

ALMA MATER STUDIORUM - UNIVERSITÁ DI BOLOGNA
CAMPUS DI CESENA
SCUOLA DI SCIENZE

CORSO DI LAUREA IN SCIENZE E TECNOLOGIE
INFORMATICHE

**REALIZZAZIONE DI UNA VISITA INTERATTIVA
DEL MUSEO CIVICO DI GALEATA (FC) TRAMITE
MAYA E UNITY3D**

Relazione finale in
Metodi numerici per la grafica

Relatore
Damiana Lazzaro

Presentata da
Giovanni Frassinetti

III Sessione
Anno Accademico 2012-2013

Indice

Introduzione	5
Capitolo 1 Storia della Computer Graphics	9
1.1 Gli inizi	9
1.2 La ricerca di nuove tecniche	12
1.3 I videogame e la diffusione su larga scala	13
1.4 L'esplosione del 3D	15
1.5 L'inserimento della componente narrativa.....	17
1.6 L'affermazione della Computer Graphics	19
1.7 Computer Graphics e storia: l'ultima frontiera	21
Capitolo 2 Descrizione delle tecnologie utilizzate	25
2.1 Descrizione del software Autodesk Maya 2014	25
2.1.1 L'interfaccia	25
2.1.2 Modellazione	27
2.1.3 Animazione	31
2.1.4 Rendering	33
2.2 Descrizione del software Unity3D	34
2.2.1 L'interfaccia	35
2.2.2 Gli Assets	36
2.2.3 I Game Object	38
2.2.4 Gli Scripts	39
Capitolo 3 Realizzazione del progetto	41
3.1 Il Museo Civico Mambrini di Galeata (FC)	41
3.2 Creazione del modello 3D	47
3.2.1 Struttura portante: Mura, pavimenti, tetto	47
3.2.2 Dettagli	49
3.2.3 Il chiostro	52
3.2.4 Scale	53
3.2.5 I reperti	54
3.2.6 Applicazione dei materiali e delle texture	58

3.3 L'interattività	61
3.3.1 Importazione del modello 3D	61
3.3.2 Importazione degli Assets	62
3.3.3 Illuminazione	64
3.3.4 Sorgenti audio	65
3.3.5 Scripts	66
Capitolo 4 La visita	73
Conclusioni	77
Sitografia e bibliografia	79
Ringraziamenti	81

Introduzione

Questa tesi è stata realizzata con l'obiettivo di fornire un metodo alternativo e innovativo per la divulgazione e l'apprendimento della Storia, mediante la realizzazione di una visita virtuale al Museo Civico Mambrini di Galeata (FC), ottenuta facendo uso della Computer Grafica.

La scelta del Museo Civico Mambrini di Galeata è stata dettata dalla possibilità di valorizzare il patrimonio storico-culturale di una vallata, quella del fiume Bidente, che dispone di una grande quantità di siti archeologici e edifici dalla grande importanza storica.

In un periodo attuale in cui lo studio della Storia è, purtroppo, spesso subordinato a quello di altre materie, un'integrazione con uno strumento così diffuso (soprattutto tra le nuove generazioni) come la grafica computerizzata può contribuire a migliorarne la diffusione e lo studio.

Nel corso degli anni la Computer Grafica è stata utilizzata come supporto in svariati ambiti come quello scientifico o cinematografico diventandone una componente fondamentale.

Questo progetto nasce proprio con l'intenzione di offrire degli spunti per una possibile integrazione tra Computer Grafica e Storia allo scopo di facilitarne lo studio e la diffusione.

Il progetto realizzato è una visita virtuale interattiva in cui l'utente può muoversi liberamente all'interno del museo, visitare le varie stanze, osservare i reperti e visualizzare informazioni su di essi.

La visita è strutturata proprio come un classico videogioco FPS (First Person Shooter) in cui l'utente comanda direttamente la telecamera con i tasti W, A, S, D, che gli permettono di muoversi nelle quattro direzioni, e con il mouse che permette la rotazione dello sguardo.

Il progetto è stato realizzato in due fasi: creazione del modello 3D del museo (per la quale è stato utilizzato il software Autodesk Maya 2014) e realizzazione della parte interattiva (facendo uso del motore di gioco Unity3D).

Esse verranno di seguito analizzate in dettaglio e verranno forniti degli esempi di come son stati creati i modelli di alcuni oggetti nonché degli script utilizzati per programmare l'interattività.

La tesi è così strutturata:

Nel primo capitolo verrà tracciata la storia della Computer Grafica a partire dalla prima immagine digitale della storia, nel 1962, fino ai tempi recenti analizzando come essa si inserisce nel panorama artistico mondiale. Verranno illustrati i vari ambiti che hanno fatto uso della Computer Grafica a partire da quello scientifico fino a quello storico, oggetto della tesi.

Nel secondo capitolo verranno descritte le tecnologie utilizzate per la realizzazione della visita virtuale. Seguendo l'ordine logico di realizzazione del progetto (creazione del modello 3D-programmazione dell'interattività) verrà descritto in dettaglio il software Autodesk Maya e gli strumenti di base più utilizzati. Successivamente verrà descritto il software Unity3D ponendo particolare attenzione ai suoi 3 componenti fondamentali: Assets, Game Object e Script

Il terzo capitolo è dedicato alla realizzazione del progetto.

Verrà inizialmente fornita una descrizione del museo, della sua storia, e di come sono strutturati i reperti al suo interno.

Successivamente verrà descritto il processo di creazione del modello 3D del museo partendo dall'edificio fino ai reperti più importanti.

Esso verrà infine importato Unity3D (il software permette l'importazione di file creati con Maya) per la creazione della visita vera e propria.

Verranno descritti gli Assets utilizzati e mostrati gli Script che sono stati scritti per poter gestire l'interattività.

Nel quarto ed ultimo capitolo verranno mostrati alcuni screenshot della visita virtuale eseguita su pc.

Capitolo 1

Storia della Computer Graphics

Questo primo capitolo tratta la storia della Computer Grafica a partire dalla prima immagine digitale realizzata nel 1962 fino ai tempi recenti. Verrà posta particolare attenzione alle varie tecniche che si sono susseguite negli anni, alla loro evoluzione e al modo in cui hanno influenzato il panorama artistico mondiale. Verranno, inoltre mostrati i vari campi in cui la Computer Grafica ha trovato maggiore utilizzo a cominciare da quello scientifico passando per il campo cinematografico, fino all'ultima frontiera rappresentata dall'ambito storico.

1.1 Gli inizi

La storia della computer grafica inizia nel 1962 presso la *Boeing Company* americana quando l'azienda incominciò a ricercare e sperimentare una tecnologia che le permettesse di visualizzare simulazioni scientifiche dei calcoli effettuati presso i suoi laboratori. In quel periodo un ingegnere informatico di nome *William Fetter* riesce a intuire le potenzialità che il computer, all'ora agli inizi della sua diffusione, avrebbe avuto nella ricerca di tale tecnologia. *Fetter* riuscì infatti a creare la prima immagine digitale della storia raffigurante due piloti nella cabina di pilotaggio. (Fig. 1.1)

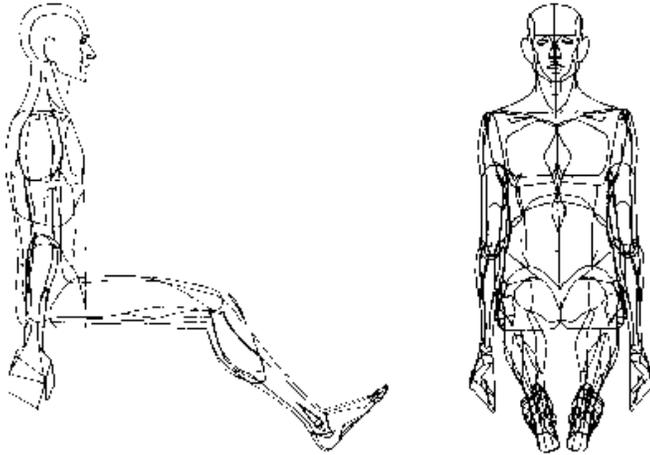


Fig. 1.1 William Fetter - Piloti in cabina di pilotaggio

Parallelamente a tale avvenimento nacque anche l'Arte Programmata la quale proponeva nuove modalità di fare arte basandosi su un concetto di democratizzazione della stessa. Essa proponeva infatti delle opere, spesso di carattere matematico, dinamiche e che potevano a loro modo interagire con gli osservatori. Nel corso degli anni Sessanta l'Arte Programmata ebbe dunque una notevole influenza su coloro che sperimentavano il computer come mezzo di rappresentazione visiva.

Nel 1963 infatti fu creata, presso i *Bell Telephone Labs*, la prima animazione generata al computer la quale simulava il movimento di un satellite intorno alla terra assieme a un orologio che ne contava le orbite (Fig. 1.2).

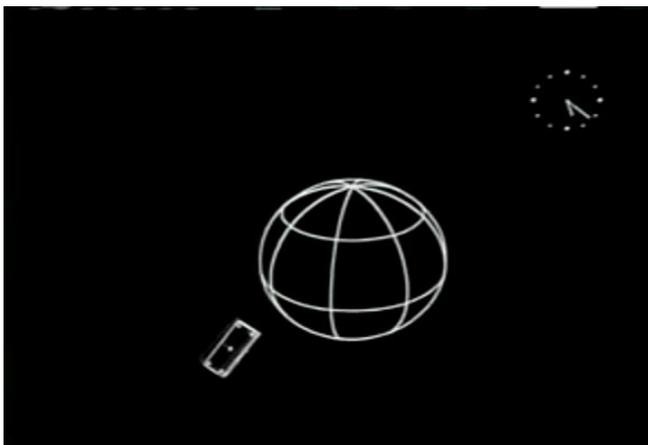


Fig. 1.2 Animazione di un satellite attorno alla terra

Grazie a questa animazione si intuirono le grandi potenzialità che la Computer Grafica poteva avere nell'ambito delle simulazioni scientifiche. Tutt'ora infatti l'ambito scientifico rappresenta uno dei più interessanti e innovativi nell'utilizzo della Computer Grafica. [1]

Il primo decennio dunque fu caratterizzato da una forte ondata di sperimentazione della Computer Grafica sia in ambito scientifico che in ambito artistico e visivo. E' infatti difficile tracciare una linea di demarcazione precisa tra la Computer grafica sviluppata a scopi scientifici o industriali e quella nata con finalità artistiche o estetiche. Un esempio delle grandi innovazioni introdotte in questo periodo è lo sviluppo di nuove tecniche di animazione quali l'interpolazione, la trasformazione di un segno in un altro o di un'intera immagine in un'altra. Risale infatti al 1968 il primo Morphing della storia, creato da *Charles Csuri*, il quale mostra la trasformazione di un disegno caotico in un colibrì. (Fig. 1.3). [2]



Fig. 1.3 Primo morphing della storia

1.2 La ricerca di nuove tecniche

Successivamente alla realizzazione delle prime immagini e animazioni al computer ci fu una continua ricerca di nuove tecniche e nuovi approcci alla Computer Grafica. Inizialmente infatti furono realizzati alcuni microfilm utilizzando il linguaggio di programmazione Fortran che successivamente fu sostituito dal *Beflix*, il primo linguaggio di programmazione specializzato per le animazioni al computer. Esso veniva utilizzato per generare immagini attraverso una serie di punti a diversi livelli di grigio calcolando dei numeri da 0 a 7 che ne indicavano il livello di luminosità.

Altri approcci alla Computer Grafica si basavano sullo studio della percezione, furono realizzati infatti dei film basati su dei 'rumori' generati dalla funzione random del calcolatore, oppure sul movimento di forme astratte.

L'approccio dominante era dunque quello astratto e le opere realizzate di quel periodo erano caratterizzate da forme, suoni e colori espressi armonicamente tra di loro. Un esempio di quest'approccio è un'opera realizzata nel 1975 da *John Whitney Sr* chiamata *Arabesque* (Fig 1.4), in cui sono ben visibili combinazioni di forme, musica e colori. A quegli anni risale inoltre il primo studio sull'illusione della tridimensionalità.



Fig. 1.4 John Whitney Sr - Arabesque (1975)

Altri procedimenti di animazione al computer introducono invece l'astrazione all'interno delle stesse immagini figurative. *Sunstone* (1979) di *Ed Emshwiller* per esempio è un film costituito da continue variazioni di colore ed è stato realizzato utilizzando un programma di animazione dinamica in cui il regista può cambiare manualmente dei parametri, in questo caso il cambio dei colori e la simulazione di strutture simili a tessiture. Con opere come questa si entra pertanto nel campo più concettuale dell'animazione al computer, dove la tecnica utilizzata influenza gli artisti anche nella messa a punto del soggetto. [3]

1.3 I videogame e la diffusione su larga scala

È solo a partire dagli anni Settanta che, in contrapposizione agli approcci visti finora, comincia a svilupparsi un approccio tendente al realismo piuttosto che all'astrazione. La nascita nel 1976 di due grandi aziende come *Microsoft* e *Apple* contribuisce, infatti, a rendere l'approccio realistico quello predominante nella realizzazione di opere in Computer Grafica. Questo cambiamento è determinato non solo dalle esigenze del contesto artistico dell'epoca ma soprattutto dalla fabbricazione di nuovi sistemi che permettono la modellazione e l'animazione di immagini tridimensionali sullo schermo. Le funzioni di rendering e di animazione per esempio diventano degli standard inseriti in tutti i computer segnando così l'inizio di una diffusione più vasta delle tecniche di modellazione e animazione al computer. Nella prima metà degli anni Ottanta si registra un notevole allargamento del bacino di utenza e la ricerca esce dai centri deputati alla sperimentazione, i grandi laboratori pubblici e privati che soprattutto negli USA sono particolarmente attivi, almeno dal punto di vista dello sviluppo tecnologico. Il linguaggio delle immagini di sintesi incontra il mercato e si democratizza. Artefice di questo progressivo avvicinamento al pubblico è da una parte la *Apple*, che nel 1984 lancia sul mercato il *Macintosh*.

L'anno dopo inoltre la *Commodore* mette sul mercato l'*Amiga*, un personal computer molto flessibile ad alta qualità video. L'*Amiga* e il *Macintosh* diventano un fenomeno molto diffuso tra i giovani, che in questo mezzo trovano uno strumento amichevole e di semplice utilizzo se paragonati alla complessità delle prime workstation.

Contribuisce a questa espansione anche la neonata industria videoludica che, grazie ad aziende come *Namco* e *Nintendo*, 1981 vede la nascita di due personaggi, *Pac-Man*(Fig. 1.5) e *Super Mario*(Fig. 1.6), che diventano subito dei classici del genere.

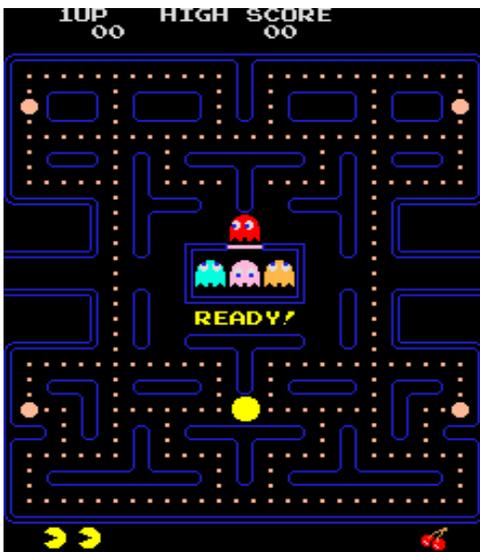


Fig. 1.5 Namco - PacMan

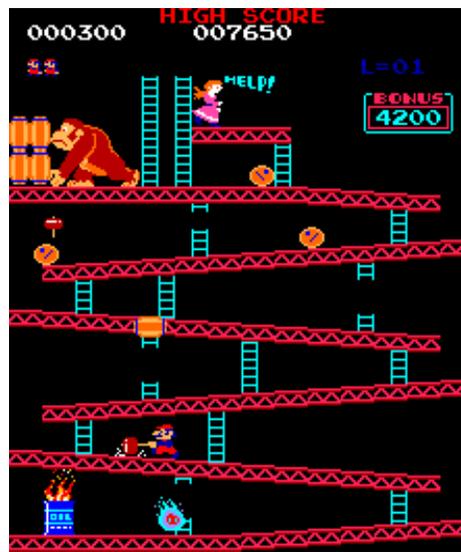


Fig. 1.6 Nintendo - Donkey Kong

I primi videogiochi sono ovviamente in 2D, ma si rivelano un ottimo laboratorio di sperimentazione tecnica e di diffusione del gusto e ciò porta da un lato alla crescita del mercato dei personal computer e dall'altro apre le porte alla produzione digitale interattiva.

Nel frattempo vengono inoltre condotte ricerche sulle animazioni in 3D e si cerca di produrre algoritmi specifici per poter gestire al meglio questa tecnica.

In questi anni nasce anche il *Siggraph*, una fiera mondiale dedicata unicamente al settore della Computer Grafica, che contribuisce allo sviluppo e alla divulgazione di queste tecniche. [4]

1.4 L'esplosione del 3D

Inizialmente in *Siggraph*, nato nel 1974, era seguito da un pubblico ristretto costituito da soli appassionati del settore ma successivamente subì una forte crescita quando, negli anni Ottanta, furono create le prime immagini rappresentanti oggetti in 3 dimensioni (inizialmente forme geometriche).

Nel frattempo *Myron Krueger* conduce delle prime ricerche sulla Realtà Artificiale e, con la sua opera *Videoplace*, riuscì a produrre un primo esempio di mondo virtuale interattivo.

Man mano che il tempo passa, le varie ricerche e sperimentazioni producono risultati che sembravano irraggiungibili come per esempio la rappresentazione della natura, del mondo animale e dell'uomo.

I risultati di tali ricerche portarono, nel 1984, alla creazione da parte della *Pixar* (all'ora divisione della *LucasFilm*) del primo cortometraggio d'animazione in 3D della storia: *Andrè and Wally b* (Fig. 1.7).



Fig. 1.7 Pixar – André and Wally b.

Anche il Giappone si affaccia sulla scena virtuale grazie alla prima simulazione di un animale, che si muove all'interno di una breve fiction di fantascienza, creata nel 1984 da *Yoichiro Kawaguchi*.

