

ALMA MATER STUDIORUM - UNIVERSITA DI BOLOGNA
SCUOLA DI INGEGNERIA E ARCHITETTURA

Dipartimento di Informatica - Scienza e Ingegneria - DISI
Corso di Laurea Triennale in Ingegneria Informatica

TESI DI LAUREA

in

ELETTRONICA

**RILEVAZIONE ED ELABORAZIONE DEI
DATI DEL MONDO REALE PER LA
REALTÀ VIRTUALE: VRBIKE**

CANDIDATO

Lorenzo Andraghetti

RELATORE

Chiar.mo Prof. Bruno Riccò

CORRELATORE

Prof. Massimo Lanzoni

Anno Accademico 2016/17
Sessione II

Ai miei nonni e ai miei genitori.

I miei pilastri.

Indice

1	Introduzione	7
1.1	Struttura della tesi	7
2	Realtà Virtuale	9
2.1	Come si crea la Realtà Virtuale	10
3	Architettura del sistema	13
3.1	Analisi del sistema hardware	13
3.1.1	Sensori del manubrio	14
3.1.2	Sensori del volante	15
3.1.3	Microcontrollore	15
3.1.4	Ricevitore Bluetooth	16
3.1.5	OSVR	16
3.2	Analisi della Parte Software	18
3.2.1	Blender	18
3.2.2	Unity	22
3.2.3	Motocontroller script	24
3.2.4	ThreadDTWSlideWindow script	24
4	Realizzazioni sperimentali e valutazione	25
4.0.1	OSVR e le impostazioni iniziali	25
5	Direzioni future di ricerca e conclusioni	27
	Bibliografia	29
A	Documentazione del progetto logico	31

Capitolo 1

Introduzione

La tesi si prefigge la realizzazione di una applicazione che riveli le informazioni del mondo reale per elaborarle così da modellare una realtà virtuale.

Si tratta quindi di un programma che permetta all'utente, dotato di visore per la realtà virtuale, la possibilità di muoversi all'interno di un mondo tridimensionale. La possibilità di movimento è data da una cyclette, in modo da dare all'utilizzatore la sensazione di muoversi lungo un percorso su una bicicletta. La cyclette è fornita di particolari sensori che ne determinano velocità di pedalata e rotazione del manubrio. Lo scopo della tesi è quello di far comunicare questi sensori con un computer in modo da pilotare la bicicletta virtuale.

In futuro si cercherà di utilizzare una bicicletta vera, applicata su rulli che diano un ritorno di forza per implementare la sensazione di fatica in casi di salita o discesa.

1.1 Struttura della tesi

La tesi è strutturata nel modo seguente.

Nella sezione *Realtà Virtuale* si illustrerà lo stato dell'arte nell'ambito della realtà virtuale. Lo studio di questo ambito è stato reso necessario per la progettazione.

Nella sezione *Architettura del sistema* si mostrerà il progetto nella sua interezza con la descrizione di tutti i moduli.

Nella sezione *Realizzazioni sperimentali e valutazione* si descriveranno al-

cune prove di calibrazione dei valori relativi alla fisica del mondo virtuale. Nelle *Direzioni future di ricerca e conclusioni* si riassumono gli scopi, le valutazioni di questi e le prospettive future.

Capitolo 2

Realtà Virtuale

Ultimamente si parla sempre più spesso di *Realtà Virtuale* ma cos'è, come si crea e a cosa serve? Il termine *Realtà Virtuale* fu coniato nel 1989 da Jaron Lanier, una delle prime persone che iniziò a lavorarci. Con il termine Realtà virtuale (Virtual Reality) si indica una Realtà simulata che si realizza attraverso la ricostruzione di ambienti o oggetti in modo da dare al soggetto una percezione il più possibile realistica della loro esistenza. A questa definizione tecnica si può affiancare una più generale, che vede la Realtà virtuale come una Realtà parallela. È possibile distinguere due tipi di Realtà virtuale: una immersiva e una non immersiva. Si definisce immersiva la Realtà virtuale che è in grado di assorbire l'utente, realizzando una vera e propria immersione dei sensi nell'ambiente tridimensionale generato dal computer. Ciò è possibile attraverso l'utilizzo di dispositivi particolari: un visore, che permette la visualizzazione delle immagini tridimensionali e l'isolamento dall'ambiente esterno, ed un tracker, per il rilevamento di posizione e movimento dell'utente. Oltre a questi ne esistono molti altri, di cui alcuni ancora in fase di studio e perfezionamento. La Realtà virtuale non immersiva, invece, è caratterizzata dall'utilizzo di un monitor per la visualizzazione delle immagini tridimensionali e non si serve dell'ausilio di un tracker determinando nell'utente la sensazione di vedere il mondo tridimensionale, creato dal computer, come attraverso "una finestra" e in maniera quindi non partecipativa.

