

**Progetto di Stage presso Vegan Solutions SRL
Via Contrà del Monte 13, Vicenza**

**Cailotto Massimo (880763@stud.unive.it)
Università Ca' Foscari Venezia**

**GitHub progetto:
https://github.com/maxcail8/progetto_stage**

**Documentazione di progetto
versione 1.0**

Demo AR

14 febbraio 2022

Indice

| | |
|--|-----------|
| 1. Introduzione | 3 |
| 1.1. Overview del progetto | 3 |
| 1.2. Deliverables del progetto | 3 |
| 1.3. Materiali di riferimento | 3 |
| 2. Definizioni e descrizioni | 4 |
| 2.1. Definizioni utili delle tecnologie utilizzate | 4 |
| 3. Evoluzione e testing | 8 |
| 3.1. Test AR.js | 8 |
| 3.2. Test A-Frame | 9 |
| 3.3. Marker personalizzati | 10 |
| 3.4. Esecuzione di XAMPP sullo smartphone | 11 |
| 3.5. Oggetti 3D personalizzati | 12 |
| 3.6. Test Unity | 13 |
| 3.7. Esportazione: Unity-gLTF | 13 |
| 3.8. Test AR.js e formato gLTF | 14 |
| 3.8.1. Interazione con gli oggetti 3D | 16 |
| 3.8.2. Oggetti 3D multipli | 17 |
| 3.9. AR.js e geolocalizzazione | 18 |
| 3.10. Test Three.ar.js | 19 |
| 4. Demo | 20 |
| 4.1. Demo Marker | 20 |
| 4.2. Demo Credit Card | 20 |
| 4.3. Demo Geo | 20 |
| 5. Tesi | 20 |
| 6. Riferimenti | 21 |

1. Introduzione

1.1. Overview del progetto

Il progetto di stage consiste nella gestione di modelli/contenuti tridimensionali (3D) e/o bidimensionali (2D) che saranno successivamente mostrati in realtà aumentata (AR) attraverso il web.

- Progettazione e modellazione dei contenuti 3D/2D attraverso l'utilizzo del motore grafico **Unity**
- Gestione e importazione di modelli in formato **glTF** per la visualizzazione sul web
- Creazione di pagine web con l'utilizzo dei linguaggi **HTML**, **CSS** e **Javascript** (con l'aggiunta di differenti framework e librerie)

In particolare, per l'interazione tra i contenuti creati digitalmente ed il web vengono utilizzate diverse tecnologie come **AR.js**, **A-Frame**, **three.js**,

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

1.2. 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

1.3. 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).

2. Definizioni e descrizioni

2.1. Definizioni utili 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)

La realtà aumentata è **un'esperienza interattiva** effettuata in un ambiente del modo reale. Gli oggetti presenti all'interno del mondo virtuale sono migliorati attraverso delle informazioni percettive generate dal computer e per fare ciò possono essere utilizzate più modalità sensoriali (legate ai cinque sensi).

La realtà aumentata può essere definita come un sistema avente principalmente le seguenti caratteristiche:

- Combinazione di mondi reali e virtuali
- Interazioni in tempo reale
- Registrazioni tridimensionali (3D) accurate di oggetti reali e virtuali

Un'esperienza di realtà virtuale è perfettamente intersecata con il mondo reale, infatti, deve essere percepita come un **aspetto immersivo** dell'ambiente reale. Essa è collegata ad altri due termini:

- Realtà mista (Mixed Reality, MR) detta anche realtà ibrida, ovvero l'unione di mondo reale e mondo virtuale per la creazione di nuovi ambienti all'interno dei quali fisico e digitale coesistono e operano in real-time
- Realtà mediata dal computer (Computer-Mediated Reality), ovvero la capacità di manipolare la propria percezione della realtà attraverso dispositivi portatili (smartphone) o computer indossabili

Realtà virtuale (Virtual Reality, VR)

La realtà virtuale è **un'esperienza di simulazione** che può essere simile o completamente differente rispetto al mondo reale. Gli ambienti di applicazione della realtà virtuale sono molteplici, ad esempio:

- Intrattenimento (video games)
- Educazione (addestramento militare/medico)
- Commerciale (meeting virtuale)

I sistemi di realtà virtuale fanno uso di **visori** per la realtà virtuale oppure di ambienti sviluppati ad hoc per generare immagini, suoni e altre sensazioni per l'utente all'interno di un ambiente virtuale. Pertanto, una persona che utilizza questo tipo di materiale sarà in grado di guardarsi attorno, muoversi e interagire nel mondo artificiale appositamente creato.

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 obiettivi:

- 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 può essere utilizzata AR.js e ci sono principalmente tre modalità: Image Traking, Location Based e Marker Based (vedi sezione 2.1 [AR.js]).

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 multiplatforma 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 Javascript 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 multiplatforma e libera, considerata come uno tra i più popolari ambienti di sviluppo PHP.

Unity

Unity è un motore grafico multiplatforma che può essere utilizzato per creare contenuti tridimensionali (3D) o bidimensionali (2D) ed esperienze interattive. Adottato principalmente nel mondo video games, esso si può estendere in altri ambiti come film, automazione, architettura, ingegneria, costruzioni, ecc...

Prefab (Prefab Assets) in Unity

In Unity un Prefab Asset può essere considerato l'equivalente di un template. A partire da un Prefab Asset si possono creare un qualsiasi numero di Prefab instances, che a loro volta possono essere salvate come parte della Scene oppure istanziate a runtime.