Di grande importanza in quel periodo sono gli studi condotti da *Bob Abel* che permettono alla Computer Grafica di essere utilizzata anche in ambito pubblicitario. La sua opera più celebre è *Sexy Robot* (1985) raffigurante un'inquietante e sensuale robot mentre mostra prodotti alimentari.

Successivamente vengono prodotti una grande serie di spot pubblicitari aventi un forte impatto visivo caratterizzati da un ritmo incalzante di immagini e suoni e incredibili movimenti di camera.

La Computer Grafica vive dunque un periodo di grande sviluppo e espansione grazie alla pubblicità, ma soprattutto al *Siggraph* che rappresenta non solo il punto d'incontro delle massime autorità del settore ma anche un trampolino di lancio per chi vuole sperimentarne nuovi utilizzi e tecniche. E' questo il caso infatti di *John Lasseter* e *Bill Kroyer* che introducono l'utilizzo della Computer Grafica anche nel mondo dei cartoni animati. [5]

1.5 L'inserimento della componente narrativa

John Lasseter esordì al *Siggraph* nel 1986 con *Luxo Jr*, la celebre animazione raffigurante una lampadina-bambino alle prese con una palla (Fig. 1.8).



Fig. 1.8 John Lasseter – Luxo Jr.

Fin dai primi secondi di questa animazione si intuì che era già presente oltre che notevoli capacità creative e tecniche, anche una grande componente narrativa.

Con quel primo cortometraggio prodotto dalla Pixar, *John Lasseter*, dimostrò che la character animation poteva essere realizzata perfettamente anche con tecnologie mai utilizzate come il computer. Il suo lavoro segnò una svolta nella storia della grafica computerizzata e dei cartoni animati e dimostrò che i tempi erano ormai maturi per l'evoluzione di un'arte, quella appunto dei cartoon, che fino ad allora era rimasta ancorata alle tecniche classiche.

L'approccio utilizzato da *Bill Kroyer* invece è di tutt'altro genere, infatti si basa sul tentativo di nascondere l'utilizzo del digitale nei cartoni animati ricorrendo a una buona integrazione tra tecniche tradizionali e nuove.

Un esempio di questo approccio è il cortometraggio *Technological Threat* del 1988 (Fig. 1.9) il quale, dietro un'animazione in 2D apparentemente tradizionale, nasconde appunto tecniche innovative di gestione dei movimenti dei personaggi e della camera.

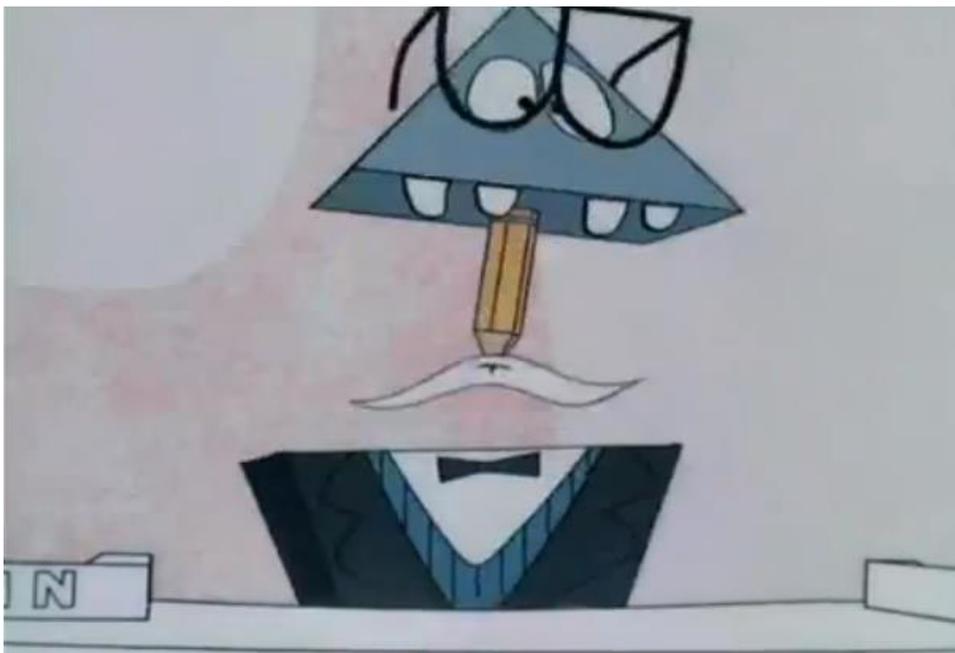


Fig. 1.9 Bill Kroyer – *Technological Threat*

Altre due opere molto importanti di quel periodo furono *Vol Libre* (1980) di *Loren Carpenter*, un viaggio virtuale all'interno di montagne generate con i calcoli frattali, e *VPL Research* (1985) di *Jaron Lanier* (primo esempio di Realtà Virtuale) entrambe primi esempi di animazione in tempo reale. [6]

1.6 L'affermazione della Computer Graphics

Grazie soprattutto alle nuove tecnologie immesse sul mercato, la seconda metà degli anni Ottanta vede l'esplosione della Computer Grafica nel mondo del cinema. Se negli anni passati si era cercato di simulare giochi di luce, figure geometriche o il movimento di personaggi in stile cartoon, ora si ricerca un sempre maggior realismo sia dei personaggi, sia dei loro movimenti.

Un obiettivo delle ricerche condotte in quel periodo è stato quello di creare personaggi umani e di rendere il più realistici possibile non solo il loro aspetto ma anche i movimenti della pelle, dei muscoli e perfino delle espressioni del viso.

Un esempio di questo approccio è il cartoon *Tony de Peltrie*, creato da *Daniel Langlois*, raffigurante una figura umana che parla e suona il pianoforte.

(Fig. 1.10)



Fig. 1.10 Daniel Langlois – Tony de Peltrie

Nel frattempo la *Disney* in accordo con la *Pixar* crea il suo primo dipartimento di Computer Grafica e nel 1995 esce *Toy Story* (Fig. 1.11), il primo film realizzato interamente in grafica computerizzata. Il film, che vincerà nel 1996 l'Oscar per i migliori effetti speciali, segna l'entrata in scena della neonata Disney-Pixar nel campo della realizzazione di film in Computer Grafica.



Fig. 1.11 Pixar – Toy Story

Anche la televisione contribuisce alla diffusione della grafica computerizzata tridimensionale, non solo venivano realizzati spot pubblicitari (come accennato in precedenza) ma anche le sigle di molti programmi tv erano realizzate con le stesse tecniche di animazione usate per i film.

Sicuramente però, come già accennato, il maggior mezzo di diffusione, ma anche di utilizzo, della Computer Grafica è l'industria cinematografica che ormai non la utilizza solo come un costoso effetto speciale aggiuntivo ma come parte integrante della storia. [7]

1.7 Computer grafica e storia: l'ultima frontiera

La Computer Grafica ha dunque attraversato varie fasi, da un utilizzo rivolto unicamente a sperimentazioni scientifiche a quello a scopo artistico, dall'integrazione con trame narrative fino alla definitiva consacrazione grazie all'industria cinematografica e videoludica. Tale espansione è dovuta, oltre che a costanti ricerche di tecniche e metodi di espressione, anche al costante incremento della potenza di calcolo dei calcolatori ha permesso il miglioramento della qualità delle esperienze visive create dalla Computer Grafica fino a raggiungere livelli attualmente sempre più vicini alla realtà.

Ciò ha portato ovviamente a un utilizzo massiccio della CG permettendo la sua introduzione in altri settori come per esempio quello pubblicitario, industriale, storico o archeologico. Una delle ultime frontiere della Computer Grafica infatti è quella riguardante lo sviluppo, la divulgazione e la valorizzazione del patrimonio culturale, artistico, storico.

La modellazione 3D rende, infatti, possibile la ricostruzione virtuale di opere d'arte o di interi edifici (per esempio un museo o un edificio storico) in maniera dettagliata e fedele alla realtà. La Computer Grafica infatti si propone come mezzo di divulgazione e apprendimento della storia in modo alternativo, creativo e anche divertente (basti pensare ai numerosi film d'animazione o documentari a tema storico prodotti negli ultimi decenni e realizzati interamente in grafica virtuale).

Il concetto alla base dell'utilizzo della Computer Grafica in un ambito complesso e ampio come quello storico è appunto il miglioramento dei metodi di presentazione e di apprendimento della storia.

La modellazione 3D in ambito storico è stata utilizzata finora principalmente allo scopo di creare immagini raffiguranti edifici o opere d'arte, ormai decaduti, per come apparivano nell'epoca in cui furono costruiti (Fig. 1.12, Fig. 1.13).

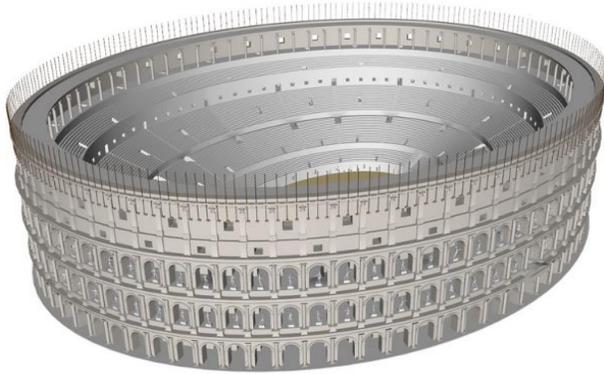


Fig. 1.12 Il colosseo

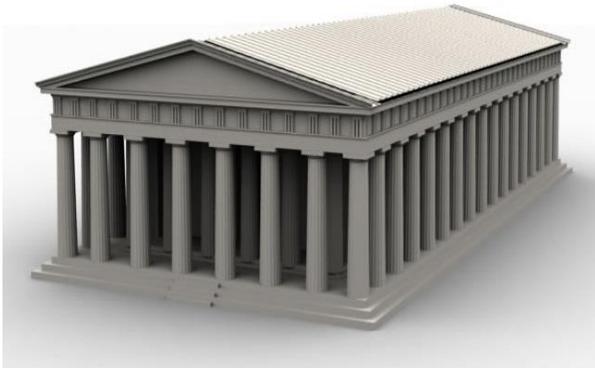


Fig. 1.13 Il Partenone

Grazie però alle moderne tecniche di modellazione grafica e ai programmi di game engine (motori di gioco) è possibile realizzare delle vere e proprie visite virtuali e permettere all'utente non solo di osservare l'ambiente di interagire esso. I programmi di Game Engine (come per esempio Unity 3D) permettono di definire, attraverso la scrittura di opportuni script, il comportamento degli oggetti presenti nella scena in relazione all'accadimento di un particolare evento come per esempio il click del mouse oppure la pressione di un tasto sulla tastiera. Un esempio di combinazione tra Computer Grafica orientata alla storia e integrazione di essa con programmi di game engine è sicuramente il famoso videogioco Assassin's Creed, il quale (tralasciando la trama e le meccaniche di gioco che non sono oggetto del progetto) presenta al suo interno ricostruzioni precise e dettagliate di intere città in varie epoche storiche (celebri sono le ricostruzioni di Firenze e Venezia nel periodo del Rinascimento Italiano. (Fig. 1.14, Fig. 1.15).



Fig. 1.14 Assassin's Creed II (Ubisoft) – Venezia



Fig. 1.15 Assassin's Creed II (Ubisoft) – Firenze

Ovviamente trattandosi di un videogame l'attenzione degli utenti non è rivolta all'apprendimento della storia, ma è possibile comunque cogliere le potenzialità che le due tecniche citate poco fa offrono, sia per quanto riguarda la parte di modellazione 3D sia per la parte di interattività.

L'interattività è un fattore molto importante nella realizzazione di visite virtuali, che infatti possono essere realizzate in due modi: statico o interattivo. Nelle visite statiche l'utente è obbligato a seguire un percorso prestabilito dalla telecamera come se stesse guardando un film. In altre parole l'utente non può interagire in nessun modo con l'ambiente. Le visite virtuali interattive invece permettono all'utente di spostarsi liberamente all'interno della scena e, tramite l'integrazione con programmi di Game Engine, di interagire con essa per esempio egli può aprire una porta o una finestra oppure, trattandosi di una visita a un museo, visualizzare informazioni relative agli oggetti esposti.

Anche se come detto in precedenza questo è solo uno degli ultimi utilizzi della Computer Grafica, esso offre grandi potenzialità soprattutto in un periodo come quello attuale in cui l'apprendimento della storia è subordinato a quello di altre materie. La Computer Grafica può contribuire alla valorizzazione della storia e può introdurre un nuovo metodo di studio basato sull'osservazione e interazione. Nel capitolo successivo verrà fornita una descrizione dei due software utilizzati per la realizzazione del progetto: Autodesk Maya per la parte di creazione del modello 3D del museo e Unity3D per la gestione della parte interattiva della visita virtuale. Verranno descritti in dettaglio gli strumenti e le componenti base dei due software ponendo particolare attenzione a quelli più utilizzati per la realizzazione del progetto

Capitolo 2

Descrizione delle tecnologie utilizzate

In questo capitolo verranno descritte in dettaglio le tecnologie utilizzate per la realizzazione del progetto: Autodesk Maya 2014 e Unity3D.

Di seguito verranno descritti il funzionamento di questi software e i dettagli riguardanti gli strumenti più utilizzati per la realizzazione delle varie parti del progetto.

2.1 Descrizione del software Autodesk Maya 2014

Autodesk Maya è un software professionale di modellazione, animazione e rendering 3D che permette, non solo, la creazione di modelli 3D, ma anche l'animazione di questi ultimi. Esso infatti è molto diffuso e utilizzato nella realizzazione di film, videogiochi (un esempio su tutti è il film Toy Story, primo film di animazione realizzato interamente con Maya) grazie anche alla gestione eccellente della Character Animation. [8]

Come già accennato Maya comprende al suo interno vari moduli che possono essere raggruppati nelle tre categorie modellazione, animazione e rendering tra di loro strettamente collegate infatti nella maggior parte dei casi il processo di costruzione di un lavoro (come può essere per esempio un film) passa proprio attraverso di esse.

Prima di procedere con la descrizione delle caratteristiche di questi moduli è opportuno descrivere l'interfaccia del programma.

2.1.1 L'interfaccia

L'interfaccia di Maya può essere divisa in tre componenti: la *view* (scena), i *tool* (strumenti) e le *action* (azioni). (Fig. 2.1)

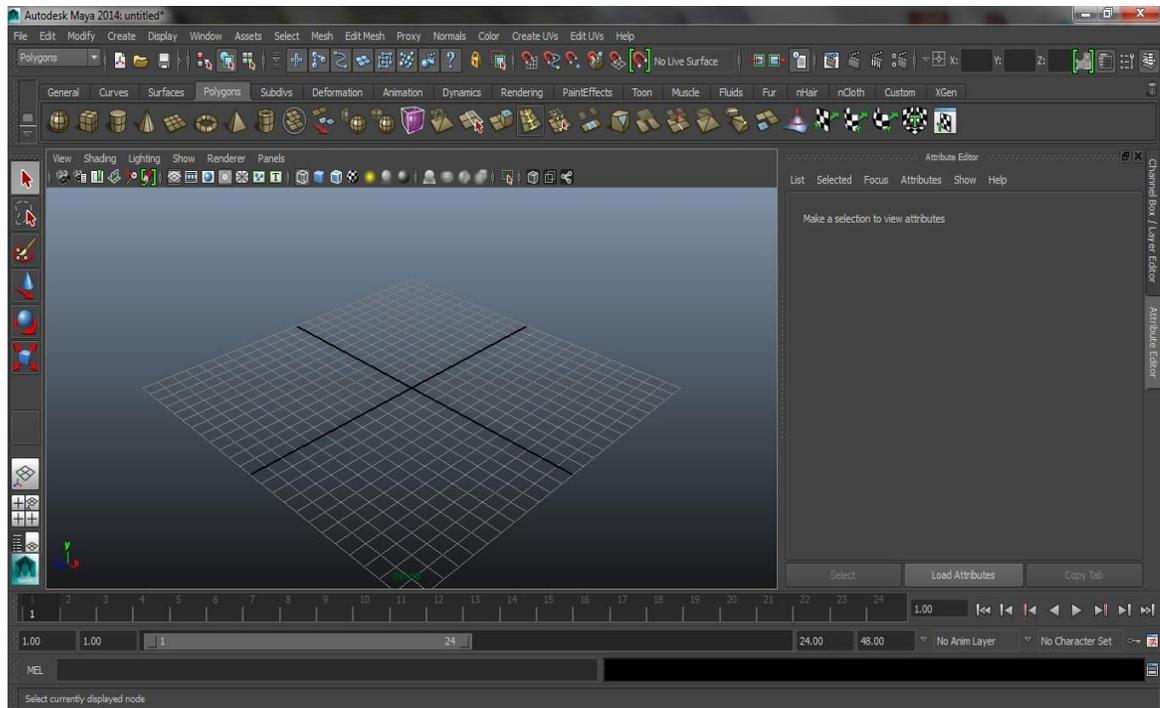


Fig. 2.1 Interfaccia di default di Autodesk Maya

La *view* è ben visibile al centro dell'interfaccia e contiene al suo interno una griglia che rappresenta il 'pavimento' del mondo virtuale con gli assi cartesiani x e z in evidenza che si incontrano al centro, detto 'Origine'. Inoltre in basso a sinistra sono ben visibili gli assi cartesiani colorati x, y e z.

E' possibile inoltre cambiare la vista della *view* in front (frontale), top (dall'alto) o side (di fianco) premendo la barra spaziatrice, o cliccando sul menù '*panels->orthographic*'.

Nella parte sinistra dell'interfaccia sono presenti i *tool* disposti per colonna, essi sono divisibili in due categorie: *tool* di selezione e *tool* di trasformazione.

I *tool* di selezione permettono di selezionare uno o più oggetti presenti nella scena mentre i *tool* di trasformazione applicano le trasformazioni di traslazione, rotazione e scalatura agli oggetti selezionati.

La selezione si può effettuare non solo sugli oggetti interi, ma anche sulle loro componenti. E' possibile infatti effettuare selezioni sui vertici, i lati o le facce.

Nella parte alta dell'interfaccia sono presenti le azioni le quali possono essere selezionate sia dai menù a tendina classici sia dalle icone di scelta rapida situate sopra la *view*.