2.1 Come si crea la Realtà Virtuale

La Realtà Virtuale è possibile grazie a strumenti, programmi e linguaggi di programmazione appositi. Nella tesi proposta, si è scelto di utilizzare una Realtà Virtuale di tipo immersivo. Questa tipologia richiede una strumentazione più complessa e performante rispetto alla tipologia non immersiva. Gli strumenti che richiede, sono i seguenti:

- **Computer:** tutte le periferiche utilizzate fanno capo a un hardware, se non integrato, che funge da fulcro per l'elaborazione e lo smistamento dei dati. Questo fulcro è spesso rappresentato da un computer dalle prestazioni elevate, in termini di CPU e GPU. La sua funzione, oltre a mantenere uno stato aggiornato, è quella di ricevere dati da sensori esterni, utili per recepire le azioni dell'utilizzatore, e inviare informazioni di feedback delle azioni elaborate. Ad esempio l'aggiornamento delle immagini visualizzate e l'attivazione di attuatori per altri tipi di sensazioni (ad esempi tattili).
- **Visore:** costituito da due pannelli LCD impostati per la vista binoculare, questo isola dal mondo esterno e costituisce una specie di casco. Questa impostazione dà la sensazione di tridimensionalità. Il visore può essere dotato di sensori in grado di rilevare i movimenti dell'utente come la rotazione del capo per inquadrare un'altra area del mondo virtuale. In alternativa, esistono visori economici che non sono altro che contenitori, muniti di lenti, che permettono di inserire lo smartphone e utilizzarlo come schermo LCD e come fulcro di tutte le funzioni hardware e software.
- **Auricolari:** solitamente integrati nei visori, permettono di udire i suoni emessi dal mondo virtuale. Il software deve variare i suoni emessi in base alla posizione dell'utilizzatore.
- **Sensori:** per il riconoscimento dei movimenti e delle azioni dell'utilizzatore. I sensori presenti nel visore danno informazioni sulla posizione, sul movimento della testa e del corpo. I sensori sono di vario tipo e sono in continuo sviluppo. I più conosciuti sono: guanti, in sostituzione dei canonici gestori di input (joystick, mouse, tastiere, ecc.); tute, in grado di trasferire le posture e i movimenti dell'utente

nella rappresentazione e Virtuix Omni, una piattaforma che permette di muoversi camminando nell'ambiente virtuale.

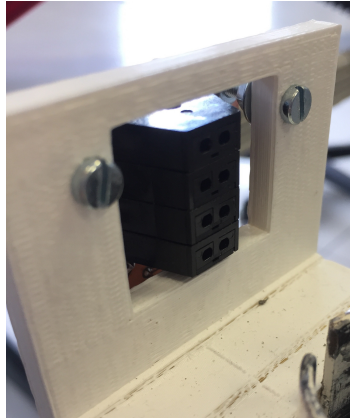
Capitolo 3

Architettura del sistema

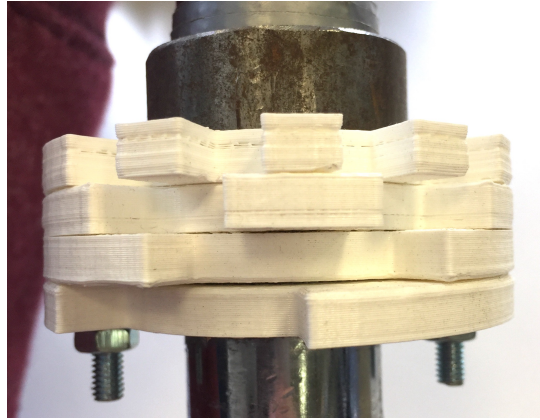
In questo capitolo si descriverà la progettazione del sistema creato. Inizialmente si tratterà l'analisi del progetto e le scelte effettuate per quanto riguarda gli strumenti da utilizzare. Successivamente si descriveranno tutti i moduli che compongono l'architettura del sistema. La trattazione è divisa in parte hardware e parte software.