Per effettuare la creazione di un Prefab Asset si può utilizzare direttamente un GameObject, facendo così il GameObject compreso di tutte le sue componenti e sottocomponenti diverranno un nuovo Asset all'interno del progetto.

glTF (Graphics Language Transmission Format)

glTF è un formato di file utilizzato per la trasmissione e il caricamento in modo efficiente di scene e modelli tridimensionali (3D) da parte di motori grafici e applicazioni. Esso riduce anche al minimo le dimensioni delle risorse 3D e l'elaborazione runtime necessaria per poterle utilizzare.

glTF definisce un formato estensibile che semplifica i flussi di lavoro di creazione e i servizi interattivi, consentendo l'uso interoperabile di contenuti 3D.

Ci sono due possibili estensioni di file (.gltf e .glb) e si può fare riferimento a risorse binarie e texture esterne.

3. 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.

Un esempio di realtà virtuale, legata al web, è la tecnica di aggiungere (in tempo reale) degli oggetti virtuali ad immagini provenienti da una sorgente di acquisizione video (es. webcam).

Procedimento per 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/possedere 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

3.1. Test AR.js

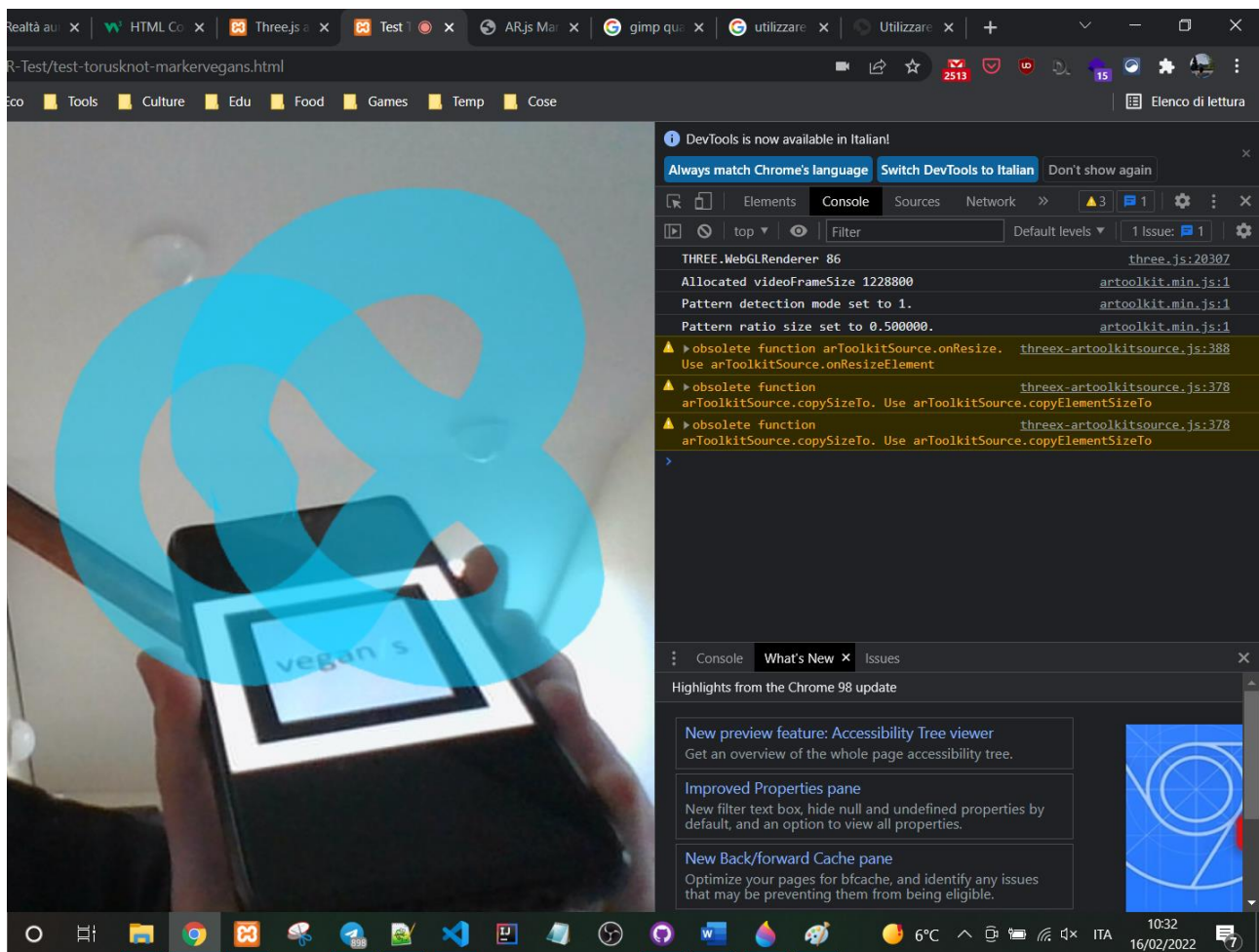
In questa sezione vengono effettuati i primi test legati alla tipologia di realtà aumentata (AR) Marker Tracking attraverso l'utilizzo di AR.js.

