**Progetto di Stage presso Vegan Solutions SRL**

**Via Contrà del Monte 13, Vicenza**

**Cailotto Massimo (880763@stud.unive.it)**

**Università Ca’ Foscari Venezia**

**Documentazione di progetto**

versione 1.0  
 **Demo AR**

14 febbraio 2022

**Indice**

[1. Introduzione 3](#_Toc96096422)

[*1.1.* *Overview del progetto* 3](#_Toc96096423)

[*1.2.* *Deliverables del progetto* 3](#_Toc96096424)

[*1.3.* *Materiali di riferimento* 3](#_Toc96096425)

[2. Definizioni e descrizioni 3](#_Toc96096426)

[*2.1.* *Definizioni/abbreviazioni delle tecnologie utilizzate* 3](#_Toc96096427)

[*Realtà aumentata (Augmented Reality, AR)* 3](#_Toc96096428)

[*Realtà virtuale (Virtual Reality, VR)* 3](#_Toc96096429)

[*HTML (HyperText Markup Language)* 4](#_Toc96096430)

[*CSS (Cascading Style Sheets)* 4](#_Toc96096431)

[*Javascript* 4](#_Toc96096432)

[*Realtà aumentata (AR) sul web* 4](#_Toc96096433)

[*AR.js* 5](#_Toc96096434)

[*WebGL* 5](#_Toc96096435)

[*Three.js* 5](#_Toc96096436)

[*Three.ar.js* 5](#_Toc96096437)

[*A-Frame* 6](#_Toc96096438)

[*Marker* 6](#_Toc96096439)

[*XAMPP* 6](#_Toc96096440)

[*Unity* 6](#_Toc96096441)

[3. Evoluzione e testing 6](#_Toc96096442)

[*3.1.* *Testing AR.js* 6](#_Toc96096443)

[*3.2.* *Testing A-Frame* 7](#_Toc96096444)

[*3.3.* *Marker personalizzati* 7](#_Toc96096445)

[*3.4.* *Esecuzione di XAMPP sullo smartphone* 8](#_Toc96096446)

[4. Esempi di utilizzi 9](#_Toc96096447)

[5. Riferimenti 9](#_Toc96096448)

# Introduzione

## *Overview del progetto*

Realizzazione di modelli 3D/contenuti 3D (o modelli 2D/contenuti 2D) che saranno successivamente mostrati in realtà aumentata (AR) attraverso una pagina web dedicata.

Per la progettazione e modellazione dei contenuti 3D/2D vengono utilizzate diverse tecnologie, ad esempio Unity, AR.js, A-Frame, ecc….

Per la creazione della pagina, invece vengono utilizzati principalmente i linguaggi HTML, CSS e Javascript (con un possibile utilizzo di differenti framework, ad esempio Vue.js, Angular.js, ecc…).

In particolare, il percorso legato al progetto di stage può essere riassunto nei seguenti punti:

* Studio delle varie tecnologie e dei loro utilizzi
* Sviluppo demo con marker (vedi sezione 2.1 [Marker])
* Possibile sviluppo demo con geolocalizzazione

## *Deliverables del progetto*

Le scadenze del progetto sono state programmate nel seguente modo:

* Data inizio progetto: 14/02/2022
* Data fine progetto: 10/03/2022

## *Materiali di riferimento*

Durante l’interezza del progetto ci sarà l’appoggio a diverse tipologie di materiale, per quanto riguarda le tecnologie verranno utilizzate principalmente le documentazioni ufficiali.

Maggiori dettagli nella sezione Riferimenti (sezione 4).

# Definizioni e descrizioni

## *Definizioni/abbreviazioni delle tecnologie utilizzate*

Di seguito sono elencati i vari termini tecnici delle tecnologie utilizzate all’interno di questa documentazione di progetto, correlati da una descrizione.

### *Realtà aumentata (Augmented Reality, AR)*

Realtà aumentata: 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).

### *Realtà virtuale (Virtual Reality, VR)*

Realtà virtuale:

### *HTML (HyperText Markup Language)*

HTML è il linguaggio di markup standard utilizzato per le pagine Web. Il suo utilizzo principale è infatti la progettazione e realizzazione della struttura logica, definita appunto markup, di una pagina Web.

HTML è un linguaggio di pubblico dominio, deriva dall’SGML e la sua sintassi è regolata e stabilita dal World Wide Web Consortium (W3C).

HTML5 è l’ultima versione sviluppata che ha principalmente i seguenti obbiettivi:

* Estensione delle funzionalità che in precedenza erano fruibili solamente attraverso estensioni proprietarie
* Garantire maggiore compatibilità tra i vari browser
* Espansione verso i dispositivi mobili

### *CSS (Cascading Style Sheets)*

CSS è il linguaggio utilizzato per la formattazione di documenti HTML, XHTML e XML. La regolamentazione per la sua composizione è stata emanata dal W3C.