3.1 Analisi del sistema hardware

L'obiettivo del progetto è quello di creare uno strumento che permetta all'utilizzatore di pedalare, sterzare e osservare un luogo in un mondo virtuale. In primo luogo, era necessario creare un sistema simile ad una bicicletta. Ai fini di sperimentazione della tipologia di progetto, si è scelto di utilizzare una cyclette. La suddetta, permette di semplificare notevolmente il sistema elettronico e il sistema software, poiché questi non devono tenere conto dell'attrito e del ritorno di forza, in quanto una pedalata farà sempre ruotare il volano. La cyclette è inoltre sprovvista di freni, i quali potrebbero essere utilizzati per frenare la bicicletta virtuale. Si è quindi scelto di ottenere solo le informazioni relative alla pedalata e alla posizione del manubrio. Queste informazioni devono essere elaborate da un microcontrollore che ottiene i dati da tutti i sensori e genera una macro-informazione da inviare al sistema software.



(a) I 4 fotodiodi



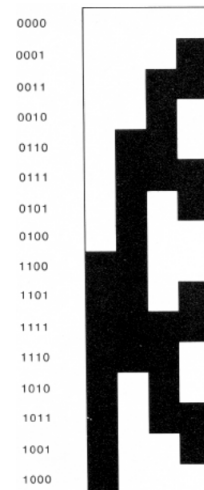
(b) Le 16 posizioni in permutazione ciclica

Figura 3.1: Sensori del manubrio

3.1.1 Sensori del manubrio

Per ottenere le informazioni relative alla posizione del manubrio è stato scelto un angolo massimo di rotazione di 120° . Questo angolo è stato diviso in 16 posizioni.

Per poter rilevare la posizione corrente è stata necessaria una codifica in 4 bit. Per ottenere questa codifica, si è fatto uso di 4 fotodiodi¹ che permettono di rilevare i 4 spazi che compongono una delle delle 16 configurazioni. I fotodiodi permettono di distinguere uno spazio vuoto da uno spazio pieno creando un segnale elettrico. Le configurazioni sono state realizzate con un oggetto stampato con stampante 3D, come si vede in figura 3.1b, in permutazione ciclica (figura a lato), che permette di accorpare gli *uni* e gli *zeri* e ridurre l'alternanza di *vuoti* e *pieni*.



¹Il fotodiodo è un particolare tipo di diodo fotorilevatore che funziona come sensore ottico sfruttando l'effetto fotovoltaico, in grado cioè di riconoscere una determinata lunghezza d'onda dell'onda elettromagnetica incidente (assorbimento del fotone) e di trasformare questo evento in un segnale elettrico di corrente applicando ai suoi estremi un opportuno potenziale elettrico. Esso è dunque un trasduttore da un segnale ottico ad un segnale elettrico.

3.1.2 Sensori del volano

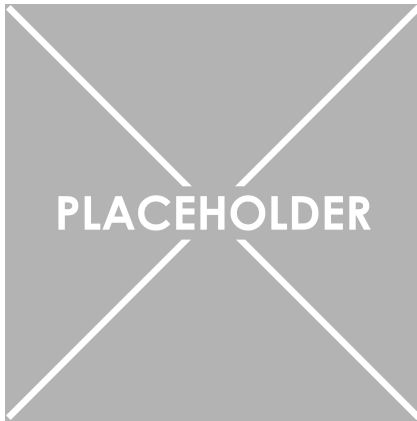


Figura 3.2: Il volano della cyclette e l'accelerometro

La bicicletta virtuale deve muoversi ad una velocità consona alla rotazione del volano della cyclette. Quest'ultimo è stato quindi munito di un accelerometro che fornisce informazioni sulla velocità di rotazione. Sui pedali sono stati inoltre installati due fotodiodi che permettono distinguere se la rotazione è oraria o antioraria. La direzione di rotazione viene ottenuta distinguendo quale dei due fotodiodi viene oscurato per primo. Ad esempio, se consideriamo due fotodiodi disposti orizzontalmente:

- Se un oggetto passa da sinistra a destra, si oscurerà prima il fotodiodo sinistro poi il destro, quindi assumendo un oggetto che abbia il fulcro di rotazione sotto i fotodiodi, la sua rotazione è **oraria**
- in caso contrario, la rotazione è **antioraria**.

3.1.3 Microcontrollore

Il cuore della parte hardware è la scheda Arduino che elabora tutti i dati ricevuti via cavo o via bluetooth. La scheda con il microcontrollore è in continua comunicazione in input e in output con il modulo bluetooth poiché deve ricevere l'informazione sulla rotazione dall'accelerometro e deve inviare all'elaboratore esterno una stringa contenente tutte le informazioni sotto forma di stringa.



