

olympics - Progetto Computer Graphics A.A.
2023/24

Filippo Brajucha - 1130613

Febbraio 2025

Contents

1	Introduzione	3
2	Descrizione	3
2.1	Struttura	3
2.2	Librerie	6
2.3	Comandi	6
2.3.1	Dat.GUI	7
2.3.2	Tastiera	7
2.3.3	Mouse	8
2.3.4	Touch-Screen	8
3	Funzionalità WebGL	8
3.1	Pipeline di Caricamento	9
4	Particolarità	9
5	Bibliografia	10

1 Introduzione

Documentazione del progetto di Computer Graphics 2023/2024, C.d.S. Informatica, Università di Bologna.

Il progetto sviluppato si chiama `olympics` e si tratta di un medagliere della Nazionale Italiana alle Olimpiadi di Parigi 2024.

L'idea è quella di raccogliere in un unico sito web i traguardi più importanti delle olimpiadi di Parigi 2024, utilizzando degli oggetti e modelli grafici per renderla più accattivante e interessante.

Nella pagina di focus viene visualizzato un breve articolo dove sono segnati i titolari delle vittorie con una piccola gallery e un'oggetto renderizzato in 3D "caratteristico" di quello sport.

2 Descrizione

Il sito web è organizzato in diverse pagine `.html` collegate tra loro tramite link ipertestuali a cui sono collegati degli script scritti in codice `javascript`.

Il sito è pensato per essere fruibile da qualunque dispositivo, con diverse funzionalità.

La navigazione è molto semplice ed è facilitata da una navbar che permette di accedere a quasi tutte le pagine.

Lo stile utilizzato ricalca quello scelto dal CONI, bianco e blu (`rgb(27, 33, 46)`) come nel sito parigi2024.coni.it e come font viene utilizzato Olympic Headline.

2.1 Struttura

L'organizzazione dell'applicativo è la seguente:

1. index.html
2. tab.html
 - focus.html
3. doc.html
4. about.html

La pagina 1 è accessibile da url, le pagine 2, 3, 4 sono accessibili da navbar partendo dalla 1, mentre la pagina "focus" viene visualizzata quando si clicca sulle canvas della pagina 2.

Nella pagina **1** viene mostrato il logo delle Olimpiadi 2024 fisso al centro della canvas, anche la luce è fissa che lo illumina mentre gira.

Nella pagina **2** è presente una tabella con le 10 discipline dove la Nazionale Italiana ha ottenuto la medaglia d'oro, gli oggetti (le medaglie) partono da ferme e cambiano posizione nel tempo se viene triggerato l'evento `mouseover`

della canvas specifica, ruotando. Anche in questa scena la luce è fissa e proviene dallo stesso lato della camera. Le medaglie sono presenti nella directory `res/models/medals`, essa contiene tutti i file `.obj` e `.mtl` e anche tutte le immagini di texture che sono applicate nel centro della medaglia e sono i pitogrammi originali sviluppati per le Olimpiadi 2024 (link).

La pagina 3 contiene questa documentazione in formato `html` o `pdf` (scaricabile dal pulsante in basso a destra).

La pagina 4 è un piccolo `about me` dove è presente un oggetto con la mia foto, alcuni informazioni utili e contatti personali.

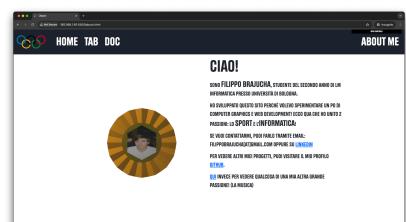
Nella pagina di `focus` viene renderizzato un articolo, il contenuto di questa pagina è dinamico, a seconda dell'oggetto selezionato nella tabella viene caricato nella canvas l'oggetto specifico di riferimento e, nella parte dell'articolo, quello specifico con le relative immagini (ogni articolo si può trovare sotto-forma di codice `.html` nella directory `olympics/res/articles`).



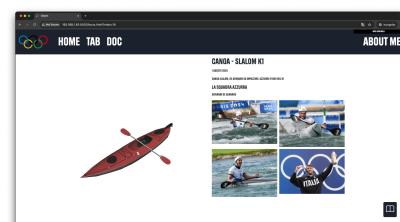
(a) Pagina 1



(b) Pagina 2



(c) Pagina 3



(d) Pagina focus

Figure 1: Le 4 pagine più importanti

Nella versione mobile non ci sono differenze per quanto riguarda la struttura del sito, se non per la navbar (fig. f) che viene ridotta ad un'icona e al pulsante in basso a destra nella pagina focus che non è presente. Ecco come cambia l'aspetto delle pagine mostrate sopra.



