



UNIVERSITÀ DI PISA

**Corso di laurea magistrale in
Informatica Umanistica
percorso
Grafica, Interattività, Ambienti Virtuali**

**Sviluppo di un sistema di interactive story
telling per ricostruzioni storiche e sua
applicazione alla ricostruzione della Domus
degli affreschi di Luni**

Candidato: *Niccolò Albertini*

Relatore: *Antonio Cisternino*

Relatore: *Enrica Salvatori*

Anno accademico 2011-2012

Indice

1 – Introduzione, motivazioni	6
<hr/>	
2 – Game engine	9
<hr/>	
3 – Unity 3d	13
3.1 Editor	18
3.2 Terrain	20
3.3 Trigger	21
3.4 Prefab	22
<hr/>	
4 – Sviluppo della piattaforma	23
<hr/>	
5 – Guida all'utilizzo	26
5.1 Il foglio xml	34
<hr/>	
6 – Panoramica sul codice,Script	37
6.1 Gestione dei trigger	37
6.2 GUI	39
6.3 Visuale su più telecamere distribuite	45
6.4 L'interprete xml	49

7 – Ricostruzione della <i>Domus</i> degli affreschi di Luni	61
<hr/>	
8 – Metodologia di lavoro	59
8.1 Scenario	62
8.2 Modelli 3d statici	62
8.3 Modelli 3d dinamici	64
8.4 Texture	65
8.5 Audio	65
<hr/>	
9 – La carta di Londra	66
9.1 La carta di Londra applicata al progetto	69
<hr/>	
10 – Ricostruzione: tra fonti e soluzioni tecniche	71
10.1 Contesto storico e archeologico	72
10.2 La <i>Domus</i> degli affreschi di Luni	91
10.3 I punti di interesse	95
10.4 I locali e l'arredamento	101
<hr/>	
11 – Conclusioni	112
<hr/>	

1-Introduzione, motivazioni

Il progetto di tesi che si va a illustrare è il proseguimento di un percorso intrapreso con la tesi per il corso di laurea triennale.

Nello specifico ci si era occupati di un progetto per la ricostruzione di un sito storico (il castello “della Brina” presso Sarzana) all’interno di *Game engine* (*Unity 3d*).¹

La motivazione di una tale ricostruzione era quella di poter unire le dinamiche tipiche dei videogiochi ad una ricostruzione storica accurata.

Il maggior contributo che i videogiochi possono dare ad una ricostruzione storica è la caratteristica principale degli stessi, ovvero l’interattività; senza nulla togliere a ricostruzioni su plastico o illustrazioni dettagliate, una ricostruzione storica su *game engine* può dare un senso di interattività e immersività irraggiungibile da qualsiasi altro mezzo statico come le illustrazioni, o dinamico, come ad esempio i filmati.

Questo perché viene data la possibilità all’utente di essere protagonista della vicenda o della visita virtuale, permettendo un punto di vista unico all’interno del mondo ricostruito.

Nella ricostruzione della Brina era ripresa una vicenda storica molto dettagliata, la presa del possesso del castello da parte di un messo del vescovo di Luni nel 1279, ed era stata trasposta sotto forma di azioni da far compiere al giocatore. Questa ricostruzione era molto funzionale, ma aveva il limite di essere un progetto unico, legato solo a questa particolare ricostruzione; non c’era la possibilità di espandere questo progetto ad altri.

Il progetto attuale riguarda invece la ricostruzione della *Domus degli Affreschi* esistente nell’area archeologica di Luni (Provincia di La Spezia).

¹ Niccolò Albertini, "Ricostruzione storica del castello della Brina in epoca altomedievale mediante *game engine*", Tesi di laurea triennale, Università di Pisa, a.a.2009/2010, relatori A. Cisternino, E. Salvatori



In questo progetto, al contrario della Brina, non si hanno dettagli su avvenimenti o fatti importanti che riguardano la struttura, oggi messa in luce a livello delle fondamenta e di una porzione limitata di alzata, ma gli scavi effettuati hanno permesso di raccogliere molte informazioni riguardo proprio alle strutture murarie e come fossero organizzati gli ambienti.

Per questa ragione si è optato per una sorta di “visita” virtuale, in cui il visitatore, avvicinandosi ad un personaggio, parla con lui e, se vuole, può approfondire quanto gli viene comunicato leggendo informazioni supplementari sullo schermo e sfogliando immagini reali dei ritrovamenti.

L’approccio quindi è molto diverso a livello di ricostruzione storica, in quanto per la Brina si era optato per uno stile più dinamico e giocoso, ma meno informativo dal punto di vista storico-culturale, mentre per la *Domus degli Affreschi* abbiamo un approccio più “museale” e preciso, mantenendo comunque l’immersività e l’interattività concesse dal mezzo.

Cosa lega allora questi due progetti così diversi tra loro? Anche se gli eventi proposti sono molto differenti e le tematiche sono affrontate con uno stile del tutto diverso, i due scenari sono legati dagli stessi “task”.

Questo significa che in linea di massima le azioni da compiere a livello tecnico sono le stesse, ritroviamo sempre questi stessi “task” (compiti) .

Ad esempio se prendiamo la prima scena di entrambe le ricostruzioni troveremo da una parte un personaggio che toccando un fiume pronuncia una frase e esegue

un animazione, dall'altra un personaggio che parla e si muove se viene toccato; azioni molto diverse a livello concettuale ma praticamente identiche a livello tecnico.

Da qui nasce l'idea di creare una sorta di piattaforma per *game engine Unity 3d* in modo da dare la possibilità ad un qualsiasi utente la possibilità di creare la propria ricostruzione senza dover scrivere una sola riga di codice.

La piattaforma in questione oltre che prevedere tutti i task standard che si possono riscontrare in un contesto del genere, ha già implementata la possibilità di poter essere utilizzata all'interno di Cave (strutture multischermo per ambienti virtuali altamente immersive) grazie al supporto per il multischermo gestito da macchine diverse.

La ricostruzione di Luni è stata realizzata una volta ultimata questa piattaforma; inoltre per dimostrarne la versatilità è stato trasposto anche lo scenario della Brina, rinnovato con il nuovo sistema.

In questo documento si espone nella prima parte la piattaforma in questione, le metodologie di lavoro utilizzate e una guida all'uso.

Nella seconda parte invece si trova la realizzazione della ricostruzione storica della *Domus* di Luni, il contesto storico e archeologico e le fonti messe a confronto con le soluzioni tecniche .

Il prodotto finale è un modello per Unity 3d dove basta disporre all'interno dell'*editor* di *Unity* i modelli 3d e scrivere un documento xml con la lista degli "avvenimenti"; così facendo si sarà già creata la ricostruzione, senza doversi preoccupare d'altro.

2-Game engine

Un *game engine* è un sistema sviluppato e progettato per lo sviluppo di videogiochi su varie piattaforme.

In linea di massima offrono un motore grafico 3d, un motore fisico per la rilevazione delle collisioni, motore per l'audio, intelligenza artificiale, strumenti per lo sviluppo e il supporto di animazioni, *script*, rete e *scene graph*.

In questo modo si può rendere lo sviluppo di videogiochi più veloce, economico e meno complesso. Gli ultimi *game engine* disponibili di solito offrono la possibilità di sviluppare diverse versioni di gioco su varie piattaforme senza dover sviluppare versioni dedicate.

Rendono anche più semplice lo sviluppo del videogioco, fornendo molti strumenti pronti per soddisfare le richieste dei game designer.

I primi *game engine* vengono sviluppati negli anni '80 ma solo dai primi anni '90, si incominciò a definirli come tali.²

Dagli anni '90 in poi sono stati sviluppati *game engine* sempre più potenti, fino ad arrivare a *tool* di altissimo livello che permettono di creare videogiochi in maniera molto semplice (sempre rispetto ai primi software ideati).

In genere possiamo dire che i *game engine* sono un insieme di strumenti per lo sviluppo di videogiochi; questi strumenti, di solito visuali, si trovano tutti all'interno di un IDE (ambiente di sviluppo integrato).³

Quasi sempre sono progettati con un approccio modulare, in maniera da poter sostituire gli strumenti predefiniti con strumenti più specializzati o ampliare parti del motore dove ve ne è la necessità.

² Definizione di "*Game engine*" da Wikipedia ENG, 2012

http://en.wikipedia.org/wiki/Game_engine

³ Un **integrated development environment (IDE)**, o **ambiente di sviluppo integrato**, è un software che supporta i programmatori nello sviluppo del codice. Di solito è costituito da varie componenti: un *editor* di codice sorgente, un compilatore, un interprete, un tool di building automatico e un debugger.

Questo perché si deve mantenere sempre una certa flessibilità, visto che i *game engine* possono trovarsi a dover rispondere ai compiti più disparati, tra i quali, oltre che videogiochi, simulazioni e dimostrazioni.

Questi sono i componenti più importanti di un *game engine*:

- il **motore grafico** che ci permette di avere una resa grafica del mondo virtuale in 3D (ma anche 2D);
- di solito viene integrato anche un **motore fisico**, non solo per la simulazione del modello fisico di riferimento, ma anche soprattutto per la gestione delle collisioni tra oggetti;
- in linea di massima questi due strumenti dispongono di *editor* che visualizza il risultato e la composizione della scena gioco.
- Nella maggioranza dei casi i *game engine* dispongono di un *IDE* o comunque di un **supporto allo scripting**, per permettere ai *game designers* di scrivere codice dedicato, così da poter creare scene in maniera totalmente libera.
- Una componente importante nei videogiochi è l'**audio**, che tecnicamente ha già raggiunto da molti anni (possiamo dire nella seconda metà degli anni '90) un livello di complessità e fedeltà praticamente paragonabile a quello della realtà. Il **motore audio** offre variabili di tutti i tipi per controllare qualunque parametro legato ai suoni possa venire in mente (distanza, propagazione, effetti..).
- La **gestione delle animazioni** invece è qualcosa che si trova tra il motore grafico, quello fisico e lo *scripting*, e permette la gestione dello scheletro del modello in modo da animarlo come si vuole.
- Nei *game engine* moderni troviamo sicuramente un componente per la **gestione della rete**, di solito perché i giochi possiedono ormai quasi sempre una componente multiplayer, comunque qualche sistema che prevede lo scambio di dati tra macchine tramite la rete (partite, classifiche, profili etc); nel progetto esposto in questa tesi, la componente di gestione del network di *Unity 3D* è stata utilizzata per

assolvere un compito particolare, ovvero quello di gestire più telecamere su macchine differenti.

- Anche il sistema di **IA (intelligenza artificiale)** è fondamentale, in quanto permette di gestire gli NPC⁴; ovviamente si cercherà sempre di dare un comportamento quanto più “intelligente” e reso possibile in vari modi, di solito con gli ASF (automi a stati finiti) che permettono agli NPC di assumere comportamenti coerentemente ai cambiamenti dello stato del mondo. Molte volte per la gestione degli NPC vengono usate le cosiddette “scene *scriptate*”, che non vanno confuse con la IA (intelligenza artificiale), in quanto tramite *script* si decide a priori cosa dovrà accadere nella scena, e non dinamicamente.
- Con il passare del tempo i videogiochi sono diventati sempre più complessi, fino ad arrivare alla costruzione di scene con centinaia se non migliaia di elementi. Per questo un elemento di fondamentale importanza per un *game engine* è lo **Scene Graph**, una struttura organizzata nella maggior parte dei casi in maniera gerarchica rappresentante la scena; questo strumento permette non solo una maggior semplicità di gestione, ma anche una maggiore efficienza di calcolo, permettendo di velocizzare il *rendering*.

La struttura come abbiamo detto è gerarchica, per permettere una suddivisione della scena molto precisa, in maniera che se vengono effettuate operazioni su un nodo padre, queste verranno applicate direttamente su tutti i nodi foglia (ad esempio se un oggetto è totalmente fuori dalla visuale potremmo stabilire di non renderizzarlo, e in questo modo tutti i poligoni che lo costituiscono non verranno renderizzati, senza doverlo specificare per ogni singola entità).

In definitiva i *game engine* sono strumenti che permettono di creare *task* molto complessi, ma allo stesso tempo semplificano il lavoro di creazione di *videogame*, ma anche di applicazioni educative, simulazioni militari o mediche.

⁴ Non Player Character, ovvero personaggi non giocanti, sono i personaggi che sono controllati dal gioco e non dal giocatore.



Soprattutto negli ultimi anni, grazie alla larga diffusione di dispositivi portatili e nuovi metodi di *input* i *game engine* si sono ampliati ulteriormente per gestire il nuovo *hardware*: dispositivi *touch*, accelerometri, interfacce aptiche etc..

Attualmente esistono molti *game engine*, e la scelta di solito dipende dall'applicazione finale che si ha in mente, distinguendo tra tipologia (gioco di ruolo, sparattutto) o visuale (in prima o terza persona, dall'alto, isometrica..).

Per la realizzazione di questa piattaforma dedicata alle ricostruzioni storiche si è scelto *Unity 3D*, perché tra i tanti disponibili ha molti punti a suo vantaggio, tra cui la grande flessibilità per quanto riguarda i vari *tool* (soprattutto l'*editor*, molto immediato) che per le potenzialità del motore di *scripting*.

3-Unity 3D



Abbiamo parlato fin' ora delle caratteristiche principali di un *game engine*, di quanti ne esistano, e delle differenze che ci sono tra l'uno e l'altro.

Il progetto realizzato è stato interamente basato su *Unity 3D*, un *game engine* molto potente ma soprattutto flessibile.

Questo perché *Unity*⁵ è uno dei pochi nel suo genere che può vantare il supporto a praticamente tutte le piattaforme presenti sul mercato; lasciando immutato il codice, o con piccole modifiche relative a particolari periferiche, si possono creare giochi ed applicazioni per Windows, MacOS, Xbox 360, PlayStation 3, Wii, Adobe Flash.

Inoltre è possibile creare applicazioni in *streaming* sul *web*, che vengono rese disponibili tramite un *WebPlayer*⁶, questa è una soluzione particolare molto utilizzata in alternativa alla creazione di un eseguibile, anche perché è la maniera più semplice per rendere fruibili contenuti di questo genere sul *web*, senza bisogno di installazioni o configurazioni (complessi algoritmi di compressione decidono in base all'*hardware* dell'utente di quali dati effettuare lo *stream*).

Attualmente *Unity* viene utilizzato principalmente per lo sviluppo di applicazioni su dispositivi mobili, visto che offre la possibilità di sviluppare direttamente sia su Android che su Ios⁷.

L'*editor* visuale è completo e molto dettagliato, ma rimane di facile utilizzo; questo anche per merito della "live preview" ovvero la possibilità di vedere in

⁵ Unity Technologies, Unity 3d Game Engine <http://unity3d.com/>

⁶ Plugin per browser che permette di usufruire lo stream del videogioco, senza doverlo installare

⁷ I sistemi operativi per piattaforme *mobile* più diffusi al momento

tempo reale la scena di gioco senza la necessità di dover attendere la compilazione del codice completo ogni volta.

L'importazione è molto differente rispetto a molti altri *game engine*, infatti basta copiare i contenuti che vogliamo dentro la cartella del progetto e questi verranno automaticamente importati, ma non solo: se aggiorniamo i nostri contenuti, questi vengono reimportati e aggiornati automaticamente all'interno della scena, senza bisogno di nessuna modifica.

Il compito di creazione dei contenuti è facilitato dal fatto che *Unity* supporta la stragrande maggioranza dei formati esistenti, e si interfaccia molto facilmente col i più utilizzati programmi per l'*editing* (che siano programmi per *editing* di modelli 3d,audio,*texture* etc..).

Oltre che queste caratteristiche particolari *Unity* ha un *engine* grafico sempre aggiornato e molto leggero (un altro punto di forza per il cross-platform) con supporto a Directx (per Windows) OpenGL (Windows e Mac) e OpenGL ES (iOS e Android).

Di base sono messi a disposizione decine di *shaders*⁸, dai più semplici ai più complessi, in modo da replicare il comportamento della luce su qualsiasi materiale; viene usato il linguaggio *ShaderLab*, che permette di definire varie versioni degli *shaders*, in modo che *Unity* possa usare quello giusto a seconda delle specifiche *hardware* dell'utente.

E' stato anche implementato il Surface Shader, che permette di scrivere solo una volta gli *shaders* necessari, senza dover effettuare modifiche a seconda della piattaforma.

Sempre di base troviamo parecchi effetti *post-processing*, che coprono tutte le possibili modifiche all'immagine finale; effetti come blur (sfocatura), correzione del colore o effetto lente sono applicabili direttamente dall'*editor*.

Viene utilizzato il Deferred Render, un tipo di *pipeline* grafica presente nei moderni *game engine*, che separando il *rendering* della geometria dal *rendering*

⁸ E' uno strumento della computer grafica 3D utilizzato per determinare l'aspetto finale ,in base alla luce, della superficie di un oggetto.

delle luci permette una più veloce gestione di quest'ultime, permettendo di rendere tutto il processo molto più veloce.

In questo modo non si notano particolari cali di prestazioni anche con molte ombre in tempo reale o attivando effetti di solito molto pesanti come l'HDR⁹ (High dynamic range imaging) o SSAO¹⁰ (Screen Space Ambient Occlusion). Per la gestione dell'illuminazione dello scenario è implementato il *Lightmapping* (struttura dati che contiene le informazioni sulla luminosità dell'ambiente. Sono pre-calcolate e utilizzati per oggetti statici) tramite Beast, in più si può usare il *Dual Lightmapping*, tecnica che permette di avere due tipi di *lightmap* per un oggetto permettendo di scalare molto facilmente la qualità dell'illuminazione. Per quanto riguarda l'*Occlusion Culling* (la gestione degli oggetti fuori camera che quindi non vanno renderizzati) *Unity* ha sviluppato Umbra, una soluzione proprietaria sviluppata proprio per rendere altamente efficiente il *rendering* finale.

Il particle system, ovvero il meccanismo che si occupa di simulare il comportamento di particelle (polvere, fumo) è lo Shuriken, sistema proprietario che permette modifiche in diretta agli effetti ed è completamente visuale.

Per quanto riguarda l'*engine* fisico il team di *Unity* ha optato per l'utilizzo di NVIDIA Physx, un motore molto avanzato utilizzatissimo negli ultimi anni da molti *game engine*, per la sua flessibilità ma soprattutto per la qualità della fisica proposta.

Physx supporta molte tecniche avanzate per la fisica come la gestione di SoftBodies (elastici..), Rigidbodies (metallici..), Ragdolls (fisica dei corpi umanoidi) e *real time cloth* (simulazione fisica dei tessuti).

⁹ High Dynamic Range o Ampia Gamma Dinamica, è una usata nella grafica che consente di effettuare I calcoli dell'illuminazione in grandi spazio, in modo da aumentare la gamma della luce rappresentabile.

¹⁰ Screen Space Ambient Occlusion, è una tecnica di rendering molto efficiente che si occupa di cosa calcolare, a seconda del fatto che l'oggetto da rappresentare sia occluso da un'altro oppure no.

Come motore audio è stato scelto FMOD, un *engine* molto completo che racchiude strumenti per l'*editing* sonoro e una gran quantità di effetti (High/Low pass, Distortion, Chorus, Echo, Reverb solo per nominarne alcuni).

Una caratteristica particolare è che in *Unity* gli effetti, le dinamiche e il campo del suono vengono rappresentati visualmente all'interno dell'*editor* e possono essere modificati sempre visivamente.

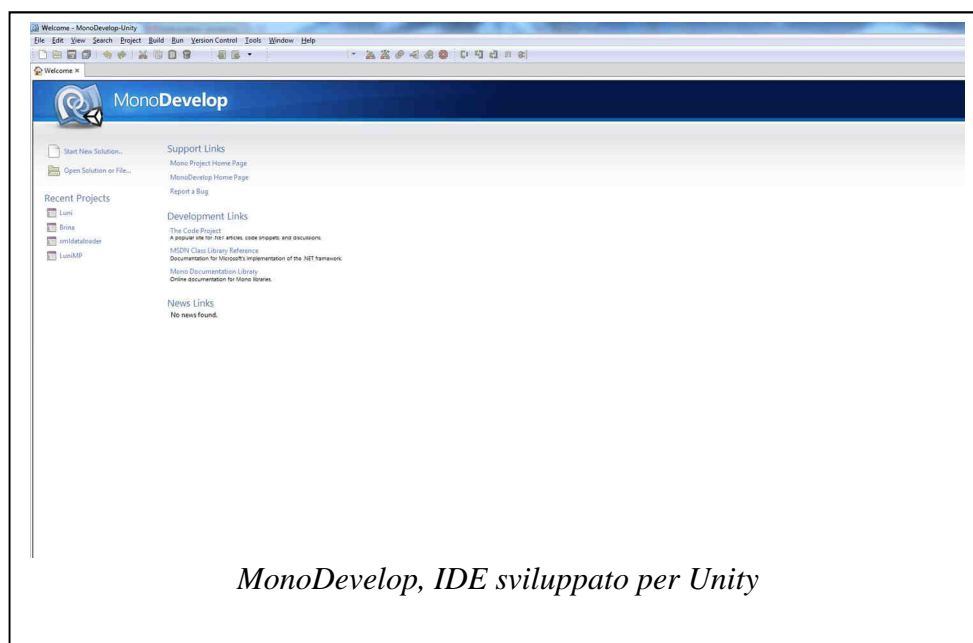
Per il supporto alla rete viene utilizzato Raknet, un potente gestore che permette sia di scambiare dati come classifiche e chat, sia l'interazione tra più giocatori; tramite vari algoritmi vengono scambiati dati compressi su posizione di oggetti, animazioni, suoni...

Utilizzare questo componente è veramente immediato, infatti si può trasformare un progetto da *singleplayer* a *multiplayer* semplicemente usando qualche *script*; tutta la gestione della rete è a carico di *Unity*, bisognerà solo impostare cosa vogliamo sincronizzare e come.

Come già accennato in precedenza *Unity* dispone di un componente chiamato *WebPlayer*: tramite questo *plugin* per *browser* si può effettuare uno *streaming* del progetto, come se si trattasse di un video.

Unity supporta tre linguaggi per lo *scripting* che si interfacciano tutti con le librerie .NET: *Javascript*, *C#*, *Boo*; i linguaggi possono interagire tra loro, quindi si possono creare *script* in linguaggi diversi ed usarli insieme.

L'ambiente di sviluppo è il MonoDevelop, una variante del più famoso Mono, basato su .NET, in modo da avere un IDE completamente integrato.



Gli *script* possono essere associati alle entità semplicemente trascinandoli all'interno dell'*editor*, e nell'*inspector* possiamo vedere tutte le variabili che possono essere cambiate direttamente senza dover modificare lo *script*; il *debugging* avviene direttamente all'interno dell'*editor* e il codice può essere corretto "al volo".

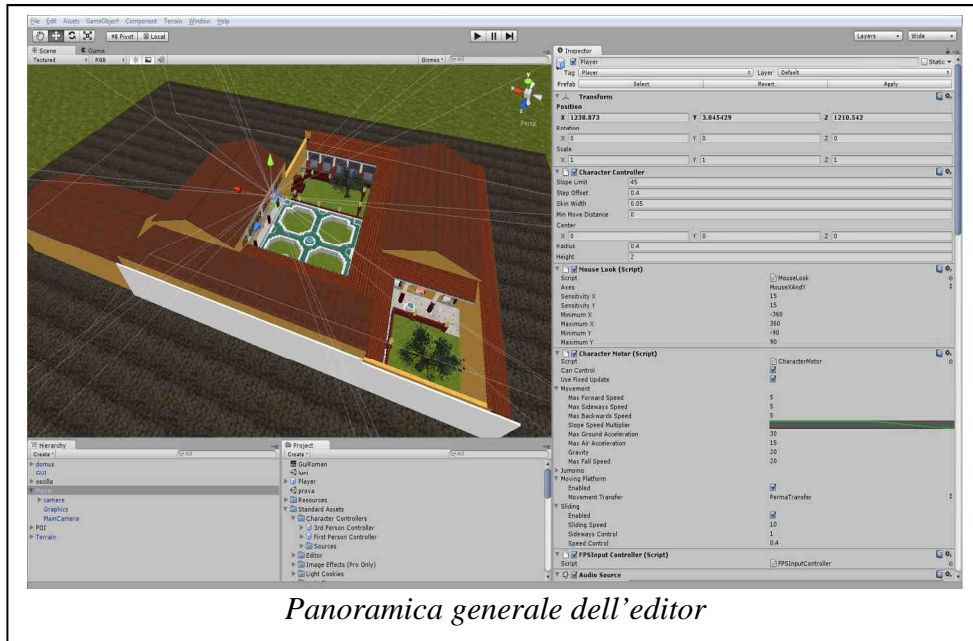
Anche il passaggio tra le diverse versioni dell'*engine* è indolore: se si prova ad aprire un vecchio progetto con una nuova versione di *Unity* questo viene automaticamente aggiornato.

La documentazione fornita con *Unity* è molto chiara e dettagliata, e attualmente il *forum* della compagnia gode di una commUnity molto frequentata; si trovano facilmente progetti di esempio e numerosi video e tutorial che spiegano le tecniche da utilizzare.

Dopo questa panoramica su *Unity* che aiuta a capire il perché sia stato scelto vediamo nel dettaglio i componenti di cui bisogna aver maggior conoscenza per l'utilizzo della nostra piattaforma.

3.1-Editor

L'editor, come già accennato in precedenza, è la nostra finestra sul mondo del gioco, da qui possiamo modificare visivamente la scena, spostare oggetti, aggiustare le fonti di illuminazione e così via.



Panoramica generale dell'editor

Esistono diversi *layout*, ma hanno tutti quattro schede principali:

1. la prima è la scheda di **visualizzazione della scena**, dove vediamo gli oggetti, li spostiamo e modifichiamo.

Quello che rende particolare questa finestra in *Unity* è la *live preview*, che facilita il lavoro di correzione e revisione del progetto, facendo vedere in tempo reale cosa stiamo effettivamente modificando.

2. L'inspector è la parte dell'editor che ci permette di ispezionare l'oggetto selezionato, mostrando tutte le informazioni: quali *script* sono associati, che valori hanno, le coordinate spaziali, la rotazione e la scala rispetto al mondo.

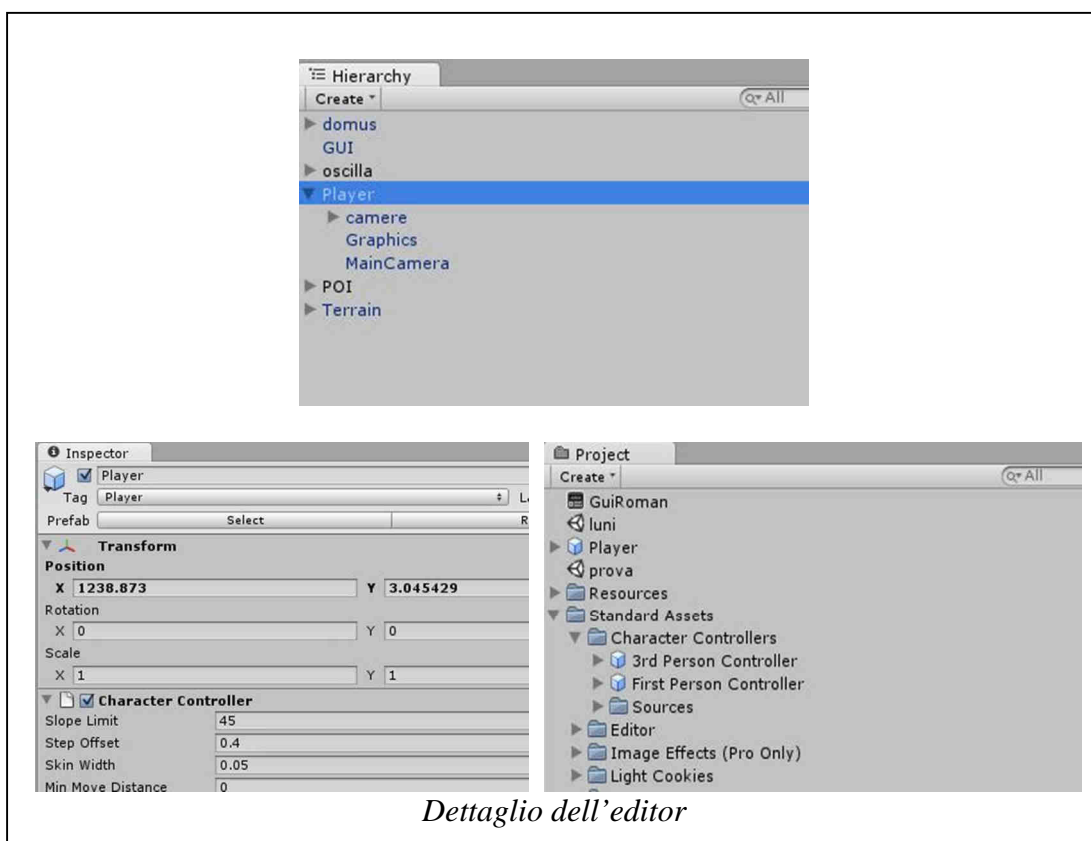
C'è anche la possibilità di trasformare l'oggetto in un *prefab*.

I *prefab* sono molto importanti e semplificano la creazione di scene con entità multiple molto complesse.

Dopo aver creato un *prefab*, questo può essere posizionato molte volte all'interno della scena; se viene fatta una modifica ad uno di questi *prefab*, automaticamente vengono modificati tutti i *prefab* posizionati, ed il *prefab* predefinito, permettendo così di modificare molte identità cambiandone una sola.