Figura 3.3: La scheda Arduino Micro e il modulo bluetooth

Stringa

La stringa elaborata dall'Arduino è composta come segue:

- **Posizione del manubrio:** un valore compreso tra 0 e 15 che indica la posizione del manubrio
- **Velocità di rotazione:** un valore compreso tra 0 e 40 che indica la velocità di rotazione della pedalata.
- **Senso di rotazione:** un segno + o - che indica senso orario o antiorario di rotazione della pedalata.

Un esempio pratico è **12+27.5**: significa che si sta registrando una pedalata in senso orario con velocità 27.5 e il manubrio è nella posizione 12.

3.1.4 Ricevitore Bluetooth

La ricezione della stringa avviene tramite un ricevitore bluetooth collegato ad una porta USB del computer generale in cui viene eseguito il software. Utilizzando un programma terminale apposito, chiamato HyperTerminal è stato possibile leggere la porta seriale COM a cui vengono inviati i dati a 9600 bit/s e testare la parte hardware.

3.1.5 OSVR



OSVR é l'acronimo di *Open-Source Virtual Reality* ed identifica la piattaforma software open-source per la Realtà Virtuale e quella aumentata.

È stata sviluppata dalla Razer e da Sensics. La prima è l'azienda leader nel settore del gaming, mentre la seconda lo è nel settore della realtà virtuale. Affinché questa piattaforma sfondasse nel mercato, le società sopracitate hanno deciso di rendere open-source sia il software sia l'hardware per i programmatori. È infatti reperibile su github il codice sorgente. OSVR consente in modo davvero semplice la configurazione e la gestione dei vari dispositivi: occhiali VR, inseguitori di posizione, periferiche di gioco e vari altri.

Caratteristiche

L'OSVR è composto da un display montato su una visiera, o meglio head-mounted (HMD), da 2 lenti amovibili, da un display OLED da 5.5 pollici con risoluzione 1920 x 1080 pixel, 60 fps di refresh rate e campo visivo di 100 gradi. Contiene una scheda madre riprogrammabile con accelerometro e giroscopio integrati. Il visore OSVR dispone di doppie lenti che consentono di ridurre la distorsione delle immagini.

A differenza di altri visori, questo dispone di una cintura elastica che porta i cavi fino all'altezza del muscolo trapezio dell'utilizzatore, in modo da non limitarne il movimento e quindi l'esperienza di gioco. Il punto forza di questa piattaforma è la facile integrazione con dispositivi e con software aggiuntivi. Ad esempio, utilizzando una telecamera eye-tracking è possibile adoperare il software fornito dal produttore della fotocamera per calcolare la direzione dello sguardo. È inoltre possibile utilizzare il visore in praticamente il 90% dei sistemi operativi. In questo modo lo sviluppatore non ha più bisogno di scegliere anticipatamente un particolare sistema operativo per la sua applicazione, la cui realizzazione richiederà meno tempo sfruttando i plug-in dei quali l'OSVR dispone. Si permette così allo sviluppatore di concentrarsi su essa piuttosto che sull'interfacciamento. Sono inoltre reperibili all'interno di Github, tutti i plug-in di integrazione con i vari motori grafici quali Unity e Unreal Engine.



Figura 3.4: I componenti del visore OSVR HDK1

Integrazione

L'integrazione dell'OSVR verrà trattata nella sezione *Analisi della Parte Software* in cui si descriverà il procedimento per installare, configurare ed utilizzare il visore.

3.2 Analisi della Parte Software

La parte software è stata scritta interamente nel linguaggio C# e si avvale del motore grafico Unity. L'applicazione crea un'ambientazione virtuale in cui viene posizionata una bicicletta a cui viene collegato uno script che legge continuamente la porta COM su cui il microcontrollore invia la stringa contenente l'informazione di movimento. Lo script parsifica la stringa e dà i valori in pasto al motore grafico che muove la bicicletta virtuale.

3.2.1 Blender

Blender è un software libero e multiplatforma di modellazione, animazione, compositing e rendering di immagini tridimensionali. Inoltre dispone di funzionalità per map-pature UV e simulazioni di rivestimenti adatte alla creazione di applicazioni e giochi 3D. All'interno di Blender tutte le funzioni e le interfacce possono essere richiamate con scorciatoie e per questo motivo quasi tutti i tasti sono collegati a numerosi comandi. La sua interfaccia si basa essenzialmente sulla finestra principale di lavoro, in cui è possibile visualizzare un modello 3D e modificarlo.