Test iniziale della libreria AR.js

- Creazione di un sito web in locale ed esecuzione dello stesso su un server locale, prendendo spunto dal sito web <https://stemkoski.github.io/AR-Examples/>
- Utilizzo di XAMPP con la creazione di una nuova directory virtuale

Test delle librerie AR.js-three.js [Cartella AR-Threejs-Test]

- Test di creazione e visualizzazione oggetto Torus Knot
- Utilizzo webcam del pc e immagine del marker pattern 'Hiro' sul cellulare per una corretta visualizzazione
- Utilizzo di three.js



Creazione pagina web AR.js [Cartella AR-samples] per capire il funzionamento di AR.js

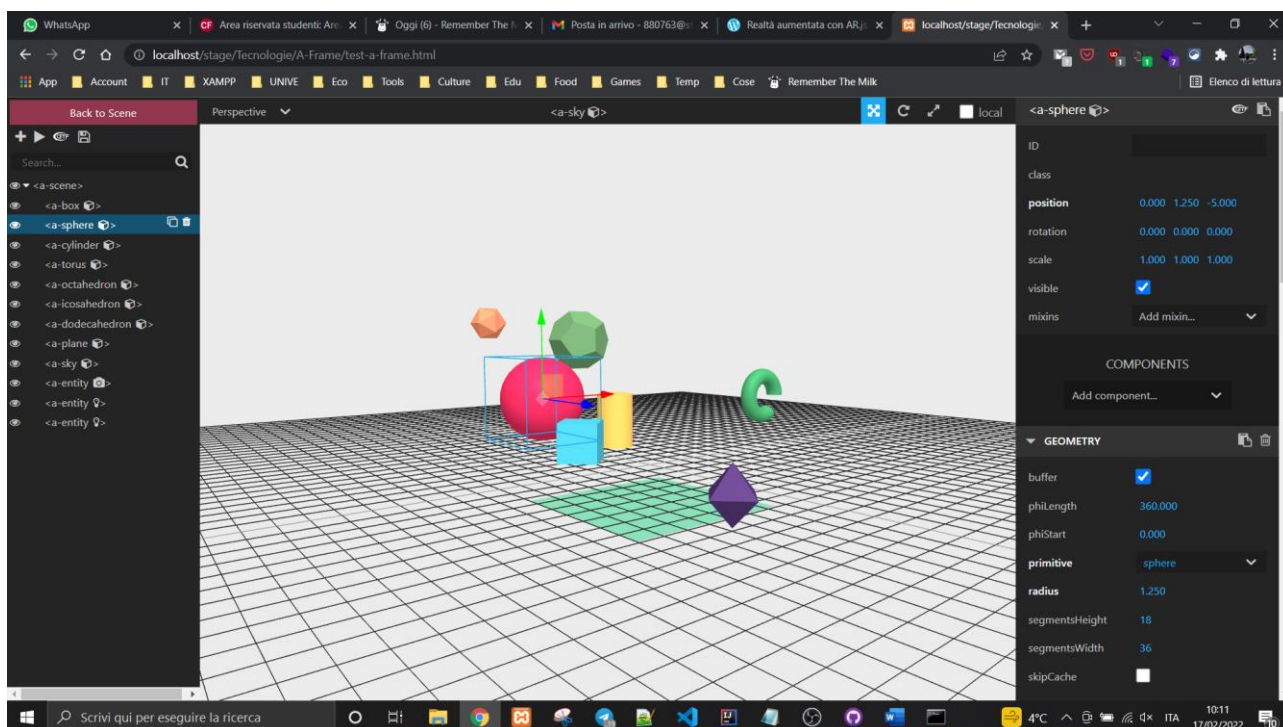
- Creazione pagina web e commenti per spiegare funzionamento

Attraverso la creazione di pagine di test 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 può notare l'adattamento dell'oggetto 3D al movimento (sinistra-destra/rimpicciolimento-ingrandimento).

3.2. Test A-Frame

Di seguito alcuni test per capire il funzionamento di A-Frame [Cartella A-Frame]

- Creazione di una semplice pagina web attraverso le primitive di A-frame
 - Aggiunta di alcuni elementi 3D
 - Ispezione attraverso lo strumento A-Frame inspector [cntrl + alt + i] per la visualizzazione ed il test dello spostamento dei vari elementi creati
- Creazione di una pagina web in locale per testare elementi, prendendo spunto dal sito <https://glitch.com/edit/#/aframe-model-viewer?path=index.html%3A4%3A4>



Successivamente le cose iniziano a diventare più complesse, complice il fatto della documentazione non troppo dettagliata di AR.js. Il tutto è capibile essendo comunque definite come cutting-edge technologies, ovvero tecnologie nuove e presenti da poco.

3.3. *Marker personalizzati*

In questa sezione si vuole sottolineare la possibilità di creare dei marker personalizzati, rendendoli più 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 [Cartella Marker]

- Non tutti i tipi di immagine creano un marker riconoscibile dal software
- Le immagini non devono essere troppo complesse
- Possono esserci colori, ma deve esserci un contrasto elevato tra sfondo e simboli
- Il file che verrà utilizzato da AR.js per riconoscere il pattern del marker è il .patt
 - 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)
- Il pattern ratio durante la creazione ≤ 0.75 (per risultati migliori)

Test di creazione di vari marker con sfondo bianco/grigio

- Creazione di un marker con sfondo bianco e scritta nera [Riconoscimento dal software: NOT SUCCESS]

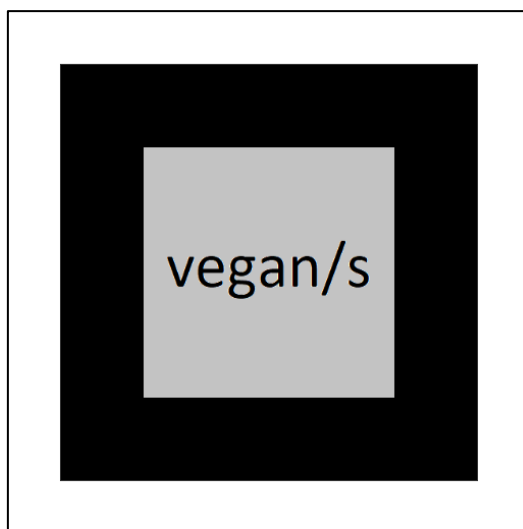
- Creazione di tre marker con sfondo grigio e scritta nera (più o meno dettagliati)
[Riconoscimento dal software: NOT SUCCESS]