3. La scheda **Project** permette di vedere tutti i file che abbiamo importato nella cartella del progetto, le loro proprietà e la struttura delle cartelle e dei file.
4. L'ultima scheda è quella della **Hierarchy**, che come dice il nome mostra la gerarchia del progetto, mostrando gli oggetti in gioco e come sono organizzati tra loro.

Per finire in alto abbiamo gli strumenti base che permettono di spostare, ruotare e scalare gli oggetti in gioco; nei menù invece si trovano vari strumenti, da quelli per la creazione dei *terrain* fino a quelli per l'inserimento di *skybox* (metodo per creare lo sfondo dei livelli di gioco in maniera da far sembrare l'ambiente molto più grande) e luci.



3.2-Terrain

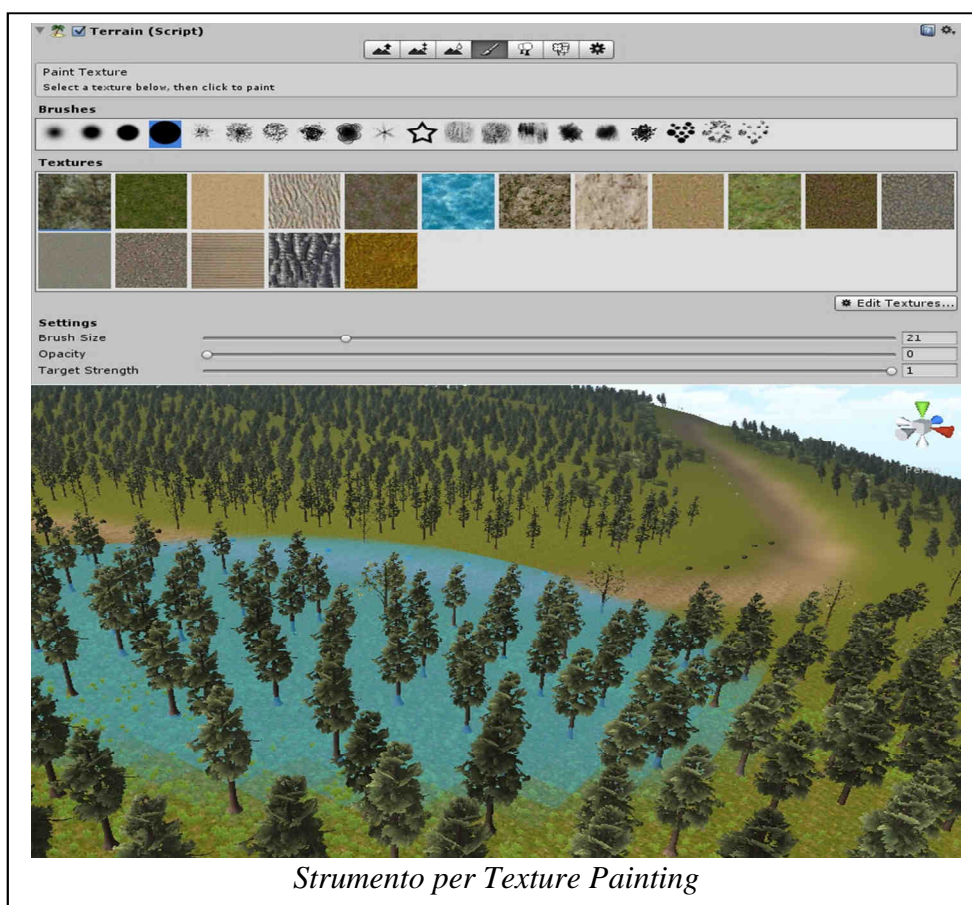
Un *terrain* è un'ambientazione, lo scenario del nostro ambiente di gioco.

In *Unity* vengono forniti gli strumenti per creare da zero uno scenario, tramite strumenti dedicati alla modifica e texturizzazione del terreno.

Gli strumenti si presentano sotto forma di “pennelli” che sono utilizzati per alzare o abbassare il terreno, per il *Texture Painting* (l'applicazione della *texture* come se si dipingesse direttamente sul modello) e per distribuire vegetazione o altri elementi.

Le *lightmaps* per il *terrain* vengono generate automaticamente, senza bisogno di doverle disegnare.

Nella nostra piattaforma è presente uno scenario di prova, in modo da poterla testare fin da subito.



Strumento per Texture Painting

3.3-Trigger

Il meccanismo che fa sì che possano avvenire gli eventi all'interno della scena (personaggio che parla, suono che viene riprodotto) è chiamato *trigger*.

In linea di massima è costituito da un solido non renderizzato di cui si conserva la geometria per rilevare le collisioni.

In generale si parla di *trigger system*, per intendere tutto il sistema di azioni e reazioni collegato ad un *trigger*.

Concettualmente un *trigger system* è un sistema che a partire da un'azione innesca una reazione programmata; questo tipo di azione può essere singola e semplice (l'utente preme un tasto) oppure complessa e legata a più fattori (se la luce della scena è sotto la soglia stabilita, l'utente preme un tasto ed è in riproduzione un file audio..).

Nel nostro caso i *trigger* sono limitati ai *task* utili ai nostri fini, ovvero alla rilevazione delle collisioni, che nel nostro caso sono relative al contatto tra il giocatore e geometrie invisibili (come le *bounding box* dei personaggi).

I *task* di base di una ricostruzione prevedono sempre che si “tocchi” con il giocatore un altro oggetto nel mondo di gioco, che sia un muro o un personaggio; il nostro *trigger* prevede tutte le condizioni in cui il giocatore viene a contatto con una certa entità, permettendo di abilitare gli *script* collegati ai *task*.

3.4-Prebab

I *Prefab* sono una delle caratteristiche che contraddistinguono *Unity*.

Se si ha all'interno della scena un oggetto molto complesso, sia graficamente che a livello di codice, ricreare lo stesso oggetto molte volte in maniera identica o simile è dispendioso.

Con i *prefab* invece possiamo rendere un oggetto predefinito, in modo che tutte le sue caratteristiche vengano salvate all'interno di un file nella cartella del progetto; si crea un'istanza dell'oggetto.

Quando vorremo usare di nuovo quell'oggetto basterà trascinarlo nella scena e replicarlo quante volte vogliamo.

Inoltre i *prefab* relativi ad un certo oggetto sono tutti collegati tra loro, in maniera che se viene apportata una modifica ad uno di essi, questa si ripercuote su tutti, evitando di doverla effettuare manualmente.

Possono essere importati in qualunque progetto o scena, visto che tutto quello che riguarda un *prefab* viene memorizzato insieme ad esso.

Nella nostra piattaforma si è scelto di usare i *prefab* di tutte le entità principali, per la loro semplicità così da diminuire la complessità per l'utente finale.

4-Sviluppo della piattaforma

Dopo aver affrontato più ricostruzioni di tipo storico su *game engine*, ci si è resi conto di come, dal punto di vista tecnico, la realizzazione richieda la risposta ad alcune esigenze costanti, indipendenti dall'oggetto che si vuole realizzare.

Si è così formata l'idea di creare una piattaforma, basata su *game engine*, che permetta anche a chi non possiede competenze sufficienti nel campo della programmazione di sviluppare progetti simili, basandosi su un approccio molto semplice e totalmente visuale, senza dover scrivere codice.

L'unica cosa da definire sono gli eventi veri e propri, che l'utente descrive all'interno di un semplice file xml.

Le fasi d'utilizzo quindi sono due: la prima consiste nella disposizione dei modelli 3d, dei personaggi e dei *prefab* predefiniti all'interno della scena; la seconda vede impegnato chi ricostruisce a "scrivere la storia" della propria scena in un file xml.

Si è dapprima condotto uno studio su quali task andassero effettivamente implementati per i nostri scopi, e quali invece riguardassero un'utenza generale (*task di base*).

Tra le numerose fasi del progetto; la prima, e forse la più importante e complessa è stata quella di codifica di tutti gli *script* per i task di base.

Gradualmente, per ogni task, si sono analizzati gli eventi da riprodurre e le situazioni che potevano essere interessanti.

Alcuni task riguardano le azioni generiche:

- **Animazione**, permette di far animare un personaggio con le animazioni assegnate;
- **Parla**, riproduce un file audio per far "parlare un personaggio" o un oggetto in scena;
- **Muovi**, sposta un personaggio lungo un percorso personalizzato;
- **IllustraPOI**, sposta la visuale del giocatore sul punto di interesse e fa apparire un'interfaccia con informazioni e foto definite dall'utente;

- **Musica**, variante di Parla, utilizzata per mettere una musica diffusa in tutta la scena.

Per la ricostruzione di Luni in particolare sono stati implementati i POI (Point Of Interest), ovvero dei punti nei quali, avvicinandosi all'oggetto interessato o ad un personaggio, è possibile ricevere informazioni direttamente sull'interfaccia, e si ha anche la possibilità di sfogliare un album con delle immagini.

Oltre che per i task si sono definiti *script* per dare all'utente un'interfaccia predefinita, in modo da offrire un'usabilità molto maggiore dell'applicazione. Dopo aver definito tutti gli *script* si sono creati gli elementi per costituire la scena: un *terrain* di base predefinito, vari *gameobject* e materiale di esempio. Una volta ultimate tutte le entità aggiuntive si sono creati i *prefab* per la GUI, per il giocatore, per i *trigger* e per i POI.

Terminato questo processo si è passati alla creazione di un progetto *Unity* standard che racchiudesse tutto ciò, così da offrire uno strumento già pronto per essere utilizzato.

Sempre all'interno del progetto è racchiusa una cartella "Other" dove è possibile inserire la documentazione e qualunque file sia utile o sia stato utilizzato per la realizzazione della ricostruzione.



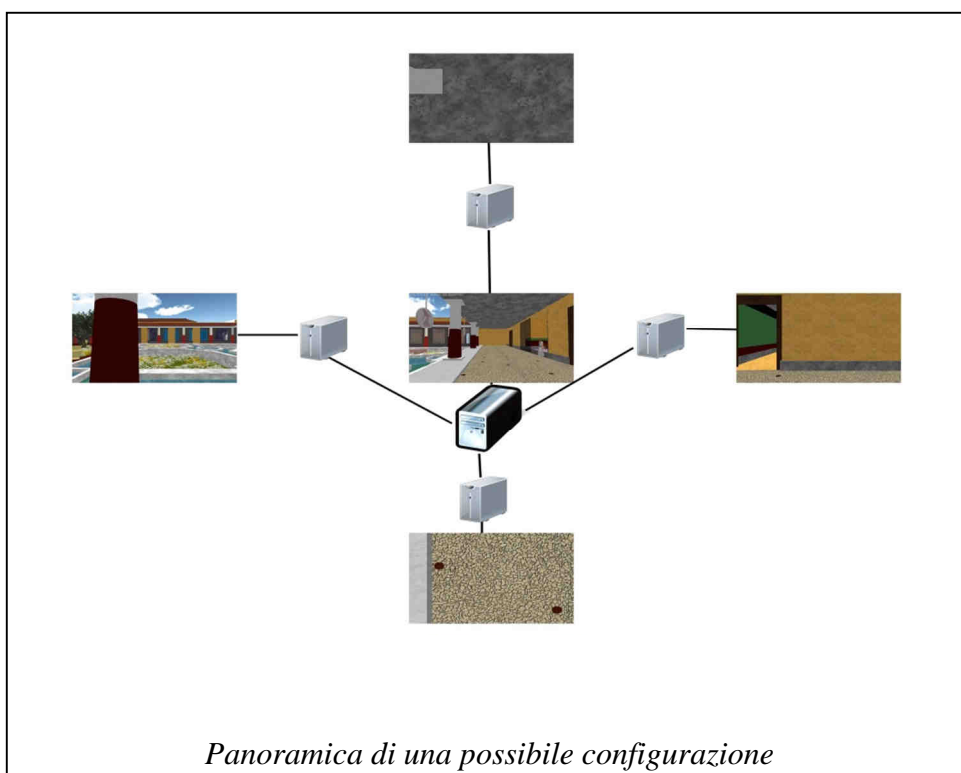
Se qualcuno avesse un progetto già esistente questo modello generico è integrabile senza problemi, visto che è stata creata anche la versione package, un archivio particolare riconosciuto da *Unity* che racchiude il progetto con tutti gli elementi e le loro proprietà.

Visto che le ricostruzioni storiche sono uno degli argomenti più utilizzati all'interno degli ambienti virtuali è stato ritenuto molto utile sviluppare la piattaforma in maniera da poter essere utilizzata all'interno di ambienti multischermo dedicati alla realtà virtuale come i CAVE.

L'approccio, come accennato in precedenza, è impostato sul calcolo distribuito delle visuali.

In parole povere si è creata un'infrastruttura *multiplayer* (*script MultiMenu* mostrato più avanti), dove abbiamo una macchina principale, che fa da Host per le altre macchine e controlla il giocatore, e una serie di macchine secondarie (fino a 6 compreso l'host, se si vogliono visualizzare tutte viste possibili) che si collegano avendo la possibilità di osservare la scena da una telecamera scelta, che può essere a destra, sinistra, dietro, sopra o sotto il giocatore.

Il codice in questione è stato sviluppato anche grazie all'utilizzo di un componente particolare di *Unity*, chiamato "NetworkView"; grazie a questo componente è stato possibile sincronizzare il movimento del giocatore tra le macchine, in modo da spostare le visuali su un oggetto solidale a tutte le telecamere.

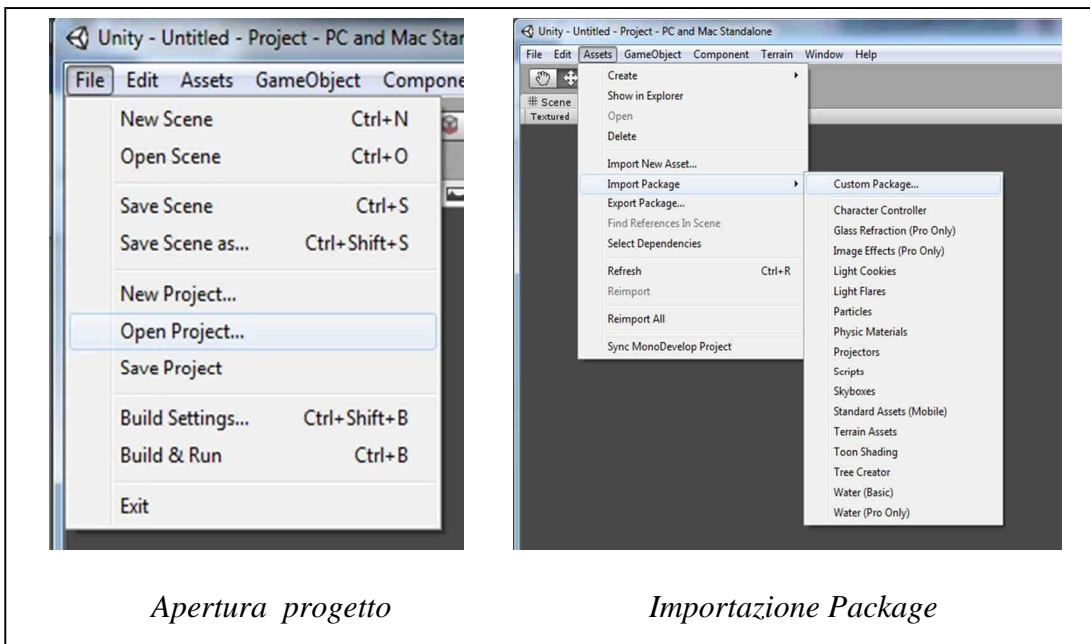


5-Guida all'utilizzo

Vediamo ora nel dettaglio le modalità d'uso della piattaforma sviluppata; passo per passo si analizzeranno gli elementi nel dettaglio e poi si fornirà una panoramica generale per vedere come effettivamente dovranno essere utilizzati all'interno della scena.

Come già detto ci sono due fasi di utilizzo: la prima è quella di disposizione dei nostri modelli 3d all'interno della scena e dei *prefab* generici; la seconda, che vedremo dopo, è la stesura del foglio xml con gli eventi della scena.

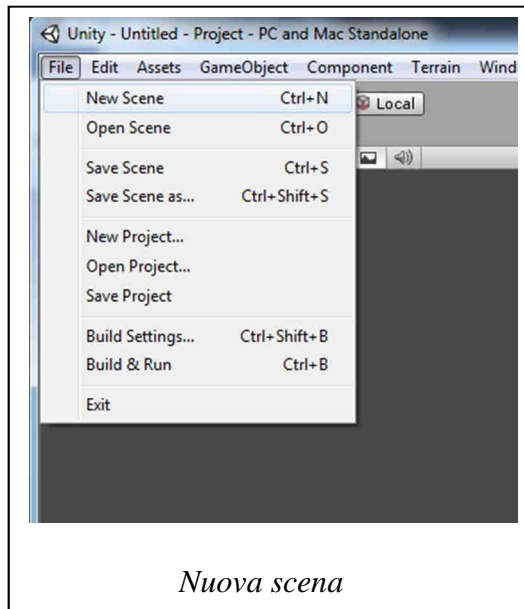
Incominciamo con l'impostare il nostro progetto; possiamo aprire direttamente in *Unity* il progetto predefinito



oppure caricare il file package se abbiamo un progetto già iniziato

Ora copiamo nella cartella "Resources" che troviamo nella "Assets" del progetto tutti i nostri file audio, le immagini e i modelli 3d.

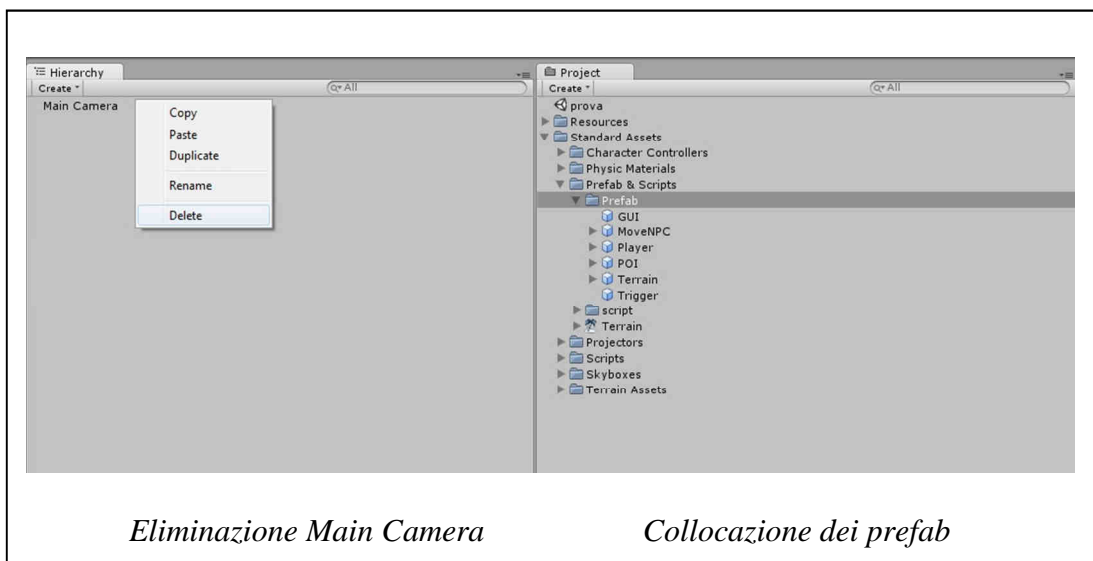
Procederemo con la creazione di una nuova scena, ma tutto quello che viene descritto può essere effettuato anche in scene già esistenti.



Nuova scena

Cominciamo con l'eliminare tutte le telecamere già presenti nella scena, come ad esempio la MainCamera assegnata alle nuove scene, altrimenti il funzionamento dell'audio potrebbe essere compromesso (*Unity* non tollera due entità che possono ascoltare l'audio di gioco).

Analizziamo i *prefab*, che si trovano nella cartella "Standard Assets\Prefab & Scripts\Prefab".



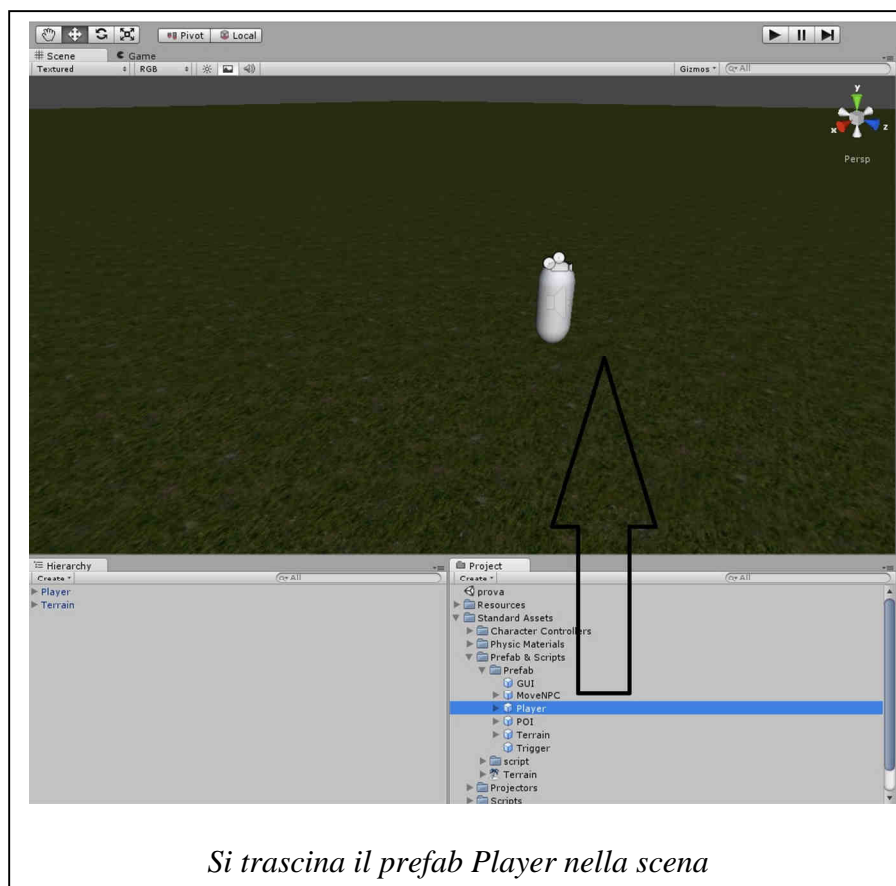
Eliminazione Main Camera

Collocazione dei prefab

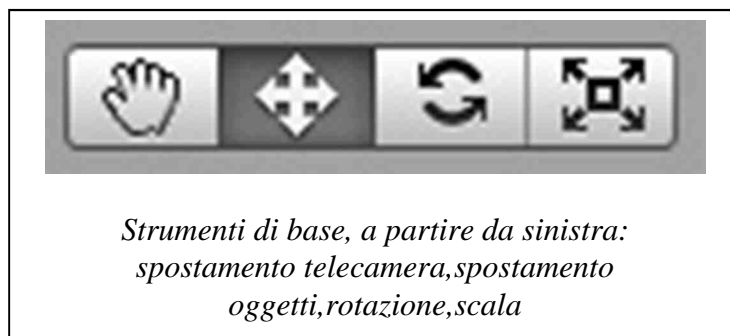
Se non si ha dimestichezza con la creazione di *Terrain* è possibile utilizzare il *prefab* relativo.

Due *prefab* sono fondamentali e devono essere inseriti nella scena, mentre gli altri variano in base alle esigenze; stiamo parlando di "GUI" (*prefab* che gestisce

tutta la GUI, dall'interfaccia multischermo al menù di pausa) e "Player" (il nostro giocatore, l'entità che permette di visualizzare e muoversi nel mondo). L'inserimento è molto semplice, basta trascinare quello che si vuole direttamente nel punto interessato della scena.



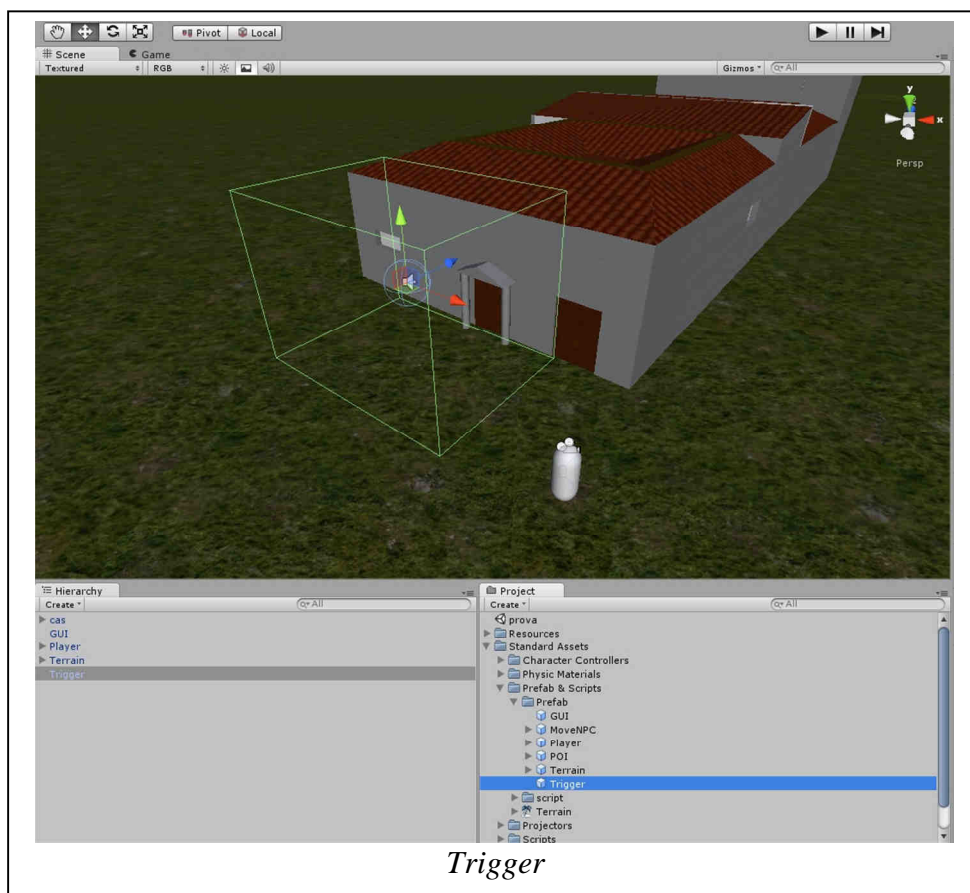
A questo punto, una volta disposti i *prefab* di base e i propri modelli 3d, e dopo averli ridimensionati e ruotati secondo le proprie preferenze tramite gli strumenti di base



si passa alla disposizione dei *trigger*, gli “interruttori” che abiliteranno gli eventi nella nostra scena.