L'*Object Mode* è la modalità che permette di visualizzare le mesh² dell'oggetto, mentre l'*Edit Mode* permette di modificarlo. Vi sono altre modalità che non sono state utilizzate, ma risultano utili per una modellazione più dettagliata. Per dotare il modello di texture o di materiali, ovvero l'applicazione di colori, trasparenza, lucentezza o altro, Blender implementa diverse finestre come la *UV/Image Editor* nella quale è possibile scegliere la texture da applicare all'oggetto. Blender è stato utilizzato per creare la

²Una mesh poligonale, anche detta maglia poligonale, è una collezione di vertici, spigoli e facce che definiscono la forma di un oggetto poliedrico nella computer grafica 3D e nella modellazione solida.

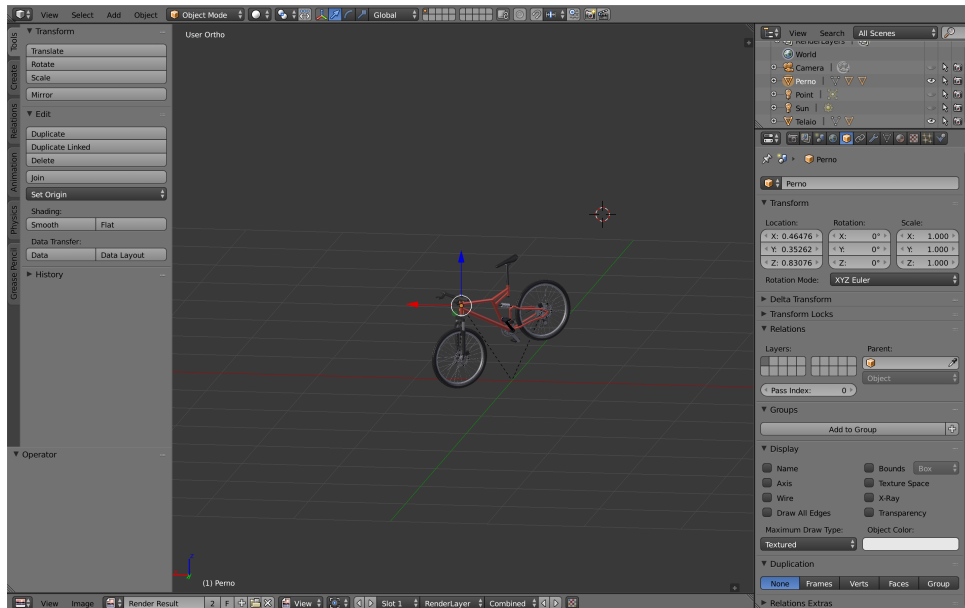


Figura 3.5: Interfaccia Principale di Blender

mesh della bicicletta virtuale da manovrare attraverso script collegati ad un motore grafico.

Mesh della bicicletta

La mesh proviene dal sito TurboSquid. È stata scaricata gratuitamente con estensione FBX³ ed è stata adattata per l'utilizzo necessario ai fini del progetto. È stata ridotta in scala ed è stata centrata alle coordinate (0,0,0). La mesh scaricata era già suddivisa in tutti i componenti di una bicicletta classica. Le uniche parti che devono essere dissociate ai fini della tesi sono il manubrio e il telaio. Sono state quindi assemblate tutte le mesh che compongono il telaio e tutte quelle che compongono il manubrio. Blender permette di associare un centro ad ogni mesh con la funzione **Set Origin > Origin to Geometry**. Questa permette di fornire, a tempo di esecuzione, una rotazione attorno al centro così definito. Sono state quindi lasciate libere le ruote e sono state centrate nel loro origine, in modo da

³FBX è un formato di file di Autodesk che viene supportato dai più comuni software di grafica 3D in quanto è in grado di immagazzinare non solo geometrie, ma anche dati di texture e di animazioni.



poter dare una rotazione attorno al proprio asse. La mesh della bicicletta è stata quindi suddivisa come segue:

- **Telaio:** comprende tutto il telaio della bicicletta escluso il manubrio e le ruote.
 - **Ruota posteriore:** è stata separata per darle possibilità di ruotare sul suo asse.
- **Perno:** consiste nell'unione tra ruota e manubrio. Questo oggetto permette di ruotare tutto il manubrio mantenendo però un asse di rotazione inclinato. Comprende:
 - **Manubrio:** contiene il manubrio stesso, la forcella e l'asse di sterzo.
 - **Ruota anteriore:** è stata separata per darle la possibilità di ruotare sul suo asse.