La distinzione tra *tool* e azioni è che i *tool* lavorano in maniera continua mentre le azioni sono istantanee.

Le componenti situate in basso e a destra dell'interfaccia sono rispettivamente: i controlli dell'animazione (la linea del tempo e i controlli di playback) e i box contenenti le proprietà degli oggetti selezionati ma siccome queste due componenti sono variabili e dipendono dall'ambito in cui si sta lavorando, verranno descritte dettagliatamente in seguito.

2.1.2 Modellazione

Obiettivo della fase di modellazione è la creazione di oggetti 3D complessi, a partire da primitive matematiche messe a disposizione da Maya.

Esistono due tipi fondamentali di modellazione: La modellazione Poligonale e la modellazione NURBS.

L'elemento di base della modellazione Poligonale è appunto il Poligono il quale è a sua volta composto dalle seguenti componenti fondamentali: *vertex* (vertice), *edge* (spigolo), *face* (faccia).

Le varie tecniche che vengono utilizzate nella modellazione poligonale si occupano di approssimare più o meno verosimilmente gli oggetti 3D creati a oggetti reali più o meno complessi. Il processo di modellazione poligonale prosegue per approssimazioni successive partendo da un risultato molto grezzo e spigoloso diventando via via sempre più preciso e dettagliato. [9]

La modellazione NURBS invece è un tipo di modellazione matematica basata su delle curve particolari (NURBS) e che permette un controllo perfetto di quanto si va a creare.

Le NURBS (Non-Uniform Rational B-Spline) infatti sono superfici matematiche perfettamente smussate, non caratterizzate dalla tipica sfaccettatura delle

superfici poligonali e sono facilmente modificabili e controllabili attraverso i punti di controllo chiamati *Control Vertex*. [10]

Nella realizzazione di questo progetto è stata utilizzata la modellazione poligonale per due fattori principali: la non necessità di rappresentare oggetti perfettamente smussati e per ridurre il carico di lavoro in fase di importazione del modello in Unity 3D.

Le primitive matematiche poligonali messe a disposizione da Maya sono: *cube*, *sphere*, *cylinder*, *cone*, *plane*, *torus*, *prism*, *pyramid*, *pipe*, *helix*, *soccer ball* e *platonic solid*.

Per eseguire una corretta modellazione partendo da suddette primitive si fa uso dei vari *tool* e *action* presenti nei menù a sinistra e in alto nell'interfaccia del programma.

Visto il grande numero di queste componenti verranno di seguito descritte solamente quelle utilizzate ai fini della creazione del modello 3D del museo (la descrizione dettagliata del modello del museo verrà effettuata in seguito).

Le *action* mostrate di seguito son state applicate per semplicità su un semplice cubo allo scopo di mostrare il loro funzionamento e i loro effetti ai fini della modellazione. Esse sono: *Insert edge loop*, *Interactive split* e *Extrude*.

Per istanziare un cubo si può cliccare su '*create->poligonal primitives->cube*' oppure sull'icona raffigurante il cubo presente nel menù di scelta rapida, esso verrà posizionato di default al centro della scena. (Fig. 2.2)

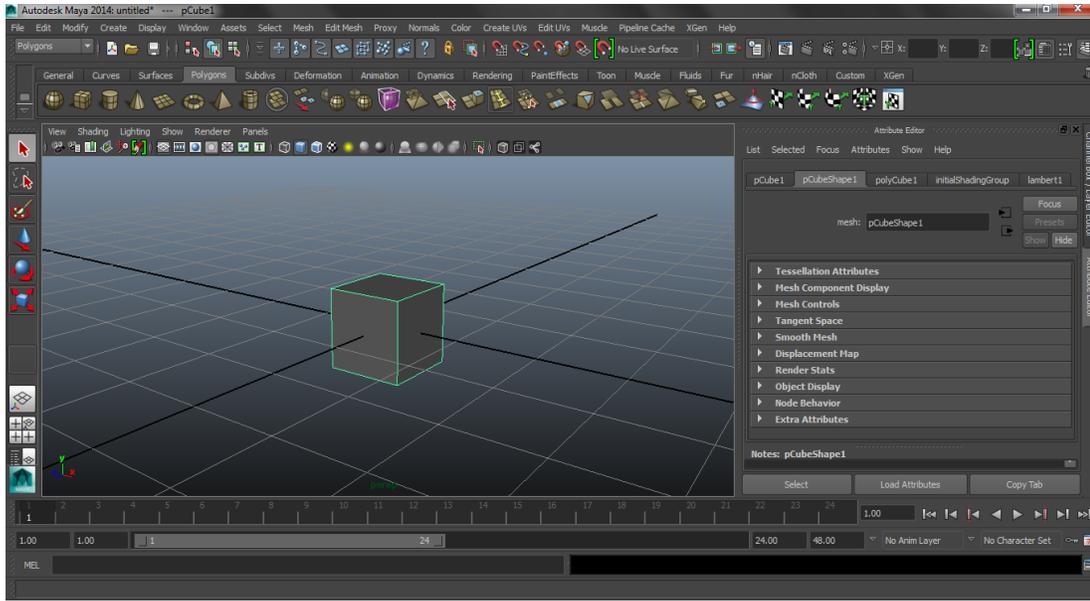


Fig. 2.2 La primitiva Cube al centro della 3d view

Di seguito verrà descritto il funzionamento delle azioni elencate in precedenza, per poterle selezionare occorre cliccare dal menù 'Edit Mesh'. (Fig. 2.3)

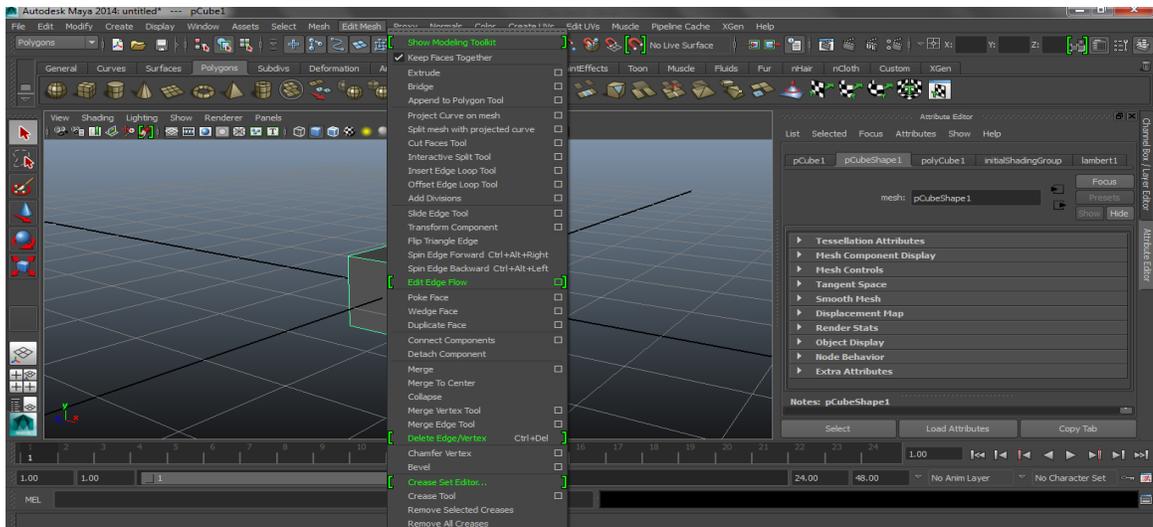


Fig. 2.3 Il menù Edit Mesh

Insert edge loop: Cliccando su un *edge* aggiunge un'altra serie di *edge* seguendo il perimetro dell'oggetto. E' molto utile qualora si voglia aumentare il numero di facce di un poligono o modificarne la forma solo in una parte. (Fig. 2.4)

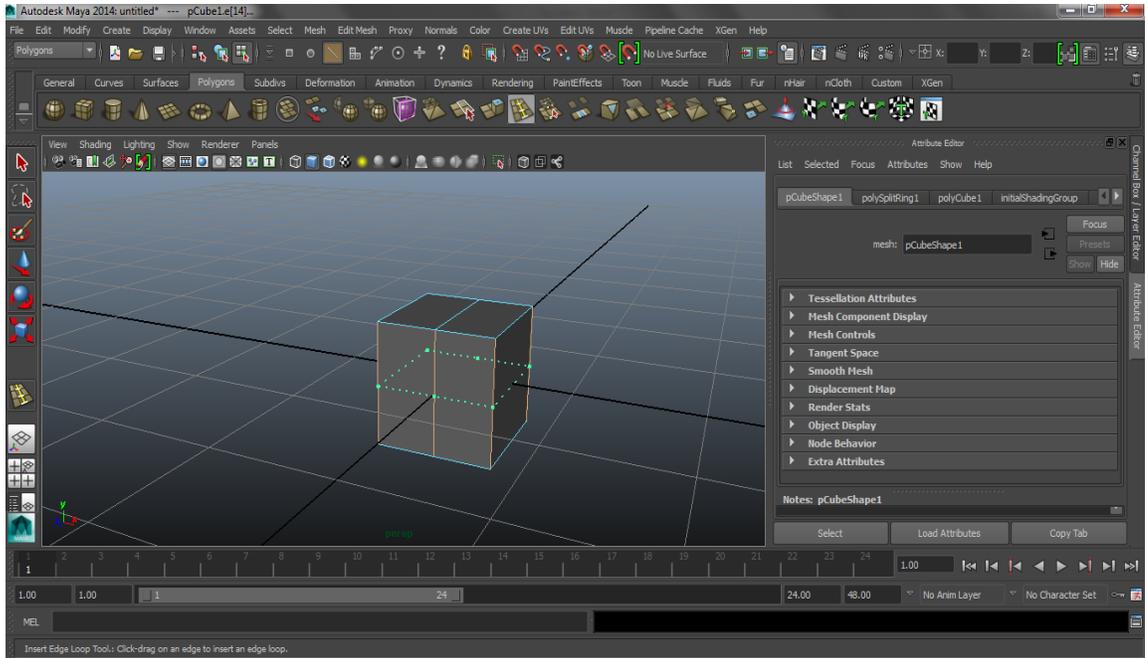


Fig. 2.4 Esempio dell'istruzione Insert Edge Loop

Interactive split: selezionando due punti appartenenti alla medesima faccia di un poligono crea un *edge* tra di essi. (Fig. 2.5)

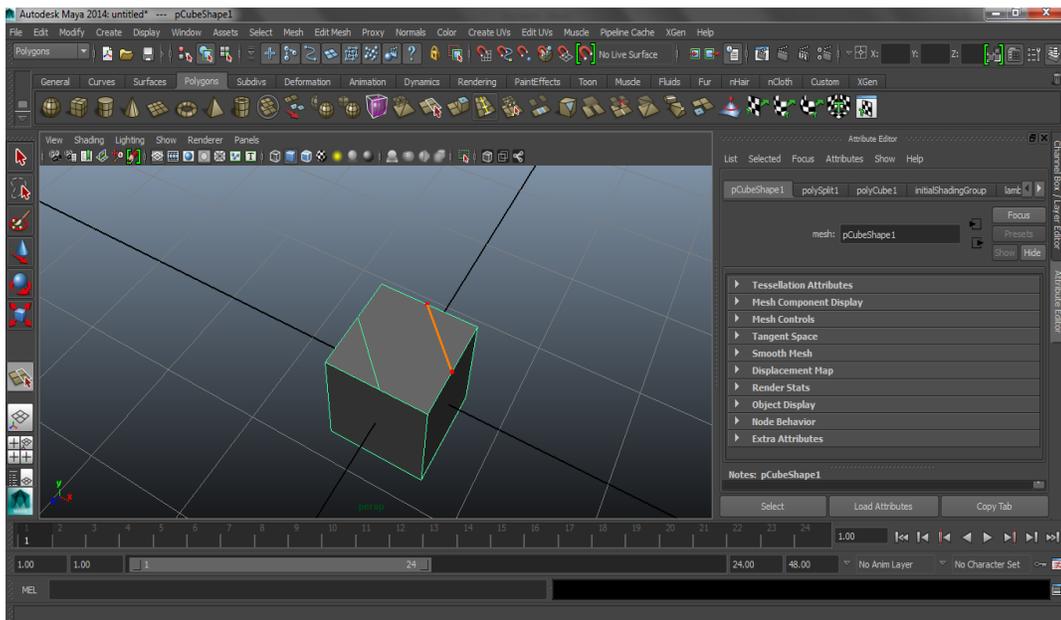


Fig. 2.5 Esempio dell'istruzione Interactive Split

Extrude: Selezionando una componente di un poligono, in questo caso una faccia, permette di estrarre la faccia lasciando invariato il resto del poligono. (Fig. 2.6)

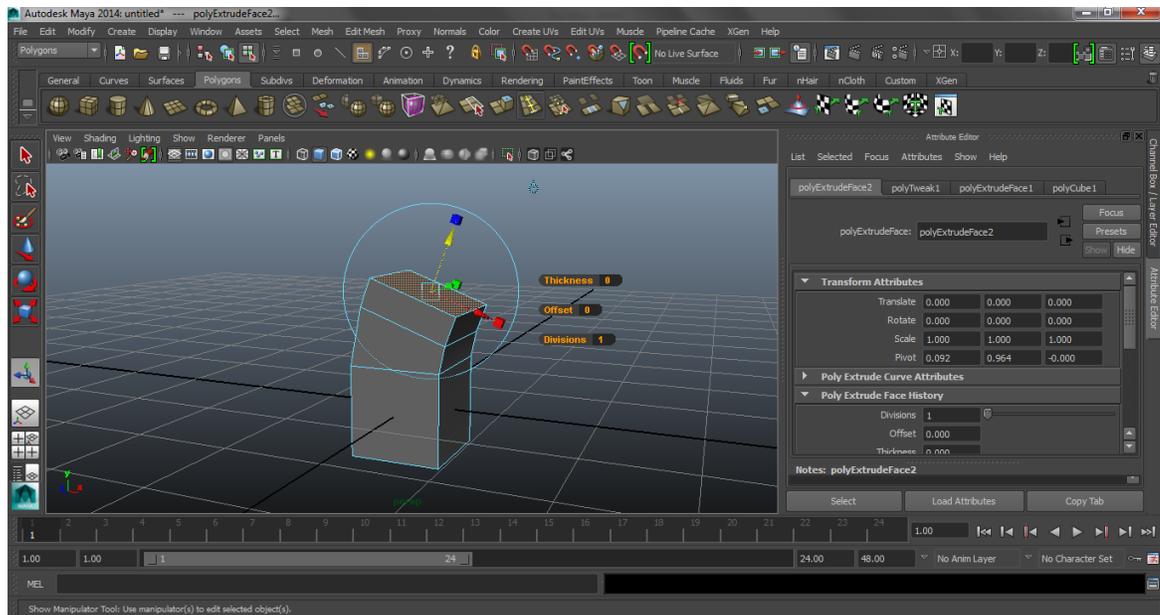


Fig. 2.6 Esempio dell'istruzione Extrude

2.1.3 Animazione

Maya è famoso anche grazie alla gestione eccellente della fase di animazione, esso infatti mette a disposizione vari strumenti che permettono di controllare in modo semplice e intuitivo i modelli 3D precedentemente creati durante la fase di modellazione.

Tra questi il più importante è il *'Joint tool'* (Fig. 2.7) che permette di creare dei veri e propri *'scheletri'* i quali una volta posizionati opportunamente in relazione all'oggetto da animare, tramite il comando *'skin->bind skin->smooth bind'*, serviranno da controlli per poter muovere l'oggetto.

In questa fase entrano in gioco i componenti presenti nella parte bassa dell'interfaccia, ovvero la linea del tempo e i pulsanti di playback.

La linea del tempo è suddivisa in frame e tramite il comando *'animate->set key'* è possibile, dopo aver selezionato uno scheletro, impostare una 'chiave' (key) definendo la sua posizione in un determinato frame. Se si effettua la stessa operazione per i frame successivi, con uno scarto di 10 frame alla volta Maya calcola per interpolazione la posizioni nei frame centrali ottenendo un'animazione continua.

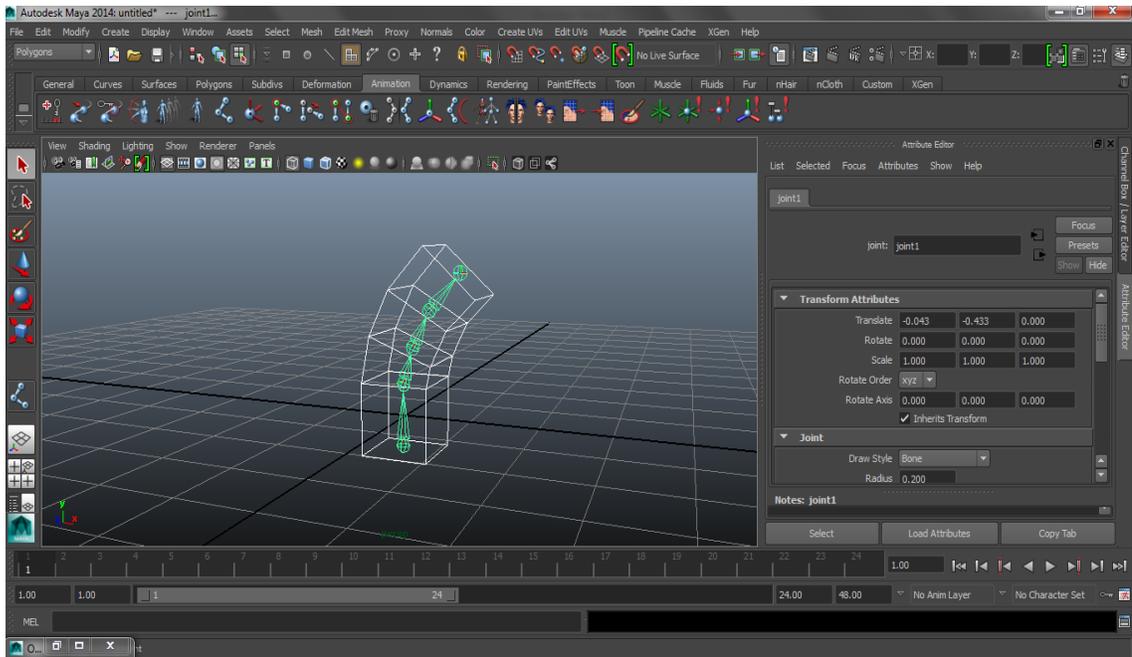


Fig. 2.7 Esempio di uno scheletro formato da quattro joint

2.1.4 Rendering

Il rendering è quel processo che si occupa della creazione di immagini 2D partendo da una descrizione matematica di una scena tridimensionale interpretata da algoritmi che definiscono il colore di ogni pixel dell'immagine digitale.