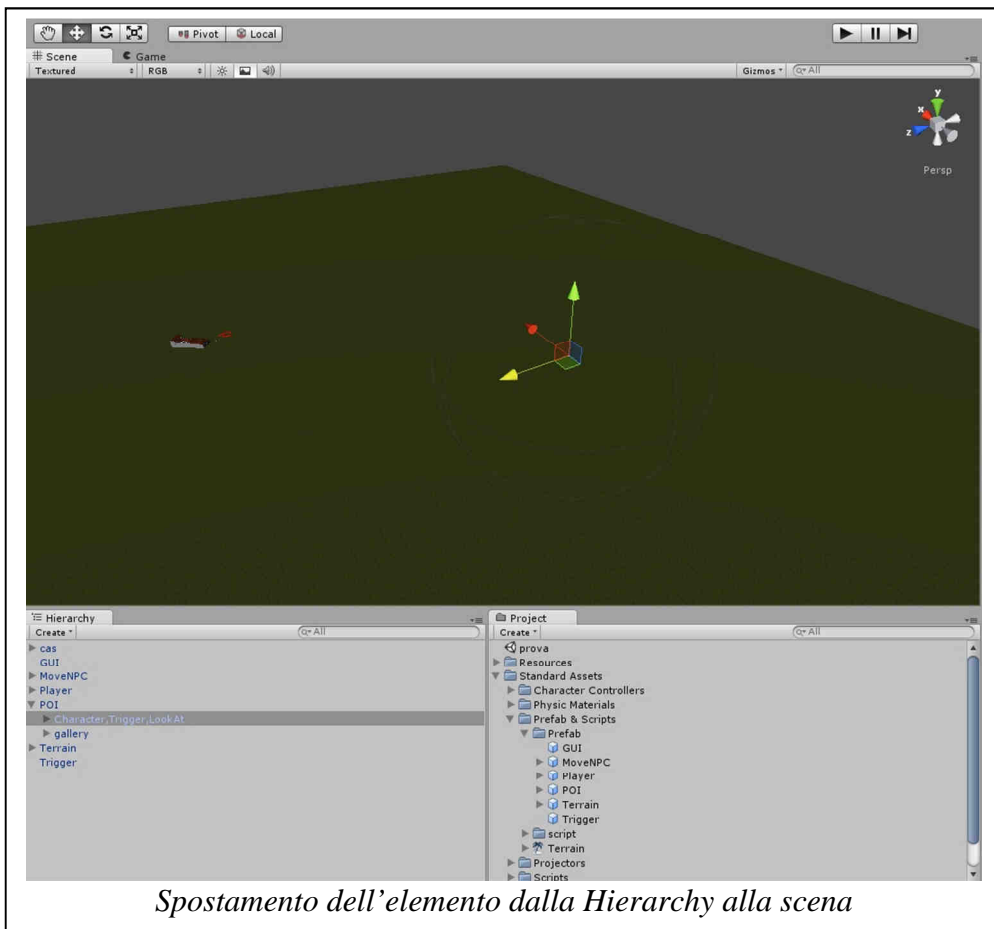
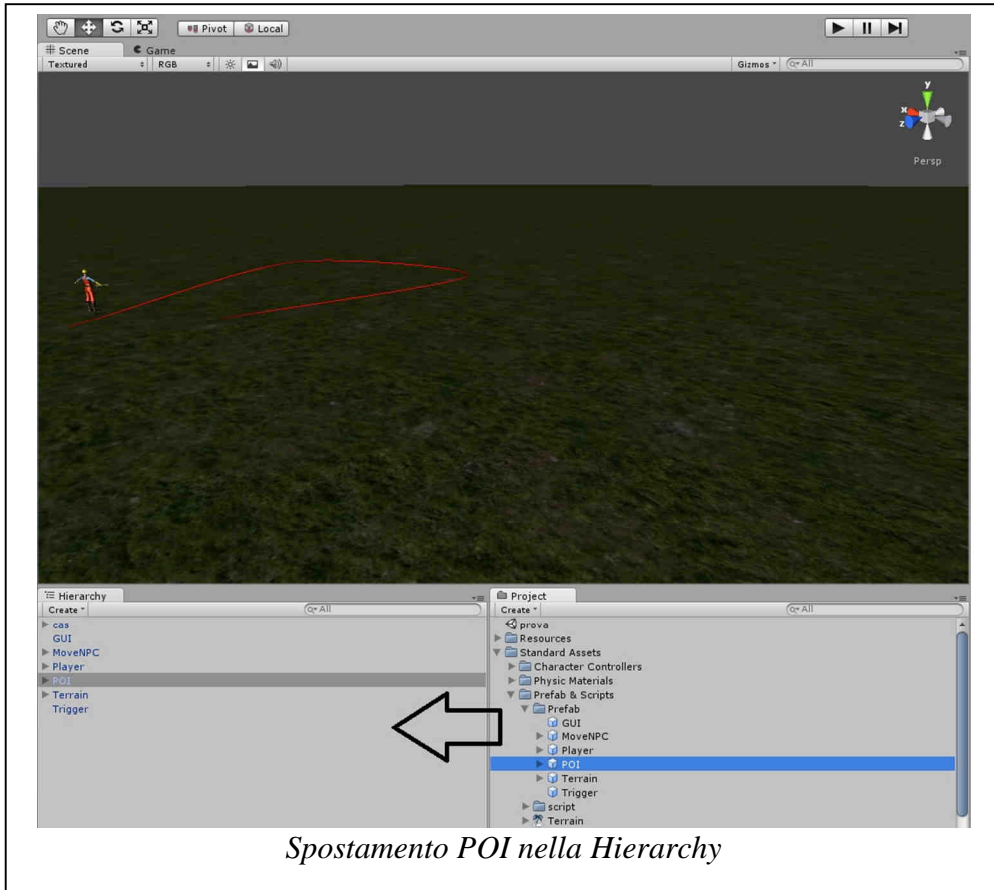
Nell’esempio, disponiamo il *prefab* “*Trigger*” davanti alla porta di una casa, perché vogliamo abilitare un evento (non importa se un suono, un’animazione o un movimento, si tratta sempre della stessa entità) quando il giocatore si avvicinerà alla casa.

Il cubo verde rappresenta l’area in cui l’evento può essere attivato.



A questo punto non resta solo che rinominare il *trigger* come preferiamo (click destro>rename) e definire gli avvenimenti all’interno del foglio xml, ricordandosi che “Nome*Trigger*” è il nome appena assegnato.

Per i punti di interesse la procedura è leggermente diversa: invece di spostare il *prefab* “*POI*” direttamente nella scena si deve spostare all’interno della Hierarchy; successivamente si sposta l’entità “*Character,Trigger,LookAt*” all’interno di *POI* nel punto desiderato.



Se si vuole spostare anche il punto dove viene spostato lo sguardo del giocatore basterà muovere l'oggetto "LookAt"

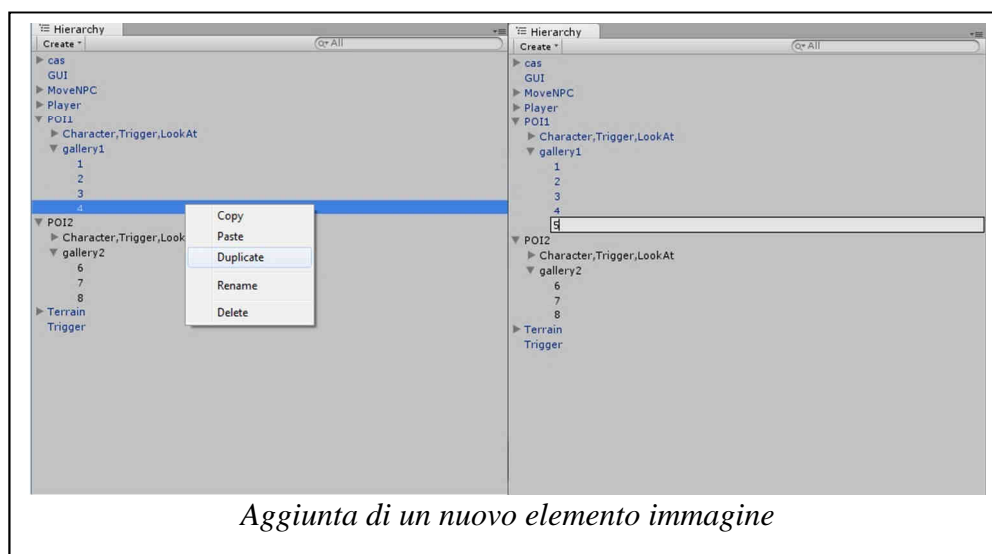
Ci potrebbe essere la necessità di cambiare l'NPC di base con un altro NPC o con un oggetto; per farlo basta trascinare il nostro personaggio all'interno della scena, cancellando quello predefinito.

Per impostare la galleria, ci si sposta sul rispettivo oggetto che si trova dentro POI; dentro troveremo vari oggetti aventi come nome un numero progressivo.

Se si vuole aggiungere altre immagini alla galleria basterà creare delle copie di questi oggetti e rinominarli con numeri crescenti.

Se abbiamo più gallerie useremo numero crescenti sempre diversi, quindi ad esempio, potremo avere una galleria1 con le immagini 1,2,3,4,5 e la galleria 2 con le immagini 6,7,8.

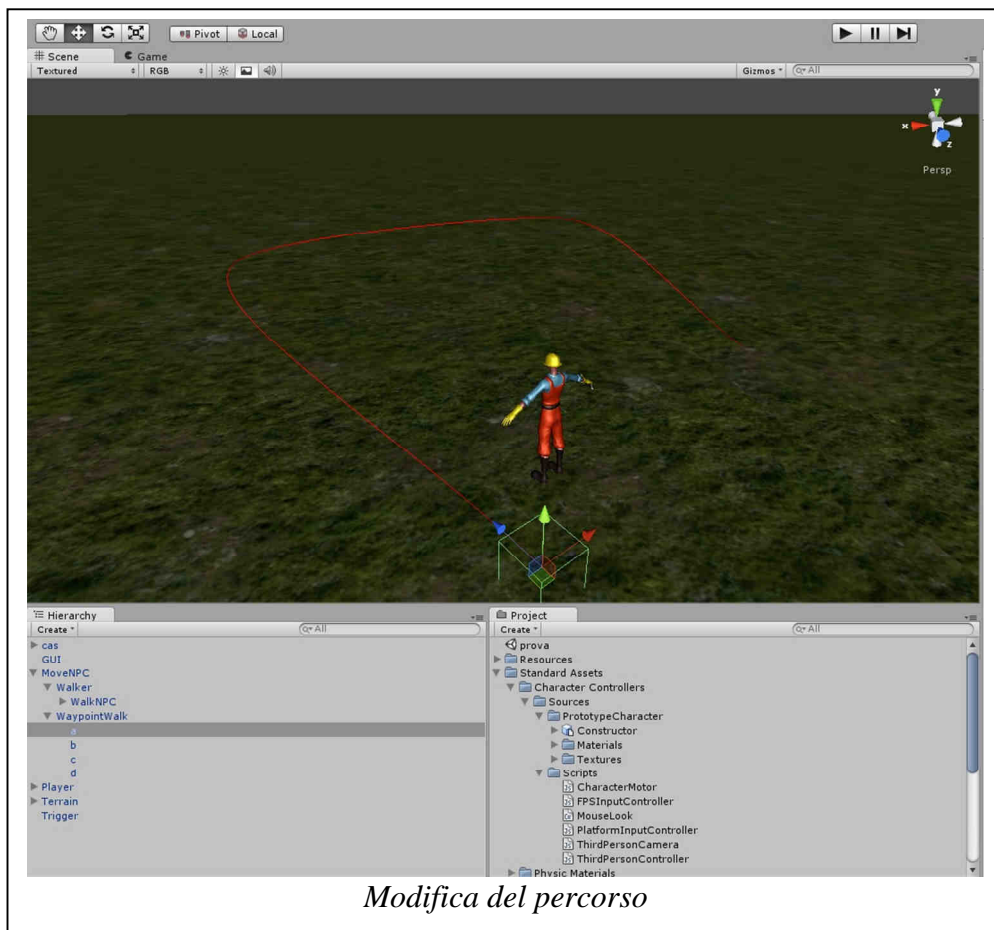
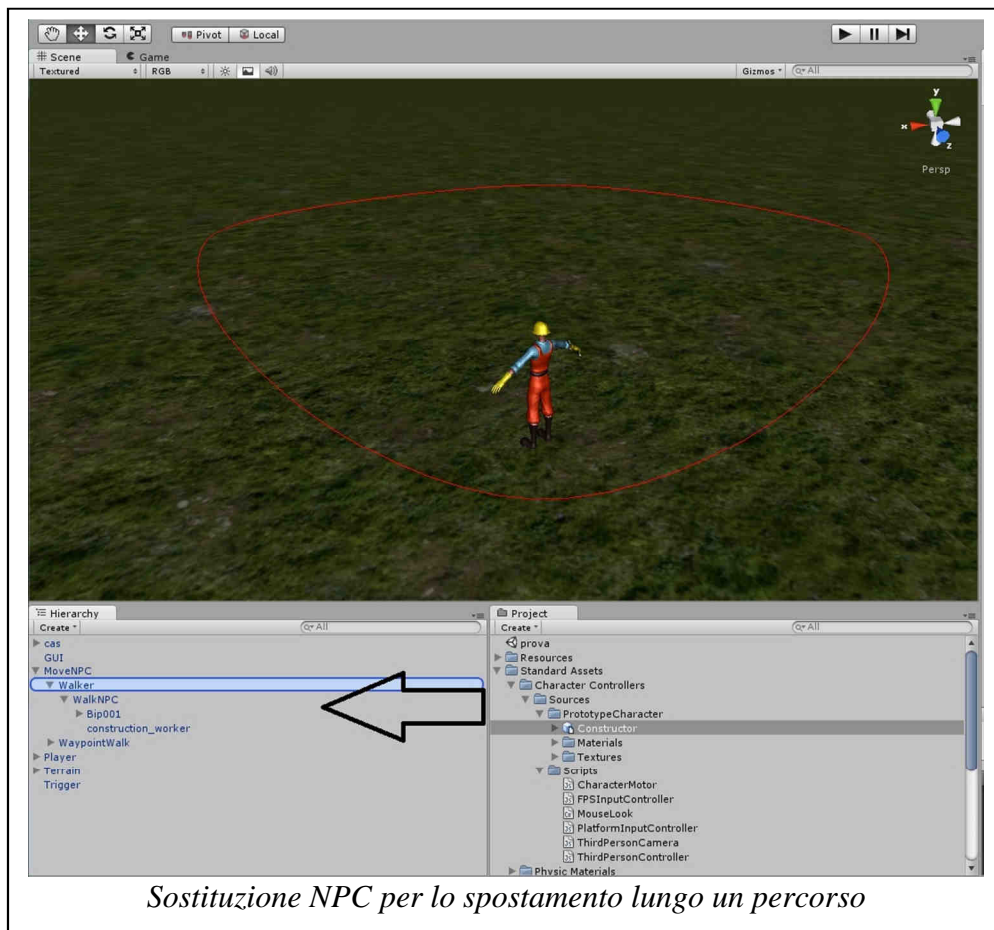
Nell'esempio vediamo l'aggiunta di un'ulteriore immagine.



L'ultimo *prefab* da analizzare è *MoveNpc*, che permette far percorrere ad un personaggio un percorso deciso da noi.

Per sostituire il personaggio predefinito basta trascinare il nostro NPC all'interno dell'entità "Walker" (che chiaramente può essere rinominato).

A questo punto possiamo definire il percorso che vogliamo; l'entità da modificare è "WaypointWalk" che è costituita da oggetti disposti in ordine alfabetico.



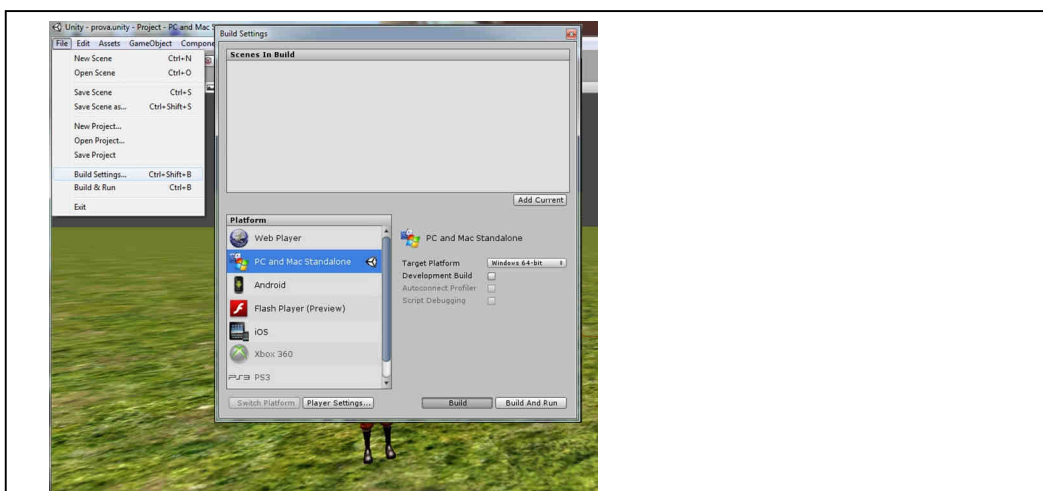
E' possibile aggiungere quante "lettere" vogliamo, a seconda di quanto complicato deve essere il percorso; ora possiamo spostare gli oggetti facendogli seguire il percorso che ci interessa.

L'unica raccomandazione è quella di stare sempre molto attenti ai nomi dati ai personaggi e alle entità in modo da associarli correttamente all'interno del file xml.

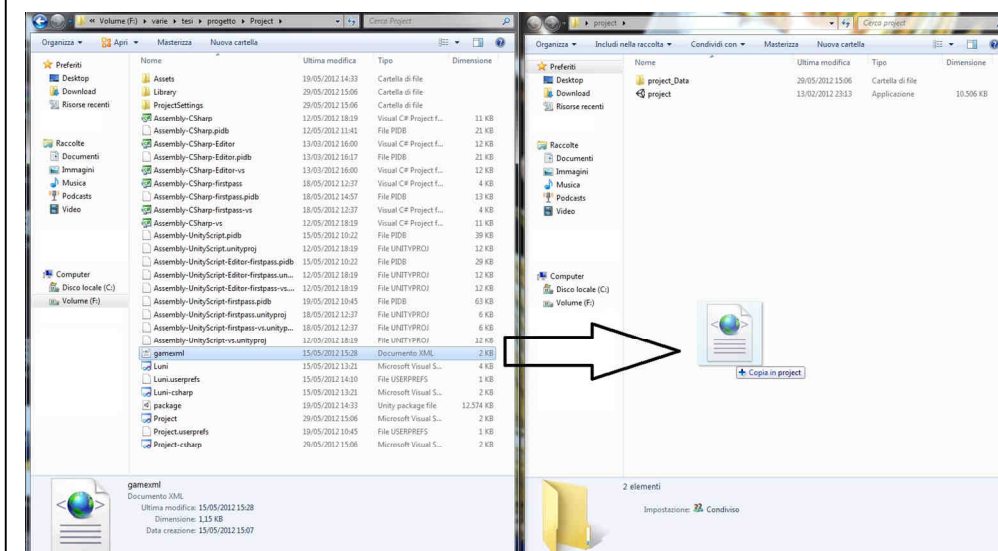
Una volta settata la scena, basterà creare il file xml, tra poco vedremo come.

Una volta ultimato il progetto procederemo a creare l'eseguibile, come mostrato nell'immagine.

Quando avremo la cartella con il nostro eseguibile basterà copiarvi dentro il file "gamexml.xml" creato in precedenza.



Creazione dell'applicazione finale



Copia del file xml

5.1- Foglio XML

Il foglio xml permette all'utente di descrivere la scena come se fosse una lista di avvenimenti, permettendogli di non dover settare manualmente la scena.

Il file si trova nella directory principale del progetto con il nome di "gamexml.xml"; può essere modificato con qualunque *editor* di testo.

I gruppi e le azioni utilizzabili sono:

Scenario: identifica lo scenario in cui avviene la ricostruzione. Ha come attributi "Nome" che definisce il nome.

Scena: identifica la scena, per rendere più leggibile il foglio, e per racchiudere gli avvenimenti in un certo contesto. Anche qua abbiamo l'attributo "Nome" che definisce il nome.

Musica: permette di inserire musica, e al contrario di Parla, il suono non è localizzato ma è diffuso su tutta la scena.

Animazione: riproduce un animazione su un npc. "NomeTrigger", "NomeNpc", "AnimazioneEntrata" e "AnimazioneUscita" stabiliscono rispettivamente il nome del *trigger* che attiva l'animazione, il nome dell'npc che la riproduce, il nome dell'animazione da riprodurre in entrata e in uscita.

Parla: riproduce un file audio localizzato che permette di far "parlare" un personaggio. Gli attributi sono "NomeTrigger", "NomeNpc", "NomeAudio". Il primo e il secondo assolvono sempre lo stesso compito che hanno in Animazione, mentre l'ultimo stabilisce il nome del file audio da riprodurre.

Muovi: sposta un oggetto o un personaggio lungo il percorso settato precedentemente. Troviamo sempre "NomeTrigger" per definire il *trigger* che fa partire il movimento, "Chi" per definire chi o cosa deve essere mosso,

“Percorso” per assegnare il nome del percorso da seguire e “DurataSecondi” che stabilisce quanto l’entità ci debba mettere a percorrere il percorso.

Per quanto riguarda i punti di interesse abbiamo:

IllustraPOI: sposta la visuale del giocatore sull’oggetto di interesse. Come attributi troviamo “NomeTrigger” che si riferisce al *trigger* appartenente al POI e “POI” dove va inserito il nome del POI su cui spostare la visuale.

Informazioni: si stabilisce il testo da inserire nelle informazioni aggiuntive del punto di interesse. Troviamo “POI” per stabilire a cosa si va ad assegnare il testo; abbiamo a disposizione due pagine, identificate da “TestoP1” e “TestoP2”.

GalleriaFoto: permette di settare i parametri della nostra galleria di immagini. “NomeGallery” assegna il nome della galleria, “POI” il nome del punto di interesse a cui la galleria è assegnata, mentre “NumeroPrimaImg” e “NumeroUltimaImg” si riferiscono al numero con cui inizia la galleria e a quello con cui finisce.

Img: assegna i file immagine ai gameobject che servono per visualizzare le foto all’interno della GUI. Con “NomeFileImg” si stabilisce il nome del file immagine, e con “NumeroGameObject” il nome (che come detto dovrà corrispondere ad un numero)

Nella prossima pagina vediamo un esempio di file Xml ben strutturato.

```

<?xml version="1.0" encoding="utf-8" ?>

<Scenario Nome= "Terrain">

  <Musica NomeAudio= "music" />

  <Scena Nome= "provaTRIGGER">

    <Animazione NomeTrigger="Trigger" NomeNpc="WalkNPC"
    AnimazioneEntrata= "run" AnimazioneUscita= "idle" />

    <Parla NomeTrigger="Trigger" NomeNpc="WalkNPC" NomeAudio=
    "frase1" />

    <Muovi NomeTrigger="Trigger" Chi= "Walker" Percorso="WaypointWalk"
    DurataSecondi="20" />

  </Scena>

  <Scena Nome= "provaPOI">

    <IllustraPOI NomeTrigger="triggerPOI" POI= "LookAt"/>

    <Animazione NomeTrigger="triggerPOI" NomeNpc="character"
    AnimazioneEntrata= "run" AnimazioneUscita= "idle" />

    <Parla NomeTrigger="triggerPOI" NomeNpc="character" NomeAudio=
    "frase2" />

    <Informazioni POI= "LookAt" TestoP1="prova" TestoP2="Prova2"/>

    <GalleriaFoto NomeGallery= "gallery" NumeroPrimaImg= "1"
    NumeroUltimaImg= "4" POI= "LookAt" >

    <Img NomeFileImg= " provaimg 1" NumeroGameObject= "1" />
    <Img NomeFileImg= " provaimg 2" NumeroGameObject= "2" />
    <Img NomeFileImg= " provaimg 3" NumeroGameObject= " 3" />
    <Img NomeFileImg= " provaimg 4" NumeroGameObject= "4" />

  </GalleriaFoto>

```

6-Script

Vediamo nel dettaglio gli *script* sviluppati per il progetto, prima con una breve spiegazione per ogni *script* e poi con i commenti all'interno del codice.

Gli *script* sono stati sviluppati in JavaScript con l'IDE MonoDevelop integrato con Unity.

Nei casi degli *script* "SmoothLookAt" e "SplineController", già compresi in Unity e scritti in C#, si è provveduto alla modifica e all'adattamento per la piattaforma.

Si possono individuare varie categorie di *script*: per gestire *trigger*, per la visualizzazione della GUI, per la gestione della visione su più macchine, per *script* di supporto e l'interprete xml.

6.1 Gestione dei *trigger*

Spline On Trigger

Attiva il componente SplineController sull'oggetto o personaggio che deve percorrere il percorso assegnatogli se rileva la collisione tra Player e il *trigger*.

Più avanti verrà mostrato in dettaglio lo *script* SplineController.

```
var SplineControllerObject:GameObject;

var on=true;

function OnTriggerEnter(collision: Collider)
{
    //si riproduce il suono assegnato all'oggetto quando il giocatore collide con il suo collider

    if (collision.tag == "Player")
    {
        SplineControllerObject.GetComponent("SplineController").enabled=true;
    }
}
```

Animation On Trigger

Muove il personaggio definito dalla variabile `animateobject` all'entrata e all'uscita del *trigger*, con le animazioni `AnimationOnEnter` e `AnimationOnExit`

```
var animateobject: GameObject;  
  
//l'oggetto da animare  
  
var AnimationOnEnter="";  
  
//nome dell'animazione in entrata del trigger  
  
var AnimationOnExit="";  
  
//nome dell'animazione in uscita del trigger  
  
function OnTriggerEnter (collision : Collider)  
{  
    if (collision.gameObject.tag == "Player")  
  
//se avviene il trigger con l'oggetto player riproduce l'animazione definita prima  
    {  
        animateobject.animation.CrossFade (AnimationOnEnter);  
    }  
}  
function OnTriggerExit (collision : Collider)  
{  
    if (collision.gameObject.tag == "Player")  
    {  
        animateobject.animation.CrossFade (AnimationOnExit);  
    }  
}
```

Sound On Trigger

Riproduce un suono definito dalla variabile `AudioSource` quando il giocatore (Player) collide con il *trigger* a cui questo *script* è assegnato

```
var Audiosource:AudioSource;  
  
function OnTriggerEnter(collision: Collider){  
  
//si riproduce il suono assegnato quando il giocatore collide con il trigger  
    if (collision.tag == "Player"){  
        Audiosource.audio.Play();  
    }  
}
```

6.2 GUI

Hide Cursor On Start

Nasconde semplicemente il cursore all'avvio dell'applicazione, visto che durante l'uso dell'applicazione non si deve vedere.

```
function Start(){  
  
    //nasconde il cursore all'avvio dell'applicazione  
  
    yield WaitForSeconds(1);  
  
    GameObject.Find("GUI").GetComponent("Cursor").showingCursor=true;
```

Cursor

Definisce la visibilità del cursore, che è attivato o disattivato a seconda se appaia una GUI con cui interagire oppure no.

```
var showingCursor = true;  
  
//variabile che definisce se il cursore è visibile o no  
  
function Update(){  
  
    //nasconde o mostra il cursore  
  
    if(showingCursor == false){  
        Screen.showCursor = false;  
        showingCursor = false;  
    }  
  
    else if(showingCursor == true){  
        Screen.showCursor = true;  
        showingCursor = true;  
    }  
}  
}
```

Questi tre *script* sono legati tra loro, in quanto gestiscono insieme la pausa dell'applicazione, che avviene con il tasto esc.

Il primo rileva se è stato premuto il tasto ESC, disabilita il giocatore e abilita il secondo che fa apparire una GUI con due opzioni: Play per continuare e Esci per chiudere l'applicazione (in questo caso viene attivato Quit).

ESC

```
function Update () {  
  
  //gestisce il menù di pausa  
  
  if(Input.GetKeyDown(KeyCode.Escape)){  
  
    //se si preme il tasto esc  
  
    GameObject.Find("Player").GetComponent("EnablePlayer").enable=false;  
  
    //disabilita il giocatore  
  
    GameObject.Find("GUI").GetComponent("PauseGUI").on=true;  
  
    //abilita il menù di pausa  
  
  }  
}
```

Quit

```
var on=false;  
  
function Update () {  
  
  //se lanciata termina l'applicazione  
  
  if(on) Application.Quit();  
  
}
```


Pause GUI

```
var on=false;

//variabile che stabilisce se lo script è attivo o no

function OnGUI(){

//menù pausa

    if (on){

        var SHCursor= GameObject.Find("GUI").GetComponent("Cursor");

        //si assegna alla variabile lo script che gestisce il cursore

        SHCursor.showingCursor=true;

        //si imposta il cursore come visibile

        if (GUI.Button (Rect (Screen.width/2, Screen.height/3 ,100,60), "Play")) {

            //pulsante play

            SHCursor.showingCursor=false;

            //si nasconde il cursore

            GameObject.Find("Player").GetComponent("EnablePlayer").enable=true;

            //si ri-abilita il giocatore

            on=false; //si disabilita lo script stesso

        }

        if (GUI.Button (Rect (Screen.width/2, Screen.height/2 ,100,60), "Esci")) {

            //pulsante esci

            GameObject.Find("GUI").GetComponent("Quit").on = true;

            //si termina l'applicazione

        }

    }

}
```

Standard GUI

E' l'interfaccia iniziale che permette di scegliere se usare normalmente l'applicazione oppure con più visuali gestite da diverse macchine

```
function OnGUI()
{
    if (GUI.Button (Rect (Screen.width/2, Screen.height/3, 150,100), "PLAY")){
        StandardGui();
    }

    if (GUI.Button (Rect (Screen.width/2, Screen.height/2, 150,100), "MULTISCR"))
    {

        GameObject.Find("Player").GetComponent("MultiMenu").enabled=true;
        GameObject.Find("GUI").GetComponent("StandardGUI").enabled=false;
    }
}

function StandardGui() {
    GameObject.Find("Player").GetComponent("PlayerOnStart").enabled=false;
    GameObject.Find("Player").GetComponent("EnablePlayer").enable=true;
    GameObject.Find("Player").GetComponent("MultiMenu").enabled=false;
    GameObject.Find("GUI").GetComponent("CursorOnStart").enabled=false;
    GameObject.Find("GUI").GetComponent("Cursor").showingCursor=false;
    GameObject.Find("GUI").GetComponent("StandardGUI").enabled=false;
}
```

Text On Gui

Gestisce la visualizzazione della GUI relativa alle informazioni (sia testo che immagini) del punto di interesse.

```
var GalleryObject:GameObject;

//definisce l'oggetto a cui è assegnato lo script per attivare la galleria

var titolo=" TITOLO";

//titolo del box
var nPag=0;

//contatore per il numero di pagina attuale

var TxtPag1=" ";

//testo prima pagina (max 4100 caratteri)

var TxtPag2=" ";

//testo seconda pagina

var temp="";

//variabile temporanea per memorizzare il contenuto della prima pagina mentre la
seconda viene mostrata e viceversa

var showLabel : boolean = false;

var PersonalSkin : GUISkin;

//GUISkin che definisce il nostro stile

var DisablePOILookAndFeelGuiObject:GameObject;

function OnGUI(){

//fa apparire il menù sulla gui

GUI.skin = PersonalSkin;

if (showLabel) {
```

```

GUI.Label (Rect (Screen.width/8, Screen.height/10 ,600,450), titolo);

//label con titolo

GUI.Label (Rect (Screen.width/7, Screen.height/7 ,600,350),TxtPag1);

//label con contenuto prima pagina

GUI.Box (Rect (Screen.width/8, Screen.height/8 ,650, 400), "");

//box in trasparenza

}

if (GUI.Button (Rect (Screen.width/35, Screen.height/10 ,100,60), "Altro..")) {

//si scorrono le pagine

showLabel = true;

    if (nPag==0) {temp=TxtPag1;TxtPag1=TxtPag2;nPag=1;}

//se ci troviamo nella prima pagina,la memorizza nella temp e mostra la seconda

    else if (nPag==1) {TxtPag2=TxtPag1;TxtPag1=temp;nPag=0;}

//se ci troviamo nella seconda pagina,la memorizza nella temp e mostra la prima
}

if (GUI.Button (Rect (Screen.width/35, Screen.height/3 ,100,60), "Esci")) {

//pulsante esci

DisablePOILookAndGuiObject.GetComponent("DisablePOILookAndGui").enabled = true
//disabilita il menù e riabilita il controllo giocatore
}

if (GUI.Button (Rect (Screen.width/35, Screen.height/5 ,100,60), "Foto..")) {

//pulsante foto

this.enabled = false;

//disabilita il testo sulla GUI

GalleryObject.GetComponent("ActivateGallery").enabled=true;
//attiva la galleria di immagini
}

}

```

Permanent GUI

Scritte permanenti che vengono visualizzate indipendentemente dalla GUI visualizzata in quel momento.

