

Qu'est-ce qu'un marqueur AR et comment fonctionne-t-il avec AR.js ?

Un marqueur AR est une image ou un motif spécial qui est reconnu par la technologie AR, comme AR.js, comme un déclencheur pour une expérience AR. Lorsqu'il est utilisé avec un marqueur AR, AR.js peut permettre des expériences AR qui sont déclenchées lorsque le marqueur est détecté par l'appareil de l'utilisateur.

Les marqueurs AR sont utilisés pour placer des objets et du contenu virtuels dans le monde réel. AR.js utilise la caméra d'un appareil utilisateur pour identifier un marqueur AR, puis superpose le contenu virtuel sur le monde réel via la vue de la caméra. Les marqueurs AR peuvent être n'importe quelle image ou motif, mais pour AR.js, il est recommandé d'utiliser des codes QR en noir et blanc.

Lorsque le marqueur est détecté par l'appareil de l'utilisateur, l'expérience AR associée au marqueur sera déclenchée, permettant de superposer du contenu virtuel sur le monde réel. Cela peut inclure des modèles 3D, des animations ou des éléments interactifs.

[Liste des marqueurs](#)

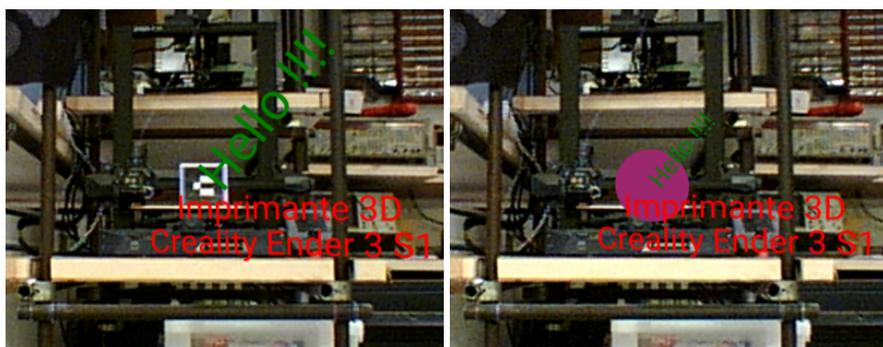
Tableau des marqueurs

Lorsque l'on utilise des marqueurs dans son fichier index.html il faut adapter le code en fonction de la taille des marqueurs. Ce tableau est à titre indicatif, il renseigne les différentes marges à respecter en fonction des tailles des marqueurs mais il est à modifier si besoin.

Taille marqueurs(cm)	Distance min(m)	Distance max(m)	
3	0,30	0,70	10 à 30
5	0,70	1,20	20 à 40
8	1,20	2	30 à 50
15	2	6	50 à 70
30	+6		

Il est important aussi de régler les rotations et les positions des axes x, y et z utilisés pour identifier des points dans un espace 3D sous la forme de coordonnées (x,y,z). [Comprendre les valeurs de la position et de la rotation](#)

Voici l'exemple suivant avec le marqueur de visible et le marqueur caché par un cercle mauve.



Ce code HTML est une page web simple qui utilise A-Frame et AR.js pour créer une application de réalité augmentée (AR). Voici une explication détaillée de chaque partie du code :

<head>

<title> : Définit le titre de la page web qui s'affiche dans l'onglet du navigateur. Ici, le titre est "Ma première app AR".

<script src="https://aframe.io/releases/1.7.0/aframe.min.js"></script>:

Charge la bibliothèque A-Frame, un framework pour créer des expériences de réalité virtuelle (VR) et augmentée (AR) en utilisant des éléments HTML.

`<script src="https://raw.githubusercontent.com/AR-js-org/AR.js/master/aframe/build/aframe-ar.js"></script>` :

Charge AR.js, une extension d'A-Frame qui permet d'ajouter des fonctionnalités de réalité augmentée, comme la reconnaissance de marqueurs.

`<body>`

`<a-scene>`:

C'est l'élément principal d'A-Frame qui représente la scène 3D. Il contient divers attributs pour configurer la scène AR :

`embedded`: Indique que la scène est intégrée dans la page web plutôt que d'occuper tout l'écran.

`arjs`: Configure les paramètres spécifiques à AR.js :

1. ``sourceType: webcam`` : Utilise la webcam comme source vidéo.
2. ``detectionMode: mono_and_matrix`` : Détecte à la fois les marqueurs simples et les matrices de marqueurs.
3. ``matrixCodeType: 3x3`` : Spécifie le type de matrice de marqueurs (ici, une matrice 3x3).
4. ``trackingMethod: best`` : Utilise la meilleure méthode de suivi disponible.
5. ``changeMatrixMode: modelViewMatrix`` : Change le mode de matrice pour le rendu.

`renderer`: Configure les paramètres du rendu :

1. ``sortObjects: true`` : Trie les objets pour un rendu correct.
2. ``antialias: true`` : Active l'anti-aliasing pour des bords plus lisses.
3. ``colorManagement: true`` : Gère les couleurs pour un rendu plus précis.
4. ``logarithmicDepthBuffer: true`` : Utilise un tampon de profondeur logarithmique pour améliorer le rendu des objets proches et éloignés.

`vr-mode-ui="enabled: false"`: Désactive l'interface utilisateur du mode VR.

`smooth`: Lisse les mouvements de la caméra pour une expérience plus fluide

`smoothCount`, `smoothTolerance`, `smoothThreshold` : Paramètres pour contrôler le lissage.

`sourceWidth`, `sourceHeight`, `displayWidth`, `displayHeight`: Définit les dimensions de la source vidéo et de l'affichage.

Ce code configure une scène AR de base qui utilise la webcam pour détecter des marqueurs et afficher des objets 3D en réalité augmentée. Pour une application complète, vous devrez ajouter des éléments 3D (comme `<a-box>`, `<a-sphere>`, etc.) à l'intérieur de la balise `<a-scene>`.

Article extrait de : <https://lesporteslogiques.net/wiki/> - **WIKI Les Portes Logiques**

Adresse : https://lesporteslogiques.net/wiki/ressource/application_au_fablab?rev=1742897240

Article mis à jour: **2025/03/25 11:07**