Il colore da assegnare al pixel è determinato da vari fattori tra i quali il materiale assegnato all'oggetto, la luce che lo colpisce (o l'ombra), la luce riflessa da altri oggetti etc.[11]

I principali algoritmi che si occupano di ciò sono: Radiosity e Ray Tracing.

L'algoritmo Radiosity suddivide la scena da rendere in più superfici (patch) e studia le interazioni luminose tra le patch calcolando la luce che ognuna di esse riceve dalle altre. Una parte di luce viene assorbita mentre un'altra parte viene riflessa nell'ambiente per il successivo passaggio dell'algoritmo. (Fig. 2.8)[12]

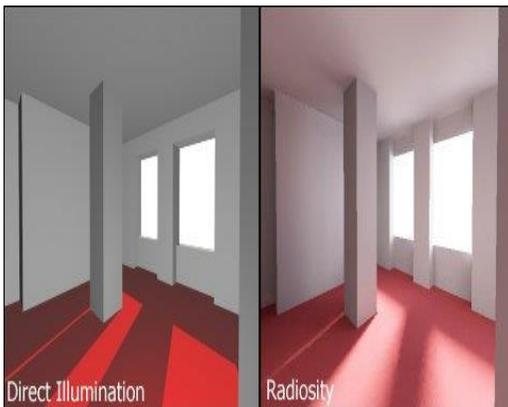


Fig. 2.8 Esempio dell'algoritmo Radiosity

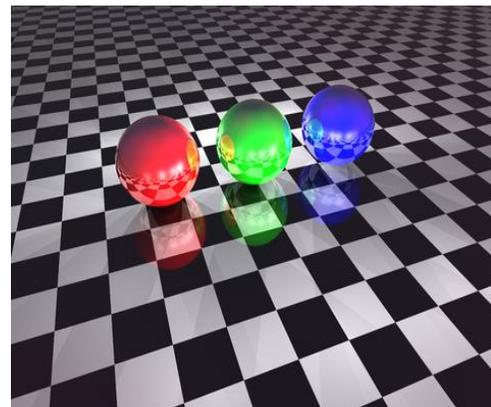


Fig. 2.9 Esempio dell'algoritmo Ray Tracing

L'algoritmo Ray Tracing invece determina il colore da assegnare a ogni pixel producendo una serie di raggi che, a seconda del punto in cui incontrano gli oggetti presenti nella scena, vengono assorbiti o riflessi interagendo con il resto dell'ambiente. (Fig. 2.9)[13]

2.2 Descrizione del software Unity 3D

Unity 3D è un motore di gioco (game engine) multipiattaforma utilizzato principalmente per la creazione di videogiochi 3D (nella nuova versione anche 2D) o altri contenuti interattivi quali visualizzazioni architettoniche o animazioni in tempo reale.

Grazie al fatto che Unity supporta un gran numero di formati di file, provenienti dalla maggior parte dei programmi di grafica quali Maya, 3DS Max, Blender, Photoshop, esso viene spesso utilizzato in integrazione con tali programmi per gestire la parte interattiva.

In modo analogo Unity produce file che possono funzionare con la maggior parte delle piattaforme di gioco attualmente presenti sul mercato (tra i quali Playstation 3, Xbox One, Wii U, Nintendo 3DS etc), con i dispositivi mobile (iOS, Android), con i PC (Windows, Linux, OS X) e con il Web (Flash). [14]

Per la gestione dell'interattività Unity utilizza degli script definiti dall'utente che associati ai vari oggetti presenti nella scena possono definirne il comportamento in base al verificarsi di vari eventi, per esempio il click del mouse o la pressione di un tasto sulla tastiera.

Inoltre Unity dispone in maniera nativa di componenti per la gestione della fisica (per esempio l'applicazione della forza di gravità) e delle collisioni (per impedire per esempio che il personaggio passi attraverso i muri).

Di seguito verranno descritti, a partire dall'interfaccia, i principali strumenti utilizzati per la gestione della parte interattiva del progetto.

2.2.1 L'interfaccia

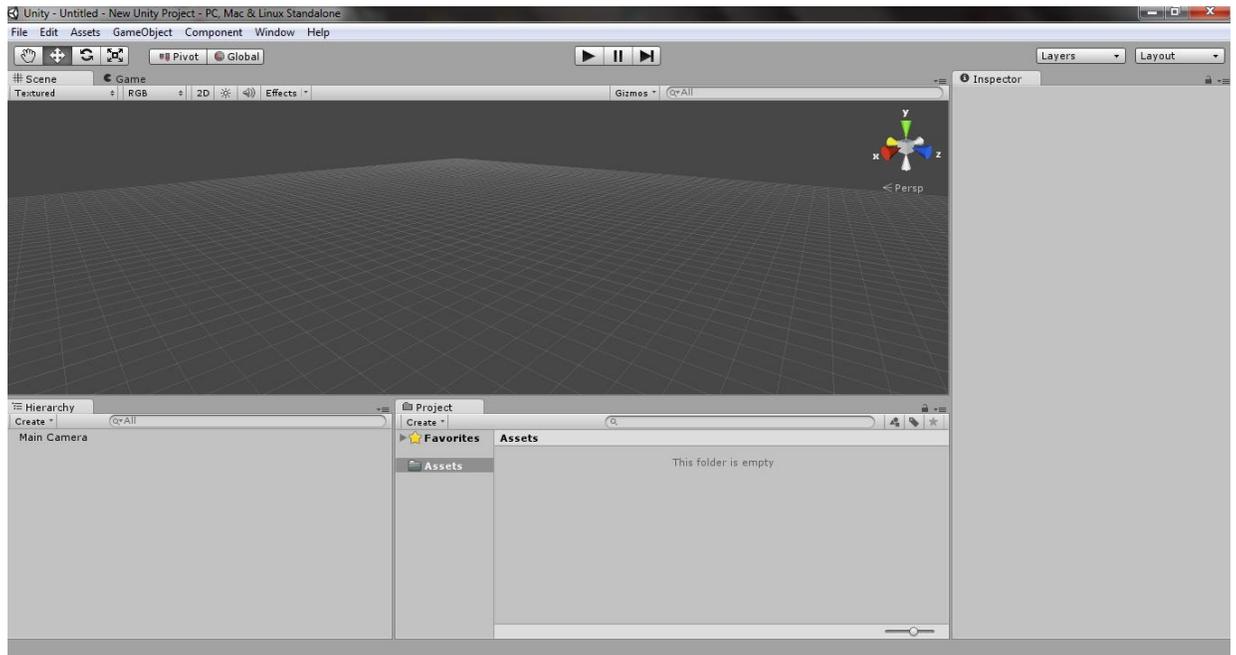


Fig. 2.10 Interfaccia di default di Unity3D

L'interfaccia di default di Unity 3D (Fig. 2.10) presenta alcune analogie con quella di Maya precedentemente descritta, infatti al centro si può notare la 3D view con indicati, in alto a destra, gli assi cartesiani di riferimento. Sopra la 3D view si trovano i *tool* fondamentali di Unity (selezione, traslazione, rotazione e scalatura) e i pulsanti che permettono l'esecuzione in tempo reale del progetto.

I due pannelli posizionati sotto la 3D view, *Hierarchy* e *Project*, contengono rispettivamente: l'elenco degli oggetti presenti nella scena e l'elenco dei file e degli 'Assets' che fanno parte del progetto inclusi modelli 3D (realizzati con Maya), script, texture e le varie componenti di gioco facenti parte di Unity.

Il pannello *Inspector*, invece, situato sulla destra rispetto alla 3D view, visualizza le proprietà e le componenti facenti parte di un oggetto selezionato. Da esso è possibile aggiungere varie componenti agli oggetti presenti nella scena, per esempio per aggiungere una componente di tipo '*Rigidbody*', si seleziona l'oggetto dalla 3D view e come si può notare compariranno sulla destra le sue proprietà. Cliccando sul tasto '*Add Component->Physics->Rigidbody*' si può

assegnare all'oggetto la componente *'Rigidbody'* che permette di renderlo soggetto alle leggi della fisica.

Di seguito verranno descritti gli strumenti di base e più utilizzati in Unity e inoltre verrà mostrato come si realizzano gli script.

2.2.2 Gli Assets

Una parte molto importante di Unity sono gli *Assets*, dei veri e propri pacchetti contenenti vari oggetti (quali scripts, modelli 3D, materiali, texture etc.) di default che servono per la gestione di una parte particolare del progetto. Nei due esempi seguenti sono stati aggiunti due *'Assets'*: *Character Controller* e *Skyboxes* i quali permettono rispettivamente la gestione del personaggio (in Prima Persona o in Terza Persona) e l'inserimento del cielo nella scena.

Per importare gli *'Assets'* all'interno del progetto basta cliccare sul menù *'Assets'* e successivamente su *'Import package->Character Controller'* per inserire le componenti per gestire il personaggio (Fig. 2.11).

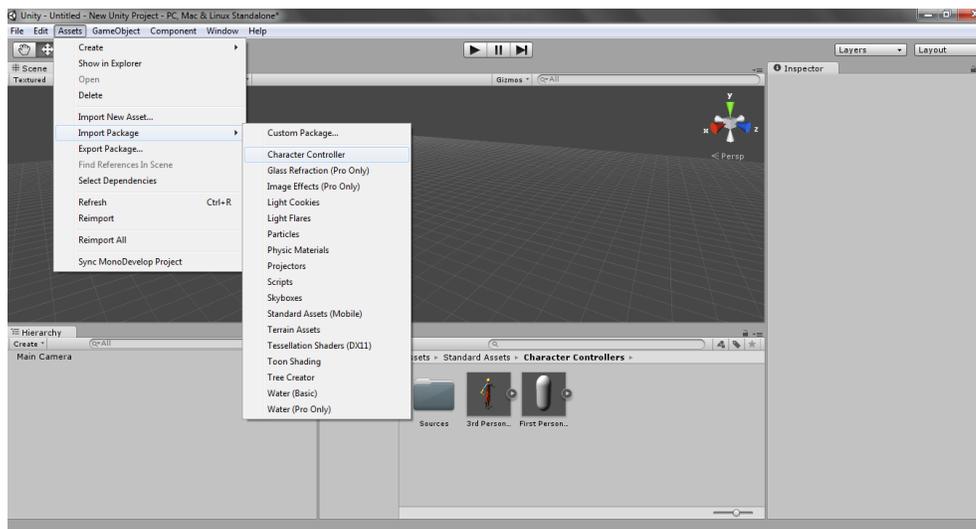


Fig. 2.11 Il menù Assets

Cliccando invece su 'Skyboxes' verranno aggiunti al pannello 'Project' tutti i materiali e le texture predefiniti relativi ai cieli presenti di default su Unity.

Per visualizzare un cielo sul gioco basta associare il relativo materiale alla telecamera di gioco che, nel caso di un gioco in Prima Persona (FPS), sarà proprio la telecamera presente di default nel Controller FPS importato precedentemente (Fig. 2.12).

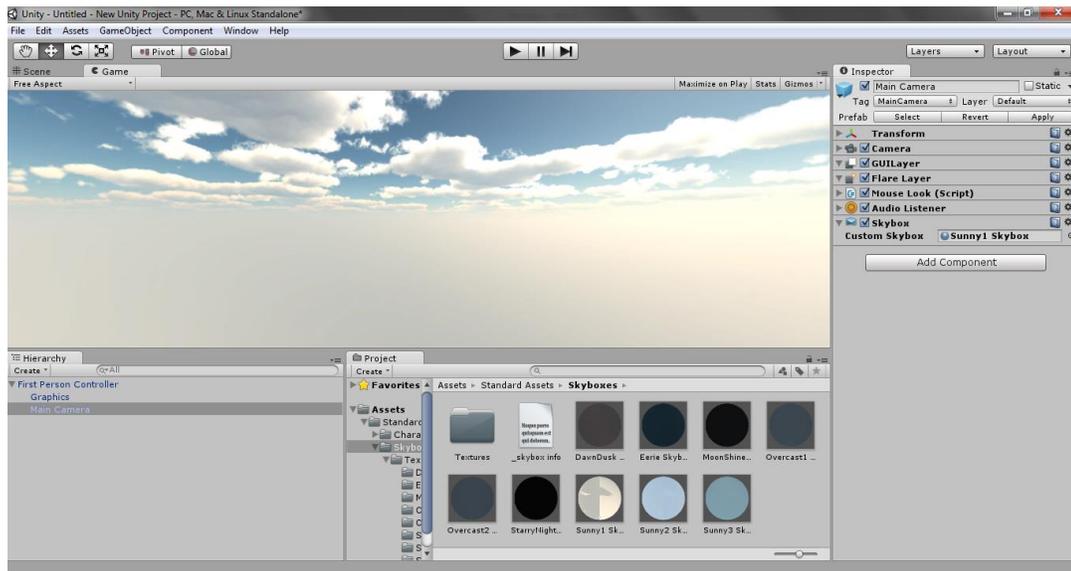


Fig. 2.12 Applicazione della componente skybox

Da notare la presenza, nel pannello Inspector, della componente 'Skybox' relativo all'oggetto 'Main Camera', appartenente al First Person Controller precedentemente importato. Per inserire il cielo è sufficiente trascinare uno dei materiali presente nella cartella 'Skyboxes' nella proprietà 'Custom Skybox'.

2.2.3 I Game Object

I Game Object sono gli oggetti primitivi che vengono messi a disposizione da Unity. Essi possono essere sia primitive matematiche (*Cube*, *Sphere*, *Capsule*, *Cylinder*, *Plane*, *Quad* come visto in precedenza per Maya) ma anche primitive di altro genere come i punti luce (Point Light, Directional Light etc.) o i terrain (Terreni).

Per istanziare un Game Object basta cliccare sul menù '*Game Objects->Create Other*' e scegliere il Game Object desiderato, in questo esempio verrà aggiunto alla scena un '*Cube*'. Si può notare come esso, una volta selezionato dal menù precedente, ora compaia nel pannello Hierarchy. (Fig. 2.13)

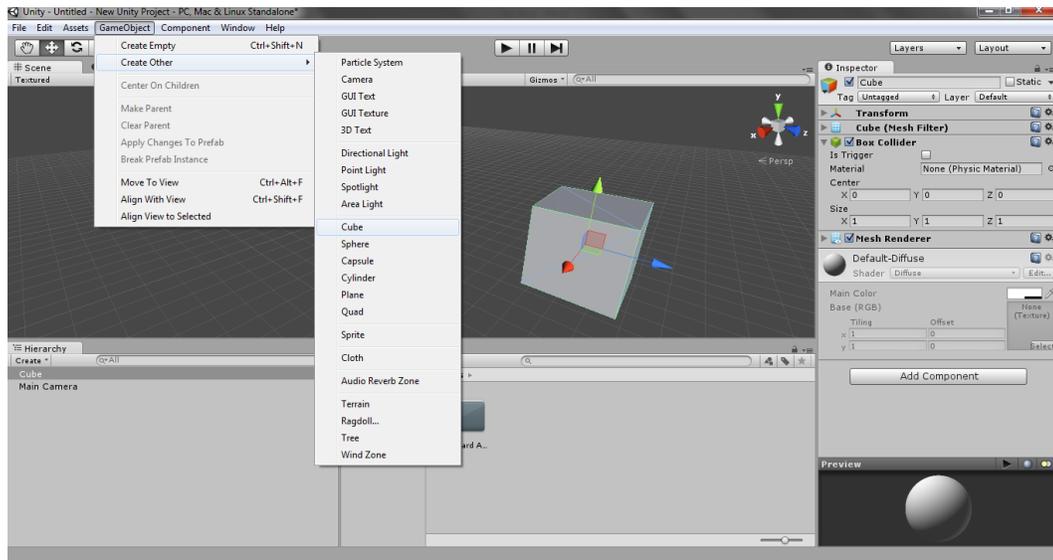


Fig. 2.13 Creazione di un oggetto Cube dal menù Game Object

2.2.4 Gli scripts

Per gestire l'interattività di un progetto, sia esso un gioco o una visita virtuale, Unity mette a disposizione un meccanismo di scripting grazie al quale è possibile scrivere degli script e associarli ai vari oggetti presenti sulla scena definendo il loro comportamento in relazione ai vari eventi che possono accadere durante l'esecuzione. Per la scrittura degli scripts Unity supporta i due linguaggi a oggetti C# e JavaScript.

L'esempio mostrato di seguito nonché gli script realizzati per il progetto finale son stati effettuati utilizzando il linguaggio JavaScript.

Unity dispone inoltre di un proprio editor di script, MonoDevelop, in maniera nativa senza dover installare software di terze parti.

Prima di procedere con la creazione degli scripts è opportuno creare un'apposita cartella, denominata Scripts, allo scopo di contenere tutti gli scripts che verranno creati durante lo sviluppo del progetto al fine di migliorare l'organizzazione tra i vari file presenti in esso.

Per creare uno script basta cliccare col tasto destro all'interno della cartella precedentemente creata e selezionare '*create->Javascript*'. Facendo doppio click sullo script appena creato si aprirà l'editor MonoDevelop e, come si può notare, sono presenti due funzioni di default: Start e Update.

La funzione Start viene eseguita solamente all'avvio dello script, mentre la funzione Update è sempre in esecuzione: in ogni frame di gioco essa esegue il codice che si trova al suo interno.

```
#pragma strict

function Start()
{

}

function Update()
{

}
```

Nel capitolo successivo verrà mostrato come è stato realizzato il progetto partendo dalla descrizione del museo fino alla creazione del modello 3D e la realizzazione degli script per gestire l'interattività

Capitolo 3

Realizzazione del progetto

Il capitolo può essere suddiviso in tre parti. Nella prima è stata fatta una breve descrizione del Museo Civico Mambrini e della sua storia. Nella seconda verrà illustrato il processo di creazione del modello 3D del museo ponendo particolare attenzione alle componenti più interessanti e difficili dal punto di vista della modellazione 3D. Infine nella terza ed ultima parte verrà descritta la fase di programmazione della parte interattiva del progetto con la descrizione degli script utilizzati.