Sono utilizzate per dare indicazioni sui tasti da usare e su come muoversi.

```
var PersonalSkin : GUISkin;  
  
//GUISkin che definisce il nostro stile  
  
var indicazioni="";  
  
function OnGUI(){  
  
    GUI.skin = PersonalSkin;  
  
    GUI.Label (Rect (2,2 ,500,500), indicazioni);
```

6.3- Visuale su più telecamere distribuite

MultiMenu

L'obiettivo era quello di avere più punti di vista sullo stesso personaggio (davanti, dietro, destra, sinistra, sopra, sotto) processati da macchine differenti.

La soluzione è stata creare una sorta di struttura multiplayer, dove la macchina principale del sistema fa da host, controlla lo spostamento del personaggio e della telecamera, mentre tutte le altre si connettono a questo host, vedendo solo la telecamera che hanno scelto tramite l'apposita GUI.

In questo modo è possibile utilizzare l'applicazione all'interno di CAVE.

```

var MainCamera:GameObject ;

//si definisce la variabile relativa alla camera primaria

var CameraRight:GameObject ;

//si definiscono le variabili delle altre camere

var CameraLeft:GameObject ;
var CameraBack:GameObject ;
var CameraUp:GameObject ;
var CameraDown:GameObject ;

var ip="127.0.0.1"; //ip dell'host

var ncamera=0;

//in base a questa variabile viene deciso quale camera verrà assegnata al client

var connected =false;

function ChangeCamera() {

//assegna una determinata camera ad un certo client

connected =true;

    if (ncamera==1){

//a seconda del valore di ncamera si lancia la funzione spectator e si abilita quella
camera specifica

        Spectator();
        CameraRight.GetComponent("Camera").enabled=true; }

    if (ncamera==2){
        Spectator();
        CameraLeft.GetComponent("Camera").enabled=true; }

    if (ncamera==3){
        Spectator();
        CameraUp.GetComponent("Camera").enabled=true; }

    if (ncamera==4){
        Spectator();
        CameraDown.GetComponent("Camera").enabled=true; }

    if (ncamera==5){
        Spectator();
        CameraBack.GetComponent("Camera").enabled=true; }

}

```

```

function Spectator ()

// in questa funzione si disabilitano vari script che non servono al client

{
    MainCamera.GetComponent("Camera").enabled=false;

    //si disabilita la camera primaria

DestroyImmediate(GameObject.Find("Player").GetComponent("SmoothLookAt"));

//si disabilita lo script relativo alla visuale sui POI

DestroyImmediate(GameObject.Find("GUI"));

//si distrugge l'oggetto GUI

Screen.showCursor = false; //si nasconde il cursore

}

function Join ()

//questa funzione gestisce l'entrata del giocatore host. si nasconde la gui relativa
alla multicamera, si abilita il giocatore e si nasconde il cursore

{
    connected = true;
    EnablePlayer();
    CursorSet()
}

function EnablePlayer()

//si abilitano gli script per permettere al giocatore di muoversi

{
    GameObject.Find("Player").GetComponent("PlayerOnStart").enabled=false;
    GameObject.Find("Player").GetComponent("EnablePlayer").enable=true;
}

function OnDisconnectedFromServer()
{
    connected = false;
}

function OnConnectedToServer()

//questa funzione gestisce l'entrata del giocatore client. si cambia camera e si
nasconde il cursore

{
    ChangeCamera();
    CursorSet();
}

```

```

function OnServerInitialized()
{
    Join();
}

function CursorSet()

//si nasconde il cursore

{
    GameObject.Find("GUI").GetComponent("CursorOnStart").enabled=false;
    GameObject.Find("GUI").GetComponent("Cursor").showingCursor=false;
}

function OnGUI()

//gui che gestisce la multicamera

{
    if (!connected)

//questo permette di nascondere questa gui appena connessi

    {
        ip = GUI.TextField (Rect (50,25 ,100,20), ip, 25);

        if (GUI.Button (Rect (50, 50 ,80,30), "CameraRight")) {

//in base al tasto gui relativo alla camera si cambia il valore ncamera che
//assegnerà di conseguenza la camera adeguata.

            ncamera=1;
            Network.Connect(ip, 5300);//ci si connette all'host
        }

        if (GUI.Button (Rect (50,85 ,80,30), "CameraLeft")) {
            ncamera=2;
            Network.Connect(ip, 5300);
        }

        if (GUI.Button (Rect (50,120 ,80,30), "CameraUp")) {
            ncamera=3;
            Network.Connect(ip, 5300);
        }

        if (GUI.Button (Rect (50,155 ,80,30), "CameraDown")) {
            ncamera=4;
            Network.Connect(ip, 5300);
        }
    }
}

```



```

if (GUI.Button (Rect (50,190 ,80,30), "CameraBack")) {
    ncamera=5;
    Network.Connect(ip, 5300);
}

if (GUI.Button (Rect (50,240, 80,30), "Host")) {

    Network.InitializeServer(10, 5300, false);

//si inizializza il server

}
}
}

```

6.4-L'interprete XML

Come già accennato l'utilizzo della piattaforma è diviso in due parti, una riguarda la disposizione dei modelli 3d all'interno dell'*editor* e un'altra la creazione di un foglio XML che verrà letto e interpretato dall'interprete di seguito.

```

import System.Xml;

//si importa la libreria per lettura/scrittura su xml

var NomeNPC = "";

//si dichiarano tutte le variabili relative ad ogni task

var NomeTrigger = "";

var AnimazioneEntrata="";
var AnimazioneUscita="";

var POI="";
var TestoP1="";
var TestoP2="";

var NomeAudio="";

```

```

var NomeGallery="";
var NumeroPrimaImg=0;
var NumeroUltimaImg=0;

var NomeFileImg="";
var NumeroGameObject=0;

var Chi="";
var Percorso="";
var DurataSecondi="";

function Start()

{
    var reader:XmlReader = XmlReader.Create("gamexml.xml");

    //leggi i dati dall'luni.xml nella cartella del progetto

    while(reader.Read()){

        //finchè ci sono dati da leggere

        if(reader.IsStartElement("Animazione"))

            // per ogni elemento come animazione,parla etc.. si leggono e si associano alle
            // variabili precedente dichiarate gli attributi

            {
                NomeNPC=reader.GetAttribute("NomeNPC");
                NomeTrigger=reader.GetAttribute("NomeTrigger");
                AnimazioneEntrata=reader.GetAttribute("AnimazioneEntrata");
                AnimazioneUscita=reader.GetAttribute("AnimazioneUscita");
                Animazione();
            }

        if(reader.IsStartElement("Parla")){
            NomeTrigger=reader.GetAttribute("NomeTrigger");
            NomeAudio=reader.GetAttribute("NomeAudio");
            Parla();
        }

        if(reader.IsStartElement("IllustraPOI") {
            NomeTrigger=reader.GetAttribute("NomeTrigger");
            POI=reader.GetAttribute("POI");
            IllustraPOI();
        }

        if(reader.IsStartElement("Informazioni")){
            POI=reader.GetAttribute("POI");
            TestoP1=reader.GetAttribute("TestoP1");
            TestoP2=reader.GetAttribute("TestoP2");
            Informazioni();
        }
    }
}

```

```

if(reader.IsStartElement("GalleriaFoto")){
    NomeGallery=reader.GetAttribute("NomeGallery");
    NumeroPrimaImg= parseInt(reader.GetAttribute("NumeroPrimaImg"));
    NumeroUltimaImg= parseInt(reader.GetAttribute("NumeroUltimaImg"));
    GalleriaFoto();
}

if(reader.IsStartElement("Img")){
    NomeFileImg=reader.GetAttribute("NomeFileImg");
    NumeroGameObject=parseInt(reader.GetAttribute("NumeroGameObject"));
    Img();
}

if(reader.IsStartElement("Muovi")){
    NomeTrigger=reader.GetAttribute("NomeTrigger");
    Chi=reader.GetAttribute("Chi");
    Percorso=reader.GetAttribute("Percorso");
    DurataSecondi=reader.GetAttribute("DurataSecondi");
    Muovi();
}

if(reader.IsStartElement("Musica")){
    NomeAudio=reader.GetAttribute("NomeAudio");
    Musica();
}
}
}

function Animazione(){

// per ogni funzione si collocano i giusti valori richiamando gli script

    var NomeNPCGO= GameObject.Find(NomeNPC);

    GameObject.Find(NomeTrigger).GetComponent("AnimationOnTrigger").animate
    object=NomeNPCGO;
    GameObject.Find(NomeTrigger).GetComponent("AnimationOnTrigger").Animati
    onOnEnter=AnimazioneEntrata;
    GameObject.Find(NomeTrigger).GetComponent("AnimationOnTrigger").Animati
    onOnExit=AnimazioneUscita;
}

function Parla(){
    var NomeAudioGO=Resources.Load(NomeAudio);

    GameObject.Find(NomeTrigger).GetComponent("AudioSource").clip=NomeAudio
    GO;
}

```

```

function Img(){
    var NomeFileImgGO=Resources.Load(NomeFileImg);

    //si carica direttamente il file dalla cartella resources

    var NumeroGameObjectINT=(NumeroGameObject+ "");
    GameObject.Find(NumeroGameObjectINT).guiTexture.texture=NomeFileImgGO;
}

function IllustraPOI(){
    var POIGO= GameObject.Find(POI);
    var POITransform= GameObject.Find(POI).transform;

    GameObject.Find(NomeTrigger).GetComponent("POILookAndGui").TargetSmoothLookAt=POITransform;
    GameObject.Find(NomeTrigger).GetComponent("POILookAndGui").TextOnGuiObject=POIGO;
}

function Informazioni(){
    GameObject.Find(POI).GetComponent("TextOnGui").TxtPag1=TestoP1;
    GameObject.Find(POI).GetComponent("TextOnGui").TxtPag2=TestoP2;
}

function GalleriaFoto(){
    var POIGO= GameObject.Find(POI);
    GameObject.Find(NomeGallery).GetComponent("ActivateGallery").DisablePOILookAndGuiObject=POIGO;
    GameObject.Find(NomeGallery).GetComponent("ActivateGallery").maxCount=NumeroUltimaImg;
    GameObject.Find(NomeGallery).GetComponent("ActivateGallery").count=NumeroPrimaImg;
}

function Muovi(){
    var ChiGO= GameObject.Find(Chi);
    var PercorsoGO= GameObject.Find(Percorso);

    GameObject.Find(NomeTrigger).GetComponent("SplineOnTrigger").SplineControllerObject=ChiGO;
    ChiGO.GetComponent("SplineController").SplineParent=PercorsoGO;
    ChiGO.GetComponent("SplineController").Duration=parseFloat(DurataSecondi);
}

function Musica(){
    var NomeAudioGO=Resources.Load(NomeAudio);
    GameObject.Find("Player").GetComponent("AudioSource").clip=NomeAudioGO;
}

```

POI Look And Gui

Questo *script* gestisce i già citati POI: si blocca il giocatore, si disabilita il menù di pausa, si sposta il suo punto di vista e si attiva lo *script* TextOnGui per le informazioni aggiuntive

```
var TargetSmoothLookAt:Transform;//var TargetSmoothLookAt:Transform;

//si definisce su quale oggetto spostare la visuale del giocatore

var TextOnGuiObject:GameObject;

//si definisce a quale oggetto è assegnato lo script TextOnGui

function OnTriggerEnter (collision : Collider){
    if (collision.gameObject.tag == "Player") {

        //se il giocatore collide con il collider dell'oggetto

        GameObject.Find("GUI").GetComponent("ESC").enabled=false;

        //disattiva menù pausa

        GameObject.Find("Player").GetComponent("EnablePlayer").enable=false;

        //disabilita il giocatore

        GameObject.Find("Player").GetComponent("SmoothLookAt").target=TargetSmoothLookAt; //imposta l'oggetto su cui spostare la visuale

        GameObject.Find("Player").GetComponent("SmoothLookAt").enabled=true;

        //sposta la visuale sull'oggetto scelto

        yield WaitForSeconds(1);

        //attendi un secondo per far apparire il menù di informazioni

        GameObject.Find("GUI").GetComponent("Cursor").showingCursor=true;

        //mostra il cursore

        TextOnGuiObject.GetComponent("TextOnGui").enabled=true;

        //attiva lo script per il menù informazioni
    }
}
```

Questa è la gestione della galleria di immagini nelle informazioni aggiuntive del POI.

Si prendono in input il numero di immagini e il nome, per poi assegnare e mostrarle a seconda dei tasti del mouse premuti.

```
var DisablePOILookAndGuiObject:GameObject;

//oggetto dove si trova lo script DisableGuiText

var count:int;

//il numero della prima immagine in galleria

var StartCount:int;
var previously : int;
var maxCount : int;
var end : int;

//il numero dell'ultima immagine in galleria

var indication="Click sinistro per prossima immagine Click destro per uscire";

//eventuali indicazioni che si danno a chi scorre la galleria

var PersonalSkin : GUISkin;

//stile della GUI per la galleria

function OnGUI(){

// definisce la posizione delle indicazioni e le mostra

GUI.skin = PersonalSkin;
GUI.Label (Rect (Screen.width/40, Screen.height/40 ,445,100), indication);
}

function Start() {

//operazioni da fare prima di tutto
StartCount=count-1;
end = maxCount + 1;

//end è uguale al maxCount + 1
```

```

    if(count != 0) Show();

    //se si vuol far partire la galleria con una immagine che non sia la prima,
    mostra subito l'immagine in questione
}

function Update() {
    if(Input.GetKeyDown(KeyCode.Mouse1)) {
        //gestisce il comando click destro del mouse (uscita dalla galleria)
        LastShow();
        Restore();
        GameObject.Find(count.ToString()).guiTexture.enabled = false;

        //si disattiva l'immagine corrente

        this.enabled = false;
        //si disattiva lo script
    }

    if(count == end) {

        //quando finisce la galleria, si lancia restore

        Restore();
    }

    if((Input.GetKeyDown(KeyCode.Mouse0)) && (count <= maxCount)) {

        //gestisce il comando click sinistro del mouse (prossima immagine)

        ++count;
        Show();
    }

    if(count != 0) previously = count - 1;

    //se il count è diverso da 0, previously è uguale al count - 1

    if(previously != 0) notShow();

    //se previously è diverso da 0, esegue la funzione notShow
}
function Show() {

    if(count != end) GameObject.Find(count.ToString()).guiTexture.enabled = true;
    //attiva l'immagine all'interno del GameObject con il nome di count
}

function notShow() {

    GameObject.Find(previously.ToString()).guiTexture.enabled = false;

    //disattiva l'immagine all'interno del GameObject con il nome di previously
}

```

```
function LastShow() {  
    GameObject.Find(count.ToString()).guiTexture.enabled = false;  
    //disattiva l'immagine all'interno del GameObject con il nome di count  
}  
function Restore() {  
    //ripristina i valori dello script e abilita lo script per uscire dalla galleria  
    count=StartCount;  
    DisablePOILookAndFeelGuiObject.GetComponent("DisablePOILookAndFeelGui").enabled=true;  
}
```


Disable Poi Look and Gui

Questo *script* esegue le stesse azioni del POILookAndGui ma esattamente al contrario: sblocca il giocatore, abilita il menù di pausa, si libera la camera principale e si disattiva lo *script* TextOnGui.

```
var GalleryObject:GameObject;

//si sceglie l'oggetto che ha lo script gallery

var TextOnGuiObject:GameObject;

function Update () {

//permette di ri-abilitare il giocatore a muoversi e disattiva il testo su GUI o la galleria

    TextOnGuiObject.GetComponent("TextOnGui").enabled = false;

//disabilita il testo su GUI

    GameObject.Find("Player").GetComponent("SmoothLookAt").enabled=false;

//disabilita lo script che obbliga la visuale del giocatore sull'oggetto di interesse

    GameObject.Find("Player").GetComponent("EnablePlayer").enable=true;

//ri-abilita gli script che permettono di controllare il giocatore

    GameObject.Find("GUI").GetComponent("Cursor").showingCursor=false;

//nasconde il cursore

    GalleryObject.GetComponent("ActivateGallery").enabled=false;

//disabilita lo script che gestisce la galleria

    GameObject.Find("GUI").GetComponent("ESC").enabled=true;

//attiva menù pausa

    this.GetComponent("DisablePOILookAndGui").enabled = false;

//per finire lo script si disabilita, in maniera da poter venire riattivato in seguito

}
```

Player On Start

Semplicemente viene stabilito se il Player può controllare all'avvio dell'applicazione; è stato utile per abilitare la StandardGUI che successivamente alla scelta proposta dà libertà di movimento.

```
var PlayerCanControl=true;

function Update () {

//abilita gli script che permettono di controllare il giocatore

GameObject.Find("Player").GetComponent("EnablePlayer").enable=PlayerCanControl;

}
```

Enable Player

Viene richiamato molte volte da *script* diversi, sia per la GUI che per altri di supporto, permette di abilitare o disabilitare gli *script* che controllano il movimento e la visuale del giocatore.

```
var enable=false;

//variabile che definisce se è possibile controllare il giocatore o no

function Update () {

//abilita gli script che permettono di controllare il giocatore

GameObject.Find("Player").GetComponent("MouseLook").enabled=enable;

//abilita la visuale via mouse

GameObject.Find("Player").GetComponent("CharacterMotor").enabled=enable;

//abilita il controllo del movimento del giocatore

}
```

7-Ricostruzione della *domus* degli affreschi di Luni

Il contesto di interesse storico riportato in vita con la ricostruzione è quello della *Domus* degli affreschi di Luni, una *domus* romana rinvenuta nell'area lunense all'interno del contesto dell'antica città di "Luna" ; la *domus* è stata abitata in più periodi, ed è stato scelto quello più "recente" (seconda metà del I sec. a.C.) perché è quello di cui sono stati trovati più reperti e di cui abbiamo più informazioni.

Inoltre questo è il periodo più fiorente dell'economia lunense e permette ricostruire un ambiente particolarmente interessante.



Come detto in precedenza abbiamo molti dati relativi a questa *Domus*, relativi a tutti i ritrovamenti archeologici e agli studi effettuati in quest'area; in questo modo è stato possibile offrire una ricostruzione molto accurata, senza dover compiere valutazioni troppo generiche.

In particolare, è stato possibile elaborare un modello 3d della *domus* dalla piantina ricavata dagli scavi effettuati sino ad oggi.

Con questa metodologia è impossibile sbagliare le misure o la disposizione dei vani nel modello 3d, e gli unici dubbi che rimangono, come in tutti gli edifici antichi di cui non rimangono altro che macerie, sono sulle altezze, ricavate facendo accurate valutazioni sul teorema di Vitruvio (spiegato anche più avanti) e dai paragoni con altre *domus* dello stesso periodo.

La precisione del modello non si limita alla struttura, ma anche alla colorazione di ogni singola stanza, visto che abbiamo molte informazioni sul colore date dai frammenti murari dei vani; anche i dettagli più particolari sono stati ricavati da ritrovamenti di determinati materiali, come ad esempio le grandi quantità di vetro trovate nel giardino, che fanno intuire l'esistenza di grandi vetrate.

Le particolarità di questa *domus*, motivo per il quale si differenzia da molte altre, sono principalmente due: gli affreschi (da cui deriva il nome) e i pavimenti; grazie ad uno stato di conservazione veramente notevole sono stati riprodotti sia i motivi degli affreschi murari, sia i pavimenti all'interno delle stanze, che nella maggior parte dei casi sono realizzati con foto originali dello scavo.

Anche l'arredamento è stato sviluppato in relazione ai ritrovamenti di vasi e oggetti di uso comune, che hanno dato modo di supporre quale fosse l'uso di determinate stanze.

La costruzione del modello si è fermata al limite degli scavi, visto che l'area è così grande che ancora oggi non è stato possibile studiarla tutta.

8- Metodologia di lavoro

Dopo aver studiato tutta la documentazione storica e archeologica, ed aver elaborato una sorta di “bozza” per ogni singola entità della ricostruzione (che sia l’intera struttura della *domus* o un oggetto d’arredamento) si è passati alla parte tecnica, ovvero alla realizzazione dello scenario, delle strutture e degli oggetti. Saranno illustrati i vari procedimenti di ricostruzione e tutte le loro fasi di sviluppo.

I programmi utilizzati sono stati:

- **Unity 3D**
- **SketchUp**, per costruire tutti i modelli 3d statici presenti nella scena: dalla struttura completa della *domus* fino agli oggetti d’arredamento più piccoli.
- **3D Studio Max**, per ricreare il personaggio del padrone di casa, i suoi vestiti e le sue animazioni.
- **Photoshop**, per disegnare le *texture* per i modelli 3d statici, dinamici e per la skybox (il cielo).
- **Audacity**, per creare e modificare i file audio all’interno della ricostruzione.

8.1- Lo scenario

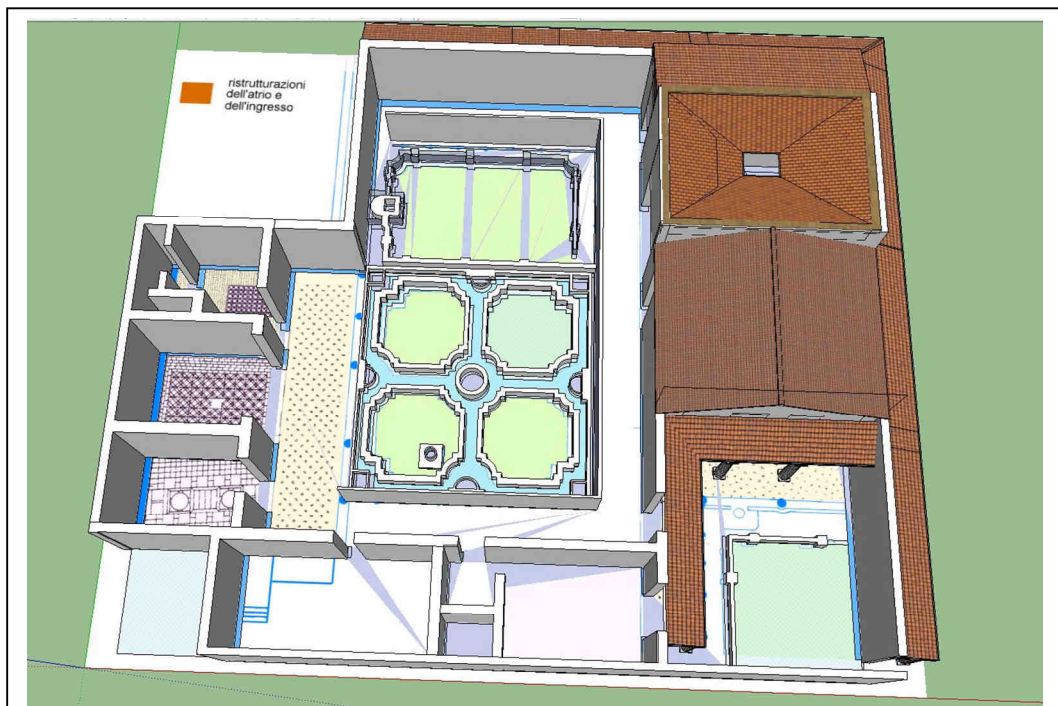
Trattandosi di una ricostruzione ambientata all'interno della *domus* non c'è stato bisogno di ricostruire tutto l'ambiente esterno, ma per realizzare un modello più completo e per osservarlo da differenti angolazioni è stata aggiunta una strada esterna in basolato, e sono stati ricostruiti tutti i muretti esterni, le staccionate di confine e le tettoie esterne.

Per creare il *Terrain* si è usato l'*editor* integrato di *Unity*, che ha fornito anche gli strumenti adatti a distribuire la vegetazione interna e gli alberi.

8.2- Modelli 3d statici

Dobbiamo differenziare due diverse modalità di realizzazione dei modelli 3d: la prima riguarda gli oggetti statici, la seconda i modelli dinamici, che nel nostro caso corrisponde al personaggio.

Per realizzare i modelli statici è stato usato SketchUp che è un ottimo strumento soprattutto per la realizzazione di strutture architettoniche.



Piantina su cui si è basata la ricostruzione 3d

La *domus*

Per la *domus* in particolare è stato usato questo procedimento: una volta avuta la piantina completa dell'edificio è stata importata come *texture* 2d all'interno di SketchUp e "stesa" sul pavimento del piano di lavoro.

A quel punto si è "ricalcata" la piantina con strumenti appositi; dopo il disegno, per ogni vano, si sono controllate le misure e si è provveduto alla realizzazione dell'alzato.

Quando si è ottenuta la struttura principale, poco per volta si sono modellati i giardini, le tettoie, i soffitti, le porte i colonnati, tutti a parte, e infine si sono riuniti in un unico modello.

Una volta ottenuto il modello definitivo è stato texturizzato secondo le informazioni a nostra disposizione .

Infine è stato posizionato tutto l'arredamento, in maniera consona all'ambiente stabilito.

C'è da dire che la modellazione della struttura principale non è avvenuta in un'unica fase: sono stati apportati continuamente piccoli cambiamenti derivati dal confronto accurato della documentazione al modello ottenuto.

Un esempio possono essere le scale del loggiato che sono state "costruite" praticamente per ultime, per capire bene dove andassero collocate rispetto al piano superiore.

Gli oggetti

Anche per la modellazione degli oggetti e dell'arredamento ci si è affidati a SketchUp; soprattutto in questa fase il confronto continuo con il materiale visivo fornito è stato determinante per la realizzazione di oggetti molto accurati dal punto di vista storico.

Qui il procedimento è il seguente: dopo il confronto con l'immagine o la descrizione si è realizza un primo prototipo di modello, che viene collocato nello scenario; a questo punto viene analizzato per vedere se in base al contesto bisogna migliorare o aggiungere qualcosa.

Infine si texturizza l'oggetto finale e si implementa nel modello principale.

8.3- Modelli 3d dinamici

Quando si parla di modelli dinamici parliamo di modelli umanoidi, che devono rappresentare personaggi o npc.

Nel nostro caso si è dovuta creare la figura del padrone di casa.

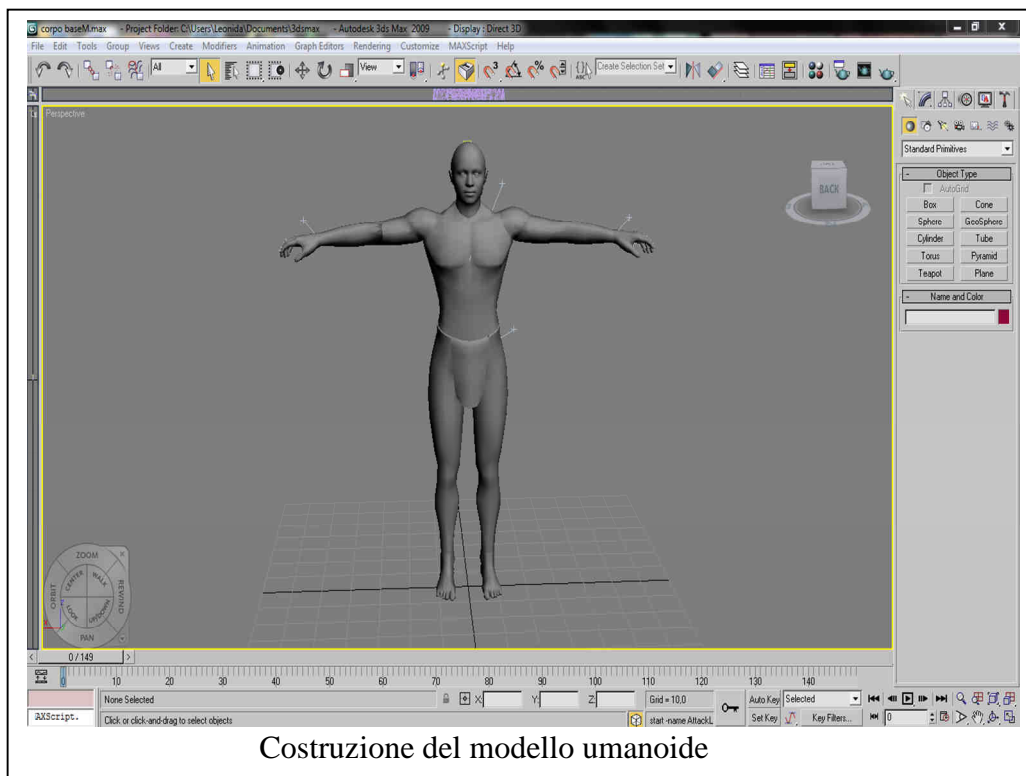
Tramite vari plugin si sono importati i modelli di umanoide all'interno di 3d Studio Max; dopo aver trovato quello più adatto si è adattato e rifinito (corporatura, connotati).

Poi, una volta texturizzato si è abbigliato e accessoriatato basandosi su ricostruzioni storiche di abiti e persone dell'epoca.

A questo punto il modello del nostro personaggio è stato esportato dentro il progetto di *Unity*.

Dopodichè si è passati alla modellazione delle animazioni sullo scheletro del personaggio, e una volta completate si sono importate anche queste e associate al modello nel progetto.