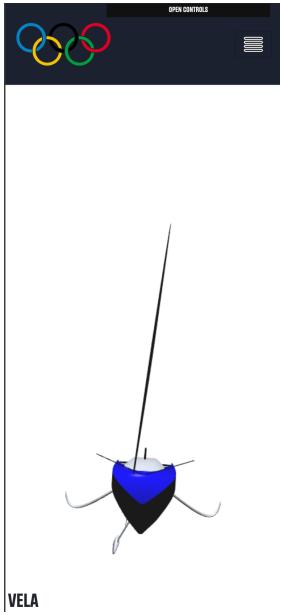
(a) Pagina 1



(b) Pagina 2



(c) Pagina 3



(d) Pagina focus



(e) Pagina focus



(f) Navbar

Figure 2: Le pagine più importanti in versione mobile e la navbar

2.2 Librerie

Per facilitare lo sviluppo dell'applicativo vengono utilizzate diverse librerie, più o meno complesse.

Per quanto riguarda la parte di impaginazione, e in generale il frontend, vengono utilizzati **Bootstrap** ([link](#)) e **BootstrapIcon** ([link](#)). Questi permettono di stilizzare in maniera facilitata i vari componenti del sito e gestire le media query.

Per la GUI viene utilizzata, come consigliato, la libreria **dat.GUI** ([link](#)) che permette di sviluppare in modo semplice un'interfaccia di comando per l'utente.

Per la renderizzazione degli oggetti 3D nelle canvas viene fatto largo uso delle librerie **WebGLFundamentals** e **m4** ([link](#)), queste facilitano di molto il lavoro con WebGL la prima e con i vari calcoli matriciali la seconda.

Il codice sviluppato contiene alcune componenti illustrate nel sito di **WebGLFundamentals**.

Nel codice sono presenti altre 2 librerie sviluppate da me che permettono di integrare i controlli alla canvas (**controls.js**) e di avere una serie di funzioni utili per la renderizzazione (**functions.js**) senza riempire troppo il codice del file **loader.js**.

Nel file **loader.js** è contenuta la funzione **render** che è il core della renderizzazione, grazie a questa vengono costantemente aggiornate le canvas.

2.3 Comandi

I comandi vengono utilizzati per spostare la vista nello spazio, proprio come se l'utente stesse usando una camera.

D indica la distanza dal centro, mentre **PHI** e **THETA** sono i 2 angoli che regolano, rispettivamente, lo spostamento verticale e orizzontale.

I comandi si possono dare tramite GUI (vedi dat.GUI), mediante mouse e

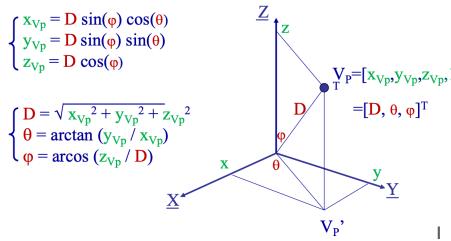


Figure 3: Schema di vista: D, PHI e THETA

tastiera, come specificato di seguito oppure attraverso touch-screen (a seconda del dispositivo).

Nella pagina focus viene renderizzato, con schermi *non-piccoli*, un modale attivabile tramite il pulsante in basso a destra con delle "istruzioni" sui tasti utili.

2.3.1 Dat.GUI

La libreria dat.GUI mette a disposizione una GUI facile da usare e implementare, agisce direttamente sui valori utilizzati dalla funzione di `render` durante la renderizzazione dell'oggetto.

Un'altra feature molto comoda di questa libreria è la creazione di limiti per i valori in input che devono rientrare in un certo range. Nel mio caso viene sfruttata per modificare le seguenti variabili con i seguenti limiti: Con i comandi si

var	min	max
D	0	30
THETA	0	2π
PHI	$-\pi/2$	$\pi/2$
Ka	0	1
Kd	0	1
Ks	0	1
xLight	-100	100
yLight	-100	100
zLight	-100	100

può agire anche sulle luci, modificando i parametri di Ka, Kd e Ks si modificano i canali di intensità, mentre con xLight, yLight e zLight la posizione della luce nello spazio in modo da illuminare meglio l'oggetto.

La GUI sviluppata con questa libreria si trova all'interno del codice della pagina `controls.js`.

2.3.2 Tastiera

La tastiera è un altro strumento di input, ci sono delle hotkey che vengono utilizzate per cambiare alcuni parametri che possono essere modificati anche con la GUI, eccoli qua di seguito: I valori modificati tramite tastiera aggiornano

hotkey	var	type
A	THETA	+
ArrowLeft	THETA	+
D	THETA	-
ArrowRight	THETA	-
W	PHI	+
ArrowUp	PHI	+
S	PHI	-
ArrowDown	PHI	-

i valori nella GUI in real-time rendendo utilizzabile anche una "combinazione" dei due sistemi di comando. Inoltre i limiti impostati nella GUI sopra descritta sono gli stessi che si devono rispettare nella modifica da tastiera.

2.3.3 Mouse

Tramite un meccanismo di *premi e trascina* è possibile utilizzare anche il mouse sulla canvas, la posizione del cursore indica le variazioni dei parametri **THETA** e **PHI**.

Con il mouse è anche possibile utilizzare la rotella per aumentare o diminuire la distanza della camera come fosse uno zoom.