3.1 Il Museo Civico Mambrini di Galeata (FC)

L'edificio in cui ha sede il museo è l'ex Convento dei Padri Minori Conventuali annesso alla Chiesa di Santa Maria dei Miracoli del borgo di Pianetto, frazione di Galeata (FC).

Il convento è raccolto attorno ad un chiostro quadrangolare dalle forme rinascimentali che fu eretto contestualmente all'annessa chiesa, ma ebbe una storia diversa da quest'ultima. I frati infatti furono mandati via nel 1784 dall'allora granduca di Toscana Pietro Leopoldo e l'intero edificio, venduto a privati, subì numerose trasformazioni che lo portarono a un lento e progressivo stato di degrado.

Era ormai in condizioni di desolante abbandono quando ne fu eseguito il recupero, negli anni '80 del Novecento, dall'Amministrazione Comunale di Galeata.

Fu Mons. Domenico Mambrini (1889-1944) che nella prima metà del Novecento si occupò del recupero e della salvaguardia delle memorie di Galeata allestendo i suoi ritrovamenti nelle sale adiacenti alla Chiesa di San Pietro (Galeata). Nel periodo successivo alla sua morte i reperti furono donati all'amministrazione

Comunale di Galeata e nel corso degli anni trovarono varie sistemazioni fino a raggiungere quella attuale nel museo civico Mambrini.

I reperti contenuti nel Museo sono situati al primo piano della struttura e sono divisi principalmente in due categorie, come mostrato in figura (Fig. 3.1).

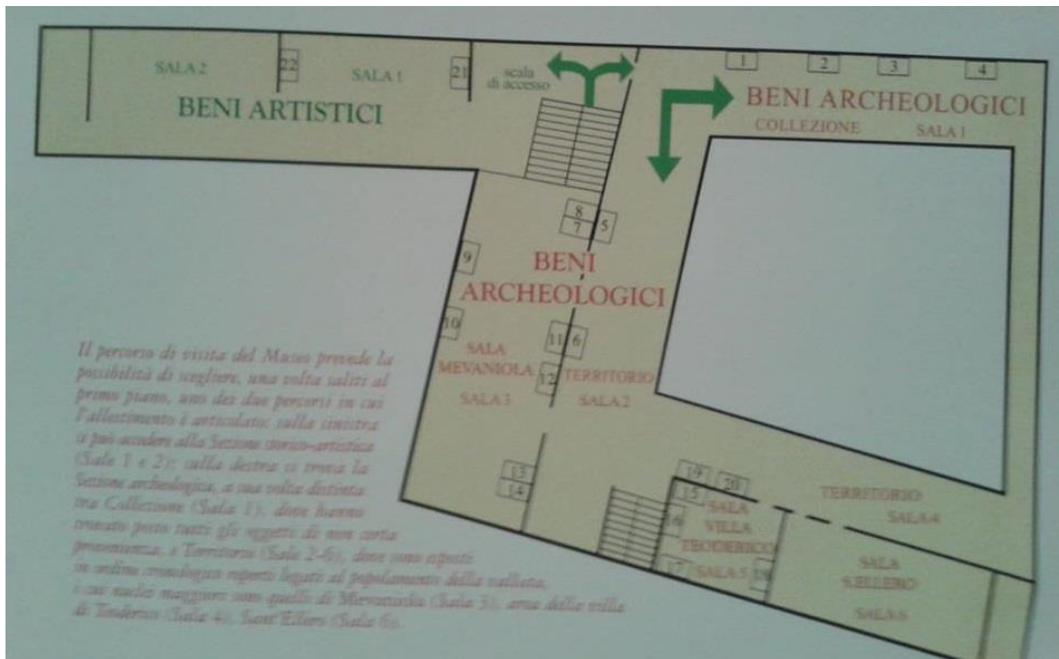


Fig. 3.1 Piantina del museo

Sezione Storico-Artistica

Sulla sinistra c'è la sezione Storico-Artistica (Fig. 3.2, Fig. 3.3) nella quale sono presenti varie opere di provenienza eterogenea, alcune infatti sono legate al territorio di Galeata e dintorni, mentre altre provengono dalla Toscana. Eterogeneo è anche il tipo di opere qui presenti, vi sono infatti alcuni dipinti, ceramiche e sculture.



Fig. 3.2 Panoramica della sezione artistica

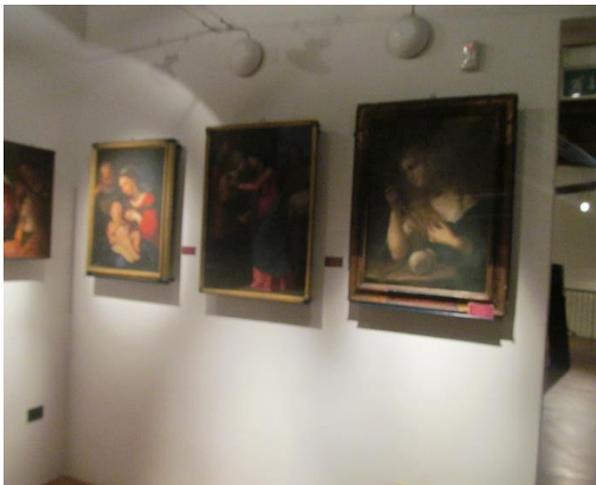


Fig. 3.3 Alcuni quadri esposti nella sezione artistica

Beni archeologici

La seconda categoria comprende i reperti archeologici i quali sono ordinati secondo un criterio cronologico, non essendo stato possibile stabilirne con certezza la provenienza e dunque l'area del rinvenimento.

Partendo dalla 'Sala 1' (Fig 3.4) si possono osservare in sequenza: fossili risalenti l'era del Miocene (23 milioni di anni fa) e del Pliocene (5,3 milioni di anni fa) seguiti da alcune opere risalenti all'antico Egitto e all'epoca romana.



Fig. 3.4 Sala 1

La 'Sala 2' (Fig. 3.5) raccoglie reperti provenienti dalla vallata del Bidente (su cui sorge appunto Galeata) risalenti all'età preistorica.



Fig. 3.5 Sala 2

Nella 'Sala 3' (Fig. 3.6) sono presenti reperti provenienti dal sito archeologico di Mevaniola, il quale sorge nelle vicinanze del museo ed è attualmente uno dei siti di età Romana più importante in Emilia-Romagna dopo Ravenna.



Fig. 3.6 Sala 3

Il percorso prosegue poi cronologicamente nella 'Sala 5' (Fig. 3.7), la quale contiene oggetti provenienti dai resti di una villa fatta costruire dal re degli Ostrogoti Teodorico attorno al 502 d.C. in occasione del restauro dell'acquedotto traiano che, seguendo il corso del fiume Bidente, portava l'acqua a Ravenna.



Fig. 3.7 Sala 5

La 'Sala 6' (Fig. 3.8) è interamente dedicata all'abbazia di Sant'Ellero, la quale sorge nei pressi di Galeata.



Fig. 3.8 Sala 6

Infine, lungo la 'Sala 4' (Fig. 3.9) sono presenti decorazioni architettoniche provenienti dall'abbazia di Sant'Ellero e dal territorio. [15]



Fig. 3.9 Sala 4

3.2 Creazione del modello 3D

Il modello 3D del museo è stato realizzato seguendo le proporzioni reali grazie alle planimetrie messe a disposizione dall'amministrazione comunale di Galeata. Esso è stato realizzato in più passaggi partendo dalla creazione delle sole mura di perimetro e del tetto, fino alla modellazione dei reperti contenuti al suo interno. E' possibile dunque suddividere questa fase in varie parti: la creazione della struttura portante, l'aggiunta dei dettagli, la creazione dei reperti e l'applicazione dei materiali e delle texture.

Dato il grande numero di oggetti presenti nel modello verranno descritti di seguito solamente quelli più complessi, per livello di difficoltà di modellazione, e quelli che rappresentano al meglio le tecniche di modellazione più utilizzate.

3.2.1 Struttura portante: mura, pavimenti e tetto

Per le mura perimetrali è stata necessaria l'importazione dell'immagine della planimetria del museo nella vista dall'alto *'top'* della scena cliccando su *'view->image plan->import Image'* (Fig. 3.10).

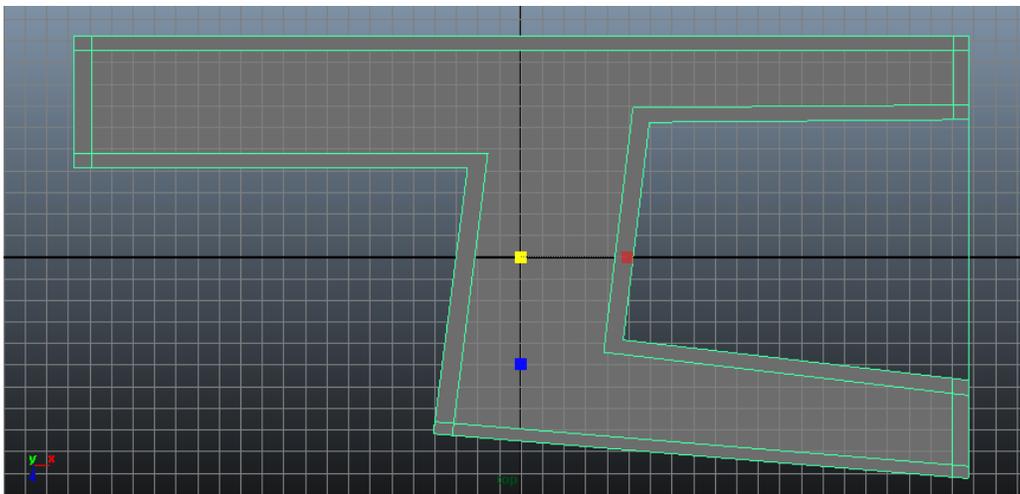


Fig. 3.10 Top view delle mura perimetrali

Successivamente è stato creato un cubo '*Create->poligon primitives->cube*' e, dopo averne ridotto opportunamente l'altezza è stato modellato opportunamente fino a ricoprire l'intera superficie della struttura utilizzando l'azione '*edit mesh->interactive split tool*'.

In questo modo ho creato il pavimento vero e proprio della struttura.

Per realizzare le mura è stato modellato un piano seguendo il perimetro dell'immagine di riferimento e successivamente, applicando l'azione '*Extrude*' sulle facce del perimetro, è stato traslato tutto lungo l'asse y.

Il pavimento del secondo piano è stato realizzato facendo una copi di quello creato in precedenza opportunamente traslata lungo l'asse y.

Per la realizzazione del tetto è stata utilizzata l'azione '*interactive split tool*'. Selezionando poi gli *edge* appena inseriti e traslandoli lungo l'asse y è possibile dare al tetto la forma opportuna. (Fig. 3.11)

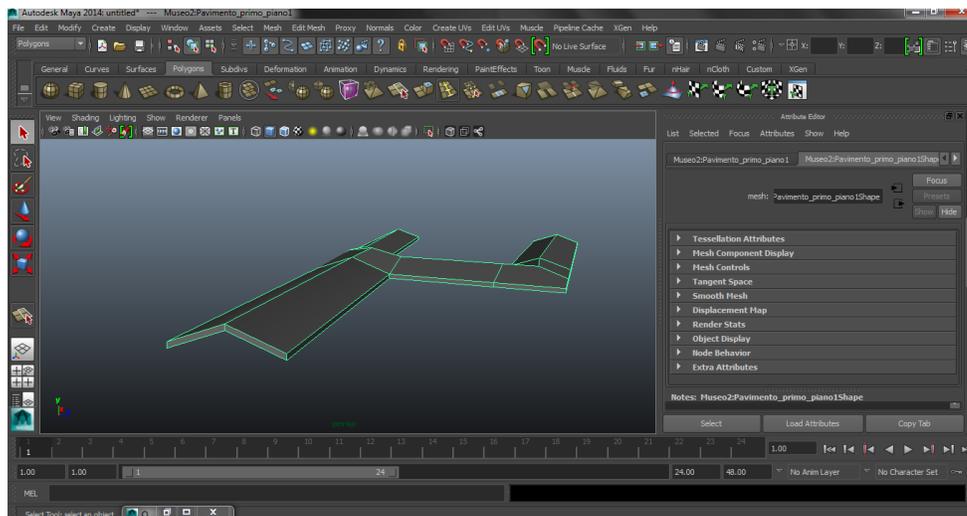


Fig. 3.11 Il tetto

3.2.2 Dettagli

Porte e finestre:

La creazione di porte e finestre si è sviluppata in due fasi, è stato necessario infatti per prima cosa creare le aperture nelle mura in cui inserire le finestre o porte vere e proprie.

Per poter fare ciò è stata utilizzata l'*action 'Insert edge loop tool'* come in figura e sono state eliminate le opportune facce creando l'apertura in cui inserire la porta o la finestra. (Fig. 3.12)

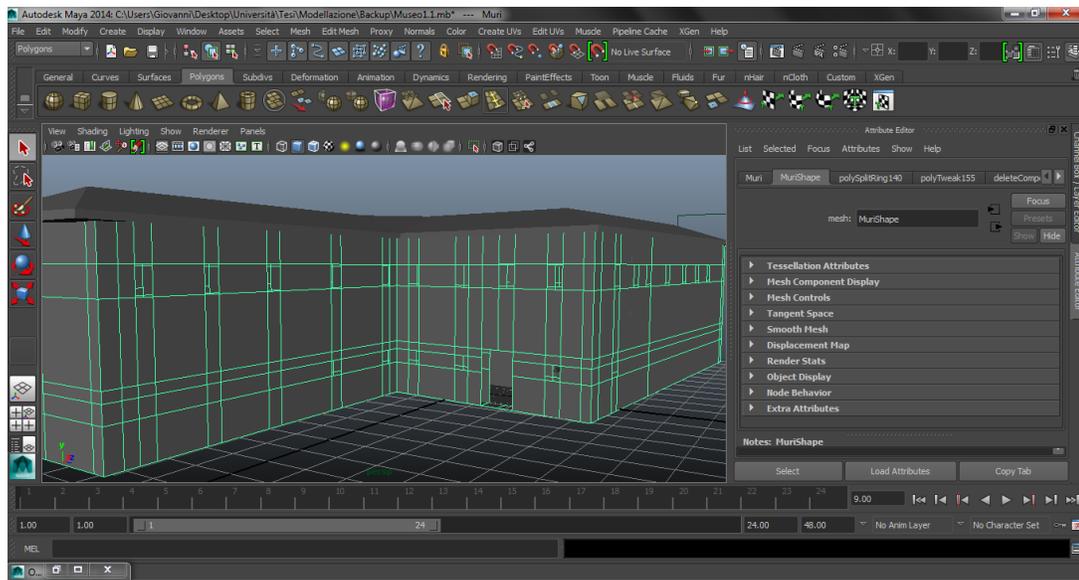


Fig. 3.12 La struttura esterna con le aperture in cui inserire le finestre e le porte

Le porte presenti nel modello sono di due tipi: ad arco e semplici. Per entrambe la tecnica di base è stata la medesima, modellando opportunamente un cubo utilizzando '*insert edge loop tool*' e '*extrude*'(Fig. 3.13, Fig. 3.14).

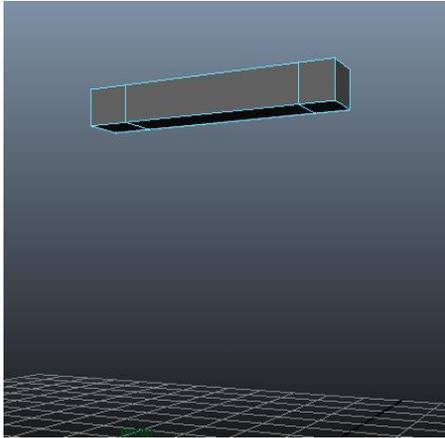


Fig. 3.13 Insert Edge Loop

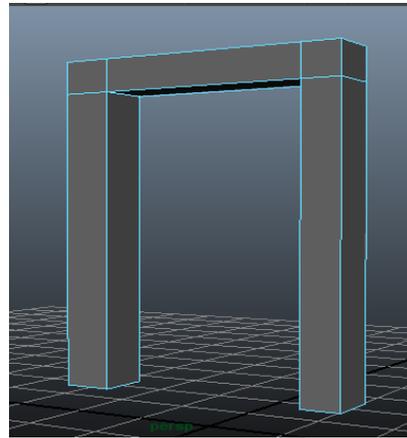


Fig. 3.14 Extrude

Per modellare l'arco è stata utilizzata l'azione *'insert edge loop tool'* opportunamente combinata con il *tool* di traslazione (stessa tecnica utilizzata per creare le arcate del chiostro interno). (Fig. 3.15, Fig. 3.16)

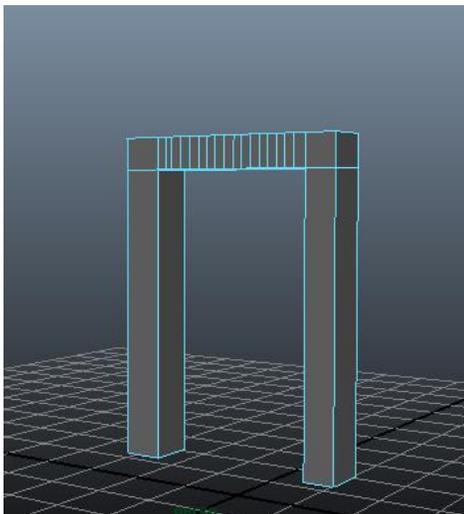


Fig. 3.15 Insert Edge Loop

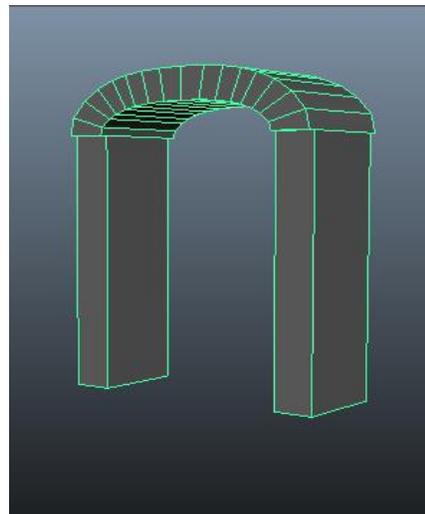


Fig. 3.16 Traslazione

Le finestre sono anch'esse di due tipologie: quelle del primo piano e quelle del piano terra. Quelle relative al primo piano sono state realizzate in due parti: la base (formata da un cubo opportunamente ridimensionato al quale son stati aggiunti degli *edge loop* nella parte inferiore) e la cornice realizzata con la stessa tecnica utilizzata per le porte semplici. (Fig. 3.17, Fig. 3.18)