Sono state apportate lievi modifiche alle *texture* direttamente da *Unity*, per rendere il personaggio compatibile con le condizioni di luce della scena.



8.4- Texture

Per il progetto si sono sviluppati numerosi tipi di *texture*, e la metodologia è stata differente a seconda se la *texture* è ricavata da un oggetto reale o è generata artificialmente.

Nel primo caso si estrapola dalla foto la porzione che ci interessa, dopodiché la si migliora o si apportano modifiche necessarie (ad esempio le *texture* dei pavimenti sono state schiarite, perché i ritrovamenti sono conservati molto bene, ma presentano comunque colorazioni troppo scure rispetto a quelle di un tempo): infine si cerca di renderla quanto più omogenea possibile, in modo da non notare il distacco netto di ripetizione della *texture*.

Nel secondo caso invece, possiamo individuare due ulteriori metodologie: o si trovava un *texture* simile a quella che si vuole ottenere e tramite gli strumenti si modifica per adattarla ai nostri scopi, oppure si genera artificialmente.

Ad esempio per la *texture* della terracotta si è riempito lo spazio dell'immagine con un colore adatto e poi si sono applicati vari filtri (rumore, graffi) fino ad ottenere un'immagine realistica.

8.5- Audio

Per completare l'esperienza un buon comparto audio è determinante; si è lavorato su diversi tipi di audio: le frasi, la musica e gli effetti sonori.

Le frasi, sono state doppiate da varie persone, per poi scegliere la registrazione più adatta allo scopo.

La lavorazione è identica per tutti questi tipi di audio: dopo aver trovato il suono (musica ed effetti sonori) o averlo registrato (frasi o effetti sonori) si è importato nell'*editor*; a questo punto si sono applicati vari filtri per migliorare il suono (eliminazione del rumore, innalzamento db etc..).

Come ultima cosa si è importato l'audio in *Unity*, e tramite il file xml si è assegnato all'entità a cui apparteneva.

A volte si è provveduto ad aggiustare un po' il volume e a regolare il suono in base a tutti gli altri suoni ed effetti nella scena.

9-La carta di Londra

La carta di Londra è un documento scritto da Hugh Denard (la versione di riferimento è quella italiana, a cura di Enrica Salvatori), e costituisce il primo tentativo di uniformare e stabilire i principi che regolano le visualizzazioni digitali, che siano riproduzioni in 2d, 3d statiche o riproduzioni 3d in tempo reale, come quella di cui stiamo parlando.

Questo per dare un rigore e un metodo alle ricostruzioni digitali che sono utilizzate sempre più spesso nel campo dei beni culturali.

Dall'introduzione:

“Dato l'uso sempre più intensivo dei metodi di visualizzazione digitale (i.e. elaborata o generata da un computer) per assistere la ricerca, la comunicazione e la preservazione dei beni culturali in una vasta gamma di contesti, è necessario fissare una serie di principi che assicurino che la visualizzazione digitale del patrimonio culturale sia intellettualmente e tecnicamente rigorosa, al pari dei metodi di ricerca sui beni culturali e di comunicazione dei risultati di tali ricerche. Allo stesso tempo, questi principi devono riflettere le proprietà distintive delle tecnologie e dei metodi per la visualizzazione digitale.”

“La Carta di Londra cerca di catturare e di costruire il consenso su questi problemi in modo da produrre un largo riconoscimento e una auspicabile conformità all'interno delle relative comunità scientifiche. Facendo questo, la Carta mira a rafforzare il rigore con il quale i metodi e i risultati della visualizzazione digitale sono usati e valutati nei contesti culturali, ossia promuovendo la comprensione e il riconoscimento di tali metodi e risultati.”

“La Carta definisce i principi per l'uso dei metodi della visualizzazione digitale in relazione all'integrità intellettuale, all'affidabilità, alla documentazione, alla sostenibilità e all'accessibilità.”

Gli obiettivi della Carta sono ambiziosi, e comprendo principi differenti che devono sia rispettare la precisione storica, sia offrire metodi affidabili e fruibili.

- **Fornire un caposaldo** che abbia un ampio riconoscimento fra gli interessati.
- **Promuovere il rigore tecnico e intellettuale** nella visualizzazione digitale dei beni culturali.
- **Assicurare che i procedimenti e i risultati della visualizzazione digitale possano essere capiti e valutati** dagli utenti.
- **Fornire autorevolezza scientifica alla visualizzazione digitale** nello studio, nell'interpretazione e nella gestione dei beni culturali.
- **Assicurare che siano stabilite e applicate strategie di accessibilità e sostenibilità.**
- **Offrire una solida base d'appoggio** sul quale le comunità impegnate nel settore possano costruire linee guida più dettagliate a implementazione della Carta di Londra.

La carta propone vari principi, che devono essere rispettati per una corretta rappresentazione digitale dei beni culturali.

Implementazione

I principi della Carta di Londra sono validi ogniqualvolta è applicata la visualizzazione digitale alla ricerca e alla disseminazione relative ai beni culturali.

Scopi e metodi

Un metodo di visualizzazione digitale dovrebbe normalmente essere usato solo quando è il metodo disponibile più appropriato per quello scopo.

Fonti della ricerca

Per assicurare l'integrità intellettuale dei metodi e dei risultati della visualizzazione digitale, le fonti rilevanti devono essere identificate e valutate in maniera documentata e strutturata.

Documentazione

Sufficienti informazioni dovrebbero essere fornite per permettere ai metodi e ai risultati della visualizzazione digitale di essere compresi e valutati in maniera appropriata rispetto ai contesti e agli scopi nei quali e per i quali sono divulgati.

Sostenibilità

Dovrebbero essere pianificate e implementate strategie per assicurare la sostenibilità a lungo termine della documentazione e dei risultati di visualizzazione digitale riguardante i beni culturali per prevenire perdite di questa parte crescente del patrimonio culturale, economico, sociale, intellettuale dell'umanità.

Accessibilità

Nella creazione e divulgazione delle visualizzazioni digitali si dovrebbero tenere in considerazione i modi in cui i risultati del lavoro possono contribuire allo studio, alla conoscenza, all'interpretazione e alla gestione del patrimonio culturale.

9.1-La carta di Londra applicata al progetto

Nel nostro progetto di ricostruzione si è cercato di attenersi il più possibile a questi principi.

Vediamo la loro applicazione al nostro progetto, analizzandoli uno per volta:

Implementazione: il progetto di ricostruzione della *Domus* degli affreschi si attiene precisamente al prodotto inteso e definito dalla Carta di Londra, essendo una visualizzazione digitale di bene culturale

Scopi e metodi: quando ci troviamo di fronte a dei resti archeologici è difficile immaginare come quella struttura o quel contesto fossero in realtà.

Questo genere di ricostruzioni, basate su un ambiente 3d altamente interattivo è sicuramente il miglior metodo per realizzare una ricostruzione accurata ma allo stesso tempo altamente fruibile.

Fonti della ricerca

Le fonti relative alla *domus* sono state valutate accuratamente, e si basano su pubblicazioni ufficiali della Soprintendenza del museo archeologico Luni.

Inoltre grazie all'aiuto della Dott. Marcella Mancusi (Soprintendenza del museo archeologico Luni) è stato possibile ricavare informazioni direttamente da molto altro materiale, come foto, disegni, etc..

Documentazione

La documentazione fornita insieme al progetto fa parte di questo testo, e verrà mostrata più avanti; nella biografia troviamo l'elenco delle pubblicazioni consultate e i riferimenti visuali.

Troviamo anche la metodologia utilizzata per ogni passaggio da documentazione a realizzazione tecnica.

Questo progetto utilizza praticamente tutti i mezzi a disposizione della visualizzazione digitale, includendo testo, audio, video, grafica 3d.

Sostenibilità

Proprio per portare avanti il progetto di ricostruzione è stata costruita una piattaforma apposita, con il proposito di portarla avanti con contenuti sempre nuovi.

Con la messa a disposizione del progetto in formato totalmente libero si spera di portare avanti il progetto anche dopo la sua realizzazione.

Il progetto è recuperabile sia online, sia negli archivi dell'Università di Pisa che in quelli della Soprintendenza del museo archeologico Luni.

Per garantire il funzionamento continuo dell'applicazione questa è resa disponibile in vari formati, inoltre è presente una guida all'utilizzo molto semplice e fruibile.

Accessibilità

Questo progetto ha un grado di accessibilità molto elevato, visto che può essere fruito da un qualunque computer.

E' possibile usare il progetto direttamente online, senza avere nessun tipo di applicativo in locale, senza il bisogno di installazioni o configurazioni particolari.

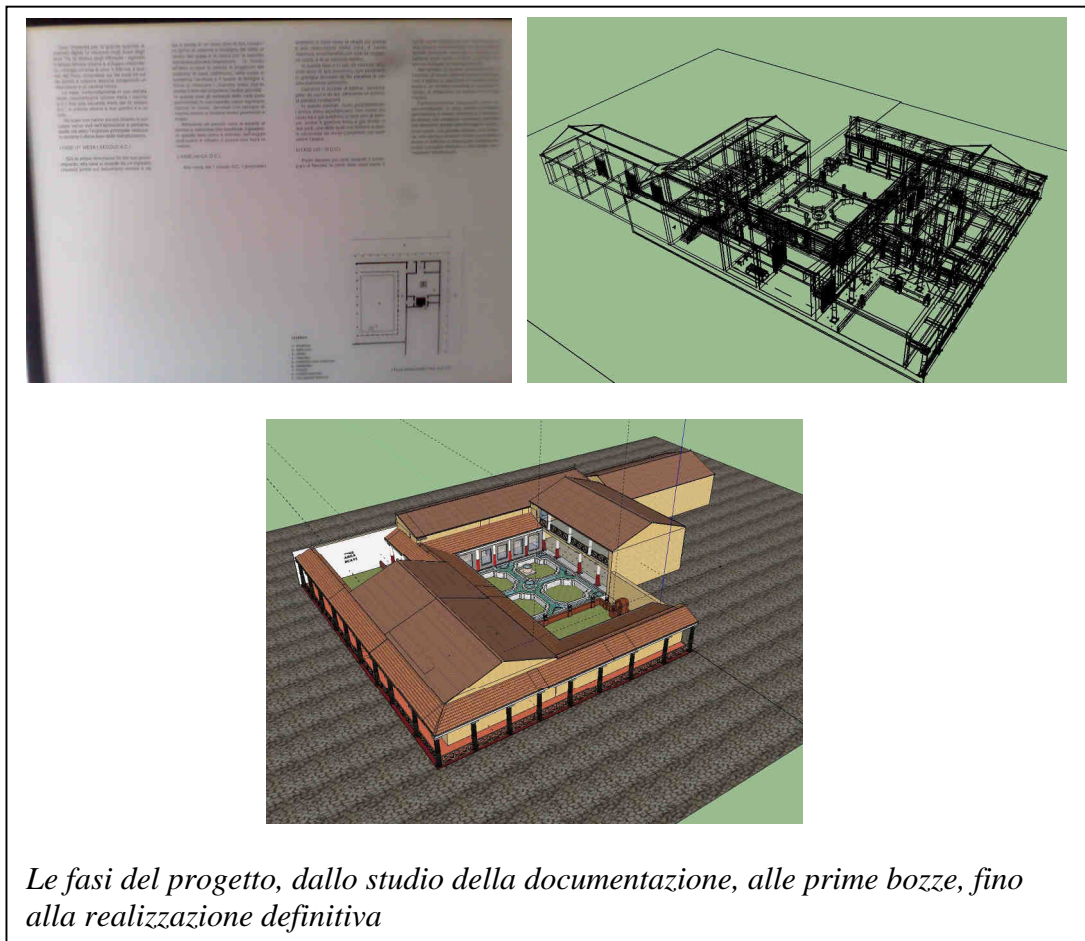
Per chi non potesse utilizzare i normali dispositivi di input viene messo a disposizione un video, in cui sono mostrati tutti i passaggi fondamentali.

10- Ricostruzione: tra fonti e soluzioni tecniche

Ogni volta che si ricostruisce un contesto storico avendo a disposizione strumenti molto potenti, in grado di restituire una resa impeccabile dal punto di vista storico e visivo, bisogna sempre seguire un metodo di lavoro chiaro e puntuale. Per ogni ambiente, stanza o oggetto dobbiamo per prima cosa valutare bene quali sono le informazioni che abbiamo a nostra disposizione.

Dopodichè possiamo passare ad un'idea dell'entità in questione e quando si sono avuti gli effettivi riscontri (confronto con altro personale che lavora al progetto, fonti testuali, visive etc..) si può passare alla realizzazione tecnica.

Di seguito vedremo ogni ambiente della *domus*, analizzato secondo il contesto storico e archeologico.



10.1- Contesto storico e archeologico

L'analisi degli scavi relativi alla *domus* risale a due periodi: la prima, più consistente risale alle pubblicazioni del 1983¹¹, la seconda è relativa a scavi aggiuntivi effettuati nel 2006¹².

Ovviamente si è tenuto conto di entrambe le documentazioni per portare avanti il progetto di ricostruzione; si analizzerà prima la documentazione dell'83 e poi quella del 2006.

Contesto Storico

I commerci mediterranei con i Greci e gli Etruschi lungo il versante marittimo, e i contatti con i fertili territori della pianura padana attraverso gli Appennini, furono la motivazione che spinse i romani (fra il 193 e il 155 a.C.) alla conquista della Liguria, appetibile proprio per la sua posizione geografica.

In questo arco cronologico durante le fasi finali del conflitto ligure-romano nel comprensorio ligure-toscano, si inserisce la costruzione della colonia di Luni. Solo però nel 155 a.C., quando saranno concluse le imprese belliche verso la Spagna, con il trionfo del console Marco Claudio Marcello, il *portus Lunae* poté assumere il suo ruolo all'interno del commercio marittimo romano.

Sul finire degli anni '70 del secolo scorso, con l'intento di riesaminare i dati già in possesso degli studiosi, si è cercato di delineare un quadro aggiornato del popolamento dal secondo secolo a.C. al primo secolo a.C. della colonia di Luni

¹¹ Annapaola Zaccaria Ruggiu, "La casa degli affreschi di Luni: fasi edilizie per successione diacronica", in "Centro studi lunensi, Quaderno 8", Lerici 1983, pp.3

¹² "Forme e tempi dell'urbanizzazione nella cisalpina (II secolo a.C. – I secolo a.C)", a cura di Luisa Brecciaroli Taborelli, Torino 2006

compresa tra il golfo di La Spezia ad occidente e il corso del torrente Parmigliola.

Questi studi ¹³ hanno appurato a Luni un vero e proprio sviluppo dell'attività edilizia pubblica e privata datata alla prima metà del I secolo d.C.

Probabilmente questo è da imputare allo sfruttamento delle cave sulle Alpi Apuane, che non solo fornivano il materiale da costruzione per Luni, ma permettevano grazie all'esportazione una florida economia.

Lo sfruttamento era finalizzato soprattutto alla costruzione di complessi pubblici e religiosi, che potevano essere strumenti di propaganda politica a causa del protrarsi dello stato di belligeranza fra le civiltà lunensi e pisane per questioni di confine.

A questo periodo risale la costruzione della *Domus* degli affreschi(40-50 d.C.).

La *domus* è orientata verso est e documenta lo stato di benessere del proprietario, sia per la sua ampiezza, sia per i materiali e le modalità di costruzione.

Questa *domus* era dotata di un quartiere estivo ed uno invernale, legati da porticati che si snodavano intorno al giardino e all'*hortus*.

Gli spazi verdi si integravano con sale di ricevimento, di studio e di riposo.

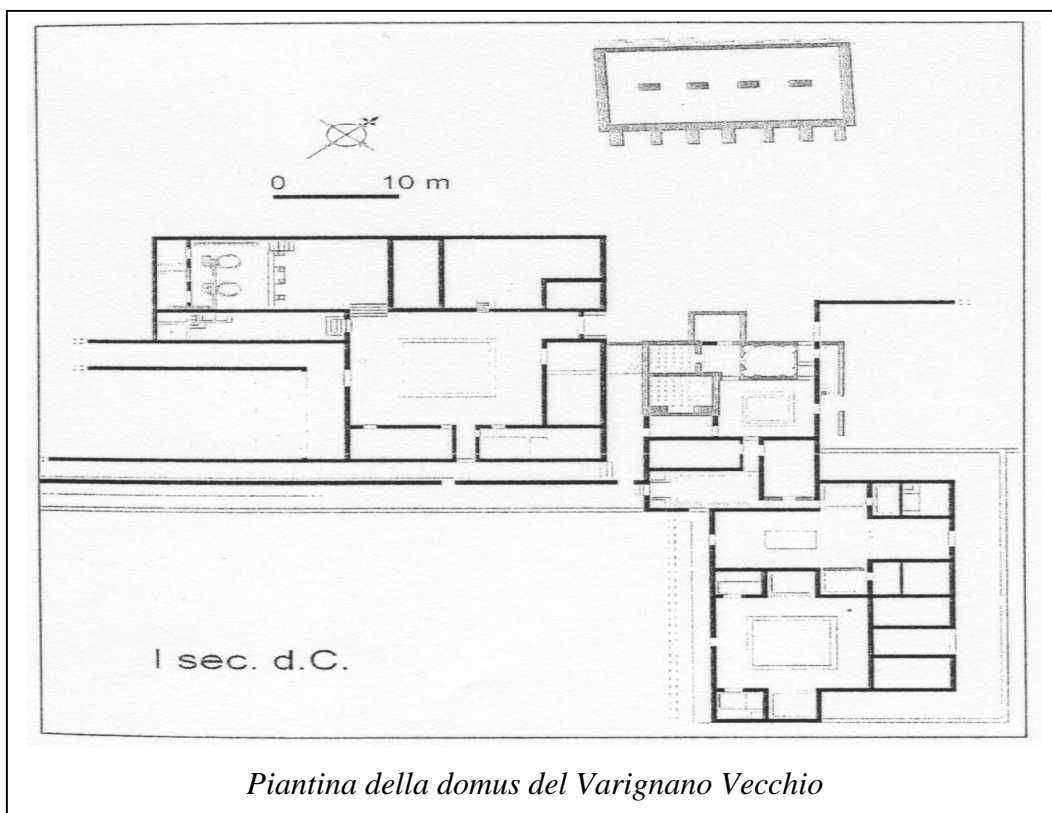
Le ville nel golfo di la spezia

Di seguito una breve panoramica sul contesto della colonia e delle altre strutture insediative nel territorio circostante; è riportato anche l'esempio di due *domus* che sono state usate anche come confronto quando si era incerti su qualche dettaglio della ricostruzione della nostra *domus*.

Nel II secolo a.C. si svilupparono, nell'area del Golfo dei Poeti e della foce del Magra, realtà insediative dedicate alla coltura dell'olivo e della vite, ma non

¹³ A.M. Durante, "Edilizia privata a Luna", in "XLIX Abitare in Cisalpina", Trieste 2001, pp. 269-295

mancarono anche ville lussuose: fra queste si trova quella del Varignano Vecchio(seconda metà del II secolo a.C.)¹⁴



Piantina della domus del Varignano Vecchio

Di questo edificio più antico sono noti sette ambienti a carattere residenziale.

Gli elementi decorativi parietali, associati a pavimentazioni ben conservate, sono in battuto di cocciopesto e graniglia in marmo decorato.

Tra i reperti rinvenuti si trovano ceramiche, olle (pentola) e brocche biansate¹⁵ o monoansate che confermano la datazione tra la fine del II e primi decenni del I secolo a.C.

Questo rende probabile una realtà produttiva fiorente che perdurò per tutte le fasi edilizie della colonia lunense.

A partire dal I secolo a.C. troviamo nei dintorni di Luni resti di fattorie, impianti rurali, necropoli e ville.

Alcuni lavori edilizi del dopoguerra hanno segnalato mosaici, anfore e monete che in seguito però sono stati dispersi.

E' rimasta comunque la struttura della Villa Maritima di Bocca di Magra realizzata a terrazze (I sec. a.C.).

Dalle ricerche di equipe di archeologi inglesi (1978-1979) in Castelnuovo di Magra si è giunti a conclusione che l'insediamento aveva carattere prettamente rurale verso la fine del II sec. a.C. e vide la sua fase di sviluppo più intensa nel I sec. a.C. grazie alla coltivazione della vite.¹⁶

Tutto questo viene dedotto anche dal ritrovamento di molti reperti di ceramiche da mensa, databili dal II sec. a.C.

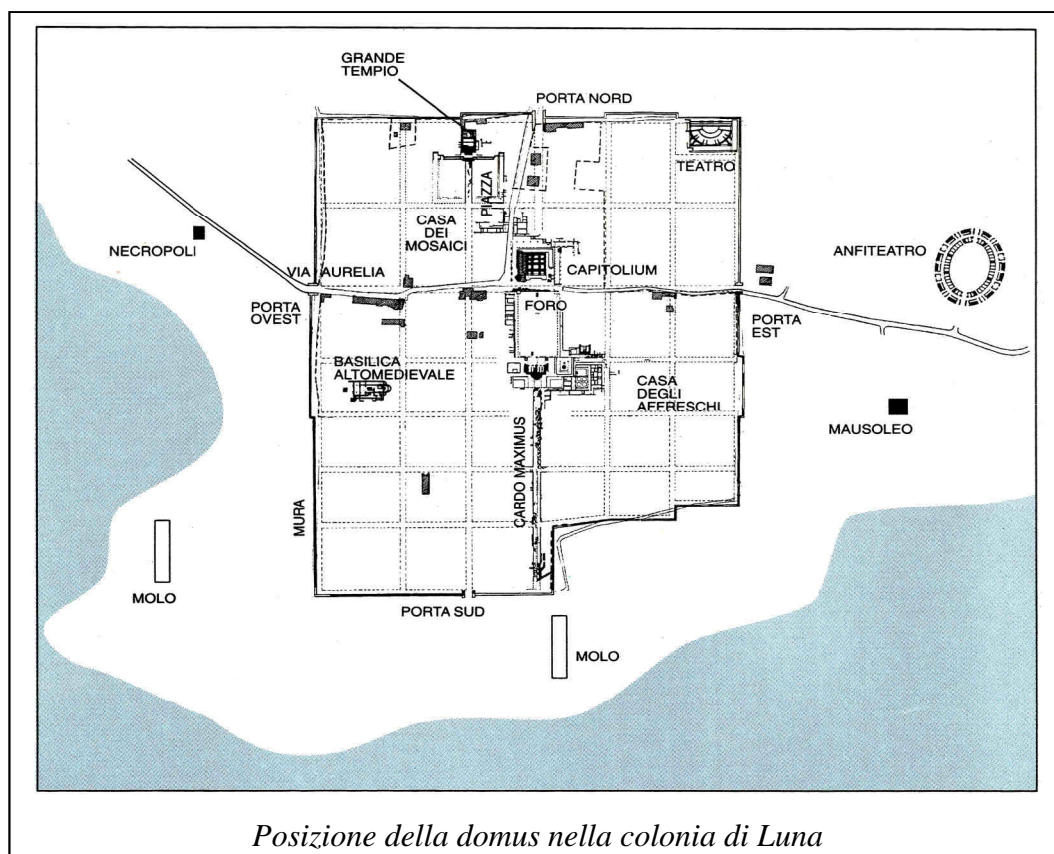
La rilettura dei materiali recuperati ha anche confermato un'occupazione di tipo agricolo-pastorale costituita da fattorie di piccole dimensioni che si ritrovano anche nell'età imperiale.

14- 15 -16

“Forme e tempi dell'urbanizzazione nella cisalpina (II secolo a.C. – I secolo a.C)”, *a cura di* Luisa Brecciaroli Taborelli, Torino 2006,pp 166-170

La *Domus* degli affreschi a Luni: analisi e fasi edilizie

La *Domus* degli Affreschi è posta a Sud del Foro tra l'incrocio di un cardo minore con un decumano minore.



Dall'analisi edilizia della villa, nelle sue diverse fasi cronologiche, è scaturita un'ipotesi di ricostruzione dell'alzato, riferita agli schemi di Vitruvio il più famoso teorico dell'architettura romana del I sec. a.C.

Gli scavi dal 1970 al 1975 hanno scoperto l'area della *domus* (1300 mt quadri) e un'area confinante costituita da una piazza a gradoni con un'area pubblica, un monumento centrale e una fontana.¹⁷

¹⁷ A.M. Durante, "Appunti per lo studio degli intonaci dipinti della *Domus* degli Affreschi", in "Romana pictura e christiana signa. Due mostre a confronto. Arte figurativa in Liguria tra età imperiale e altomedioevo", Genova 2003, pp.49

La *domus* è separata da questa zona pubblica da un *ambitus* non praticabile e da un canale di raccolta delle acque piovane.

Verso nord la casa è circondata da un portico a colonne laterizie mentre nel portico settentrionale si trovava un ingresso originariamente di età repubblicana (40 d.C.).

Lungo il portico orientale non ci sono aperture.

La casa degli affreschi è così chiamata perché possiede una grande quantità di intonaci dipinti che decorano soprattutto un porticato e la parte orientale della casa; la *domus* comprendeva ambienti su tre fasce: occidentale, centrale e orientale.

La prima comprendeva tre vani; il vano di mezzo era adibito a triclinio estivo (locale in cui veniva servito il pranzo, prende il nome dai tre cuscini su cui i padroni di casa e i loro ospiti si sdraiavano), mentre gli altri due avevano destinazioni meno chiare, ma probabilmente erano stanze per gli ospiti.

Il triclinio con ogni probabilità era dedicato oltre che al pranzo anche al riposo, per la vista che aveva sul giardino, fattore che permetteva di avere sempre molta luce in questa zona della casa.

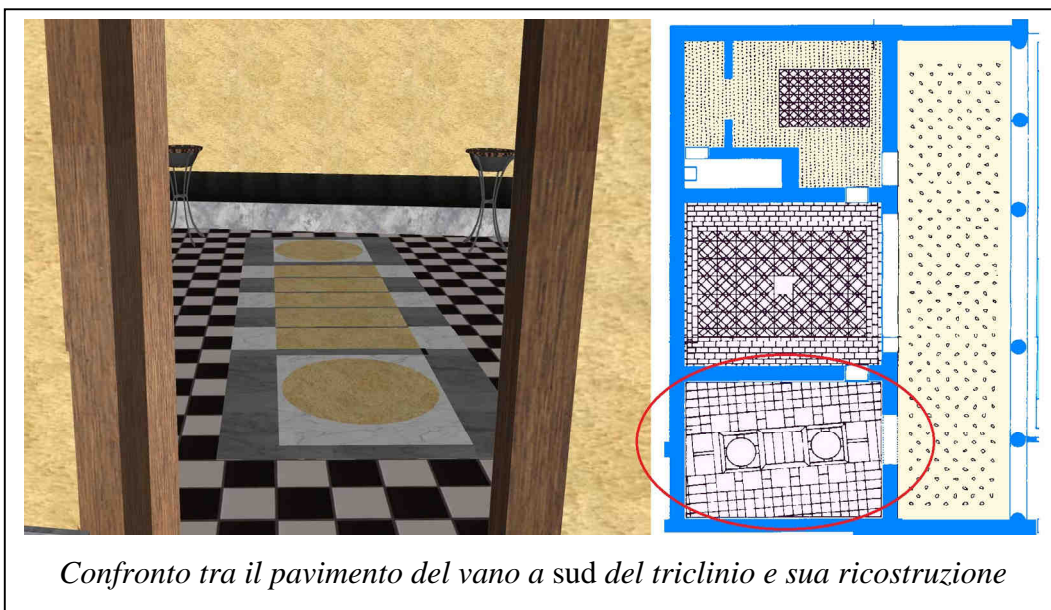
Nella parte centrale della *domus* si trovavano due giardini, di cui uno ricco di fontane e canali di irrigazione ed un portico finestrato.

Nella parte orientale si trovano il triclinio invernale, una camera da letto e un corridoio di fronte ad un giardino ribassato; proseguendo nel corridoio troviamo una stanza con pavimento in mosaico e altre due stanze di uso generico.

La pavimentazione completa e in buone condizioni si è conservata solo nei due triclinii e nella camera da letto.

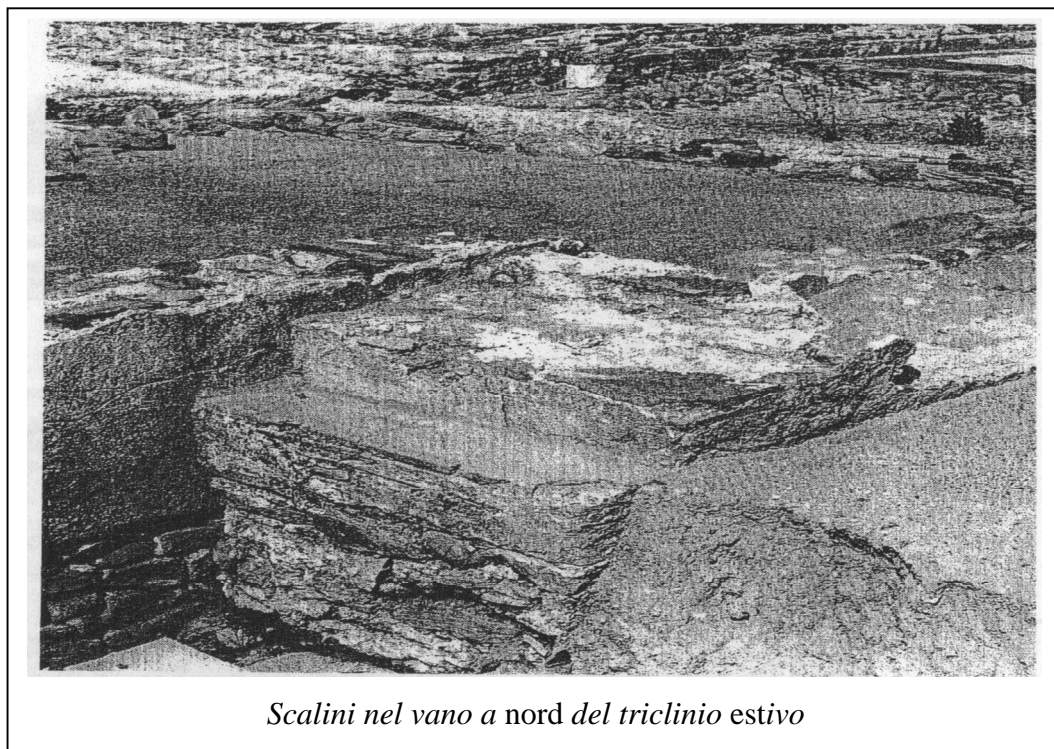
Partiamo con l'analisi degli ambienti occidentali.