Creazione marker4

- Creazione di un marker (marker4) personalizzato con sfondo bianco
- Utilizzo di marker1 (uguale a marker4 ma con sfondo grigio) come immagine per il riconoscimento
- [Riconoscimento dal software: SUCCESS]

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

- Creazione di un marker aziendale (procedimento come marker4)



3.4. Esecuzione di XAMPP sullo smartphone

XAMPP (web server Apache) è il server grazie al quale tutto il codice delle pagine web può essere testato raggiungendo <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. In caso non lo sia già seguire i seguenti passaggi.

Collegare pc e smartphone alla stessa rete WIFI

- Modificare il file di xampp -> httpd-xampp.conf (Require all granted)
- Restart di XAMPP e avvio di 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 in questo caso) e permettendo la connessione
- Controllare se collegandosi con un dispositivo nella rete all'indirizzo IP del pc appare l'homepage di XAMPP (es. 192.168.4.73)

Se si provano ad effettuare dei test collegandosi direttamente all'indirizzo probabilmente non si avrà successo. Infatti, deve essere utilizzato il protocollo https poiché 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 più pratica -> Mettere https davanti all'url e procedere

Testare la realtà aumentata da smartphone risulta particolarmente utile per quanto riguarda l'esperienza interattiva (molto più immersiva).



3.5. Oggetti 3D personalizzati

È possibile personalizzare la realtà aumentata, senza dover per forza inserire delle forme presenti sulle librerie o framework come three.js, A-Frame, ecc...

Possono essere creati degli oggetti 3D esternamente, personalizzandoli, tramite l'utilizzo di vari software come Blender, Unity, ... oppure utilizzare dei prefabbricati.

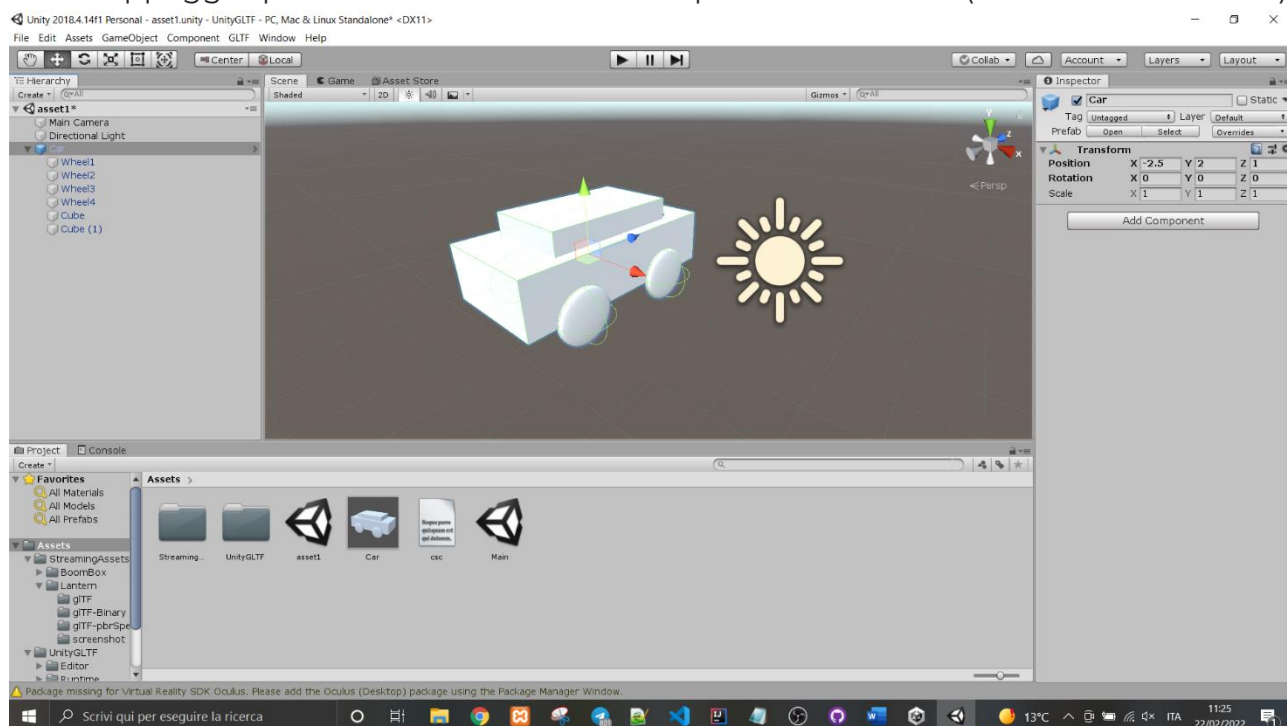
In ogni caso si dovrà poi provvedere all'importazione di tali oggetti all'interno della scena di AR.js.

Una cosa molto importante da sottolineare sono i formati accettati dal framework A-Frame e di conseguenza da AR.js, che, come da documentazione, sono i seguenti:

- **glTF** (formato consigliato)
- **OBJ**
- **COLLADA**

3.6. Test Unity

Per effettuare un primo test di creazione di un oggetto 3D personalizzato è stato utilizzato il sito <https://www.html.it/pag/383900/unity-creare-modello-3d-macchina/> come appoggio per la creazione di una semplice macchinina (stile cartoni animati).



