

TP02 : WebAR

Références avant de commencer :

- A-Frame Documentation : <https://aframe.io/docs/>
- AR.js Documentation : <https://ar-js-org.github.io/AR.js-Docs/>

Exercice 1 : Créer une scène 3D avec A-Frame

- Créez un projet sur GitHub.
- Créez un fichier `index.html` et ajoutez la bibliothèque A-Frame.
- Créez une scène 3D contenant :
 - Trois boîtes de couleurs différentes, placées à des positions différentes.
Un plan sous les boîtes pour servir de surface de base.
 - Les boîtes doivent léviter à une hauteur de **2 unités** au-dessus du plan.
- Ajoutez une caméra positionnée à **10 unités de hauteur et 15 unités de distance**, orientée vers le centre du plan, afin de voir toute la scène.

Exercice 2 : Intégrer AR.js et tester sur mobile

- Transformez la scène en une scène AR en ajoutant la bibliothèque **AR.js** et l'attribut `ar.js` dans la balise `<a-scene>`.
- Utilisez le **marqueur Hiro**. Lorsque la caméra détecte ce marqueur, une boîte semi-transparente apparaît.
- Configurez cette boîte pour qu'elle **tourne automatiquement sur elle-même**.
- Téléchargez le marqueur Hiro ici :
<https://jeromeetienne.github.io/AR.js/data/images/HIRO.jpg>
- Publiez votre projet avec **GitHub Pages**.
- Ouvrez l'URL du projet sur votre smartphone, autorisez la caméra et pointez-la vers le marqueur Hiro.

Exercice 3 : Ajouter une interaction par clic/tap

- Ajoutez un curseur à la caméra
- Configurez la boîte de l'Exercice 2 pour qu'elle change de couleur lorsqu'on clique dessus avec la souris (desktop) ou lorsqu'on tape dessus sur mobile.
- Ajoutez une animation supplémentaire : lors de cette interaction, la boîte doit tourner plus vite ou changer d'échelle.
- Testez sur mobile : en tapant sur la boîte affichée via le marqueur Hiro, sa couleur change et l'animation se déclenche.

Exercice 4 : Utiliser un marker personnalisé

- Choisissez une image de votre choix et générez un marker AR personnalisé avec l'outil :
<https://jeromeetienne.github.io/AR.js/three.js/examples/marker-training/examples/generator.html>
- Téléchargez le fichier `.patt` généré et ajoutez-le à votre projet.
- Intégrez ce marker personnalisé dans la scène AR.
Configurez l'apparition d'un objet 3D (différent de la boîte) lorsqu'il est détecté.

Exercice 5 : Combiner QR code et marker Hiro

- Générez un **QR code** qui redirige vers l'URL GitHub Pages de votre projet.
- Placez le marker Hiro au centre du QR code.
- Scénario à tester :
 - Scanner le QR code → ouvrir le site.
 - Pointez ensuite la caméra du navigateur vers le même QR code → la boîte AR s'affiche au centre.

Exercice 6 : Ajouter une interface utilisateur (UI DOM overlay)

- Dans la scène AR (avec le marker Hiro), affichez une boîte 3D.
- Créez une **interface utilisateur HTML** (DOM overlay) contenant :
Un **slider** pour changer la taille de la boîte (0.1 → 3.0 unités).
 - Une **palette de couleurs** (au moins 3 choix : rouge, vert, bleu).
- L'UI doit seulement être visible quand le marqueur Hiro est détecté :
 - Utilisez les événements `markerFound` et `markerLost` pour afficher/masquer l'UI.
- L'utilisateur peut :
 - Glisser le slider → la taille de la boîte change en temps réel.
 - Cliquer une couleur → la boîte change immédiatement de couleur.