Nel vano a sud del triclinio estivo troviamo una decorazione pavimentale che ha al centro una fascia a dischi inseriti in quadrati e stretti rettangoli riuniti a quattro per formare un quadrato in battuto di malta bianca con tessere di marmo rosso, nero e verde disposte irregolarmente.



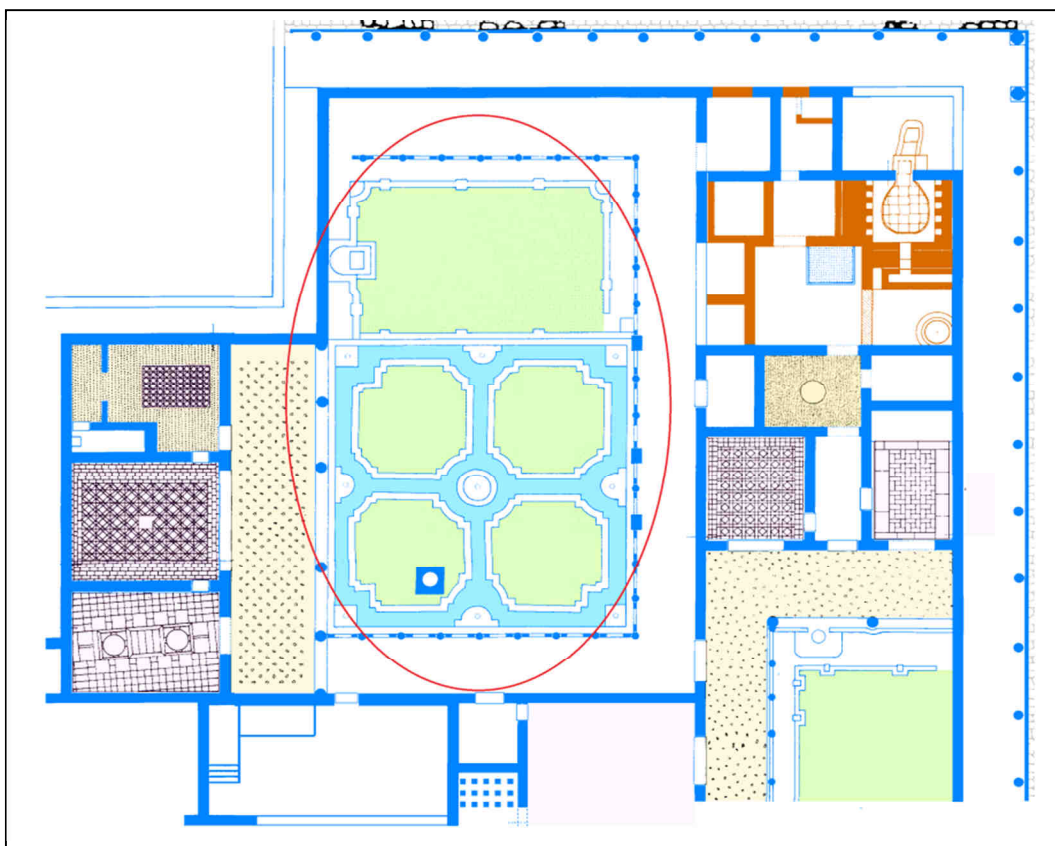
Nel vano a nord del triclinio si trova un canale che si immette nella cloaca massima, dove sono stati trovati alcuni fondi di anfora: molto probabilmente qua si trovava un magazzino di derrate alimentari o una bottega con dietro una latrina.

Sempre accanto a questo vano sono state trovate le tracce di quattro gradini, dalla cui altezza si può dedurre l'altezza del piano terra tra i 3,65 e i 4mt.



La decorazione delle pareti di tutti questi vani è costituita da affreschi con fasce verdi o nere che hanno come base lastre di pietra.

Nella parte centrale della planimetria si evidenziano due giardini.



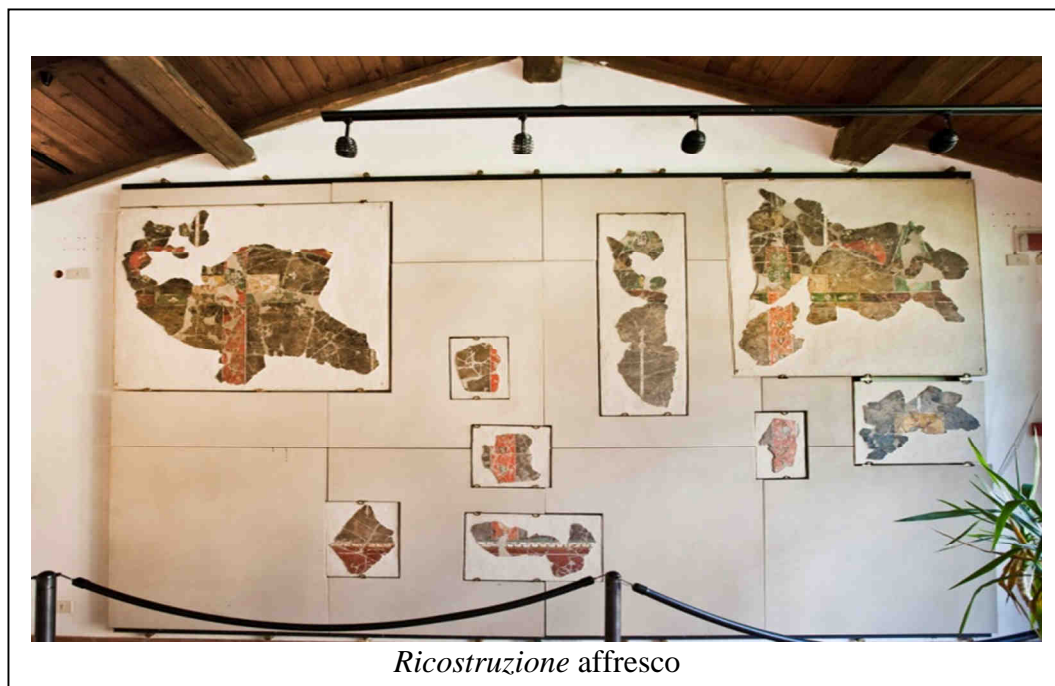
Il primo costituito da quattro aiuole circolari circondate da canali per l'acqua; troviamo sette fontane, una su ogni margine più una al centro.

E' possibile affermare che questi due giardini erano circondati per tre lati da un grande portico a vetri, grazie alle grandi quantità di frammenti di vetro¹⁸ ritrovati tutto intorno.

Percorrendo il portico si arrivava nel braccio nord-est dove si trovavano gli affreschi ritrovati nelle migliori condizioni, anche se gli strati di rinvenimento

¹⁸ Annapaola Zaccaria Ruggiu, "La casa degli affreschi di Luni: fasi edilizie per successione diacronica", in "Centro studi lunensi, Quaderno 8", Lerici 1983,pp.13

degli affreschi (sottili tralci verdi con foglie e fiori e piccoli quadrati gialli e rossi) sono stati modificati ulteriormente durante l'età traiano-adrianea. La parte superiore degli affreschi è difficile da ricostruire in quanto è stata profondamente intaccata da inserti tombali effettuati durante l'età medievale. Dalla ricostruzione dell'affresco della parete nord del portico si è calcolata l'altezza dell'alzato, che corrispondeva a 3,5mt circa.



Più tardi (tra la metà del II sec d.C. e la prima metà del IV sec. d.C.) con il crollo della *domus*, il portico è stato ridipinto, ed è anche per questo che non tutte le zone dell'affresco si sono conservate.

Tuttavia in base all'analisi delle scarse tracce di colore, si sono evidenziate tre componenti principali: zoccolo in basso rosso, fascia orizzontale gialla di mezzo, registro superiore nero.

Il portico interno si collegava con quello esterno che circondava tutta la *domus*; in questo portico erano presenti muri intervallati da colonne dipinte nella parte inferiore di rosso e in quella superiore di nero.

Nella fascia orientale della *domus*, in seguito ad una suddivisione di un atrio di età repubblicana, erano stati costruiti diversi ambienti di servizio, e stanze con ricche pavimentazioni marmoree a disegni geometrici.

Il materiale usato proveniva dalle cave di Luni, dall’Africa e dalla Grecia.

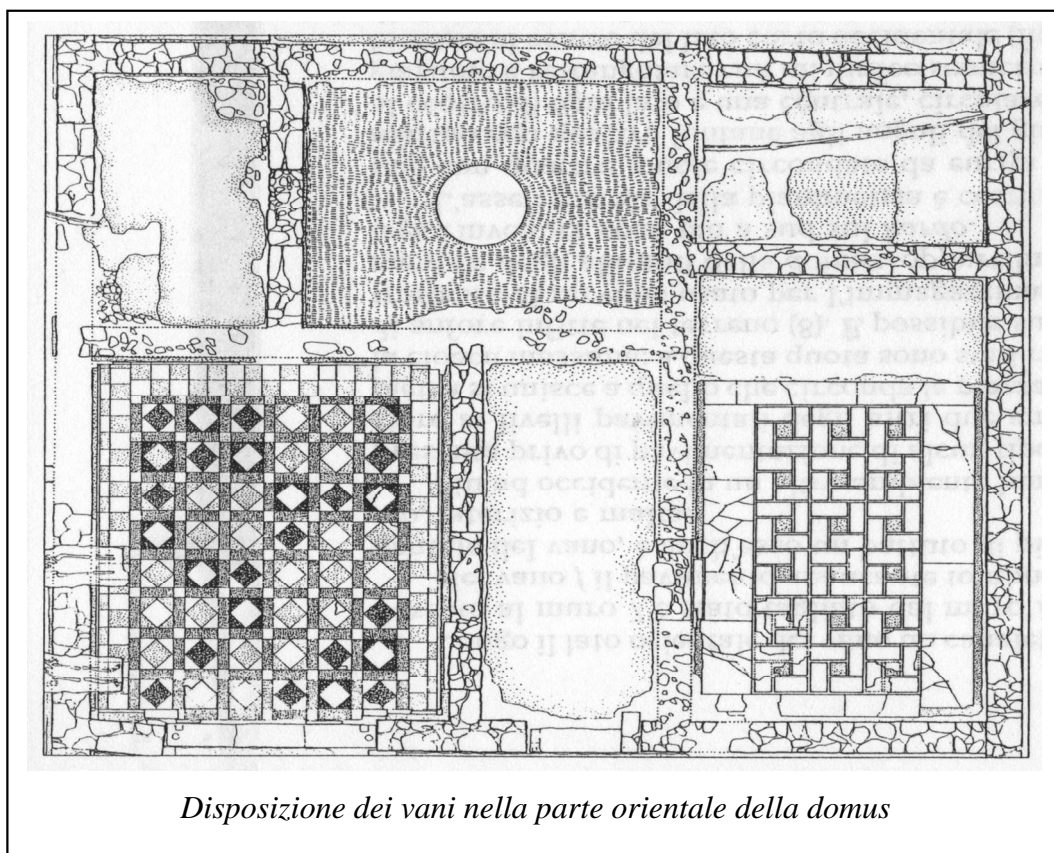
Uno dei suddetti ambienti era molto probabilmente un triclinio invernale per la sua vista privilegiata sul giardino meridionale; è probabile che in questo giardino, sulla parete di separazione del portico, si trovasse una finestra dalla quale poter usufruire durante il giorno dell’illuminazione.

Possiamo pensare che anche questo ambiente del triclinio invernale fosse adibito come quello estivo a sala di riposo diurno e di studio perché isolato dagli altri ambienti.

Dall’altra parte, superato il corridoio troviamo una stanza da letto.

Tra la stanza da letto e il triclinio invernale si trova un ambiente pavimentato a mosaico con tessere rettangolari in marmo bianco al centro di cui si trova un grande disco in marmo di Numidia, probabilmente riferibile ad un triclinio invernale di età repubblicana.

Accanto a questo ambiente è statabbiamo trovato un vano, probabilmente in uso come cucina prima e come archivio dopo.



Nella stanza da letto la pavimentazione è a rettangoli in marmo bardiglio con listelli in marmo rosso del Tenaro (Grecia)

Per la disposizione delle formelle di decorazione del pavimento marmoreo è stato pensato che in questa parte del vano ci fosse un letto in muratura o in metallo.

A nord della camera da letto si apre un vano dove è stata ritrovata una fornace, utilizzata in diverse fasi: come fornace per laterizi (resti di laterizi); come fornace vetraria (da ritrovamento di tessere in vetro); come fornace a calce (reperiti di calce cotta, blocchi di calcare e marmo).

Come possiamo vedere molte volte i reperti si sovrappongono, proprio perché appartenenti a varie fasi di utilizzo della casa.

Vediamo nel dettaglio i ritrovamenti per ogni fase.

I fase¹⁹ (prima metà I sec. a.C.)

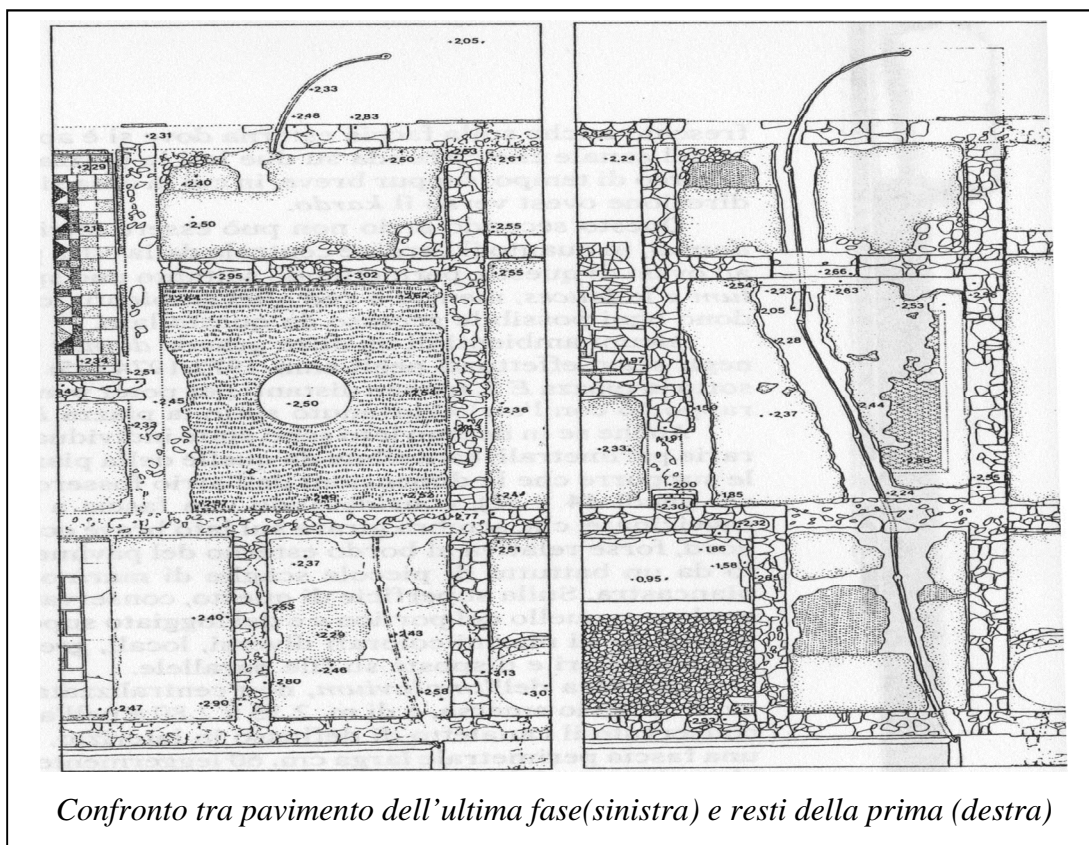
E' possibile dire che la prima fase di utilizzo della *domus* corrisponde a questo periodo grazie al ritrovamento di alcuni lacerti pavimentali in opus signinum²⁰ rinvenuti sotto il pavimento del triclinio invernale e sotto il livello del giardino. Sono stati trovati anche pavimenti risalenti sempre a questo periodo sotto la parte più ad occidente della *domus* che fanno pensare all'esistenza di una *domus* precedente.

Alla casa si accedeva dal portico del decumano minore per una porta di cui si è conservata un'anta nel muro; attraverso questa si passava nell'atrio orientale adiacente ad una sala adibita a tablino, anch'esso dotato di una pavimentazione sempre in opus signinum e tessere cubiche di marmo bianco.

Dal *tablinum*²¹, attraverso una porta si passava al portico, dotato di colonnine e finestre affacciate sul giardino.

¹⁹ Annapaola Zaccaria Ruggiu, "La casa degli affreschi di Luni: fasi edilizie per successione diacronica", in "Centro studi lunensi, Quaderno 8", Lerici 1983, pp.17

²⁰ Tipo di calce molto resistente e impermeabile



Delimitava il giardino un basso muro costituito da lesene (pilastri con sommità arrotondate, probabili appoggi di vasi, statue e altri tipi di decorazione).

Lungo l'esterno del muretto del portico si trovava un canaletto per la raccolta delle acque piovane, successivamente smantellato.

Da alcuni saggi effettuati all'interno del giardino, si può capire che dal portico finestrato si poteva accedere allo stesso attraverso un'ampia porta, situata al centro del braccio più lungo del portico; sempre da questi saggi si evince che il pozzo collocato nella parte inferiore del giardino apparteneva allo stesso periodo.

Vari indizi ci fanno presupporre che il giardino centrale risalga all'età repubblicana: la posizione delle porte del tablinum (a ovest); la posizione della porta al centro del portico; il rinvenimento sotto la pavimentazione di una fistola che portava l'acqua corrente alle fontane e agli *euripi* (canali).

Successivamente la porta occidentale del tablinum e quella d'accesso al giardino sono state chiuse.

²¹ Locale adibito a salotto solitamente posto in fondo all'atrium

Il portico che circondava su tre lati il giardino aveva in origine poche finestre abbastanza piccole visto che il muro aveva uno spessore ridotto.

Durante lavori di scavo più approfonditi su queste zone, sono stati rinvenuti reperti pavimentali (frammenti di resti a vernice nera e rossa) forse datati tra il I sec. a.C. e gli inizi del I sec. d.C.

Si sono trovati anche i ciottoli e laterizi pressati riferibili alla costruzione del portico e di un pavimento asportato in seguito.

Nella zona occidentale del giardino sono stati rinvenuti resti di un crollo, ma non è possibile capire con certezza quanto questo sia avvenuto, visto che i reperti sono troppo esegui a causa di numerose sepolture medievali.

II fase ²²(40 d.C.)

Gli scavi del 1974 hanno messo in luce l'esistenza di una fase intermedia tra quelle di età repubblicana e quelle attribuite al periodo di Claudio o Nerone (III fase).

Questa affermazione è dovuta ad un saggio di scavo sotto la pavimentazione della piazza, adiacente alle tre sale dove si trova il triclinio invernale che ha messo in luce un altro atrio.

Questo secondo atrio non poteva appartenere ad altra *domus*, in quanto il cardine centrale della città passava ad ovest di questo lasciando solo spazio per il vestibolo, mentre a sud altri ambienti (forse botteghe) chiudevano ogni possibile accesso.

Dove nella III fase troviamo la fornace è presente un atrio con un' *impluvium* (vasca quadrangolare a fondo piatto progettata per raccogliere l'acqua piovana) non centralizzato avente una fascia perimetrale rivestita di lastre di marmo e ribassata rispetto al pavimento dell'atrio.

²² Annapaola Zaccaria Ruggiu, "La casa degli affreschi di Luni: fasi edilizie per successione diacronica", in "Centro studi lunensi, Quaderno 8", Lerici 1983, pp.17

Le dimensioni dell'atrio sono ricavate sempre dal rapporto larghezza-lunghezza proposte da Vitruvio (con questo calcolo l'altezza avrebbe dovuto essere $\frac{3}{4}$ della lunghezza).

Il vano che nella III fase venne trasformato in un triclinio invernale in questa fase costituiva il *tablinum* aperto sia sull'atrio sia sul portico e le sale erano probabilmente sale da ricevimento aperte sullo stesso portico.

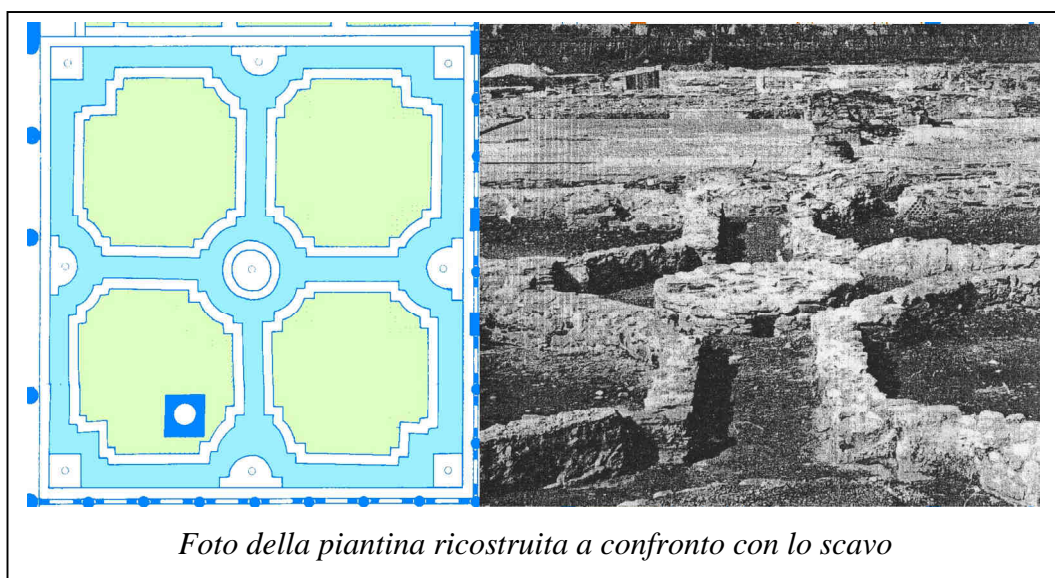
Dalla pavimentazione delle due sale laterali rimangono solo le impronte nella malta, mentre è ben conservata quella centrale; tutto quello che è arrivato ai giorni nostri è databile all'età claudio-neroniana.

In conclusione si può supporre che l'atrio sia stato costruito dopo l'età tiberiana (37-40d.C.) e la sua demolizione sia avvenuta nel 50d.C.

Per avere un panorama completo ci si rifà sempre ai calcoli vitruviani; l'ipotesi dello schema ricostruttivo è parziale, visto che non si hanno riferimenti relativi a questo periodo per quanto riguarda la parte centrale (giardino e peristilio) e orientale (ambienti con pavimenti).

Si può solo presumere che l'atrio repubblicano non fosse più in funzione ma che fosse stato diviso in tanti vani di servizio.

La divisione del giardino in due parti corrisponde proprio a questi anni.



III fase ²³(50-70 d.C.)

E' da considerare un'ulteriore fase edilizia, nella quale è opinabile il ridimensionamento della *domus* con la perdita di alcuni vani a ovest e la costruzione pavimentale dei vani adiacenti ai triclinii.

Contemporaneamente è probabile la chiusura dell'apertura del portico alla quale si accedeva attraverso il triclinio estivo.

Sopra questo triclinio e sui vani adiacenti si costruisce un piano superiore con loggiato cui si accedeva da una scala: quest'ultima era situata nel vano adiacente alla stanza a sud del triclinio estivo.

Gli scavi mostrano che la costruzione dei muretti dei giardini risale alla metà del I sec. d.C. e che a parte qualche modifica sono rimasti immutati fino alla III fase. Come detto sono presenti due giardini.

Il giardino a sud è costituito da quattro aiuole circondate da euripi rivestiti di marmo bianco delle cave lunensi; di questi canaletti rimane solo la malta molto spessa, che conserva le impronte delle lastre saldate con graffe di bronzo.

Agli angoli nord-ovest, nord-est e sud-ovest, sud-est si trovano quattro fontane di forma quadrata: solo quella che si trovava a sud-est si è conservata quasi integralmente; ad ogni metà di tutti i lati perimetrali troviamo altre quattro fontane semicircolari.

Al centro del giardino troviamo una fontana circolare che riceve l'acqua dagli *euripi*; è da notare che la parte superiore della fontana sostiene una vasca bassa di marmo con una capienza d'acqua molto ridotta; probabilmente sotto questa era poggiato il bacino della fontana vera e propria.

I muretti delle aiuole e le spallette degli *euripi* sono di materiale composto e di reimpiego come pietre scistose, laterizi, frammenti di sculture, ceramiche.

All'interno degli *euripi*, per la tenuta idraulica, si trovano strati di cocciopesto e malta.

²³ Annapaola Zaccaria Ruggiu, "La casa degli affreschi di Luni: fasi edilizie per successione diacronica", in "Centro studi lunensi, Quaderno 8", Lerici 1983, pp.23

Grazie alla stratigrafia è stata notata la mancanza di *humus*, per questo si pensa che l'area non fosse coltivata ad orto, ma molto probabilmente a prato e fiori.

Il periodo di uso del giardino, come risulta dai ritrovamenti, viene documentato dal I al III sec. d.C.; mentre alla fine del IV secolo è stato abbandonato, come il resto della casa.

Sempre tra il 50 e il 70 d.C. nel giardino a nord venne costruita una nicchia semicircolare nella quale si trovava un ninfeo più alto del muro di cinta e un bacino di fontana per lo scarico dell'acqua.

Il materiale ritrovato fa intendere che la nicchia era rivestita di lastre marmoree e di tessere vitree forse per realizzare un mosaico.

Dall'osservazione del materiale relativo al piano di calpestio (ceramiche del I sec d.C.) e mancando materiale del II e III sec. d.C. si deduce che questo spazio fosse molto praticato e perciò più pulito.

Si suppone anche che nel giardino fosse collocato un larario (luogo destinato al culto Lari).

Il culto dei Lari, che prima veniva praticato nell'*atrium* e nel *tablinum*, più tardi si stabilì all'esterno nel *viridarium* (giardino) e nel peristilio.



Busto principe Tiberio

E' stata rinvenuta una lastra decorata a rilievo nel giardino nord, dove si nota l'effigie del dio Silvano che tiene in una mano una roncola e nell'altra un ramo frondoso; al suo fianco si trovano un cane e un'ara sacrificale.

Nelle campagne di scavo del 72-74 è stato rinvenuto un ritratto in marmo di un principe della famiglia claudia, probabilmente Tiberio.

In questa fase i due giardini sono indipendenti, questo è dimostrato dalla costruzione di un pilastro nel muro, che fa pensare alla volontà di dividere le aree.

Riguardo la fascia orientale della *domus* si ritiene che l'atrio repubblicano fosse già stato suddiviso in diversi ambienti di servizio.

In generale comunque, per quanto riguarda questo periodo, i tipi litologici e le tecniche di lavorazione impiegata, fanno intuire che il proprietario apparteneva alla classe imprenditoriale legata all'estrazione del marmo; questo è confermato anche dal fatto che nella *domus* mancavano ambienti legati ad attività agricole.

Scavi recenti (2006), aggiunte, correzioni e approfondimenti

Porta Occidentale²⁴ della *Domus* degli affreschi

I prelievi dei depositi archeologici riguardo la porta occidentale denotano un sistema difensivo repubblicano con interventi di restauro, dimostrato anche dal fatto che il fronte interno delle mura è connesso con una torre.

La parte inferiore costruita con tecniche e materiali diversi (grandi e piccoli blocchi squadrati) è stata riportata alla luce per un elevato di mt 2,50.

La parte superiore, ritenuta una tarda riedificazione conservata per un'altezza di mt. 2,70, è realizzata in scisti (trasformazione di argilla sottoposta ad alte pressioni) legati con malta.

Nei livelli soprastanti il piano di cantiere è stato rinvenuto un'orlo di anfora greco-italica tarda, collocabile nella prima metà del secondo secolo a.C.

E' stata trovata anche una lucerna biconica a vernice nera e altre importazioni di prodotti iberici come ceramiche da cucina e da fuoco.

Si rilevano anche molte anfore vinarie (poiché a Luni si produceva il miglior vino d'Etruria), contenitori da trasporto e piatti da mensa.

Settore Nord-Ovest²⁵ della *Domus* degli affreschi

L'area, contaminata da scavi ottocenteschi, ha conservato l'impianto repubblicano del cardine (via che percorre la città nel senso nord-sud posta in un sistema urbanistico ortogonale suddiviso in isolati quadrangolari) e del massimo decumano (via che percorre la città nel senso est-ovest) della città.

Le strutture di un edificio con quattro ambienti realizzati in ciottoli, non legati da malta, si trovano tra il cardine e il decumano.

