**Progetto stage - Demo AR**

**Realtà aumentata (AR) sul web**

Ridotta in minimi termini: tecnica per permettere di aggiungere (in tempo reale) degli oggetti virtuali ad immagini provenienti da una sorgente di acquisizione video (es. webcam)

Come realizzare un’esperienza di realtà aumentata (AR) sul web:

* Possedere uno o più **marker**, ovvero delle immagini speciali che saranno utilizzate dal software come punti di riferimento per decidere dove posizionare gli elementi 3D all’interno dell’immagine ripresa dalla fotocamera/webcam
* Creare uno o più **oggetti 3D** da associare ai marker. Questi oggetti possono essere creati:
  + Sfruttando delle primitive messe a disposizione da A-Frame
  + Attraverso software di modellazione 3D (es. Unity, Blender, …)
* Scrivere una **pagina web** di codice che permetta di catturare l’ingresso video dentro un canvas e indicare, ad esempio ad AR.js, quali sono i marker da riconoscere
* Eseguire la pagina web generata attraverso un **web server** che “serva” la modalità HTTPS, dato che l’accesso alla webcam avviene esclusivamente se la connessione è protetta

**AR.js**

AR.js è una libreria Javascript per la Realtà Aumentata (o Augmented Reality, AR) sul Web.

AR.js offre tre differenti tipologie di realtà aumentata:

* **Image Tracking**: in questo caso quando la camera trova un’immagine 2D è possibile mostrare su di essa (o vicino) del contenuto. Questo contenuto possono essere immagini 2D, immagini 3D, GIF, video 2D, …
* **Location Based AR**: vengono utilizzati luoghi e posizioni del mondo reale per mostrare della realtà aumentata attraverso il dispositivo dell’utente.
* **Marker Tracking**: in questo caso quando la camera trova un marker è possibile mostrare su di essa (o vicino) del contenuto, come nell’Image Tracking.

AR.js è una soluzione puramente web, non serve quindi nessuna installazione ed è scritta in Javascript basandosi su three.js, A-frame e jsartoolkit5.

Funziona inoltre su tutti i telefoni con webgl e webrtc.

**Test AR.js**

La tipologia di AR utilizzata è Marker Tracking.

Creazione sito web locale per test

* Ri-creazione del sito web <https://stemkoski.github.io/AR-Examples/> in locale (tramite github <https://github.com/stemkoski/AR-Examples> ) per vederlo runnato su un server locale (utilizzato XAMPP e creazione di una nuova directory virtuale)

Creazione pagina web di esempio: test-torusknot

* Test di visualizzazione di Torus Knot (utilizzo webcam del pc + immagine del marker pattern Hiro sul cellulare per una corretta visualizzazione)
* Utilizzo di three.js
* File test-torusknot.html

Creazione sample pagina web: ar\_sample.html (funzionamento di AR.js)

* Commentato la pagina web per spiegare funzionamento

Attraverso la creazione di pagine di testing con AR.js si può già notare, seppur in modo semplice, l’utilizzo del web per vivere un’esperienza di realtà aumentata. Infatti spostando la sorgente video oppure muovendo il marker si vede che l’oggetto 3D si adatta al movimento (sinistra-destra/rimpicciolamento-ingrandimento).

**Three.ar.js**

Three.ar.js è una libreria three.js di supporto utilizzata per sviluppare esperienze di realtà aumentata attraverso **WebARonARKit** e **WebARonARCore**.

WebARonARKit e WebARonARCore sono delle app sperimentali, rispettivamente per iOS e Android, che permettono agli sviluppatori la creazione di realtà aumentata utilizzando il web.

**A-Frame**

A-Frame è un framework web per lo sviluppo di esperienze di realtà virtuale (VR).

A-Frame si basa su HTML, ma il suo punto di forza è che fornisce una struttura dichiarativa, estensibile e componibile a **three.js**.

**Test A-Frame**

* Creazione pagina web semplice A-frame (Hello WebVR) con l’aggiunta di alcuni elementi 3D
* Aprire l’inspector (cntrl + alt + i) per la visualizzazione e il testing dello spostamento dei vari elementi creati

**Marker personalizzati**

Possono essere creati dei marker personalizzati, rendendoli appropriati (es. “brandizzandoli”).

Per la creazione di un nuovo marker posso utilizzare il sito <https://jeromeetienne.github.io/AR.js/three.js/examples/marker-training/examples/generator.html>

**Riferimenti**

Realtà aumentata con AR.js: <https://devsware.wordpress.com/2018/07/25/realta-aumentata-con-ar-js/>

Documentazione ufficiale AR.js: <https://ar-js-org.github.io/AR.js-Docs/>

Github organization AR.js: <https://github.com/ar-js-org>

Github three.ar.js: <https://github.com/google-ar/three.ar.js?files=1>

Documentazione ufficiale A-Frame: <https://aframe.io/>

Github A-Frame Inspector: <https://github.com/aframevr/aframe-inspector>

Documentazione ufficiale three.js: <https://threejs.org/>