3.7. Esportazione: Unity-glTF

Dato che i formati accettati da AR.js, e più in generale da A-Frame, sono glTF, OBJ e COLLADA l'oggetto creato con Unity deve poter essere esportato in uno di questi formati.

Questo è un problema.

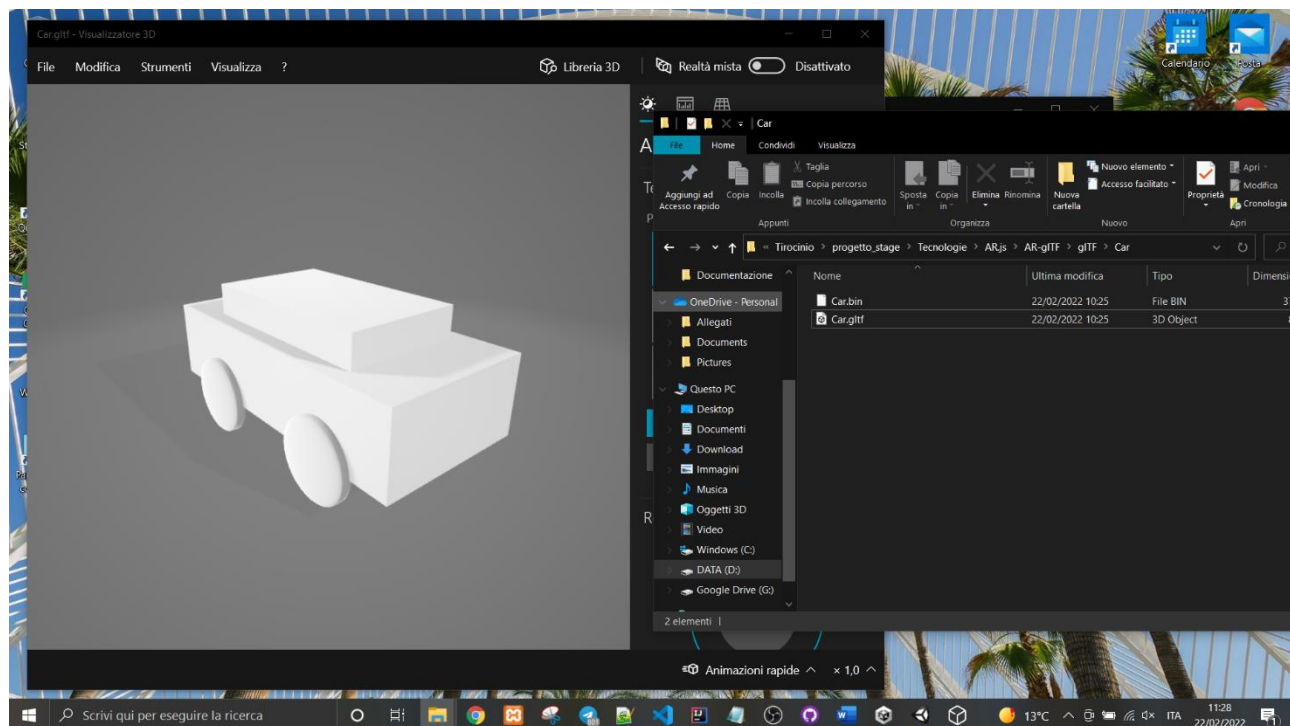
Si può creare un prefab da Unity (vedi sezione 2.1 [Prefab]), ma esportandolo direttamente si otterrà un file .prefab. Si deve quindi trovare un modo per effettuare l'esportazione da Unity ad un formato compatibile, ad esempio a glTF (tra l'altro formato consigliato da utilizzare). Già qui non è semplice, facendo alcuni test delle varie soluzioni online si può facilmente notare come il tutto non funzioni sempre e correttamente.

Tra le soluzioni principali per effettuare questa feature, troviamo:

- <https://github.com/KhronosGroup/UnityGLTF>
- <https://github.com/Siccity/GLTFUtility>
- <https://github.com/Plattar/gltf-exporter>
- <https://github.com/prefrontalcortex/UnityGLTF>

In questo caso l'esportazione è stata resa possibile dopo svariati test appoggiandosi al sito <https://github.com/KhronosGroup/UnityGLTF> tramite i seguenti passaggi:

- Clonazione della sorgente Github
- Installazione della versione Unity 2018.4.14f1 (necessaria per l'apertura del progetto UnityGLTF)
- Esecuzione di [GLTFSerializer](#) per il corretto funzionamento:
 - Apertura in Visual Studio e installazione di tutti i plugin/framework necessari
 - Compilazione della soluzione
- Apertura tramite UnityHub del progetto UnityGLTF all'interno della sorgente Github clonata
- Esempio di esportazione:
 - Dopo aver creato un nuovo GameObject convertirlo in Prefab
 - Direttamente sulla toolbar di Unity appare una nuova voce "GLTF", grazie al plugin installato
 - Cliccando lì, selezionare export selected