In uno di questi ambienti è stato ritrovato un focolare e in quello adiacente si trova un bancone realizzato in spezzoni laterizi, ciottoli, pietre scistose e calcaree.

Gli affreschi²⁶

La prima ristrutturazione effettuata sugli affreschi risale alla seconda metà del I secolo d.C., dopo una caduta dell'intonaco del braccio nord della *domus*, dovuta ad un sisma seguito da un'alluvione; nonostante tutto dal deposito di materiale degli intonaci rinvenuti è stata scoperta la posizione esatta dei motivi decorativi. Grazie ad un quadro più chiaro, è stato possibile ricostruire il motivo dell'affresco che è stato prima disegnato a mano e poi digitalizzato.

Si nota uno zoccolo a fondo rosso con pannelli gialli e verdi e una parte mediana, a fondo nero; nella parte superiore si trovano frammenti decorativi, forse più significati: fiori di loto con lunghi steli e tralci di rose selvatiche (rose canine).



Dettaglio dell'affresco

24-25-26 Gervasini L., Durante A.M., Gambaro L., Landi S., “Luna e l’ager lunensis: nuovi elementi per la conoscenza della città antica e del territorio romanizzato fra il golfo della Spezia e il portus Lunae.” In “Atti delle Giornate di Studio Forme e tempi dell’urbanizzazione nella Cisalpina (II secolo a.C.-I secolo d.C.)”, Torino 2006, pp 163-170

10.2-La Domus degli affreschi di Luni

Come già accennato la *domus* è stata ricostruita fino dove sono stati portati avanti gli scavi attuali.

Essendo la zona ancora da mettere interamente in luce si è preferito non azzardare ipotesi sul resto della *domus*, ma rappresentare solo ciò di cui si disponevano indicazioni e dati certi.

Tramite una piantina prodotta a seguito degli scavi, è stata possibile la realizzazione della struttura in maniera molto fedele a quella che si pensa essere stata la *domus*, almeno per le proposizioni.

Per la ricostruzione di affreschi, decorazioni e suppellettili ci si è basati sugli studi fatti sui reperti.

Nella nostra ricostruzione virtuale è stato scelto di riprodurre la fase più “recente” della *domus*, ovvero quella relativa alla seconda metà del I secolo d.c. , in quanto la più interessante dal punto di vista storico per la grande quantità di reperti risalenti a questo periodo²⁷.

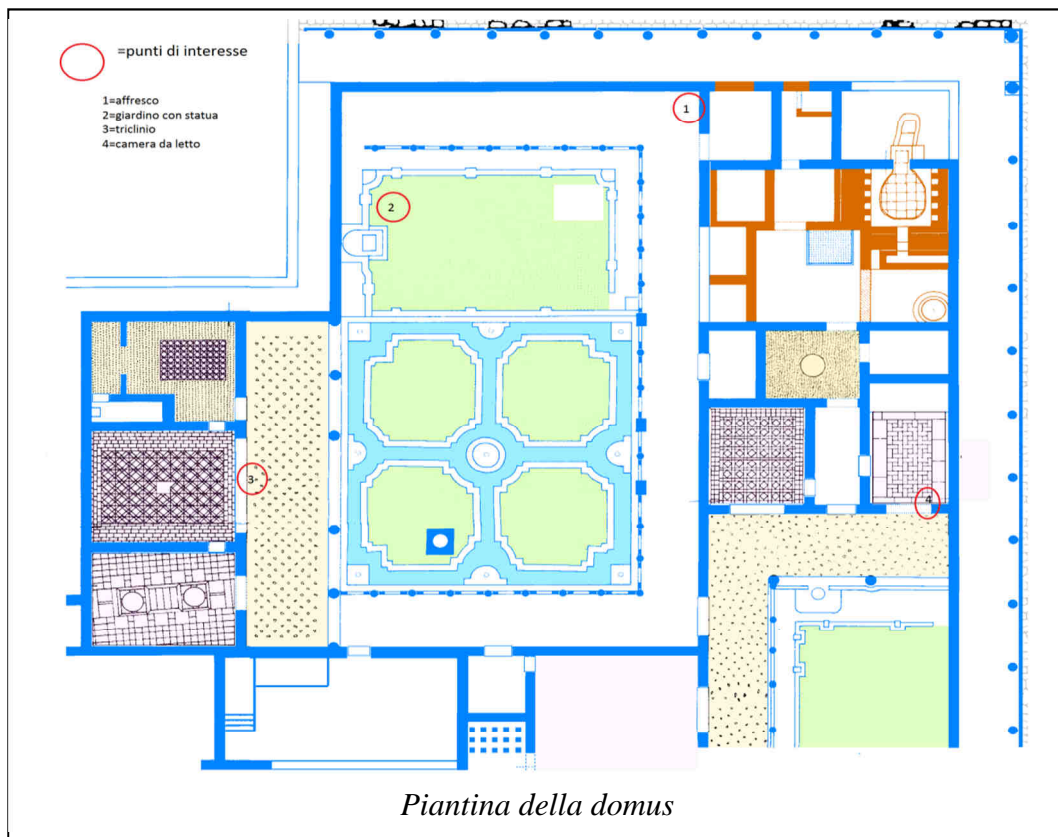
All'interno della ricostruzione troviamo quattro punti di interesse dove il visitatore può ricevere informazioni direttamente da un personaggio, che rappresenta il padrone di casa su vari canali: direttamente con il parlato, con testo aggiuntivo e con foto reali relative allo scavo; di questi punti di interesse vedremo approfondimenti dedicati.

La *domus* è composta da quattro parti principali: il triclinio estivo e le due stanze adiacenti a est, il triclinio invernale e la camera da letto a ovest, i due giardini al centro, il vano delle scale per il loggiato e un giardino a sud e per finire il portico affrescato a nord.

Nell'immagine vediamo una piantina completa.

²⁷ Ceramiche di tutti i tipi, sia da mensa che per l'igiene personale. A.M.

Durante, “Appunti per lo studio degli intonaci dipinti della *Domus* degli Affreschi” , in “Romana pictura e christiana signa. Due mostre a confronto. Arte figurativa in Liguria tra età imperiale e altomedioevo” , Genova 2003, pp.49



Per i giardini è stato possibile ipotizzare la presenza di vasi proprio per la forma dei muretti, che presentavano basi di appoggio lungo tutta la lunghezza; inoltre il ritrovamento di un vaso in più parti conferma quest'ipotesi.

Nel giardino nord troviamo una nicchia, probabilmente dedicata al culto del dio Silvano, di cui nella ricostruzione 3d è disponibile un approfondimento.



In tutti i giardini troviamo dei pozzi, dedicati alla raccolta dell'acqua, ornati da volti e motivi floreali; questi pozzi poggiano su basi quadrangolari in marmo.



Struttura dei giardini com'è attualmente

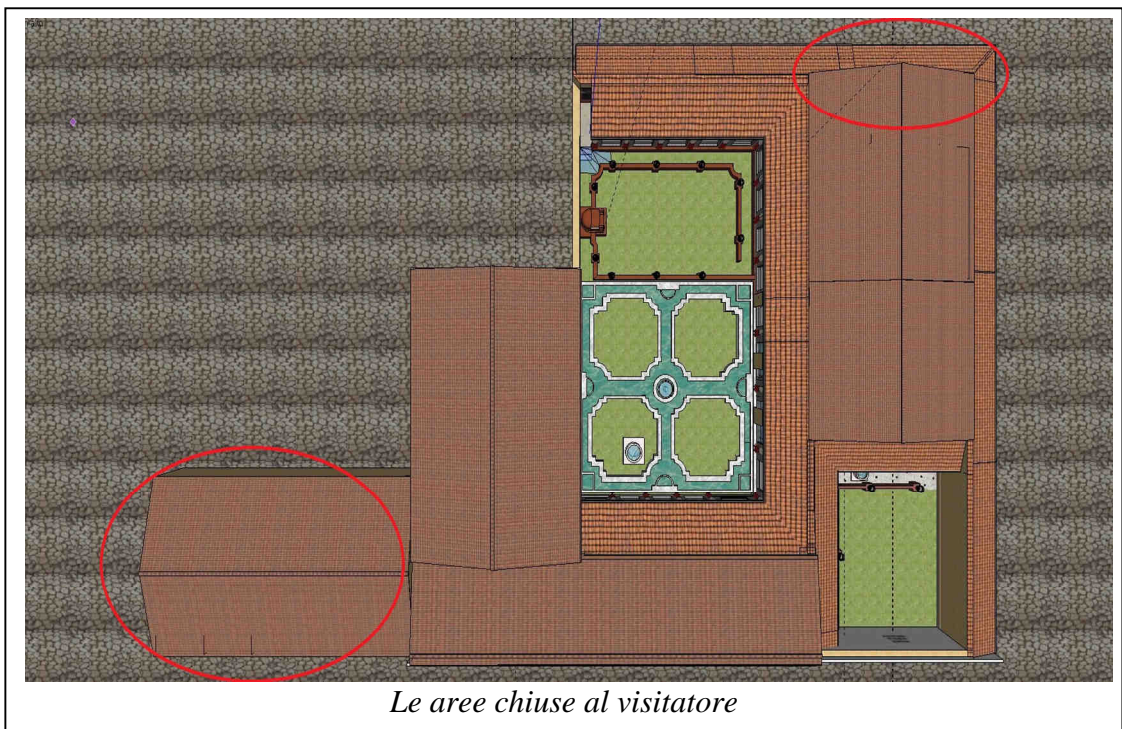
La strada all'esterno è stata realizzata in basolato, proprio perché ancora oggi è possibile vedere queste pietre intorno ai resti della *domus*.



Basolato

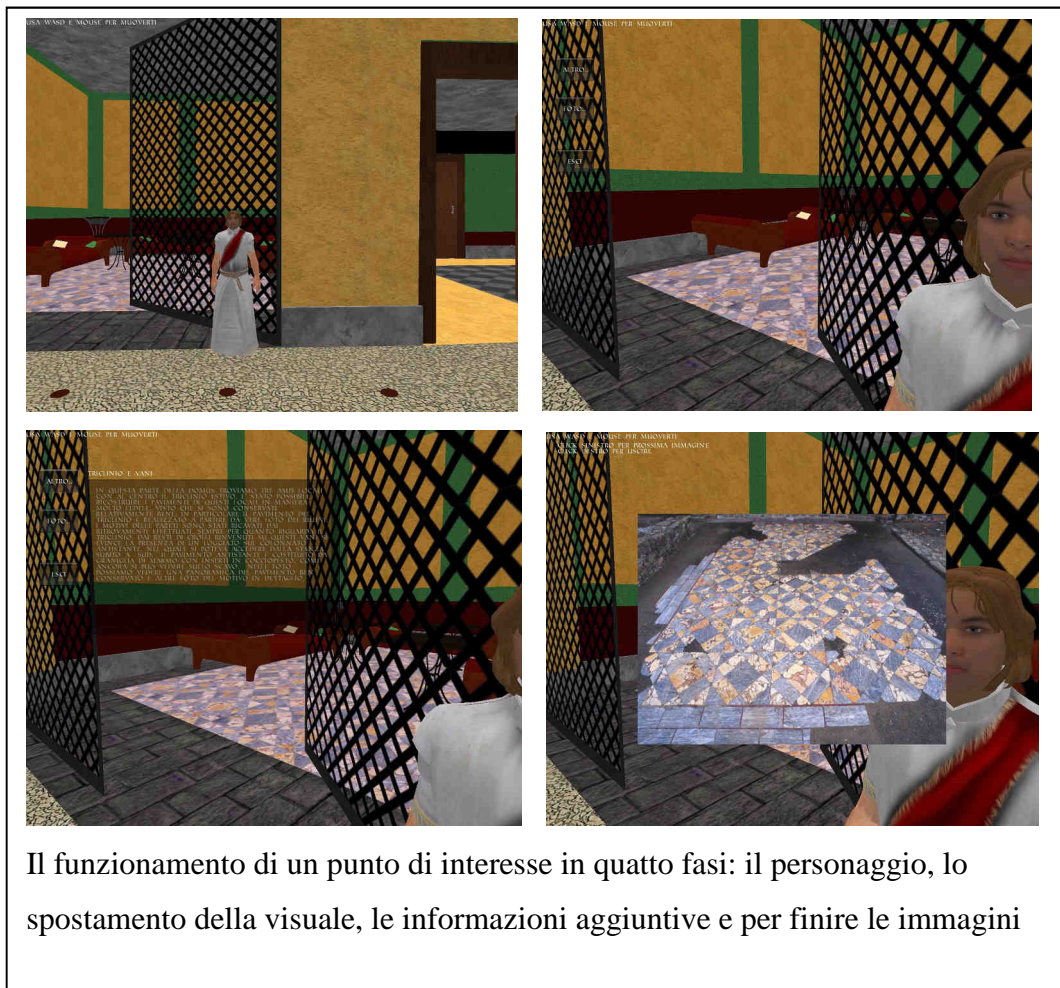
L'ambiente est della casa è stato ricostruito, ma non è aperto al visitatore; questo perché oltre il perimetro murario non è stato trovato molto materiale interessante, quindi si è preferito tenere chiusa quest'area, per non confondere con un ambiente troppo esteso il visitatore virtuale.

Un'altra parte, riprodotta ma attualmente non aperta al pubblico è la parte nord-est, dove si trovava una fornace, che si affacciava all'esterno; sempre per non disperdere troppo l'area virtuale visitabile si è preferito non riprodurla, visto che comunque non apparteneva al perimetro interno della casa.



10.3 - I punti di interesse

Per punto di interesse si intende un punto dove troviamo il personaggio principale, che rappresenta il padrone della *domus*; avvicinandoci ad esso la visuale si sposta sull'oggetto del discorso e possiamo interagire con lui, ascoltando la sua presentazione dell'ambiente in cui ci troviamo. Inoltre è possibile leggere informazioni aggiuntive relative allo scavo tramite un'interfaccia a video e vedere foto reali dello scavo.



I punti di interesse all'interno della ricostruzione sono quattro:

Punto di interesse 1 : Gli affreschi

Proprio da questi affreschi deriva il nome che è stato dato all'edificio, "*Domus degli affreschi*".

Grazie al ritrovamento di numerose parti murarie affrescate, è stato possibile ricreare i motivi e le decorazioni in maniera molto fedele sia per quanto riguarda il disegno che per i colori.

Nelle foto possiamo vedere sia i resti murari rinvenuti, composti nell'ordine esatto all'interno di un ala del museo.



Motivo dell'affresco ridisegnato a confronto con il muro ricostruito del museo

L'affresco è composto da una parte bassa di colore porpora con motivi romboidali lungo tutta questa parte; salendo troviamo una prevalenza del colore nero interrotto nella parte più bassa da una fantasia con piccoli disegni su sfondo verde, ocra o porpora.

Nella parte in alto invece il motivo ha sempre piccoli disegni ma di tipo floreale e il motivo stesso è di questo tipo; il nero prevale sempre ma anche qua compaiono l'ocra, il porpora e il verde.

Questo motivo è stato ridisegnato come immagine digitale seguendo accuratamente i resti e la ricostruzione del muro che è stata effettuata all'interno del museo.

Una volta ottenuta la *texture* questa è stata applicata sulla parte del modello 3d corrispondente al corridoio sul lato nord della *domus*.

Punto di interesse 2: Giardino con ninfeo e giardino con fontane

Dagli scavi è emerso che lungo il muretto del giardino nord si trovava una piccola nicchia, molto probabilmente dedicata al riparo di una statua di Tiberio Gemello.

Sempre nel giardino è stato rilevato un rilievo marmoreo del dio Silvano raffigurante la divinità accanto ad un ara e un cane.

Nella parte sud invece troviamo una serie di canali marmorei dedicati all'irrigazione dei giardinetti centrali.

Al centro, tra questi giardinetti, racchiusi tra i canali, vi era uno spazio dedicato ad una statua o una fontana.

Tutto il giardino interno è racchiuso in un colonnato con ampie vetrate da cui era possibile avere una buona visuale su tutta la parte esterna.

Tra le foto è possibile osservare oltre ad un'immagine di Tiberio Gemello anche il rilievo marmoreo del dio Silvano.

Inoltre si può ammirare una ricostruzione dall'alto della villa su acquerello per osservare meglio la disposizione dei giardini.



Rilievo del dio Silvano

Punto di interesse 3: Il triclinio estivo

In questa parte della *Domus* troviamo tre ampi locali con al centro il Triclinio estivo (ossia il locale in cui veniva servito il pranzo nei mesi più caldi).

E' stato possibile ricostruire i pavimenti di questi locali in maniera molto fedele, visto che si sono conservati relativamente bene; in particolare il pavimento del triclinio è realizzato a partire da vere foto dei rilievi.

Dai resti di crolli rinvenuti su questi vani si evince la presenza di un loggiato sul colonnato antistante, nel quale si poteva accedere dalla stanza subito a sud.

Il pavimento davanti alle stanze è costituito da graniglia di marmo con inserti in cocciopesto, come ancora si può vedere sullo scavo.

Nelle foto possiamo vedere una panoramica del pavimento ben conservato e altre foto del motivo in dettaglio.



Pavimento del triclinio estivo

Punto di interesse 4: Il triclinio invernale e la camera da letto

In questa parte della *domus* si trova un'ampia camera da letto e il triclinio invernale, che si affacciano sul giardino antistante, ribassato di qualche decina di centimetri.

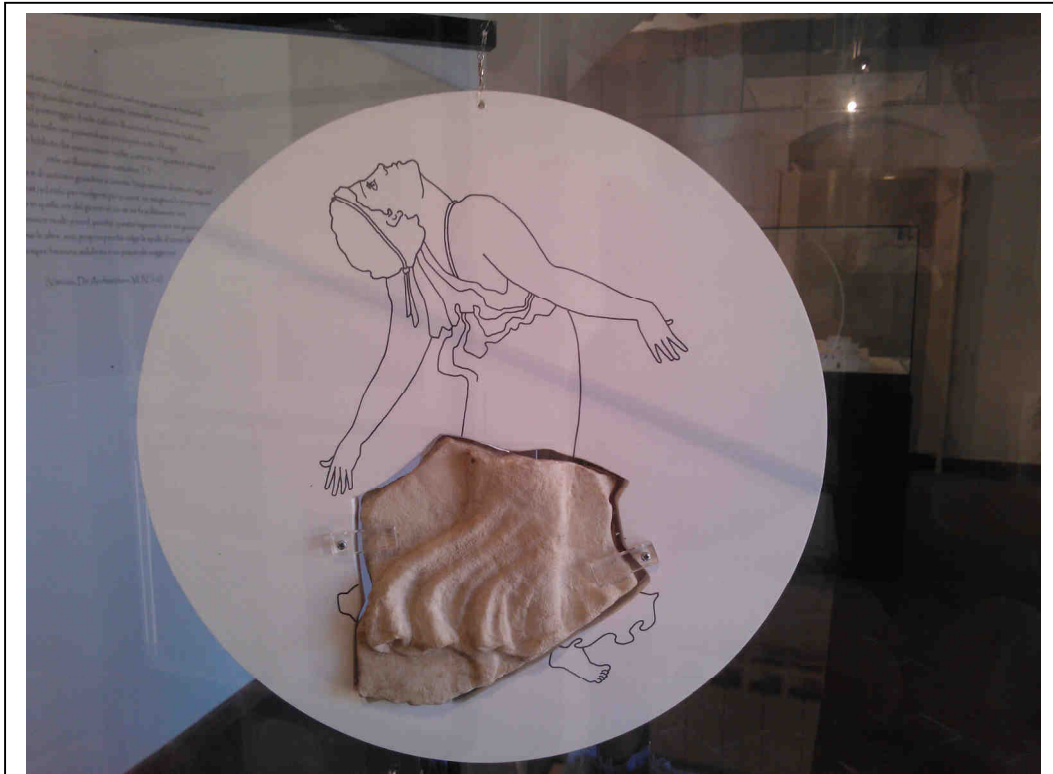
Anche in questi ambienti, come nel triclinio estivo è stata possibile una ricostruzione molto accurata dei pavimenti in quanto si sono conservati molto bene, permettendo anche qui l'utilizzo di foto attuali.

Se si procede nel corridoio è possibile osservare la stanza con pavimentazione a mosaico; al centro del pavimento si trova un disco marmoreo ben conservato, ricostruito sempre con l'utilizzo di vere immagini.

Tra le foto possiamo vedere varie panoramiche e dettagli dei pavimenti, ma non solo: vediamo nel dettaglio i ritrovamenti e i disegni su cui sono state ricostruite le oscilla, piccoli oggetti decorativi appesi tra le colonne, che appunto oscillavano con il vento.



Pavimento della camera da letto



Oscilla

10.4-I locali e l'arredamento

Oltre che i punti di interesse la *domus* era composta da molti altri vani, ed ognuno è stato ricostruito o valutando i dati relativi agli scavi, oppure, dove i dati sono risultati mancanti, paragonando ambienti simili di altre *domus* dello stesso periodo²⁸.

Analizziamo ogni ambiente per come è stato ricostruito, partendo dal lato nord-ovest e procedendo in senso antiorario; nella figura vediamo la numerazione dei vani, per capire meglio la loro disposizione:



²⁸ Casa di Nettuno e Anfite a e Casa del bicentenario a Ercolano.

A.Maiuri, "Ercolano, i nuovi scavi (1927-1958)", Roma 1958

1- Stanza nord triclinio

In questa stanza non si è potuto ricreare il pavimento a partire da foto reali, in quanto troppo rovinato e privo di una serie di mattonelle ben conservate. Grazie allo studio del motivo²⁹ è stato tuttavia possibile ricreare una *texture* molto somigliante a quello che doveva essere in realtà; la resa definitiva è costituita da un riquadro centrale con mattonelle romboidali e da una cornice di mattoni di pietra.

La medesima cosa è stata fatta per lo zoccolo alla base della stanza, ricavato dai frammenti in pietra rilevati sul luogo dello scavo.

La colorazione delle pareti è stata effettuata valutando i ritrovamenti di frammenti colorati di muro; il motivo “a fasce” è stato pensato dopo un confronto con *domus* pompeiane³⁰.

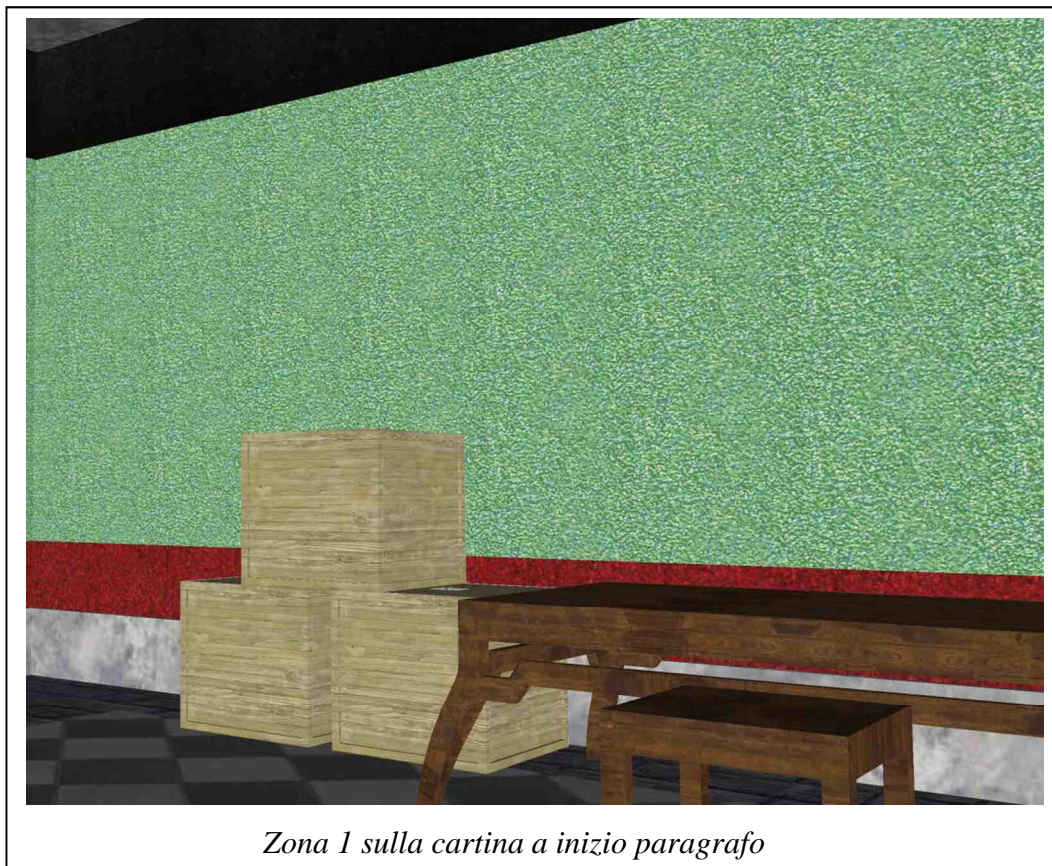
L’arredamento di questa stanza, così come le porte, è molto semplice, perché dietro la porta si trovava un collegamento con la cloaca massima, quindi probabilmente questo locale era antistante alla latrina; troviamo solo un tavolo con uno sgabello senza decorazioni e qualche cassa, visto che è molto probabile che questo locale fosse una sorta di magazzino.

Il magazzino poteva essere arredato con più oggetti ma si è preferito non sbilanciarsi verso rappresentazioni troppo dettagliate che potevano risultare inaccurate.

²⁹ A.M. Durante, “Appunti per lo studio degli intonaci dipinti della *Domus* degli Affreschi”, in “Romana pictura e christiana signa. Due mostre a confronto. Arte figurativa in Liguria tra età imperiale e altomedioevo”, Genova 2003, pp.49

³⁰ Casa di Nettuno e Anfrite a Ercolano.

A.Maiuri, “Ercolano, i nuovi scavi (1927-1958)”, Roma 1958



2- Il triclinio estivo

Questo vano è una delle stanze più importanti della casa, proprio per la sua collocazione con vista diretta sul giardino interno; sempre per la sua collocazione si deduce che sia il triclinio estivo, e proprio per questo è stata collocata al suo ingresso una porta a griglia di metallo, sempre riferendosi alle *domus* pompeiane³¹.

Qui i pavimenti si sono conservati in maniera eccezionale, tanto che è stato possibile riprodurli direttamente da foto dello scavo; anche qua troviamo la parte centrale del pavimento costituita da mattonelle in marmo bianco e nero con particolari fantasie e una cornice esterna in mattoni di pietra.

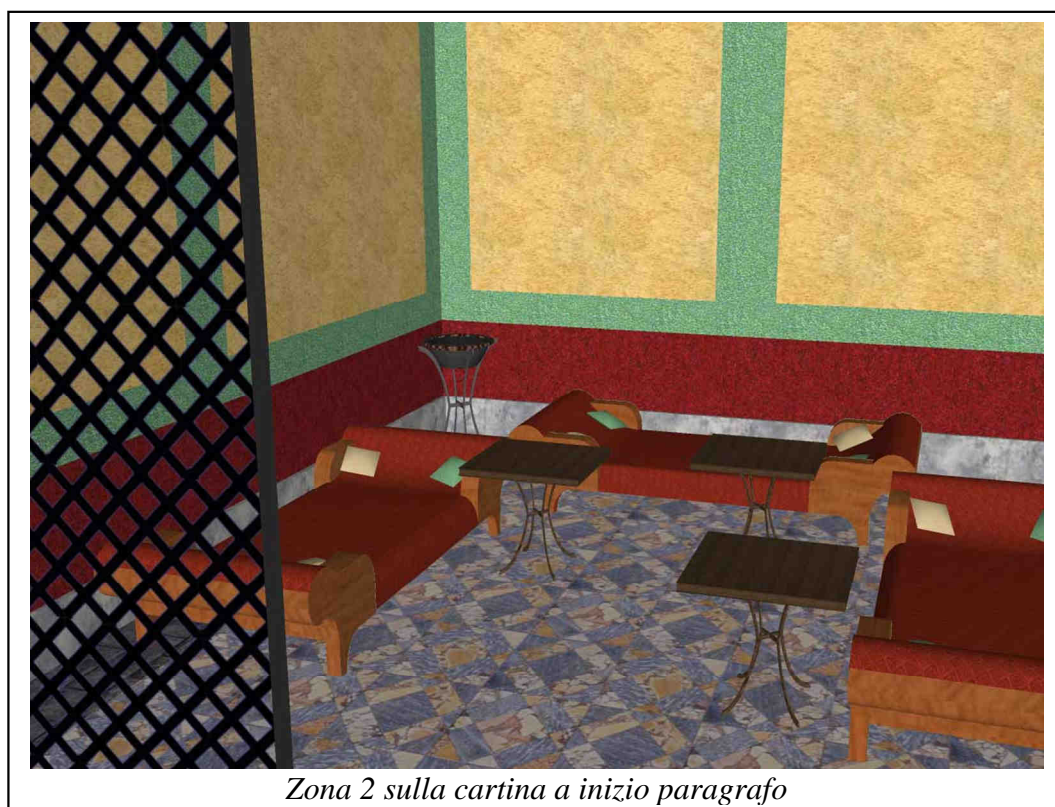
³¹ Casa del bicentenario a Ercolano.

A.Maiuri, "Ercolano, i nuovi scavi (1927-1958)", Roma 1958

Il motivo sulle pareti è stato ricavato analizzando i vari frammenti di colori diversi ritrovati all'interno di questo vano.

L'arredamento è quello del classico triclinio romano con tre divanetti disposti verso il centro della stanza dove troviamo i tavolini che venivano usati per il cibo.