Da sottolineare il fatto che si può effettuare lo zoom-in (quindi diminuire la distanza) entro un certo limite che viene calcolato a partire dagli extents dell'oggetto, viene mantenuto, quindi, sempre un certo raggio di distanza per non creare problemi con la renderizzazione.

2.3.4 Touch-Screen

La gestione dei comandi avviene anche per schermi piccoli (e quindi verosimilmente touch-screen), infatti tramite le callback `ontouchstart`, `ontouchend`, `ontouchmove`, `touchcancel` vengono gestiti gli eventi di pressione dello schermo come se fosse utilizzato il mouse.

3 Funzionalità WebGL

Viene fatto largo uso di WebGL, a partire dalla prima pagina dove è presente nel centro una canvas, nella seconda pagina vengono utilizzate ben 10 canvas che contengono un oggetto e la pagina focus che contiene un'altra canvas, come anche la pagina di about.

Le funzionalità sono differenti, la pagina iniziale e quella della tabella sono fisse, l'utente non può interagire. Per dispositivi *non touch* è sviluppata anche un'animazione nella pagina della tabella, essa fa sì che la posizione dell'oggetto (la medaglia) cambi nel caso in cui si passi (e si rimanga) sopra con il mouse. Sostanzialmente la medaglia ruota sul suo asse verticale, l'illuminazione è fissa e frontale in modo da tenere l'oggetto illuminato.

Nella pagina di focus l'utente è più libero di muovere e orientare la camera nello spazio, così può osservare meglio gli oggetti renderizzati. In questa pagina l'utente può anche spostare la posizione della luce, così da scegliere l'illuminazione più corretta per l'oggetto nella scena.

Vengono utilizzate le funzioni `parseObj` e `parseMaterial` messe a disposizione da WebGLFundamentals ([link](#)).

Mentre per la parte di luci si fa riferimento a ciò che si è visto a lezione e alla sezione "Lighting" di WebGLFundamentals ([link](#)).

Nella renderizzazione della canvas vengono utilizzati degli elementi scritti in GLSL, il `vertex-shader` e il `fragment-shader`, dichiarati all'interno del file `functions.js`, che permettono, ripetutivamente, la renderizzazione dei punti (vertici) parsati grazie alla funzione di parsing del file `.obj` e dei colori, parsati grazie alla funzione di parsing del file `.mtl`.

3.1 Pipeline di Caricamento

Dagli script associati ad ogni file .html viene richiamata la funzione `main` dello script `loader.js`. Essa prende in input:

- `objIndex` - indice grazie al quale viene recuperato l'url del file obj
- `gl` - context della canvas
- `meshProgramInfo` - il programma utilizzato da quel context (creato con `gl, fragment shader` e `vertex shader`, quindi sostanzialmente sono i riferimenti della canvas e i metodi grazie ai quali riesce a plottare gli oggetti nello spazio 3D e colorarli)

Dopo il caricamento del file vengono eseguite delle funzioni e dei calcoli per capire le dimensioni e le caratteristiche dell'oggetto, vengono inizializzate le matrici della camera e della luce.

A seconda di vari parametri come il `renderStatus`, vengono attivate o meno delle funzionalità, come la possibilità di muovere l'oggetto liberamente o la camera fissa, in particolare:

Status	Pagina	Props
0	tab	Camera fissa, luce fissa, oggetto fisso
1	index	Camera fissa, luce fissa, oggetto gira lentamente
2	tab	Camera fissa, luce fissa, oggetto gira velocemente
3	focus	Camera modificabile da utente, oggetto fisso, luce mobile
4	about	Camera modificabile da utente, oggetto fisso, luce fissa

4 Particularità

Una particolarità interessante del progetto sviluppato è che le medaglie visualizzate nella pagina "tab.html" e nella pagina "about.html" sono sviluppate da me in Blender from scratch, partendo proprio da un cilindro, applicando trasformazioni sui volumi e i diversi materiali e texture una volta ultimato.

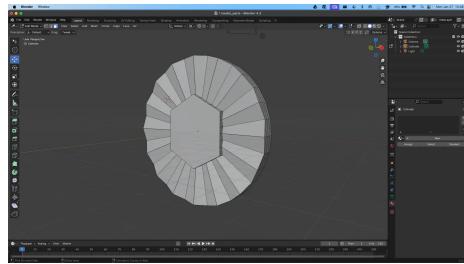


Figure 4: Modellazione Blender della medaglia

5 Bibliografia

1. <https://webglfundamentals.org/webgl/lessons/webgl-load-obj-w-mtl.html>
2. <https://parigi2024.coni.it/it>
3. <https://www.olympics.com/en/news/paris-2024-reveals-look-of-games-62-pictograms>
4. <https://webglfundamentals.org/>
5. <https://github.com/dataarts/dat.gui>
6. <https://webglfundamentals.org/webgl/lessons/webgl-3d-lighting-directional.html>
7. <https://getbootstrap.com/>
8. <https://icons.getbootstrap.com/>
9. <https://github.com/dataarts/dat.gui>

Appendix