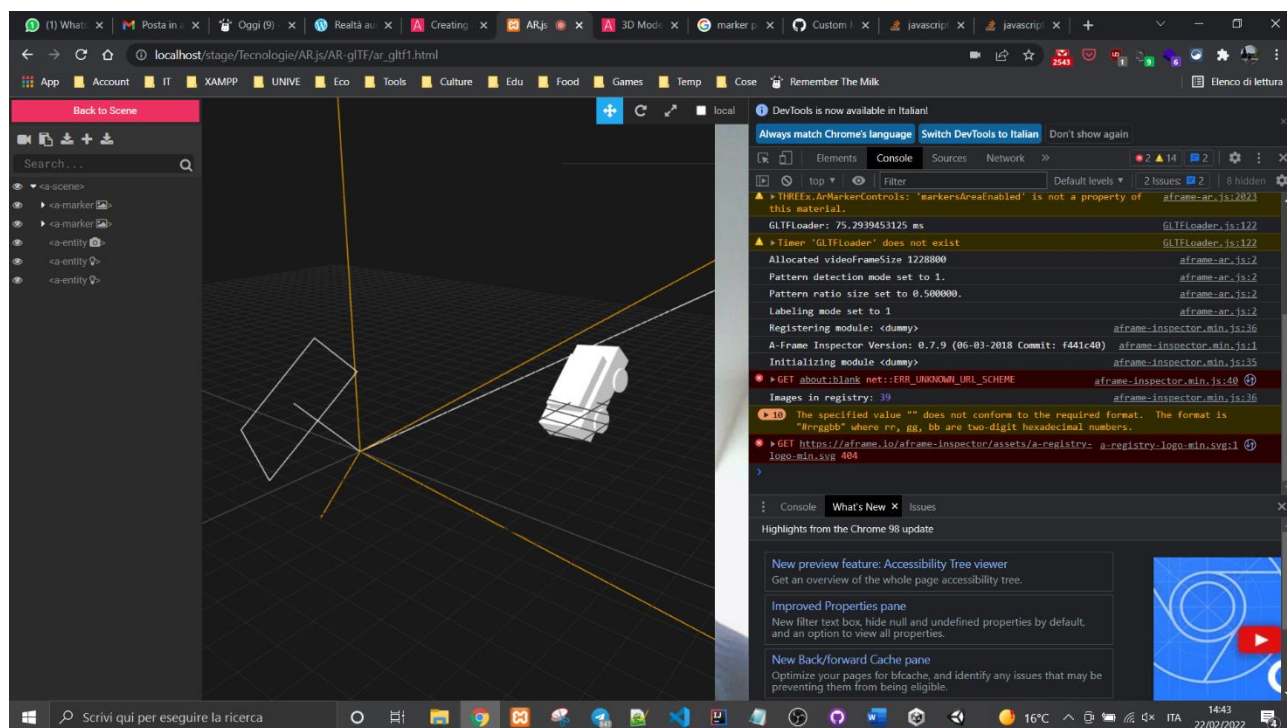
3.8. Test AR.js e formato glTF

Sezione dedicata a diversi test di AR.js con l'utilizzo di oggetti 3D formato gltf [Cartella AR-gltf]

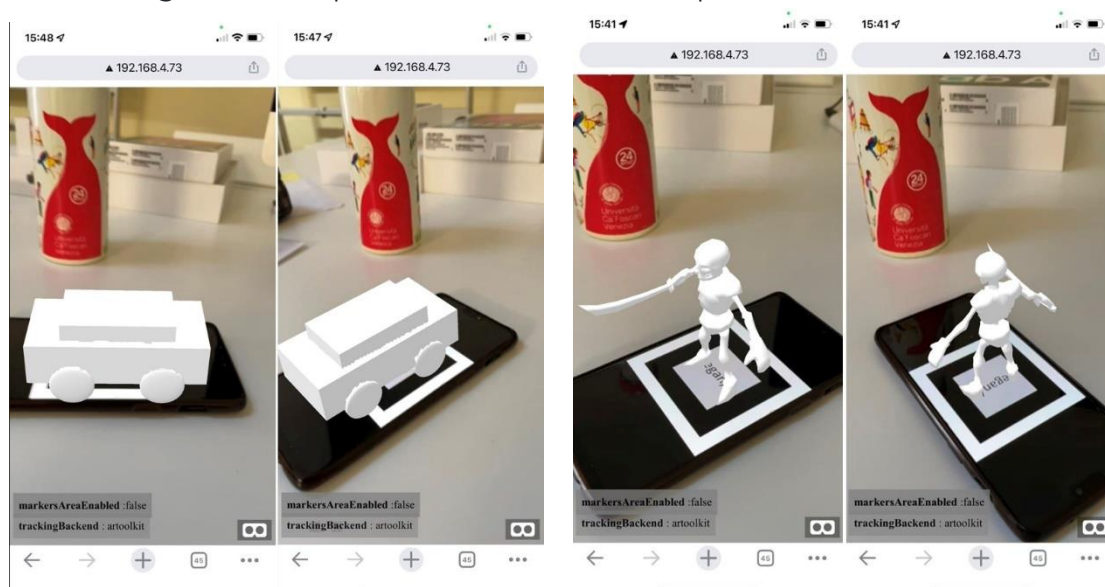
Caricamento del modello 3D

- Creazione di un proprio oggetto 3D personalizzato con formato .gltf
- Creazione di una pagina web per la corretta visualizzazione dell'oggetto (modifica attraverso i tag di A-Frame)

È possibile utilizzare lo strumento A-Frame inspector per controllare che il modello sia stato caricato correttamente.



Attraverso la sorgente di acquisizione video si vedrà quindi il modello 3D.