I CSS sono necessari per la separazione dei contenuti delle pagine HTML dalla loro formattazione o layout, permettendo così anche una programmazione più chiara.

### *Javascript*

Javascript è un linguaggio di programmazione orientato ad oggetti e agli eventi per la creazione di effetti dinamici interattivi tramite delle funzioni di script, invocate da eventi innescati dall’utente stesso.

È utilizzato maggiormente nella programmazione Web lato client, ma è stato successivamente esteso anche a lato server.

### *Realtà aumentata (AR) sul web*

Per realizzare un’esperienza di realtà aumentata (AR) sul web (attraverso AR.js) ci sono principalmente tre modalità: Image Based, Locaction Based e Marker Based (vedi sezione 2.1 [AR.js]).

Come realizzare un’esperienza di AR Marker Based 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. Inoltre, funziona su tutti i telefoni con WebGL e WebRTC.

### *WebGL*

WebGL è un’API multipiattaforma che viene utilizzata per creare grafica 3D in un browser Web. Si basa sull’utilizzo di OpenGL, GLSL e offre la familiarità dell’API OpenGL standard.

Essendo che la sua situazione avviene nell’elemento Canvas di HTML5, WebGL ha la piena integrazione con tutte le interfacce DOM (Document Object Model).

### *Three.js*

Three.js è una libreria Javscript cross-browser e un’API utilizzata per la creazione e l’animazione di computer grafica 3D in browser Web che utilizza WebGL.

### *Three.ar.js*

Three.ar.js è una libreria three.js di supporto utilizzata per sviluppare esperienze di realtà aumentata attraverso:

* **WebARonARKit**
* **WebARonARCore**.

Entrambe 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). Esso si basa su HTML, ma non è solo un linguaggio di markup, il suo punto di forza è fornire una struttura dichiarativa, estensibile e componibile a **three.js**.

A-Frame è stato progettato per sviluppare in modo efficiente contenuti VR ed attorno ad esso è nata una delle più grandi community VR, partendo da un progetto open source indipendente.

Esso supporta la maggior parte dei visori VR come Vive, Rift, Windows Mixed Reality, Daydream, GearVR, Cardboard, Oculus Go e mira a definire delle esperienze VR interattive completamente coinvolgenti che vanno oltre i contenuti base a 360°, sfruttando appieno il tracciamento posizionale e i controller.

Può essere utilizzato anche per la realtà aumentata.

### *Marker*

Nell’ambito della realtà aumentata, un marker è un’immagine che il sistema deve riconoscere per attivare i contenuti virtuali. Il software quindi, analizzando il flusso rilevato da una sorgente di acquisizione video (fotocamera/webcam) utilizzerà il marker come riferimento e posizionerà di conseguenza gli elementi 3D/2D.

Grazie all’evoluzione della tecnologia un marker può essere qualsiasi tipo di immagine.

Nel caso di AR.js i marker possono essere di tre tipologie:

* **Hiro**: marker di default (è uno dei marker predefiniti, attributo ‘preset’)
* **Barcode**: makers auto generati attraverso la computazione di matrici
* **Pattern**: makers personalizzati dall’utente

### *XAMPP*

XAMPP è una distribuzione di Apache gratuita che contiene al suo interno MySQL, PHP e Perl. È una piattaforma software multipiattaforma e libera, considerata come uno tra i più popolari ambienti di sviluppo PHP.

### *Unity*

Unity.

# Evoluzione e testing

In questa sezione sono elencati tutti i testing delle varie tecnologie e tutte le specifiche legate all’evoluzione del progetto nella sua interezza, arrivando fino alla versione finale.

## *Testing AR.js*

Testing tipologia di AR Marker Tracking

* Creazione di un sito web in locale per vederlo runnato su un server locale (utilizzato XAMPP e creazione di una nuova directory virtuale), prendendo spunto dal sito web <https://stemkoski.github.io/AR-Examples/>

Creazione pagina web di esempio AR.js-three.js [test-torusknot.html]

* 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

Creazione sample pagina web AR.js [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/rimpicciolimento-ingrandimento).

## *Testing A-Frame*

Alcuni test per capire il funzionamento di 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
* Creazione pagina web in locale per testing, prendendo spunti dal sito <https://glitch.com/edit/#!/aframe-model-viewer?path=index.html%3A4%3A4>

Successivamente le cose iniziano a diventare più complesse, complice anche il fatto della documentazione non troppo dettagliata di AR.js. C’è da ricordare però che stiamo parlando di cutting-edge technology ovvero una tecnologia nuova e presente da poco sul web.