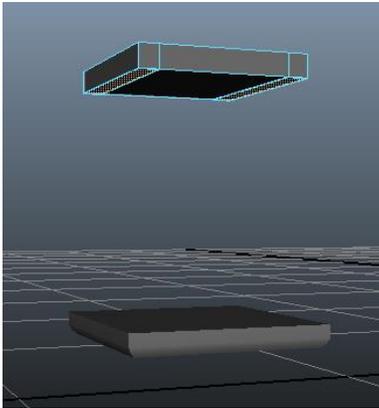


Fig. 3.17 Insert Edge Loop nella parte superiore

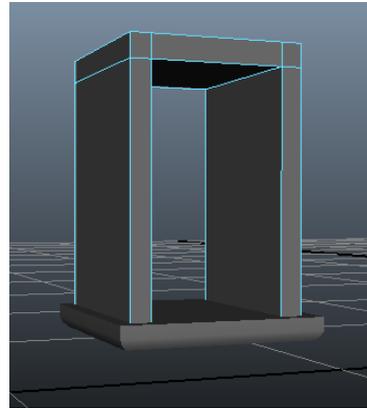


Fig. 3.18 Base e cornice complete

Le finestre del piano terra invece sono composti da due cubi, uno per la cornice e l'altro per la parte interna, anch'essi modificati facendo uso dell'istruzione '*insert edge loop tool*'. (Fig. 3.19, Fig. 3.20)

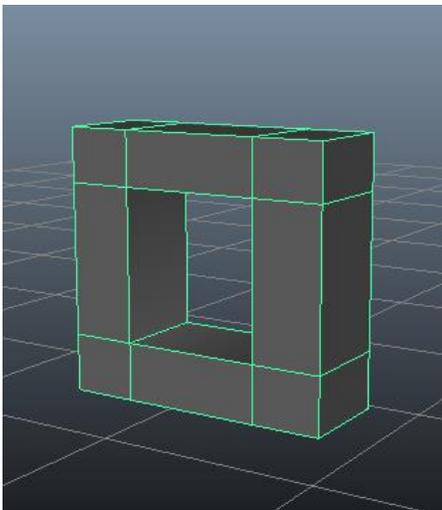


Fig. 3.19 Cornice

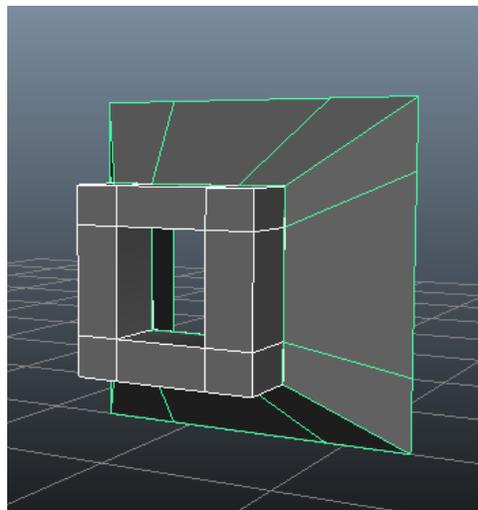


Fig. 3.20 Finestra completa

3.2.3 Il chiostro

Le arcate, come descritto in precedenza, sono state realizzate utilizzando la stessa tecnica vista per gli archi delle porte, mentre le colonne son state modellate unendo un cilindro (modificato tramite *l'insert edge loop tool*) e due cubi (la base e il capitello). (Fig. 3.21, Fig. 3.22)

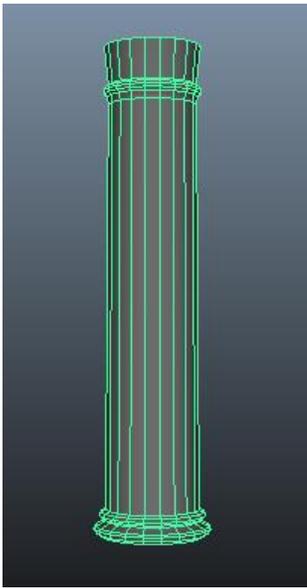


Fig. 3.21 Cilindro

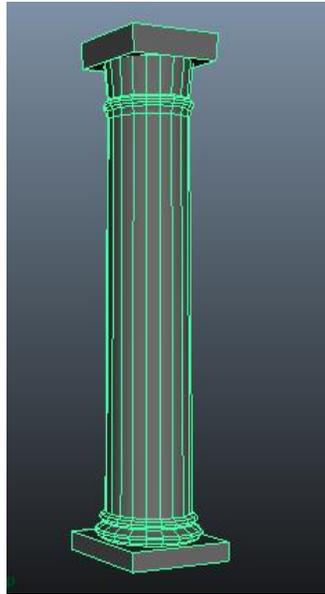


Fig.3.22 Colonna completa

Il pozzo centrale è stato ottenuto unendo un cubo (base) e un cilindro (pozzo) e, dopo aver replicato le colonne e le arcate, il risultato è il seguente. (Fig. 3.23)

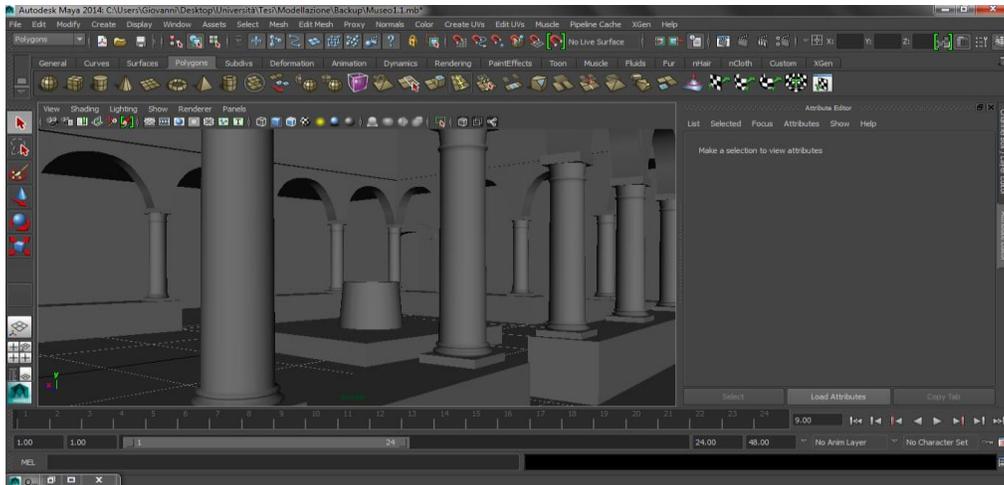


Fig. 3.23 Il chiostro

3.2.4 Scale

Le scale che portano dal piano terra al primo piano sono state realizzate partendo da un cubo e eseguendo tante suddivisioni, grazie all'istruzione '*insert edge loop tool*', quanti sono i gradini e successivamente estrudendo ogni faccia ottenuta opportunamente traslata lungo l'asse y. (Fig. 3.24, Fig. 3.25)

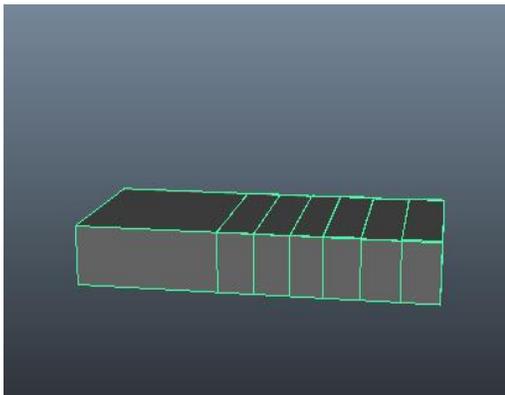


Fig. 3.24 Insert Edge Loop

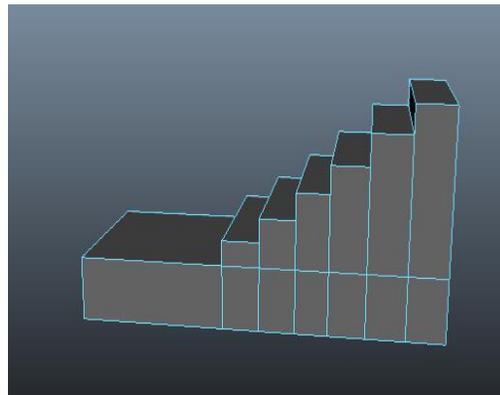


Fig. 3.25 Extrude

3.2.5 I reperti

Per la creazione dei reperti contenuti all'interno del museo è stata fatta una selezione privilegiando quelli di rilevanza maggiore (a livello di importanza storica) e di maggior visibilità. Inoltre, si è preferito effettuare la modellazione di reperti solamente di una certa grandezza, intesa come dimensioni fisiche, mentre sono stati tralasciati quelli troppo piccoli per poter essere modellati in modo ottimale.

Le teche

Le teche sono state realizzate unendo due cubi, uno per la base e uno per il coperchio il quale è stato precedentemente modellato grazie all'*insert edge loop tool* e all'*extrude* utilizzando la stessa tecnica vista per le finestre. (Fig. 3.26, Fig. 3.27)

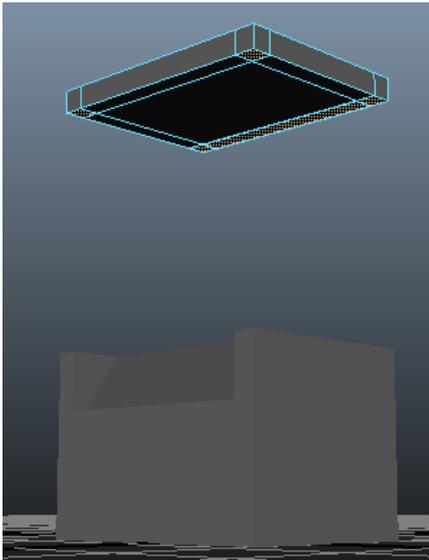


Fig. 3.26 Base e coperchio

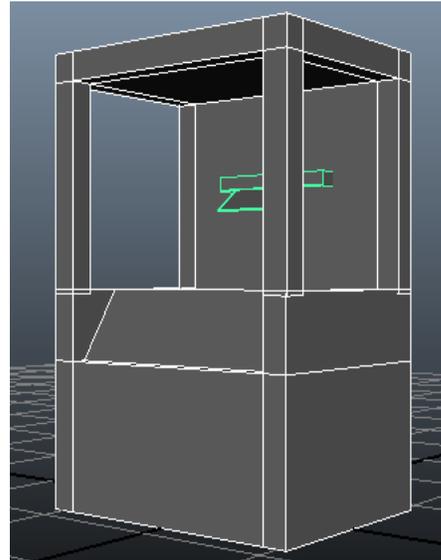


Fig. 3.27 Teca completa

Il basso rilievo

Il pezzo probabilmente più importante del museo, a causa della sua valenza storica ma soprattutto della leggenda che lo accompagna, è sicuramente il basso rilievo raffigurante il re degli Ostrogoti Teodorico con il suo cavallo inchinarsi dinnanzi alla figura di S. Ellero. Si narra, infatti, che il re nel tentativo di imporre le sue leggi cercò di raggiungere a cavallo l'abbazia dove risiedeva il monaco, che rifiutava le imposizioni volute dal re. Ma prima di arrivare alla cinta muraria il cavallo del re si fermò come impietrito e per tale prodigio egli chiese perdono e il monaco lo sollevò e lo condusse con sé.

Il basso rilievo è formato da due tavole raffiguranti le due diverse figure del re e del monaco, il tutto inserito in una struttura al centro simile alle teche già descritte in precedenza, mentre attorno è sorretta da dei piedistalli realizzati modellando un cubo con *'insert loop tool'* e *'extrude'*. (Fig. 3.28)

Le due tavole, data la loro elevata complessità a livello di modellazione, sono state realizzate semplicemente utilizzando due cubi e applicando attentamente le texture raffiguranti i due personaggi (Fig. 3.29) in modo da diminuire il carico di lavoro effettuato dal pc.

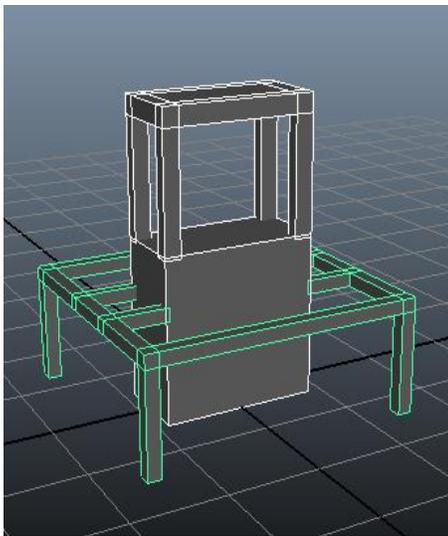


Fig. 3.28 Struttura e piedistallo



Fig. 3.29 Il basso rilievo

La chiave

Un altro reperto molto importante sia per la sua valenza storica ma soprattutto per la sua singolare fattura è la Chiave di Mevaniola. Essa è composta da due parti, una in ferro (il manico) e una in bronzo (la chiave) e la particolarità è che nel manico è raffigurata una testa canina.

Il modello è composto da un cilindro, modellato opportunamente grazie a *insert edge loop tool*, che rappresenta il manico, un cubo che forma la chiave mentre la testa canina è stata ricavata da un modello 3D di un cane [16]. (Fig. 3.30, Fig. 3.31)

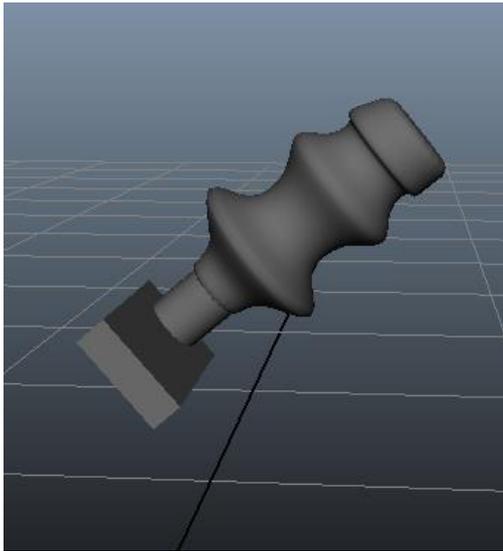


Fig. 3.30 Chiave e impugnatura

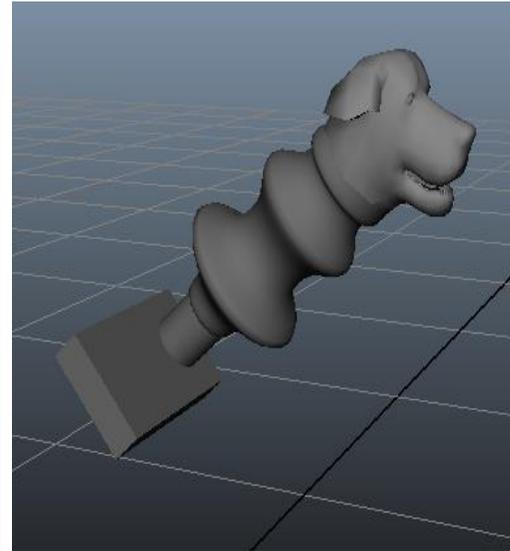


Fig. 3.31 Chiave completa

I vasi

Tutti i vasi presenti nel modello 3D sono stati creati seguendo la medesima tecnica, ovvero tramite rotazione di una curva, modellata secondo il profilo del vaso.

Si disegna una curva utilizzando l'azione *'Create->CV Curve Tool'* il quale permette di definire una curva in funzione dei propri vertici di controllo (Control Vertex), infine si esegue una rotazione della stessa di 360° utilizzando l'azione *'Surface->Revolve'* (selezionabile solo dopo aver impostato su *'Surfaces'* il tab presente in alto a destra nell'interfaccia). (Fig. 3.32, Fig. 3.33)

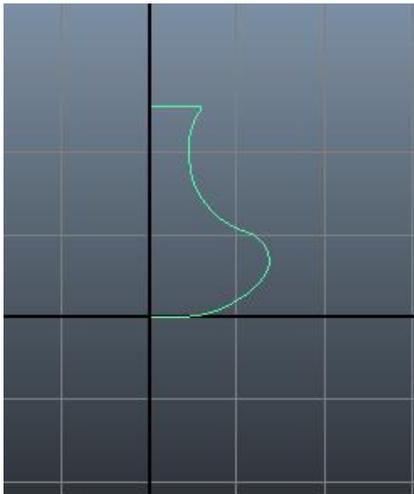


Fig. 3.32 Profilo del vaso

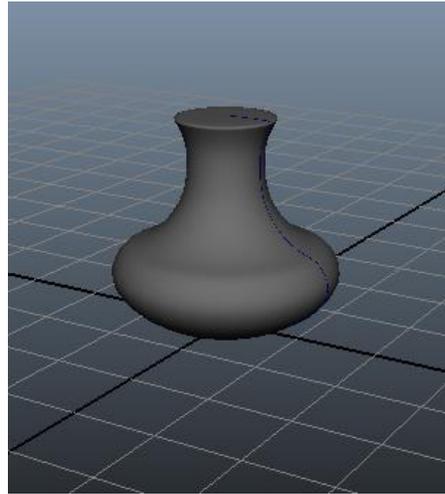


Fig. 3.33 Vaso finale

Utilizzando la medesima tecnica sono stati realizzati anche altri oggetti, qui non menzionati, che (a causa della loro forma) possono essere ottenuti attraverso la rotazione di una curva che ne definisce il profilo.