Animazione e texture del modello 3D

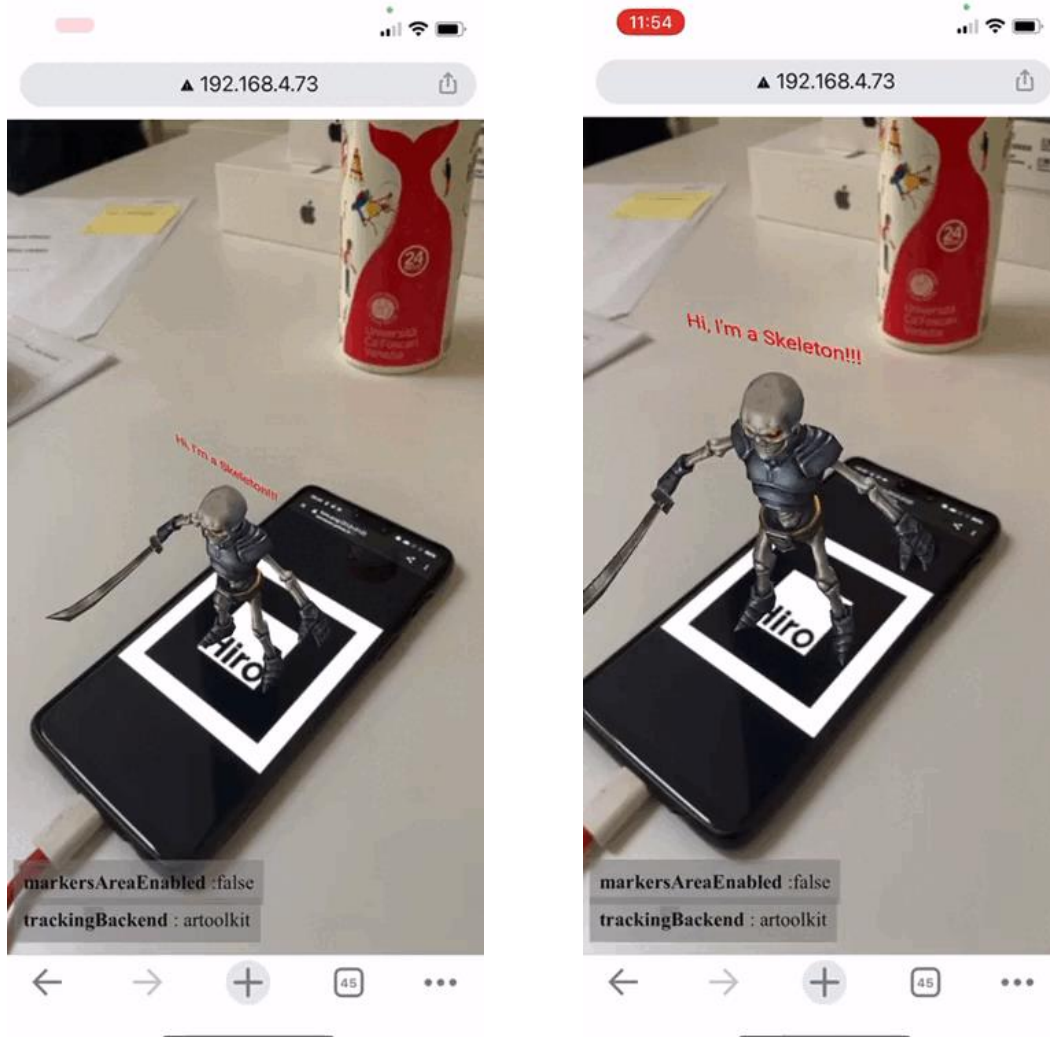
- C'è la possibilità di aggiungere un'animazione utilizzando direttamente le primitive di A-Frame (animazione di rotazione in questo caso).
- Creazione di una pagina web di test per l'aggiunta di una texture al modello 3D, utilizzo delle primitive di A-Frame combinate con la libreria three.js.

3.8.1. Interazione con gli oggetti 3D

Una volta creati gli oggetti 3D e visualizzati correttamente in realtà aumentata attraverso una pagina web è possibile migliorare l'esperienza ed inserire delle funzionalità legate all'interazione con l'utente (eventi custom, interazioni, ...).

Ad esempio, per le interazioni basilari con gli oggetti creati si può introdurre la manipolazione di contenuti tridimensionali attraverso delle gesture in AR.js.

Come da documentazione ufficiale è possibile appoggiandosi al sito github <https://github.com/fcor/arjs-gestures> per la creazione e gestione di gesture basilari.



3.8.2. Oggetti 3D multipli

È possibile caricare anche più oggetti 3D su un singolo marker. Per permettere una buona visualizzazione si possono utilizzare le proprietà scale e position per gestire la dimensione e la posizione dei vari oggetti 3D rispetto al marker.



3.9. AR.js e geolocalizzazione

Finora tutti i test di realtà aumentata effettuati sono così detti Marker Based, basati in effetti sull'utilizzo del concetto di marker (vedi sezione 2.1).

Tuttavia, l'esperienza di realtà aumentata può essere fatta anche attraverso l'utilizzo della geolocalizzazione.

Procedimento per realizzare un'esperienza di AR Location Based sul web:

- Possedere le proprie **coordinate geografiche** (latitudine e longitudine) che saranno successivamente utilizzate dal software per decidere dove posizionare gli elementi 3D
- Creare/possedere uno o più **oggetti 3D**. 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 e indicare, ad esempio ad AR.js, quali sono le coordinate geografiche da utilizzare
- Attivare il tracciamento **GPS** sul proprio dispositivo
- 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

[Cartella AR-geo]

Test iniziali AR.js Location Based

Creazione di una pagina web per test iniziali della geolocalizzazione con AR.js attraverso la visualizzazione di un testo rispetto alla propria posizione.

In questo contesto, l'aggiunta delle coordinate può essere fatta in tre modalità:

- Staticamente tramite HTML
- Staticamente tramite Javascript
- Dinamicamente tramite Javascript e APIs

Come da documentazione ufficiale, è possibile appoggiarsi al tutorial

<https://medium.com/chialab-open-source/build-your-location-based-augmented-reality-web-app-c2442e716564>, testando così le diverse tipologie.