## *Marker personalizzati*

Possono essere creati dei marker personalizzati, rendendoli appropriati (es. “brandizzandoli”). Per la creazione di un nuovo marker AR.js si può utilizzare il sito <https://jeromeetienne.github.io/AR.js/three.js/examples/marker-training/examples/generator.html>

Considerazioni per la realizzazione:

* Non tutti i tipi di immagine vanno bene per creare un marker che venga poi riconosciuto
* Le immagini non devono essere troppo complesse
* Possono esserci colori, ma deve esserci contrasto tra sfondo e simboli
* File .patt -> File che verrà utilizzato da AR.js per riconoscere il pattern del marker
  + 4 matrici bidimensionali che contengono 16x48 valori, in cui sono rappresentati i canali RGB
  + Solitamente file di pattern non validi hanno tutti 0 come valore (immagine completamente nera)
* Pattern ratio <= 0.75 (risultati migliori)

Creazione di vari marker con sfondo bianco/grigio

* Creazione di un marker con sfondo bianco e scritta nera [Riconoscimento: NOT SUCCESS]
* Creazione di tre marker con sfonfo grigio e scritta nera (più o meno dettagliati) [Riconoscimento: NOT SUCCESS]

Creazione marker4

* Creazione di un marker (marker4) personalizzato (scritta Max) con sfondo bianco
* Utilizzo di marker1 (uguale a marker4 ma con sfondo grigio) come immagine da riconosce
* Riconoscimento SUCCESS

Creazione “Company Marker”: Vegan Solutions (logo vegan/s)

* Creazione marker aziendale (come marker4)

## *Esecuzione di XAMPP sullo smartphone*

Avendo utilizzato XAMPP (web server Apache) tutto il codice delle pagine web può essere testato raggingendo <http://localhost/percorso>

Per testare il tutto anche da smartphone, avendo così un’esperienza di realtà aumentata migliore, XAMPP deve essere raggiungibile anche da mobile e in caso non lo sia già seguire i seguenti passaggi.

Quando pc e smartphone sono collegati alla stessa rete WIFI:

* Modificare file di xampp -> httpd-xampp.conf (Require all granted)
* Restart XAMPP e avviare Apache web server
* Nel caso in cui non sia già presente una Inbound Rule in Windows Firewall con la porta che utilizza XAMPP per la comunicazione:
  + Creare una Inbound Rule in Windows Firewall selezionando la porta corretta (80 nel mio caso) e permettendo la connessione
* Controllare da dispositivi nella rete l’indirizzo IP del pc con XAMPP (es. 192.168.4.73)

Tutto molto bello ma rimane il problema dell’https, ovvero per avere l’accesso alla fotocamera serve una connessione protetta.

* Soluzione migliore -> <https://ngrok.com/> (one command for an instant, secure URL to your localhost server through any NAT or firewall)
* Soluzione palesemente adottata da me -> mettere https davanti all’url 😊

# Esempi di utilizzi

In seguito un elenco di esempi di utilizzo delle varie tecnologie:

* Sito lamborghini (AR):

<https://www.lamborghini.com/sites/it-en/files/js_assets/experience/assets/ar/index.html?utm_source=3dplatform&utm_medium=qrcode&utm_campaign=augmented_reality_huracan_sto>

# Riferimenti

Per creare questo documento sono stati utilizzati come riferimento i seguenti materiali

Documentazioni/siti ufficiali:

* Documentazione ufficiale AR.js: <https://ar-js-org.github.io/AR.js-Docs/>
* Github organization AR.js: <https://github.com/ar-js-org>
* Documentazione ufficiale WebGL: <https://www.khronos.org/webgl/>
* Documentazione ufficiale three.js: <https://threejs.org/>
* Github three.ar.js: <https://github.com/google-ar/three.ar.js?files=1>
* Documentazione ufficiale A-Frame: <https://aframe.io/>
* W3school: <https://www.w3schools.com/>

Altro:

* Realtà aumentata con AR.js
  + <https://devsware.wordpress.com/2018/07/25/realta-aumentata-con-ar-js/>
  + <https://aframe.io/blog/arjs/>
* Github A-Frame Inspector: <https://github.com/aframevr/aframe-inspector>
* Xampp da mobile: <https://people.utm.my/shaharil/access-pc-localhost-xampp-server-from-mobile/>
  + <https://github.com/jeromeetienne/AR.js/issues/463>
* Unity (creazione macchina): <https://www.html.it/pag/383900/unity-creare-modello-3d-macchina/>