Il telaio è stato separato dal manubrio solo nell'asse di sterzo, ma è stato mantenuto il canotto di sterzo, in modo da poter ruotare il manubrio attorno all'asse passante per il centro del canotto. La mesh così ottenuta è stata esportata ed è stata importata

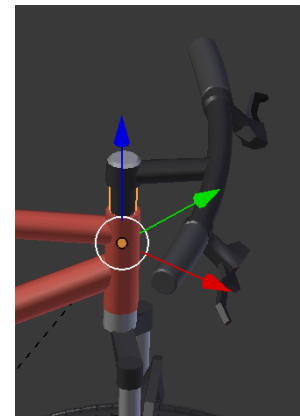


Figura 3.6: L'asse del canotto della bicicletta

in Unity come oggetto da manovrare tramite script.

Mesh dell'ambientazione

L'ambientazione è stata fornita dal Cineca: si tratta di un modello 3D dei portici di San Luca (Bologna) ottenuto tramite rilevazione laser. Il modello riporta fedelmente tutti i dettagli spaziali dei portici, ma non riportano la texture. Per il progetto che è stato creato, l'ambientazione non ha alcuna rilevanza: è possibile asportare la bicicletta e posizionarla in un qualsiasi altro mondo virtuale, purché sia adatto al movimento di una bicicletta al suo interno. L'ambientazione può essere munita di luce principale sia su blender che sul motore grafico. L'importante è non sovrapporre troppo luci per non rischiare la sovraesposizione. In figura 3.7 possiamo notare il modello dei portici in un'ambientazione vuota.

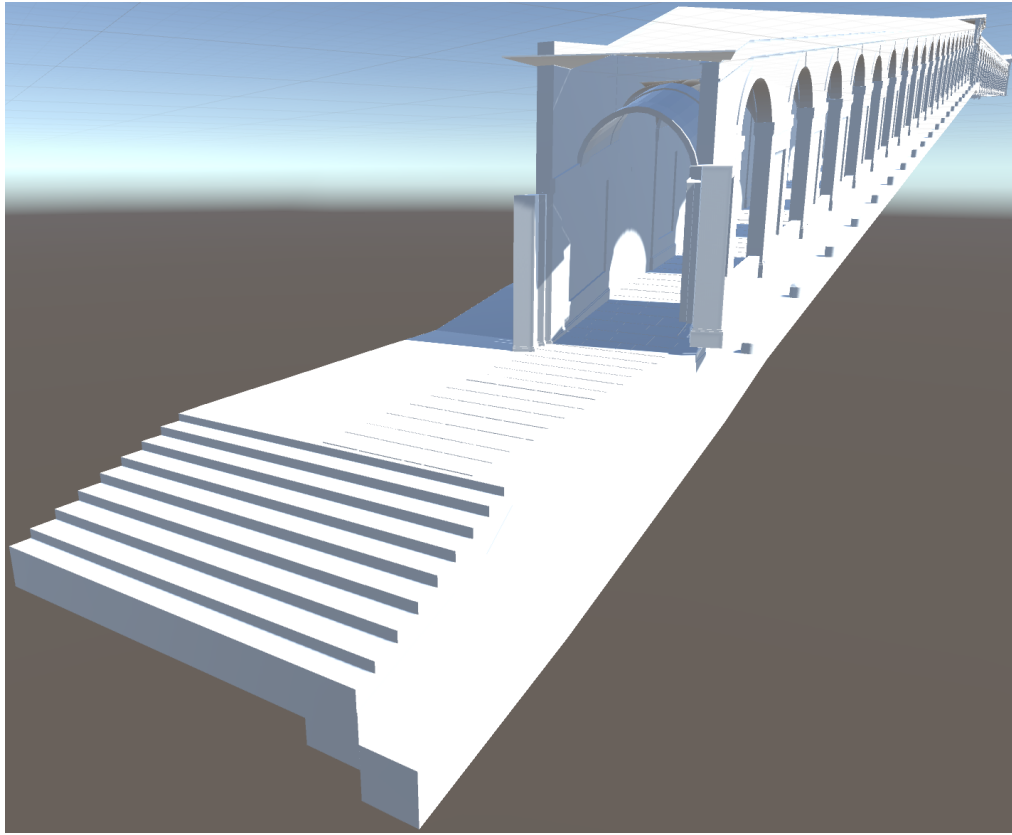


Figura 3.7: Il modello dei portici di San Luca

3.2.2 Unity

Unity è un motore grafico integrato multiplatforma per la creazione di videogiochi o altri contenuti interattivi 3D, quali visualizzazioni architettoniche o animazioni 3D in tempo reale. Unity permette di modellare l'ambientazione 3D e il modello della bicicletta virtuale attraverso script. Permette inoltre di integrare il visore per la Realtà Virtuale ed è per questo che la scelta è ricaduta su questo motore grafico: per via della facile integrazione dei visori quali OSVR e Oculus Rift.