Test GeoAR.js (statico con HTML)

Creazione di una pagina web per testare il caricamento di un oggetto 3D (pokemon) staticamente direttamente tramite HTML.

Test GeoAR.js (statico con Javascript)

- Creazione di una pagina web per testare il caricamento di un oggetto 3D (pokemon) staticamente attraverso l'utilizzo di Javascript.
- Creazione di una pagina web per eseguire lo swap di pokemon posizionati in un determinato luogo (lat, long)



Test GeoAR.js (dinamico con Javascript e APIs)

Creazione di una pagina web per testare il caricamento di link dinamicamente tramite Javascript e l'utilizzo di API (<https://foursquare.com/>).

3.10. Test Three.ar.js

Sezione dedicata a test relativi alla libreria three.ar.js. [Cartella Three.ar.js]

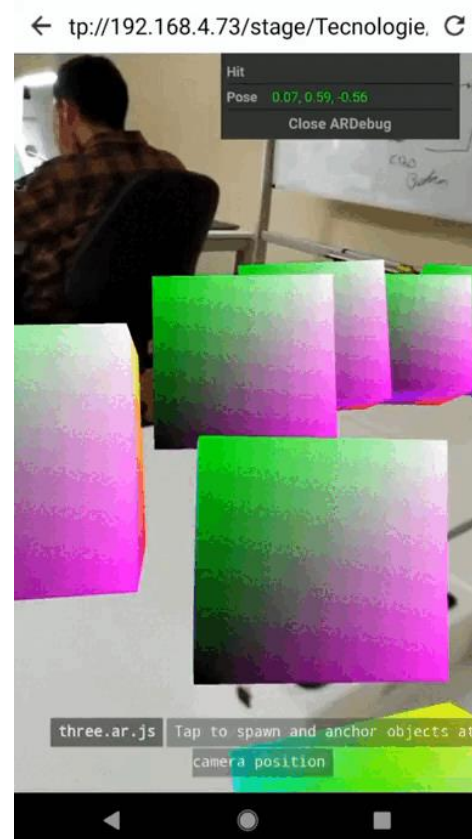
Sito github: <https://github.com/google-ar/three.ar.js>.

Test ed evoluzione:

- Installazione di ARCore e WebARonARCore su Google Pixel 2 per un funzionamento corretto
- Creazione di tre pagine web, rispettivamente:
 - Test posizionamento di oggetti 3D nel mondo reale
 - Test rilevamento "Plane" (o "Surface")
 - Test posizionamento e "Anchor" di oggetti 3D

Dai vari test effettuati si può comprendere come in questo campo le tecnologie esistenti possano ancora evolversi, le applicazioni utilizzate sono sperimentali e non possono essere sempre testate.

Inoltre, in differenti test le cose spesso non funzionano nel modo corretto o come ci si aspetti, complice il fattore di alta difficoltà relativo al tracking di oggetti nel mondo reale attraverso il web.



4. Demo

4.1. *Demo Marker*

Sito web dedicato:

- Augmented reality cartoon-style car
- Augmented reality skeleton: animation, texture, gesture
- Augmented reality vehicles: multiple marker
- Augmented reality monsters: multiple objects

4.2. *Demo Credit Card*

Sito web dedicato:

- Augmented reality credit card
Il concept iniziale era la realizzazione di una pagina web in grado di visualizzare informazioni relative ad una carta di credito direttamente inquadrandola.
Ciò è stato fatto in parte attraverso l'utilizzo di un marker dedicato.
Purtroppo, le tecnologie esistenti non permettono di effettuare attraverso il web tracking così performanti e precisi (vedi 3.10), se in futuro verranno sviluppate tecnologie di questo tipo questa demo potrebbe essere un buon punto di partenza.

4.3. *Demo Geo*

Sito web dedicato:

- Augmented reality pokemon swap
- Augmented reality points of interest

5. Tesi

Riferimenti utili per tesi

- AR
- VR
- **AR applicata al web**
- Differenza tra AR e VR
- Realtà aumentata basata sulla proiezione
- Realtà aumentata basata sul riconoscimento
- Realtà aumentata basata sulla posizione
- Realtà aumentata basata sul "tracciare"
- Realtà aumentata basata sulla sovrapposizione

Link Tesi

- <https://webthesis.biblio.polito.it/13182/1/tesi.pdf>
- http://tesi.cab.unipd.it/57105/1/Elena_Rostellato_2017.pdf

- <http://dspace.unive.it/bitstream/handle/10579/10683/842282-1210441.pdf?sequence=2>

6. 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/>
- Documentazione ufficiale Unity: <https://docs.unity3d.com/Manual/index.html>
- W3school: <https://www.w3schools.com/>
- Wikipedia
 - https://en.wikipedia.org/wiki/Augmented_reality
 - https://en.wikipedia.org/wiki/Mixed_reality
 - https://en.wikipedia.org/wiki/Computer-mediated_reality
 - https://en.wikipedia.org/wiki/Virtual_reality
 - [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))

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>
- Forum Unity-gLTF: <https://forum.unity.com/threads/gltfutility-a-simple-gltf-plugin.654319/>
- Modelli 3D
 - Unity Asset Store: <https://assetstore.unity.com/>
 - Sketchfab: <https://sketchfab.com/tags/unity3d>

Esempio di utilizzo nel sito lamborghini (AR): https://www.lamborghini.com/sites/it-en/files/js_assets/experience/assets/ar/index.html?utm_source=3dplatform&utm_medium=qr&utm_campaign=augmented_reality_huracan_sto