des boutons dans l'interface utilisateur
  - Le joystick du contrôleur gauche permet de se déplacer dans la scène, tandis que celui du contrôleur droit permet de tourner la vue
  - Dans A-Frame, on utilise le composant ***tracked-controls*** ou ***hand-controls*** pour suivre les mouvements des contrôleurs de l'utilisateur



```
<a-entity id="leftHand" hand-controls="hand: left; handModelStyle: lowPoly; color: #ffcccc"></a-entity>
```

```
<a-entity id="rightHand" hand-controls="hand: right; handModelStyle: highPoly; color: #ffcccc"></a-entity>
```

# Interactions & Controllers

## Tracked Controls

- Suivi des mouvements des contrôleurs VR (par exemple, Oculus, Vive)
- Events : *controllerconnected*, *controllerdisconnected*, *axismove*, *buttonchanged*, *buttondown*, *buttonup*, *touchstart*, *touchend*
  - permet d'utiliser les joysticks pour se déplacer et les gâchettes pour saisir ou activer des objets

## Grabbing

- Utilisation du composant *super-hands* pour gérer les actions de saisie (grab) et relâche (release) d'objets
- Les utilisateurs peuvent saisir un objet, le déplacer, puis le relâcher en utilisant les contrôleurs VR

## Raycaster

- Utilisation d'un raycaster pour émettre un rayon laser à partir des contrôleurs et interagir à distance avec des objets ou des éléments d'interface (UI)
- Les utilisateurs peuvent sélectionner des objets ou appuyer sur des boutons en pointant le rayon vers une cible

# A-FRAME Extras

- Un ensemble de composants et de fonctionnalités supplémentaires qui étendent les capacités d'**A-Frame**, notamment pour la prise en charge des environnements **VR/AR** et des contrôles de mouvement dans l'espace 3D

## Movement Controls

- Permet de déplacer facilement l'utilisateur dans un environnement VR
- Prend en charge le déplacement via joystick ou clavier dans des environnements VR
- Attributs principaux
  - `fly`: permet à l'utilisateur de "voler" dans l'espace
  - `speed`: ajuste la vitesse de déplacement

## Physics

- Intègre des effets physiques aux objets tels que la gravité, les collisions et la dynamique
- Utilise des bibliothèques comme **Cannon.js** pour gérer les aspects physiques dans la scène 3D



# A-Frame Extras vs A-Frame Physics System

## A-Frame Extras

- **Physique simplifiée** : utilise **Cannon.js** pour des effets physiques basiques tels que la gravité et les collisions
- **Contrôles de mouvement** : fournit des composants comme *movement-controls* pour faciliter le déplacement dans un environnement VR/AR
- **Utilisation**
  - Idéal pour des projets qui nécessitent des effets physiques simples et des déplacements basiques
  - **Facile à intégrer**, conçu pour des besoins rapides en VR/AR

## A-Frame Physics System

- **Système physique avancé** : prend en charge **Cannon.js** et **Ammo.js** pour des interactions physiques complexes comme les forces, les contraintes, les objets souples, etc
- **Plus grande flexibilité** : offre des fonctionnalités détaillées telles que la gestion des collisions, les liens physiques entre objets, les forces de friction, etc
- **Utilisation**
  - Idéal pour des projets nécessitant des **interactions physiques réalistes** ou des simulations physiques avancées
  - Plus complexe à configurer, mais permet des résultats très réalistes

<script src="https://cdn.jsdelivr.net/gh/c-frame/aframe-extras@7.5.0/dist/aframe-extras.min.js"></script>

# Chargement d'objets 3D

- Les formats couramment utilisés incluent GLTF/GLB, OBJ, et Collada (DAE)
- GLTF/GLB est format recommandé
  - Léger, performant, et optimisé pour les scènes interactives en temps réel
  - GLTF : format texte, facile à lire et à manipuler
  - GLB : version binaire de GLTF, plus compacte
- Charger un modèle 3D GLTF/GLB
  - Utilisez l'élément `<a-asset-item>` pour charger un modèle 3D dans votre scène via la balise `<a-assets>`
  - Ensuite, insérez le modèle dans la scène en utilisant une entité `<a-entity>`

```
<a-assets>  
  <!-- Charger le modèle 3D GLTF/GLB →  
  <a-asset-item id="treeModel" src="models/tree.glb"></a-asset-item>  
</a-assets>
```

```
<!-- Ajouter le modèle à la scène →  
<a-entity gltf-model="#treeModel" position="0 0 -5" scale="2 2 2"></a-entity>
```