Schermata Principale

La schermata principale di figura La schermata principale di Unity si compone di 5 riquadri suddivisi in:

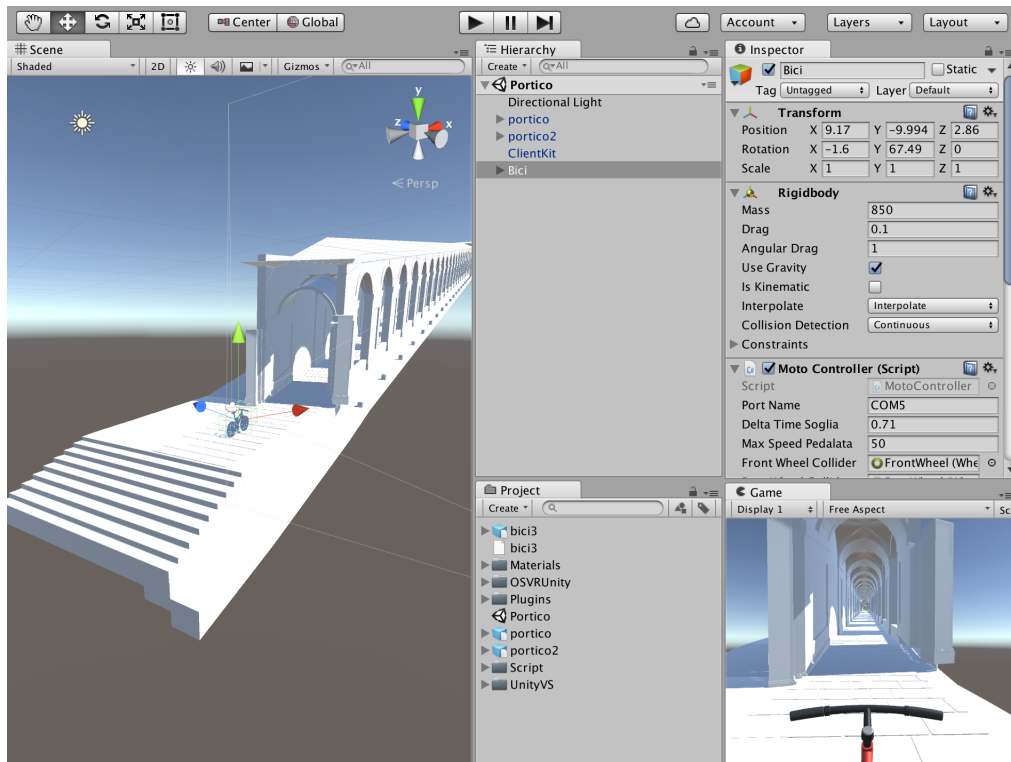


Figura 3.8: La schermata principale di Unity

- **Scene:** dove viene mostrata l'architettura dell'ambientazione e degli oggetti. All'interno di quest'area si ha la possibilità di modificare gli oggetti, scalarli, muoverli e/o ruotarli. È inoltre possibile cambiare visualizzazione e vista utilizzando mouse e tastiera.
- **Game:** in questo riquadro è possibile osservare l'inquadratura della telecamera o del visore quando la simulazione è avviata. È inoltre possibile interagire con tastiera per muovere gli oggetti o spostare la visuale.
- **Inspector:** questo pannello permette allo sviluppatore di modificare tutti i valori relativo ad un oggetto selezionato. Permette inoltre di assegnare script che modellino a tempo di esecuzione l'oggetto, come ad esempio modificandone la posizione e/o creando animazioni per muoverlo.
- **Hierarchy:** in questo punto sono elencati tutti gli oggetti presenti nella scena in ordine di parentela o di directory. Ogni componente può averne al suo interno altre (figlie) e può essere usata per trasferire

delle caratteristiche quali la posizione, la rotazione e/o vincolare quest'ultime ai comportamenti del padre. Grazie al pulsante Create si possono aggiungere nuovi elementi alla scena. Ogni elemento aggiunto può essere messo in gerarchia semplicemente trascinandolo su un altro con il mouse, questi possono essere oggetti 2D e 3D, telecamere, luci, ecc.

- **Project:** in questa sezione è possibile rintracciare tutte le cartelle e i file del progetto. Questa sezione può interagire con Hierarchy: è infatti possibile trascinare oggetti prefabbricati (come ad esempio la bicicletta creata in Blender) e posizionarli nella scena.