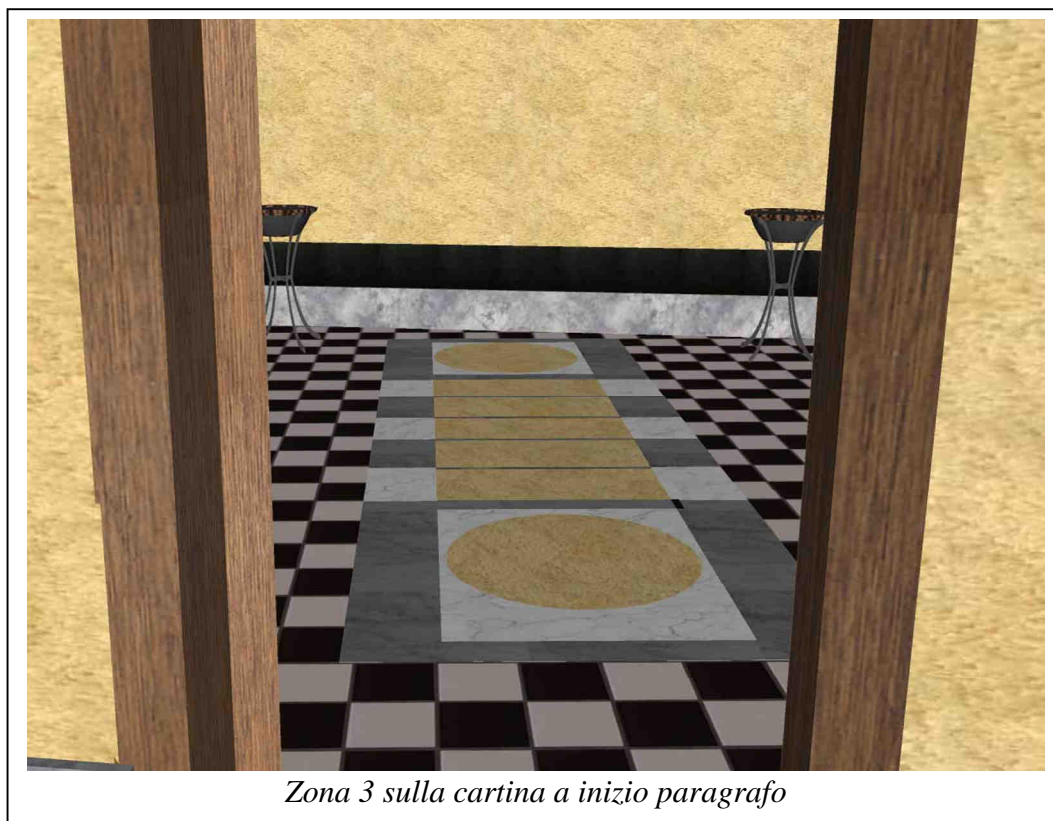
Il pavimento del colonnato antistante è in graniglia di marmo con inserti in cocciopesto, come si può vedere direttamente sullo scavo.



La destinazione di questa stanza non è molto chiara, probabilmente era la stanza degli ospiti, ma non sono stati trovati oggetti che confermassero quest'ipotesi. Quello che è certo è il pavimento, che anche se non si è conservato bene quanto quello del triclinio, mostra chiaramente la sua conformazione; lungo tutti i bordi della stanza vediamo una scacchiera di mattonelle, mentre al centro si trova un disegno particolare: una serie di mattonelle in marmo bianco e nero che racchiude alle estremità due dischi sempre in marmo.

Proprio perché non si era sicuri di cosa effettivamente ci fosse in questa stanza si è preferito arrearla solo con oggetti per l'illuminazione.

La “fascia ” nera sul muro è ricavata dai frammenti di colore nero ritrovati, così come per lo zoccolo in pietra.



Zona 3 sulla cartina a inizio paragrafo

4- Le scale e il loggiato

Gli archeologi, durante gli scavi in questa zona si sono trovati di fronte a più frammenti stratificati; questo ha fatto subito supporre la presenza di un loggiato sopra il colonnato antistante alle tre stanze del triclinio.

Per accedervi si potevano percorrere le scale che si trovavano nel vano adiacente alla stanza a sud del triclinio; questa stanza non presenta particolare arredamento, se non una ringhiera per le scale, sicuramente presente e qualche oggetto per l'illuminazione.

Il pavimento è continuo rispetto a quello esterno, in graniglia di marmo.

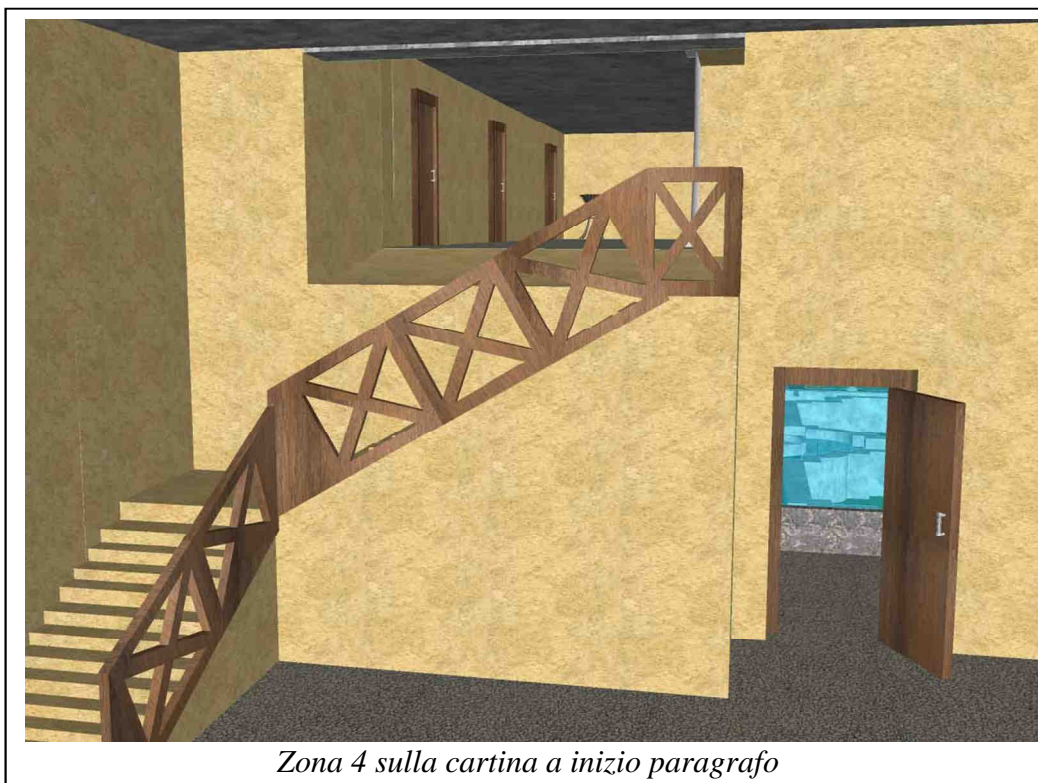
Questa stanza corrisponde al limite massimo degli scavi effettuati fino ad oggi, e si suppone che non solo il vano continui, ma che vi siano altre stanze ancora.

Salendo troviamo il loggiato, che era costituito da tre stanze in corrispondenza delle tre stanze al piano di sotto: non sono state ricostruite in quanto il crollo

totale permetteva solo di far capire la presenza del loggiato, ma non come fosse formato all'interno.

Anche qua troviamo una ringhiera di legno, per proteggere tutto il loggiato.

Potevano essere aperte le tre stanze al piano superiore, ma la loro ricostruzione sarebbe stata molto arbitraria, visto che di queste stanze sappiamo a malapena della loro esistenza grazie ai crolli sul piano inferiore.



Zona 4 sulla cartina a inizio paragrafo

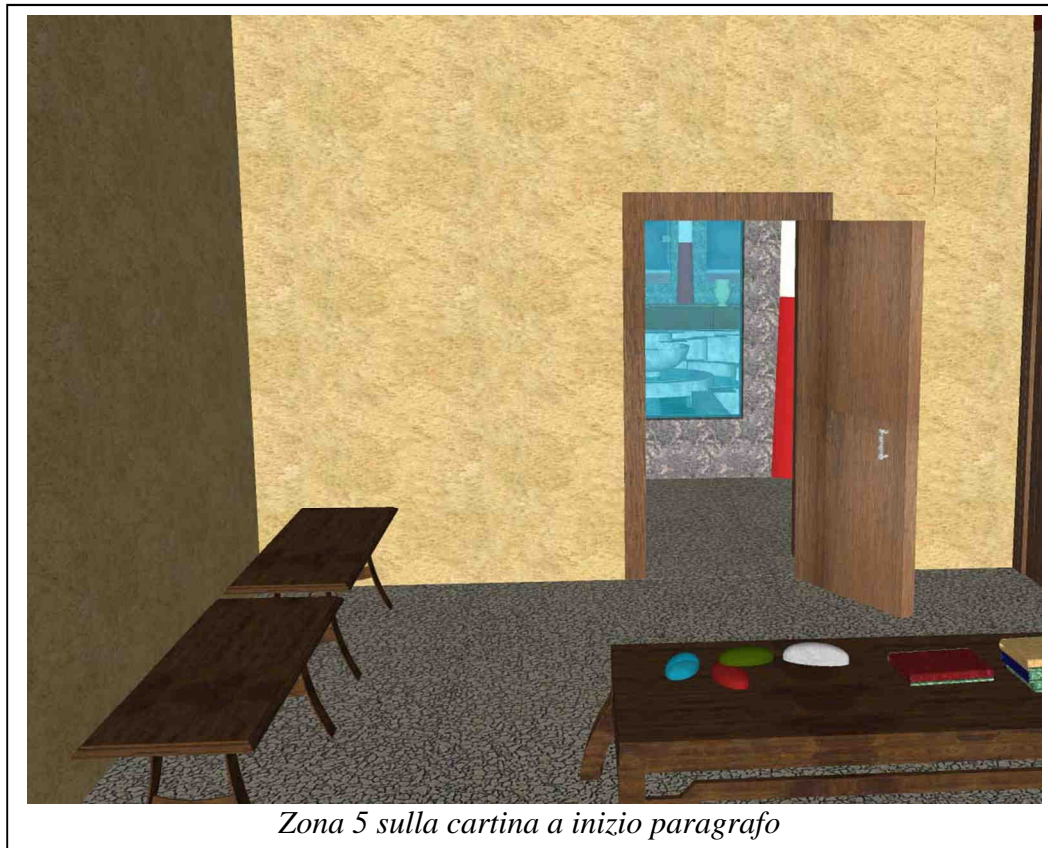
5- Vano antistante le terme

La stanza in questione era sicuramente un accesso alla stanza accanto, dove si trovavano le terme, che purtroppo sono state scavate solo in minima parte.

L'arredamento è costituito da un tavolo dove appoggiare gli effetti personali e da due panche per spogliarsi.

Il pavimento è sempre continuo rispetto al corridoio esterno, in graniglia di marmo.

Troviamo una porta un po' più importante, visto che comunicava anche con la stanza accanto, abbastanza ampia.



6- Triclinio invernale

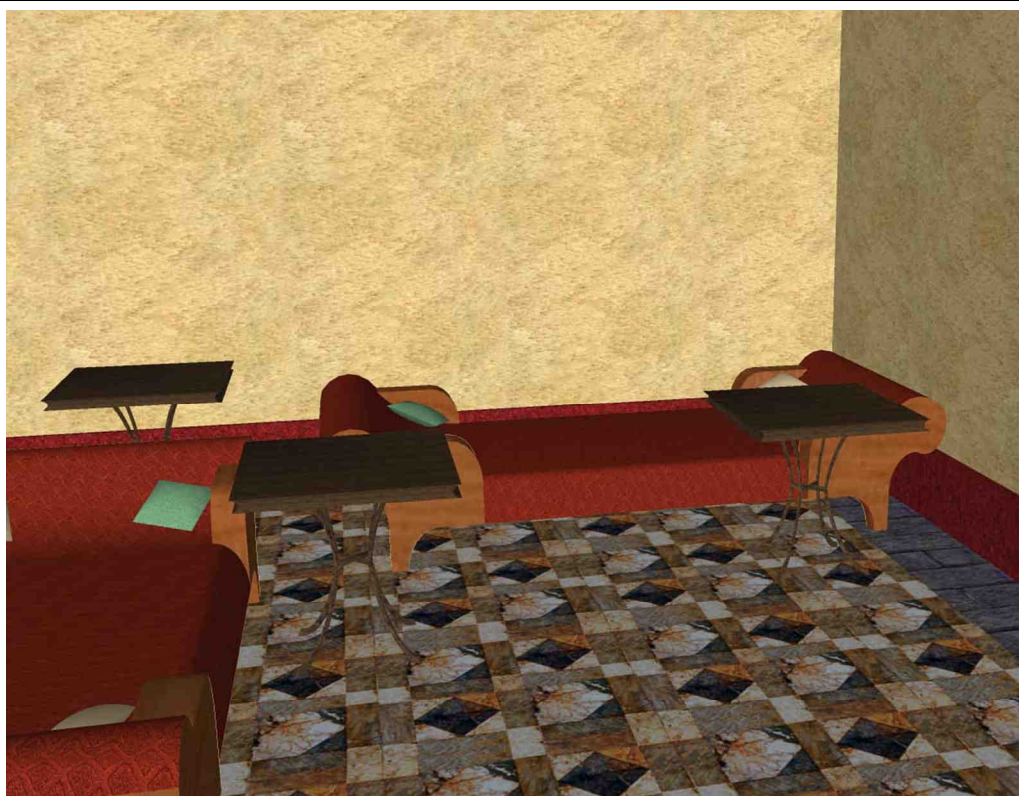
Dall'altra parte della *domus* troviamo il triclinio invernale, che si trova davanti al giardino sud ribassato rispetto a tutta la casa di qualche decina di centimetri.

L'arredamento è leggermente differente in quanto troviamo solo due divanetti con rispettivi tavoli; questo perché questa stanza aveva un'apertura sul corridoio, quindi era molto scomodo collocare un divanetto proprio davanti al corridoio.

Qua non troviamo una porta a griglia ma una porta di legno, visto che il triclinio era usato in inverno, quindi è plausibile che la stanza fosse più protetta.

Il triclinio, insieme a quello estivo e alla camera da letto adiacente, presenta uno dei pavimenti meglio conservati della *domus*; anche qua si è potuto ricavare le *texture* direttamente da foto reali.

Lo zoccolo della stanza è in pietra tinta di porpora, mentre per le pareti sono stati riscontrati solo normali intonaci.

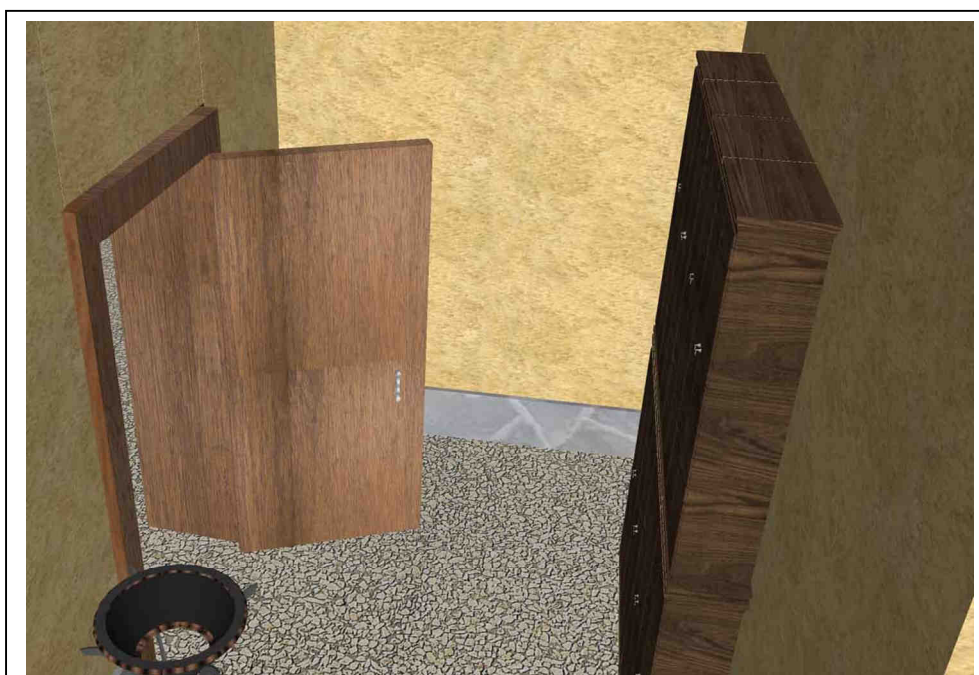


Zona 6 sulla cartina a inizio paragrafo

7- Stanza degli armadi

In questa stanza sono stati trovati resti di mobili, più precisamente degli armadi. Questi armadi, molto probabilmente, servivano per tenere i documenti del padrone di casa (contabilità etc..).

La stanza è abbastanza piccola rispetto a tutti gli altri vani della casa, per questo si pensa che non vi fossero svolte altre attività.



Zona 7 sulla cartina a inizio paragrafo

8- Camera da letto

Come detto in precedenza, questa è una delle stanze dove troviamo un pavimento conservato quasi perfettamente, ed è quindi stato possibile utilizzarlo come *texture*; è costituito da una cornice di mattonelle rettangolari di marmo bianco, mentre all'interno troviamo mattonelle quadrate con fantasia centrale; tra le mattonelle troviamo listelli di marmo di altri colori.

Si deduce che questa fosse una camera da letto per la base in muratura trovata in fondo alla stanza, che di solito in quel periodo veniva completata con un materasso e usata come letto.

L'arredamento è costituito da qualche cuscino sul letto e da un tavolino per gli effetti personali; di questi elementi non abbiamo riscontri sui reperti, ma sono stati aggiunti per l'alta probabilità di trovarli in una stanza dedicata al riposo.



Zona8 sulla cartina a inizio paragrafo

9- Stanza sul corridoio

Questa stanza si affacciava sul pavimento a mosaico, ed è molto probabilmente l'unica con un'apertura verso l'esterno; questo si è dedotto dal fatto che in quel punto della casa manca qualunque fonte di luce, mentre in tutti gli altri vani non ci sono problemi di illuminazione.

Per questo sono stati messi due scuri nella stanza, per chiudere la finestra che si affacciava all'esterno.

Troviamo un tavolo e uno sgabello, molto probabilmente dedicati al riposo e ad attività ricreative.

10- Corridoio centrale

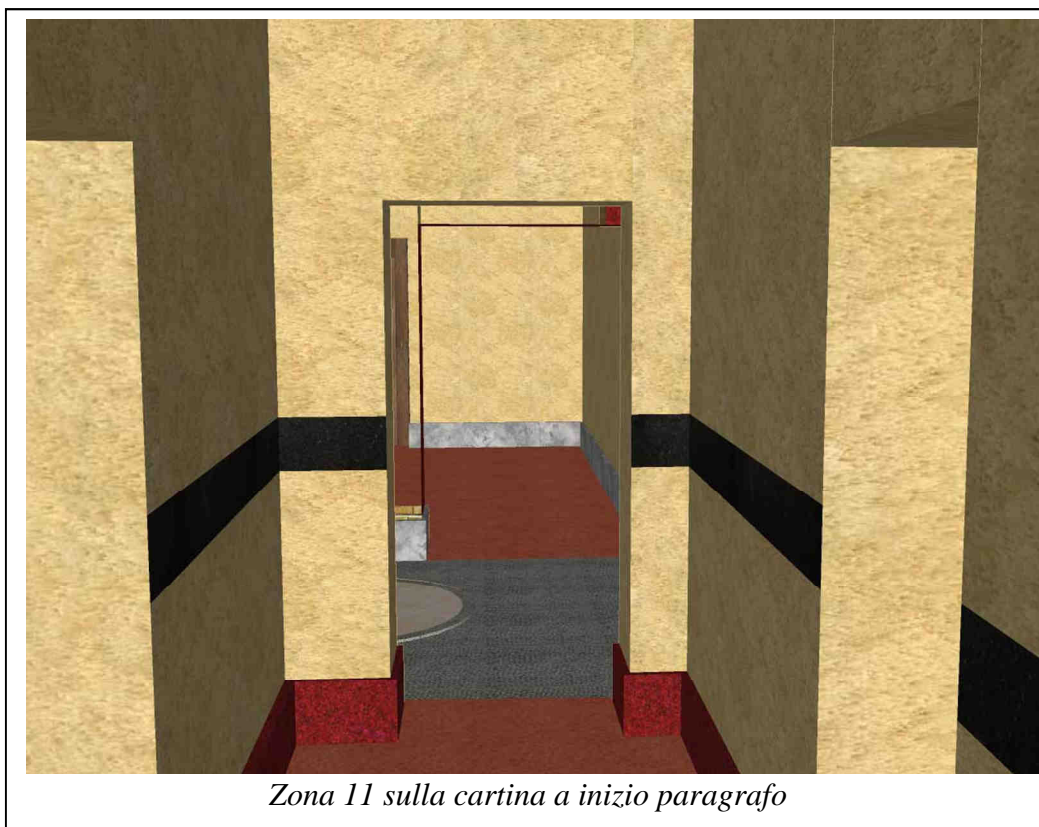
Questo corridoio mette in comunicazione tutti vani della parte ovest della *domus*.

Il pavimento è realizzato in terracotta, ed è stato ricreato con una *texture* generica di terracotta (quella presente sul luogo dello scavo era ormai troppo scura).

Le fasce ornamentali nere sul muro e lo zoccolo porpora sono stati texturizzati in questo modo per i ritrovamenti sullo scavo, che confermavano la presenza di un muro multicolore.



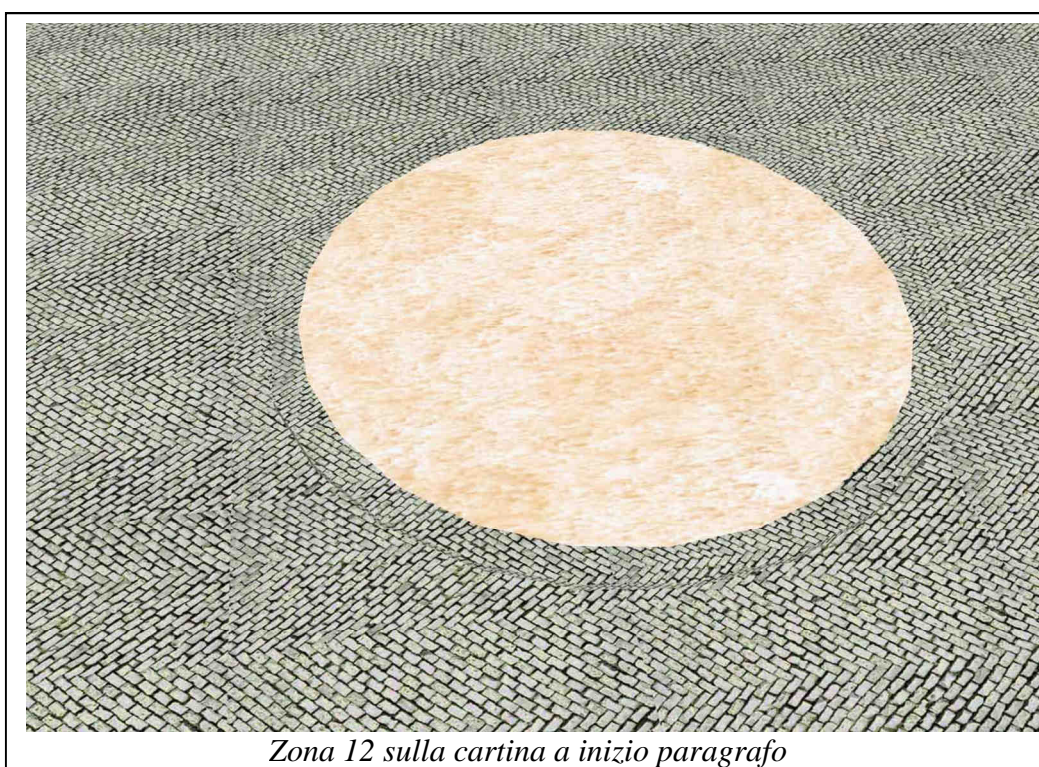
Zona 10 sulla cartina a inizio paragrafo



11- Stanza centrale del mosaico

In questa stanza si è trovato un mosaico conservato in maniera eccellente: al centro troviamo un disco di marmo bianco, circondato da altri inserti in mosaico. La *texture* applicata al pavimento corrispondere alla foto reale del mosaico, con qualche piccola modifica per sbiancare un po' il marmo.

Anche qua si può notare la presenza di uno zoccolo in pietra.



11- Conclusioni

Il fine del progetto era quello di creare una piattaforma per ricostruzioni su *game engine*, che permettesse a chi non ha competenze di programmazione di creare la propria ricostruzione storica senza problemi.

Inoltre, si è ricostruita la *Domus degli affreschi* di Luni, seguendo tutti i principi di una ricostruzione storica fedele e dettagliata.

Per dimostrare le potenzialità del progetto, anche la ricostruzione storica del castello della Brina (progetto di tesi triennale) è stata riportata e adattata alla nuova metodologia di sviluppo.

Tutti questi obiettivi sono stati raggiunti, uno dopo l'altro, visto che la struttura del progetto richiedeva che ogni passaggio fosse la naturale evoluzione di quello precedente.

Questa tesi, oltre che dimostrare il lavoro svolto e descrivere le modalità di lavoro adottate si propone come manuale d'uso per l'utilizzatore finale della piattaforma; infatti, oltre che spiegare nel dettaglio tutte le componenti di un *game engine* in maniera molto comprensibile e illustrare come è stato costruito il sistema, si spiega quali sono le entità che effettivamente bisogna utilizzare nella piattaforma.

Il testo si divide fondamentalmente in due parti:

Nella prima parte si è analizzata la piattaforma sviluppata, le problematiche affrontate e lo sviluppo verso le soluzioni adottate; poi si è illustrata una vera e propria guida, con le spiegazioni di tutti i passaggi da intraprendere, e infine sono stati descritti tutti gli *script* creati per dare vita alla piattaforma.

Nella seconda parte invece ci si è concentrati sulla ricostruzione della *Domus* di Luni, analizzando ogni singola parte dell'edificio ricostruito, paragonando sempre le fonti storiche al modello ottenuto, e motivando le scelte effettuate; infine si è fornito un approfondimento basato su tutta la documentazione utilizzata, in maniera da chiarire il contesto storico e archeologico.

Gli obiettivi, come detto, sono stati raggiunti appieno, riuscendo a sviluppare una piattaforma per ricostruzioni parallelamente allo studio e alla ricostruzione della *Domus* degli affreschi, con completezza sia tecnica che storica.

12. Bibliografia

“Zona Archeologica e Museo Nazionale di Luni”, *a cura di* L. Gervasini, A.M. Durante, Roma 2000

“Abitare a Luna, edilizia privata nel Casale Caleo”, *a cura di* L. Gervasini, Genova 2001

A.M. Durante, “Appunti per lo studio degli intonaci dipinti della *Domus* degli Affreschi”, in “Romana pictura e christiana signa. Due mostre a confronto. Arte figurativa in Liguria tra età imperiale e altomedioevo”, Genova 2003, pp.49

A.M. Durante, “Edilizia privata a Luna”, *in* “XLIX Abitare in Cisalpina”, Trieste 2001, pp. 269-295

Annapaola Zaccaria Ruggiu, “La casa degli affreschi di Luni: fasi edilizie per successione diacronica”, *in* “Centro studi lunensi, Quaderno 8”, Lerici 1983, pp.3

A.Maiuri, “Ercolano, i nuovi scavi (1927-1958)”, Roma 1958

“Forme e tempi dell’urbanizzazione nella cisalpina (II secolo a.C. – I secolo a.C)”, *a cura di* Luisa Brecciaroli Taborelli, Torino 2006

Gervasini L., Durante A.M., Gambaro L., Landi S., “Luna e l’ager lunensis: nuovi elementi per la conoscenza della città antica e del territorio romanizzato fra il golfo della Spezia e il portus Lunae.” In “Atti delle Giornate di Studio Forme e tempi dell’urbanizzazione nella Cisalpina (II secolo a.C.-I secolo d.C.)”, Torino 2006

Sitografia

Unity 3d Team, “Unity Technologies: *Unity 3d*”,2012

<http://Unity3d.com/>

Mattia Armani, “Multiplayer.it: *Unity*, la star dei tool di sviluppo” ,2011

<http://multiplayer.it/notizie/86196-gdc-2011-Unity-la-star-dei-tool-di-sviluppo.html>

Giorgio Melani, “Multiplayer.it: L’*engine Unity* verrà utilizzato dalla polizia scientifica” ,2009

<http://multiplayer.it/notizie/72571-lengine-Unity-verra-utilizzato-dalla-polizia-scientifica.html>

Definizione *Unity* “ Wikipedia ENG :*Unity (Game engine)*”,2012

[http://en.wikipedia.org/wiki/Unity_\(game_engine\)](http://en.wikipedia.org/wiki/Unity_(game_engine))

Elenco dei *game engine*, “Wikipedia ENG :List of *game engines*”,2012

http://en.wikipedia.org/wiki/List_of_game_engines

Definizione *game engine*, “Wikipedia ENG :*Game engine*”,2012

http://en.wikipedia.org/wiki/Game_engine

Dario Nei,“Multinews: Il Porting da iOS ad android è semplificato con *Unity*” ,2011

<http://multinews.altervista.org/mobile/il-porting-dei-giochi-ios-su-android-avviene-facilmente-grazie-ad-Unity/>

Gaetano Cavallaro,“Buddhagaming: *Game engine*: il top per “indietreggiare” un pò” ,2012

<http://buddhagaming.it/game-engine-il-top-per-indieggare-un-po-113129/>