3.2.6 Assegnazione dei materiali e delle texture

L'assegnazione dei materiali è un passaggio fondamentale nella creazione di un modello 3D in quanto permette di definire l'aspetto dell'oggetto. In particolare i *'Materials'* definiscono il modo in cui l'oggetto interagisce con la luce che lo colpisce, i più utilizzati sono *'Lambert'* (poco riflettente) e *'Phong'* (molto riflettente).

A ogni materiale è poi possibile associare un colore o, nel caso in cui l'oggetto da rappresentare non sia in tinta unita, una texture che definisce il 'motivo' della superficie dell'oggetto.

La seguente immagine (Fig. 3.34) mostra un tipico esempio di texture raffigurante un muro di mattoni (essa è anche la texture utilizzata per le mura esterne del museo).



Fig. 3.34 Muro di mattoni fotografato sul posto

Per creare un materiale, e associargli una texture, si utilizza una finestra di Maya chiamata *Hypershade* (Fig. 3.35) dalla quale è possibile creare materiali scegliendoli dalla lista presente sulla sinistra.

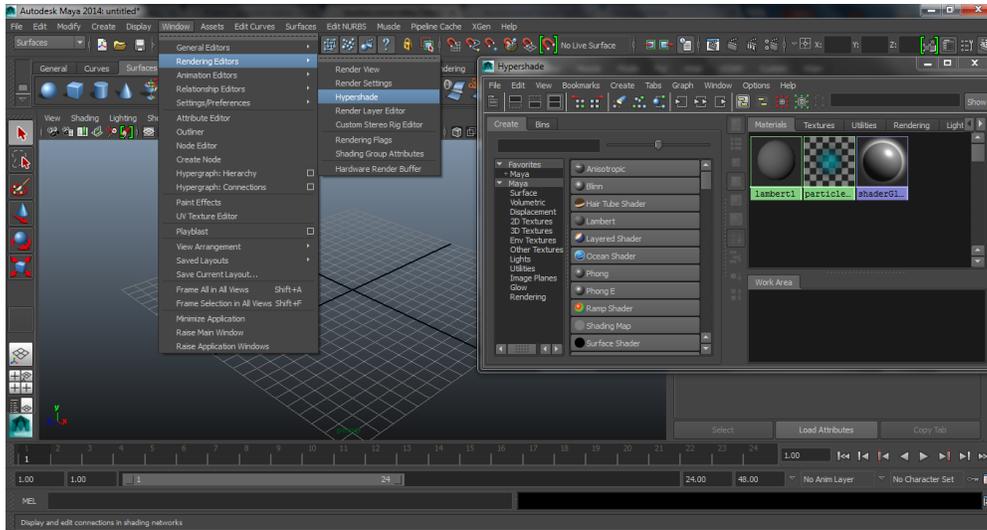


Fig. 3.35 Il menù Hypershade

Selezionando un materiale esso verrà immediatamente aggiunto alla lista. Di default i materiali vengono creati aventi colore grigio, per assegnargli una texture occorre, dopo averli selezionati, inserire il file della texture desiderata nel campo *'color'* delle proprietà.

Occorre, dunque, cliccare sul quadratino presente di fianco alla barra di selezione dei colori, scegliere *'file'* nel menù successivo e, cliccando sul simbolo della cartella che compare a destra nell'interfaccia, scegliere il file da assegnare. (Fig. 3.36)

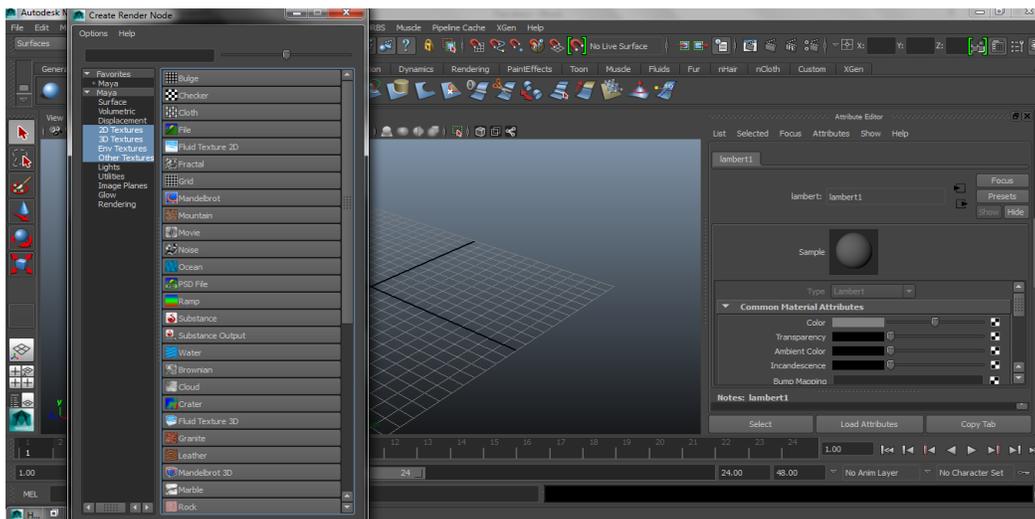


Fig. 3.36 Il menù di scelta della texture

Infine, per assegnare un materiale a un oggetto presente nella scena è sufficiente selezionare tale oggetto (o le sue facce) e, tenendo premuto il tasto destro del mouse, selezionare *'Assign existing material'* selezionando il materiale tra quelli presenti nella lista.

I materiali presenti nel modello 3D del museo sono stati tutti assegnati seguendo il procedimento appena descritto. (Fig. 3.37)

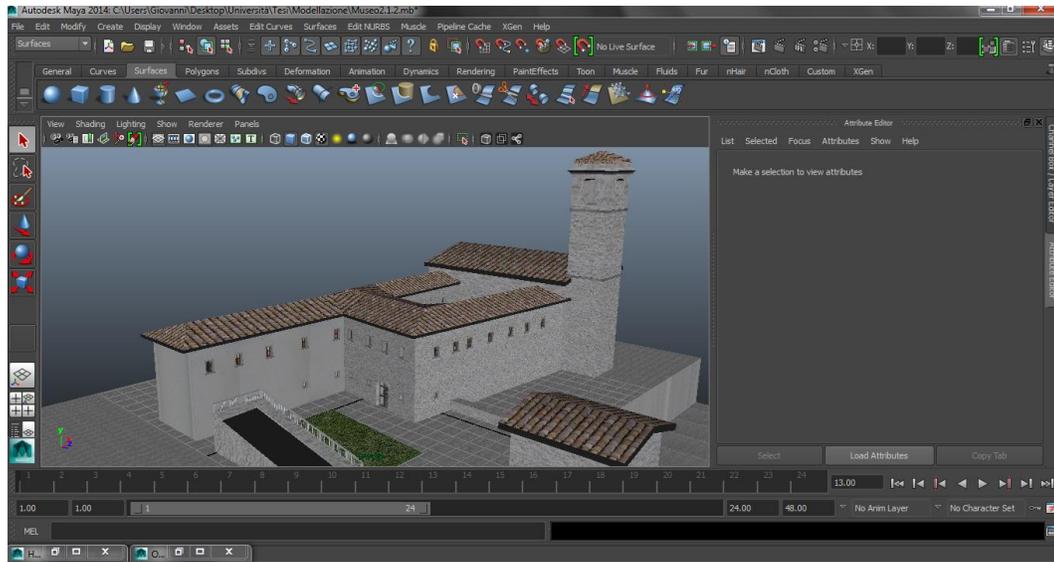


Fig. 3.37 Il modello 3D finale del museo

La maggior parte delle texture presenti nel modello 3D sono state ottenute eseguendo delle fotografie sul posto, in modo da rendere la visita virtuale il più possibile realistica.

Alcune texture invece (per esempio l'erba o la roccia che costituisce le colonne), data l'impossibilità di effettuare fotografie precise, sono state reperite eseguendo ricerche appropriate su Google immagini.

3.3 L'interattività

La parte interattiva del progetto è stata creata e gestita, utilizzando il software Unity3D, in più fasi: Importazione del modello 3D, importazione degli *Assets* necessari, scrittura degli script.

3.3.1 Importazione del modello 3D

L'importazione del modello 3D è un'operazione molto semplice in quanto è sufficiente trascinarlo dentro al pannello *'Project'*. Per avere un'organizzazione semplice e intuitiva tra i vari oggetti del progetto è opportuno, come mostrato in precedenza, creare una nuova cartella con nome *'Models'*.

Una volta importato il modello nel pannello *'Project'* si può notare che esso compare tra i file del progetto ma non nella scena, per istanziarlo occorre trascinarlo all'interno della 3D view.

Una particolarità di Unity è che mantiene tutte le proprietà con cui il modello 3D è stato creato con il relativo programma di modellazione, incluse le varie primitive di cui è composto nonché tutti i materiali e le texture. (Fig. 3.38)

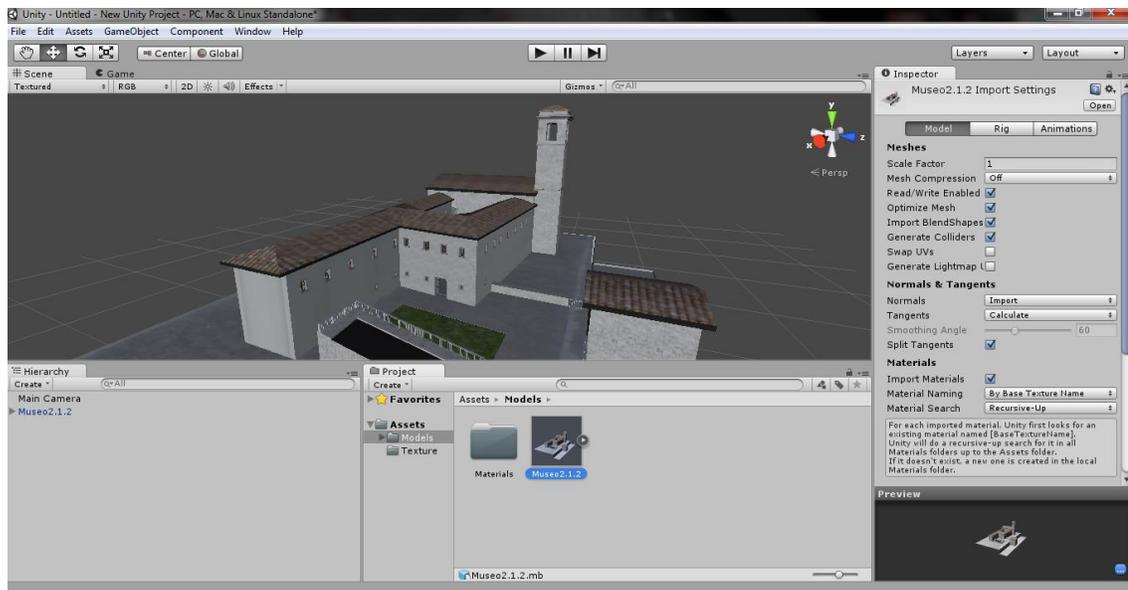


Fig. 3.38 Il modello importato in Unity3D

3.3.2 Importazione degli Assets

Per realizzare una visita virtuale è necessario gestire un *Character* (personaggio) che permetta all'utente di muoversi all'interno della scena. I *Characters* possono essere di due tipologie: First Person Controller (Prima persona) o Third Person Controller (terza persona).

Per poter inserire un *Character* all'interno della scena è necessario importare l'*Assets* relativo che Unity mette a disposizione cliccando nel menù '*Assets->Import Package->Character Controller*'.

Come accennato in precedenza tutti i file (modelli 3D, materiali, script etc.) appartenenti al suddetto 'pacchetto' verranno aggiunti immediatamente nel pannello '*Project*'.

Per poter inserire il *Character* all'interno della scena basta (come visto in precedenza per il modello) trascinarlo all'interno di essa, in questo caso è stato scelto First Person Controller. (Fig. 3.39)

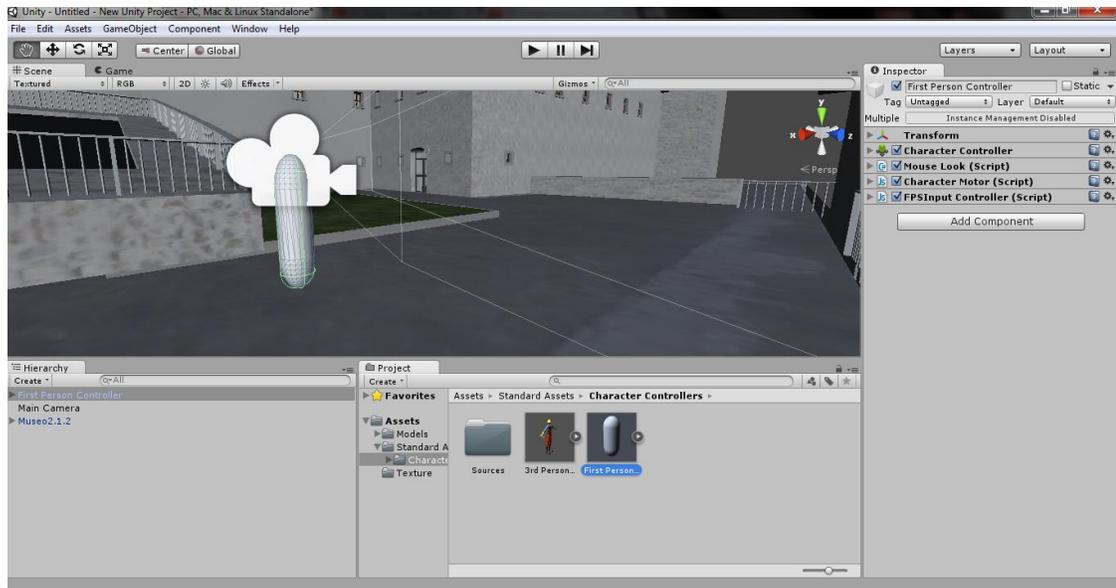


Fig. 3.39 Il First Person Controller

Associati al Controller, come si può notare dal pannello *Inspector*, sono associati di default gli script che definiscono i pulsanti di input, di default W,A,S,D per il movimento nelle 4 direzioni, e il comportamento del mouse che permette di ruotare lo sguardo del personaggio.

Sono stati aggiunti inoltre altri due *Assets*: *Skyboxes* e *Terrain*.

Entrambi hanno lo scopo di rendere il più realistico possibile l'ambiente esterno al museo (ovvero il punto di partenza della visita).

L'Asset '*Terrain*' aggiunge ai file del progetto i materiali e le texture reattive ai terreni, come per esempio l'erba o la roccia. Aggiungendo successivamente il Game Object '*Terrain*', Unity mette a disposizione vari tool atti a modellare il terreno a proprio piacimento.

In questo caso, utilizzando il '*brush tool*' appartenente all'oggetto '*terrain*' si è cercato di modellare, in modo più realistico possibile, l'aspetto delle montagne che circondano il museo. Da notare la differenza di texture tra le montagne più vicine (erba) e quelle più lontane (roccia). (Fig. 3.40)

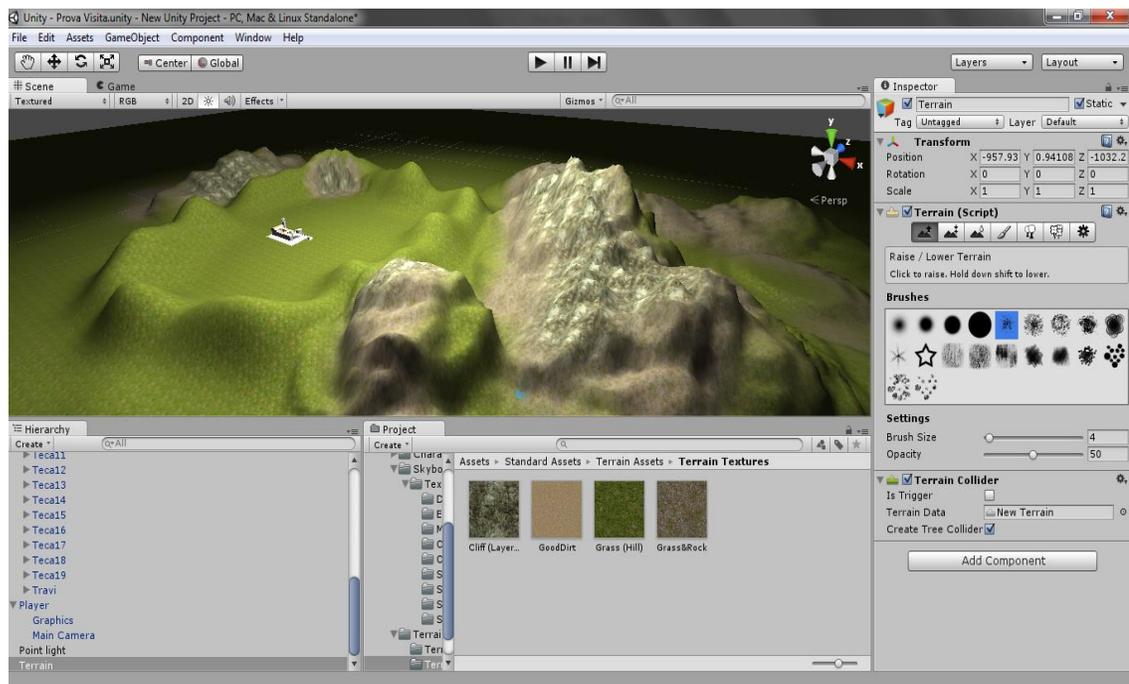


Fig. 3.40 Panoramica del Terrain finale

3.3.3 Illuminazione

Per poter illuminare l'ambiente Unity mette a disposizione quattro tipologie di luci: Point Light, Spot Light e Directional Light.

Una luce di tipo '*Point Light*' illumina un'area, definita dalla proprietà '*Range*', in modo uniforme in tutti i suoi punti e pertanto non genera ombre.

Il tipo '*Spot Light*' genera un cono di luce, a partire da un punto, di cui è possibile modificarne l'area tramite la proprietà '*Spot Angle*' e concentra la luce all'interno di esso.