Ogni oggetto che fa parte del progetto è elencato all'interno del riquadro Project ed ogni oggetto che fa parte della scena si può trovare all'interno di Hierarchy. Gli oggetti all'interno di Hierarchy vengono chiamati GameObject e ognuno di questi può essere salvato come componente Prefab, con tutti i suoi figli, e riutilizzato in tutte le scene come oggetto identicamente uguale al GameObject che lo ha originato. Questa funzione risulta utile nel caso servano istanze multiple dello stesso GameObject all'interno di una o più scene. Quando viene modificato il GameObject originario, le variazioni si propagano a tutte le sue istanze nel programma. Selezionando un GameObject da Hierarchy o anche dalla sezione Project la finestra Inspector si adatta alle caratteristiche in esso contenute. Ogni GameObject possiede al suo interno dei Component che ne definiscono caratteristiche e comportamento.

3.2.3 Motocontroller script

spieghi ogni singolo comportamento dello script

3.2.4 ThreadDTWSlideWindow script

spieghi ogni singolo comportamento dello script

Capitolo 4

Realizzazioni sperimentali e valutazione

4.0.1 OSVR e le impostazioni iniziali

Nei primi test eseguiti con l'OSVR appena installato, si è riscontrato un bug che, all'avviamento della simulazione, posizionava la telecamera ruotata di 90° verso destra e un'inversione di rotazione tra roll (su Z) e pitch (su Y). La chiarificazione di questi movimenti si ha in figura 4.1.

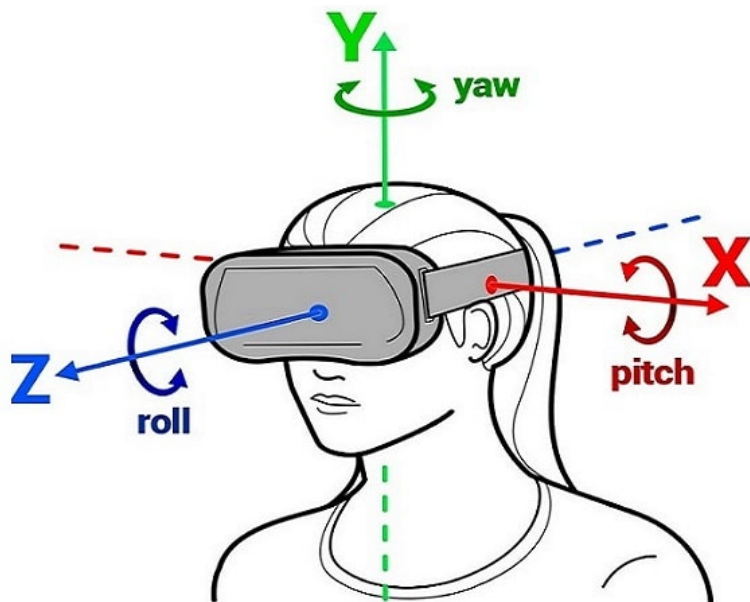


Figura 4.1: Le varie rotazioni della testa

Il problema è che l'OSVR necessita di una calibrazione iniziale che non era stata eseguita dopo l'installazione. Si è risolto il problema posizionando la bicicletta a $(0,0,0)$ con l'oggetto *VRDisplayTracked* interno alla bicicletta con posizione $(0,0,0)$ e tutte le rotazioni a 0. In questo modo le rotazioni a tempo di esecuzione erano tutte corrette.

Capitolo 5

Direzioni future di ricerca e conclusioni

Bibliografia

- [1] **TurboSquid**. <http://www.turbosquid.com/Search/3D-Models/free>.
- [2] **Virtux Omni**. <http://www.virtuix.com>.
- [3] Blender Foundation. **Blender**. <https://www.blender.org>.
- [4] Inc. Hilgraeve. **HyperTerminal**. <http://www.hilgraeve.com/hyperterminal-serial-port/>.
- [5] OSVR Developer Portal. **OSVR Guide**. <http://osvr.github.io/>.
- [6] Arduino Team. **Arduino**. <https://www.arduino.cc>.
- [7] OSVR Team. **OSVR site**. <http://www.osvr.org>.
- [8] Unity Technologies. **Unity3D**. <https://unity3d.com>.

Appendice A

Documentazione del progetto logico

Documentazione del progetto logico dove si documenta il progetto logico del sistema e se è il caso si mostra la progettazione in grande del SW e dell'HW. Quest'appendice mostra l'architettura logica implementativa (nella Sezione 4 c'era la descrizione, qui ci vanno gli schemi a blocchi e i diagrammi).