Une application AR est une application qui peut être installée sur un smartphone ou une tablette pour afficher des objets virtuels et des informations superposées au monde réel.

Le WebAR peut-il afficher de l'AR sur les anciens smartphones ?

Oui, le WebAR peut afficher du contenu AR sur les anciens smartphones tant que l'appareil dispose d'un navigateur web prenant en charge la technologie WebAR, comme Chrome ou Firefox. Cependant, la performance et la qualité de l'expérience AR peuvent varier en fonction des capacités de l'appareil et de son navigateur web.

Quelles technologies sont utilisées pour exécuter WebAR ?

WebAR utilise une combinaison de technologies pour fournir des expériences AR immersives et interactives sur le web. Ces technologies incluent :

```
  ARCore et ARKit :
  Ce sont des kits de développement logiciel développés respectivement par Google et Apple, qui permettent des expériences AR sur les appareils Android et iOS. Ils utilisent diverses technologies, telles que la vision par ordinateur et l'apprentissage automatique, pour reconnaître et suivre les objets physiques dans l'environnement de l'utilisateur, permettant ainsi de superposer du contenu virtuel sur le monde réel de manière fluide et immersive.
  WebGL : C'est une norme web qui permet de rendre des graphismes 3D dans un navigateur web. WebAR utilise WebGL pour rendre le contenu virtuel qui est superposé sur le monde réel, permettant de l'afficher de manière réaliste et interactive.
  HTML5 : C'est une norme web qui définit la structure et le contenu d'une page web. WebAR utilise HTML5 pour créer et afficher l'interface utilisateur pour l'expérience AR, permettant aux utilisateurs d'interagir avec le contenu virtuel de manière naturelle et intuitive.
  JavaScript : C'est un langage de programmation largement utilisé sur le web. WebAR utilise JavaScript pour rassembler toutes les technologies mentionnées précédemment, permettant à l'expérience AR de fonctionner et d'être contrôlée dans le navigateur web. AR.js est une bibliothèque JavaScript populaire pour l'affichage WebAR.
```

Ensemble, ces technologies permettent à WebAR de fournir des expériences AR fluides et interactives sur le web, sans nécessiter d'application dédiée.

Qu'est-ce que AR.js et comment fonctionne-t-il ?

AR.js est une bibliothèque JavaScript gratuite et open-source pour créer des expériences de réalité augmentée sur le web. Elle est construite sur les normes web WebGL et WebRTC, et utilise les marqueurs **ARToolKit** et **AR.js** pour permettre des expériences AR sur les pages web.

AR.js est conçu pour être facile à utiliser et léger, ce qui en fait un choix populaire parmi les développeurs qui souhaitent créer des expériences AR sur le web. Il est également compatible avec une large gamme de navigateurs web et d'appareils, y compris les smartphones, les tablettes et les ordinateurs de bureau. Globalement, **AR.js** est un outil puissant pour créer des expériences WebAR, et il est largement utilisé par les développeurs dans le secteur AR.

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.

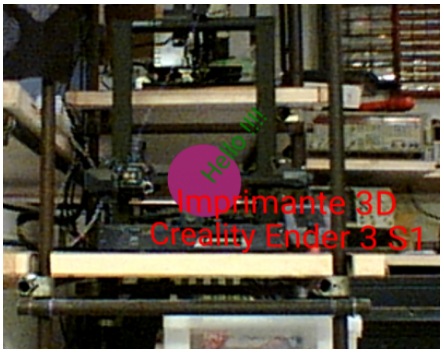
Ce tableau est à titre indicatif, il renseigne les différentes marges à respecter en fonction des tailles des marqueurs mais il est à modifier si besoin.

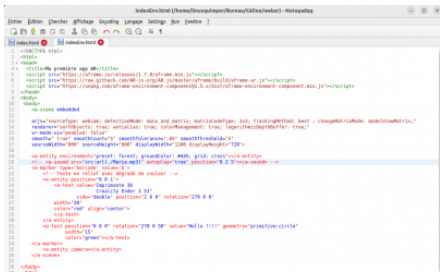
Taille marqueurs(cm)	Distance min(m)	Distance max(m)	
3	0,30	0,70	10 à 30
5	0,70	1,20	20 à 40
8	1,20	2	30 à 50
15	2	6	50 à 70
30	+6		

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.githack.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: 3×3` : Spécifie le type de matrice de marqueurs (ici, une matrice 3×3).
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, moothTolerance, 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>`. Il est aussi possible d'ajouter du son, une vidéo bref presque tout est possible de faire.

Lien externe

Lien externe

Se référencer à https://aframe.io/|Lien externe]]

Article extrait de : http://www.lesporteslogiques.net/wiki/ - **WIKI Les Portes Logiques**
Adresse : **http://www.lesporteslogiques.net/wiki/ressource/code/fonctionnement_de_la_realite_augmente**
Article mis à jour: **2025/03/27 12:00**