Il tipo '*Directional Light*' genera un cilindro di luce avente una direzione e di conseguenza se ci sono oggetti verranno proiettate le loro ombre sull'ambiente.

Per l'illuminazione dell'ambiente esterno è stata utilizzata una luce di quest'ultimo tipo dato che simula molto bene l'effetto del sole (Fig. 3.41).

Per l'interno del museo invece sono state utilizzate luci di tipo *Point Light* (Fig. 3.42)

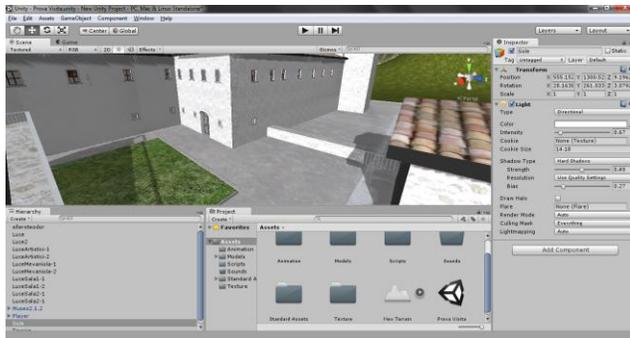


Fig. 3.41 Directional Light

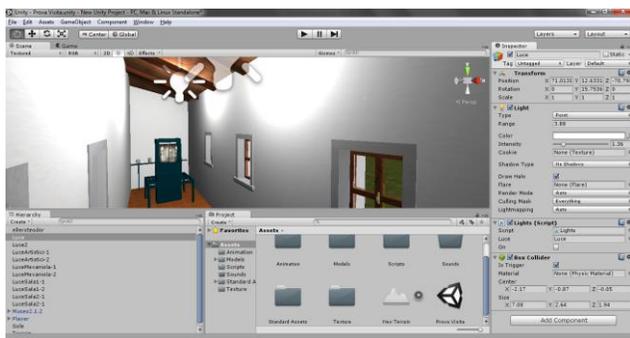


Fig. 3.42 Point Light

3.3.4 Sorgenti audio

Unity permette anche l'inserimento e la gestione di sorgenti audio.

Per inserire una sorgente sonora è sufficiente creare un nuovo Game Object vuoto, cliccando su *'Game Object->Create Empty'* ed aggiungere, dal pannello Inspector, una nuova componente di tipo *'Audio->Audio Source'*.

Una volta creata la sorgente è sufficiente associare ad essa un file audio (presente nel pannello Project) e modificarne parametri quali il volume, la riproduzione in loop oppure la distanza massima entro la quale è possibile sentire il suono emesso dalla sorgente.

Modificando inoltre i parametri relativi alle proprietà *'2D Sound Settings'* e *'3D Sound Settings'* è possibile rendere il suono più dinamico impostando per esempio il graduale abbassamento del volume (fade out) in relazione alla lontananza dalla sorgente, o viceversa. (Fig. 3.43)

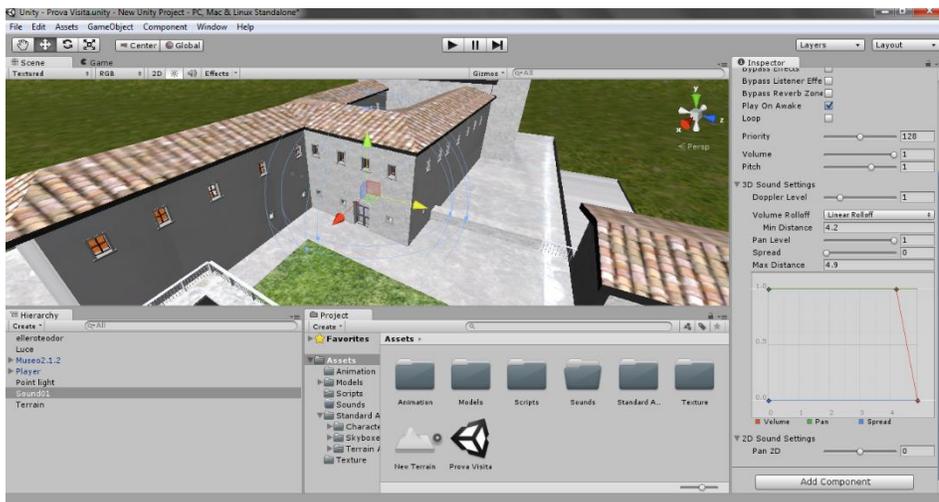


Fig. 3.43 Le proprietà della componente Audio Source

3.3.5 Scripts

Animazione della porta

Per poter aprire e chiudere le porte in maniera interattiva è necessario oltre che creare uno script, che verrà mostrato successivamente, impostare anche le animazioni relative all'apertura e alla chiusura delle stesse.

Unity dispone di un sistema di animazione del tutto simile a quello visto in precedenza per Maya: infatti una volta impostata la posizione iniziale e finale dell'oggetto, quelle intermedie verranno calcolate per interpolazione. Si procede selezionando l'oggetto da animare e aggiungendo una componente 'Animation' nel pannello Inspector. Aprendo poi la finestra 'Window->Animation' si aprirà un pannello in cui è possibile impostare le chiavi (key) nei vari frame di gioco. (Fig. 3.44)

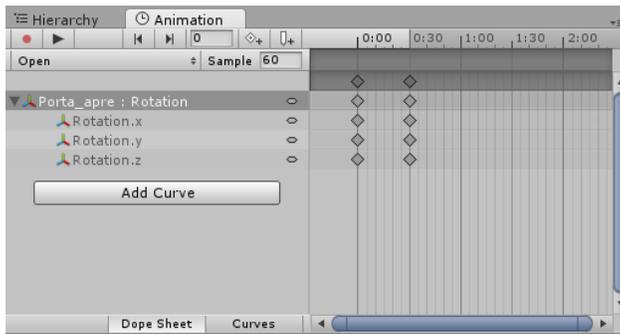


Fig. 3.44 Il pannello Animation

Nel caso dell'apertura e chiusura di una porta il parametro fondamentale è il grado di rotazione rispetto all'asse y, chiusa 0° aperta 85°.

Una volta create le animazioni è stata aggiunta alla porta una componente 'Box Collider', la quale ha la funzione di creare un'area, attorno alla porta: se il personaggio si trova all'interno di essa può aprire la porta (facendo partire le animazioni create in precedenza) premendo il tasto E sulla tastiera. (Fig. 3.45)

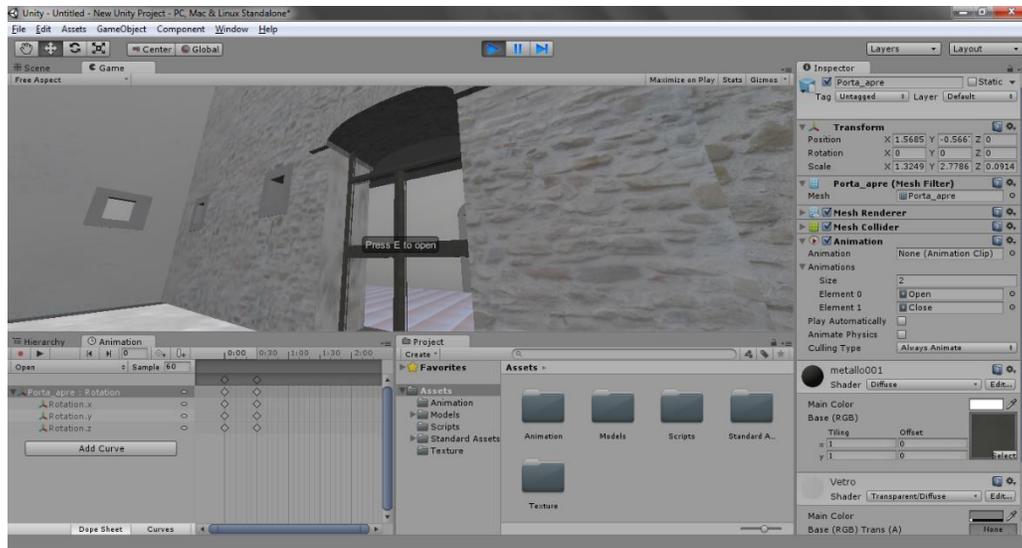


Fig. 3.45 Funzionamento dello script

Lo script che permette il verificarsi di tutto ciò è il seguente e, tralasciando i dettagli sulla sintassi del codice, grazie alle tre funzioni `OnTriggerEnter()`, `OnTriggerExit()` e `changeDoorState()` permette la gestione degli eventi appena descritti.

Le prime due funzioni eseguono un controllo sul `BoxCollider`, verificando se il personaggio è all'interno o all'esterno di esso. Nel caso in cui il personaggio entri nel `BoxCollider` viene attivata la scritta 'Press E to open' e nel caso in cui venga premuto il tasto viene richiamata la funzione `changeDoorState()` che apre o chiude la porta.

```

#pragma strict

var theDoor : Transform;
private var drawGUI = false;
private var doorsClosed = true;

function Update ()
{
    if (drawGUI == true && Input.GetKeyDown(KeyCode.E))
    {
        changeDoorState();
    }
}

function OnTriggerEnter (theCollider : Collider)
{
    if (theCollider.tag == "Player")
    {
        drawGUI = true;
    }
}

function OnTriggerExit (theCollider : Collider)
{
    if (theCollider.tag == "Player")
    {
        drawGUI = false;
    }
}

function OnGUI ()
{
    if (drawGUI == true)
    {
        GUI.Box (Rect (Screen.width*0.5-51, 200, 102, 22), "Press E to open");
    }
}

function changeDoorState ()
{
    if (doorsClosed == true)
    {
        theDoor.animation.CrossFade("Open");
        doorsClosed = false;
        yield WaitForSeconds(3);
        theDoor.animation.CrossFade("Close");
        doorsClosed = true;
    }
}

```

Accensione e spegnimento delle luci

Lo script che gestisce l'accensione e lo spegnimento delle luci presenti dentro il museo è strutturato in modo simile a quello visto in precedenza per la porta.

Anche per le luci, infatti, è stata aggiunta una componente *'Box Collider'* che svolge lo stesso ruolo visto in precedenza.

In questo caso non vi è bisogno però di richiamare un'animazione, dal momento in cui per poter spegnere o accendere le luci basta settare a true o false la proprietà *'enable'* appartenente al Point Light relativo.

```
#pragma strict

var Luce : GameObject;
var on = false;

function Update ()
{
    if(on==true)
    {
        if(!Luce.light.enabled)
        {
            Luce.light.enabled=true;
        }
    }
    else
    {
        if(Luce.light.enabled)
        {
            Luce.light.enabled=false;
        }
    }
}

function OnTriggerEnter(theCollider : Collider)
{
    if(theCollider.tag=="Player")
        on=true;
}

function OnTriggerExit(theCollider : Collider)
{
    if(theCollider.tag=="Player")
        on=false;
}
```

Visualizzazione delle informazioni sui reperti

Il principio base è lo stesso utilizzato per i due script precedenti, utilizzando un Box Collider il quale, una volta raggiunto dal personaggio, permette all'utente di premere il tasto E per visualizzare informazioni sul reperto innanzi a lui.

Per visualizzare le informazioni è stato necessario aggiungere un nuovo Game Object chiamato *'GUITexture'* il quale, una volta associatogli l'immagine relativa alle informazioni da visualizzare, permette la stampa a video della suddetta immagine.

Per fare in modo che l'immagine venga visualizzata solo in presenza di un reperto è necessario il seguente script:

```
#pragma strict

var drawInfo = false;
var drawGUI = false;
var textRil : GUITexture;

function Update ()
{
    if (drawInfo == true && Input.GetKeyDown(KeyCode.E))
    {
        draw();
    }
}

function OnTriggerEnter (theCollider : Collider)
{
    if (theCollider.tag == "Player")
    {
        drawGUI = true;
    }
}

function OnTriggerExit (theCollider : Collider)
{
    if (theCollider.tag == "Player")
    {
        drawGUI = false;
        textRil.enabled = false;
    }
}
```

```

function OnGUI ()
{
    if (drawGUI == true)
    {
        GUI.Box (Rect (Screen.width*0.5-51, 200, 200, 22), "Press E to show Info");
        drawInfo=true;
    }
}
function draw()
{
    if(drawGUI == true)
    {
        textRil.enabled = true;
    }
}

```

La funzione draw() attiva la visualizzazione della texture contenente le informazioni tramite il comando 'textRil.enabled = true' solo se il personaggio si trova all'interno del relativo 'Box Collider' che viene verificato tramite il controllo 'if(drawGUI == true)'. La variabile drawGUI infatti viene settata a 'true' nella funzione 'OnTriggerEnter', vale a dire solo se il personaggio entra all'interno del 'Box Collider'. L'immagine contenente le informazioni viene rimossa nel momento in cui il personaggio esce dal 'Box Collider' infatti il comando 'textRil.enabled = false' è inserito nella funzione 'OnTriggerExit'.

Esportare il progetto:

Per poter esportare il progetto e renderlo eseguibile anche all'esterno di Unity occorre cliccare su '*File->build*' e selezionare la piattaforma a cui il progetto è destinato. Una volta esportato il progetto è possibile eseguirlo anche in un dispositivo che non ha Unity installato, purché sia compatibile col tipo di file generato dal programma.

Nel prossimo ed ultimo capitolo verranno mostrate alcune immagini relative alla visita vera e propria eseguita sul computer.

Capitolo 4

La visita

Verranno di seguito mostrate alcune immagini riguardanti il funzionamento dell'applicazione finale.

La visita comincia all'esterno del museo (Fig. 4.1) quindi l'utente dovrà raggiungere la porta d'entrata e premere il tasto E per aprirla. (Fig. 4.2)



Fig. 4.1 Esterno del museo e punto di partenza della visita



Fig. 4.2 Apertura della porta di entrata

Una volta raggiunto l'interno del museo si può aprire la porta presente nella stanza, per poter visitare il chiostro (Fig. 4.3), oppure salire le scale sulla sinistra per raggiungere il primo piano del museo nel quale sono contenuti i reperti.



Fig. 4.3 Panoramica del chiostro

Una volta raggiunto il primo piano è possibile: girare a sinistra per visitare la sezione artistica (Fig. 4.4) oppure girare a destra e entrare nella Sala 1 (Fig.4.5) che è collegata con tutte le altre e osservare i reperti.



Fig. 4.4 La sezione artistica



Fig. 4.5 La Sala 1

Per poter visualizzare le informazioni relative a un reperto l'utente deve avvicinarsi ad esso, o alla teca relativa, e premere il tasto E. Comparirà immediatamente un'immagine con scritte le informazioni desiderate, per poter continuare la visita l'utente deve allontanarsi dall'oggetto e le informazioni scompariranno. (Fig. 4.4, Fig. 4.5)



Fig. 4.4 Il basso rilievo



Fig. 4.5 La visualizzazione delle informazioni

Conclusioni

Anche se la qualità grafica di questa visita virtuale non è paragonabile a quella che possiamo osservare nei film d'animazione o nei videogiochi, è possibile comunque cogliere le potenzialità che questo lavoro può offrire.

Lo scopo principale di questa tesi è, infatti, quello di mostrare le potenzialità che può offrire la Computer Grafica applicata all'ambito Storico e di cercare di creare degli spunti per un rinnovo dei metodi di studio e divulgazione della Storia.

Non vanno sottovalutate inoltre anche le possibilità che, grazie ad Internet, visite virtuali come questa possono offrire in termini di valorizzazione del patrimonio culturale di un paese di provincia come Galeata.

Il motore di gioco Unity3D, infatti, permette di creare applicazioni che possono essere visualizzate via Web (come i giochi in Flash presenti in molti siti Internet) permettendone quindi una rapida diffusione e, come detto in precedenza, contribuendo ad accrescere l'interesse verso un territorio ricco di storia e di siti archeologici.

Senza tralasciare però il comparto tecnico dell'applicazione che richiede la creazione di un modello 3D il più fedele possibile alla realtà e la conoscenza dettagliata dei software necessari per l'implementazione.

Sitografia e bibliografia

- [1] <http://www.meetthemediaguru.org/dalla-boeing-alla-pixar-breve-storia-della-computer-art-1/>

- [2] <http://www.meetthemediaguru.org/dalla-boeing-alla-pixar-breve-storia-della-computer-art-2/>

- [3] <http://www.meetthemediaguru.org/dalla-boeing-alla-pixar-breve-storia-della-computer-art-3/>

- [4] <http://www.meetthemediaguru.org/dalla-boeing-alla-pixar-breve-storia-della-computer-art-4/>

- [5] <http://www.meetthemediaguru.org/dalla-boeing-alla-pixar-breve-storia-della-computer-art-5/>

- [6] <http://www.meetthemediaguru.org/dalla-boeing-alla-pixar-breve-storia-della-computer-art-6/>

- [7] <http://www.meetthemediaguru.org/dalla-boeing-alla-pixar-breve-storia-della-computer-art-7/>

- [8] http://it.wikipedia.org/wiki/Autodesk_Maya

- [9] http://it.wikipedia.org/wiki/Modellazione_3D#Modellazione_poligonale

- [10] http://it.wikipedia.org/wiki/Modellazione_3D#Superfici_NURBS

- [11] <http://it.wikipedia.org/wiki/Rendering>
- [12] <http://it.wikipedia.org/wiki/Radiosity>
- [13] http://it.wikipedia.org/wiki/Ray_tracing
- [14] [http://it.wikipedia.org/wiki/Unity_\(motore_di_gioco\)](http://it.wikipedia.org/wiki/Unity_(motore_di_gioco))
- [15] Luisa Mazzeo Saracino, Il Museo Civico “Mons. Domenico Mambrini”
di Galeata
- [16] <http://tf3dm.com/3d-models/dog>

Ringraziamenti

Ringrazio tutta l'amministrazione comunale di Galeata e in particolare il Vicesindaco Radu Iftode e tutto l'ufficio Beni Culturali per aver accolto con entusiasmo il mio progetto e per avermi fornito tutto il materiale necessario alla sua realizzazione.

Ringrazio inoltre tutte le persone addette al Museo Civico Mambrini che mi hanno fatto da guida durante le mie frequenti visite.

Un ringraziamento particolare anche alla Professoressa Damiana Lazzaro per aver accolto l'idea di questa tesi e per la sua disponibilità.