

MANN ANALYTICS

Percorsi a regola d'arte

Analisi di percorsi effettuati dai visitatori di un museo, con conseguente sviluppo di una applicazione di visualizzazione per essi

Aveta Carmine N46003385
Banchini Federico N46003517
Benfenati Domenico N46003380
07/06/2019



MANN Analytics

Percorsi a regola d'arte

Sommario

| | |
|---|----|
| Introduzione | 2 |
| Creazione del modello 3D | 3 |
| Google Sketchup 8 | 3 |
| Strumento Lines | 3 |
| Strumento Shapes | 3 |
| Strumento Pull..... | 3 |
| Altri strumenti utili..... | 4 |
| Modellazione Museo Archeologico | 4 |
| Esportazione del modello | 5 |
| Diagrammi UML..... | 6 |
| Use Case Diagram | 6 |
| Class Diagram | 6 |
| Sequence Diagram di analisi per la visualizzazione dei percorsi | 7 |
| Activity Diagram per la visualizzazione dei percorsi..... | 7 |
| Creazione del Menù iniziale..... | 8 |
| Struttura del menù | 8 |
| Cambiamento della lingua..... | 9 |
| Settaggio dei colori | 10 |
| Splash Screen | 10 |
| Volume e musica..... | 11 |
| Visualizzazione 3D dei percorsi | 12 |
| Definizione dei WayPoint..... | 12 |
| Creazione delle linee | 12 |
| Creazione degli oggetti..... | 13 |
| Navigation Mesh | 13 |
| Modalità di visualizzazione | 14 |
| Applicazione Web..... | 16 |

INTRODUZIONE

L'intento di questo applicativo è quello di rappresentare una sorta di monitoraggio IoT dei visitatori di un particolare museo sito nel cuore della omonima città, quale il Museo Archeologico Nazionale di Napoli.

Ciò che è stato implementato parte da una raccolta di dati durata diversi mesi, durante i quali delle torrette dotate di interfaccia Bluetooth situati in posti chiave del museo stesso, hanno registrato il passaggio del bacino di utenza del museo, qualora il visitatore stesso sia stato in possesso di un dispositivo Bluetooth attivo e sia transitato nel raggio d'azione della torretta stessa.

Durante la registrazione, che avviene in maniera totalmente non invasiva, la torretta mantiene in memoria sia l'indirizzo fisico del dispositivo rilevato, in modo da poter identificare l'utente in modo univoco, e l'orario in cui viene effettuata la rilevazione.

Tutte le registrazioni portano alla composizione di un dataset conservato all'interno di un server, che è la base di partenza per lo sviluppo dell'applicazione di visualizzazione. Il dataset è composto di diverse righe, ognuna delle quali contiene il percorso del visitatore espresso per punti tramite delle lettere che identificano la relativa torretta che ha effettuato il rilevamento, la durata totale della visita e la relativa data.

L'applicazione consente, tramite una opportuna interfaccia utente, di visualizzare tutti i percorsi contenuti nel dataset, all'interno di un ambiente 3D, dopo aver preliminarmente caricato il dataset stesso.

L'applicativo consente anche di applicare qualche filtro per decidere i percorsi da poter importare, come selezionare una specifica data, un orario particolare, o il passaggio nel raggio d'azione di una specifica centralina su cui si vuole fare un'analisi.

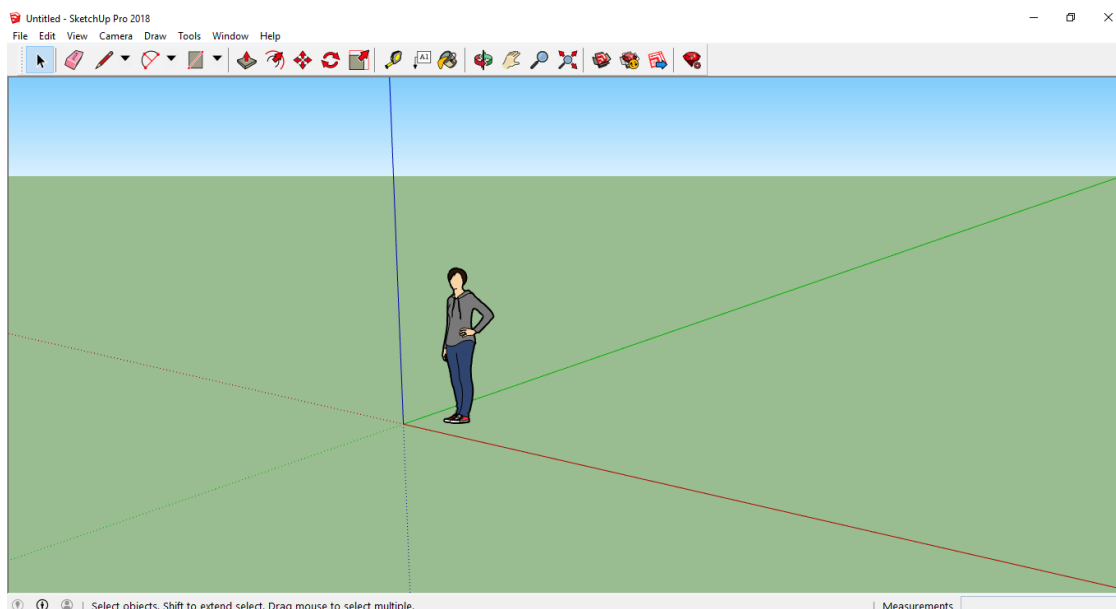
Il modello 3D del museo è stato interamente realizzato tramite l'applicativo Google Sketchup 8 Pro, software di progettazione sviluppato da Google, grazie al quale è stato possibile riprodurre facilmente il museo in toto, mantenendo un alto grado di fedeltà e una precisione grafica quasi paragonabile alla realtà. Grazie alle modalità di export del software stesso è stato possibile utilizzare lo stesso modello in Unity 3D, applicativo che è stato utilizzato per lo sviluppo dell'intera applicazione di visualizzazione dei percorsi registrati dalle torrette.

CREAZIONE DEL MODELLO 3D

Il modello 3D del MANN è stato sviluppato utilizzando l'applicativo Google Sketchup, per la modellazione 3D di oggetti. Vediamone brevemente il funzionamento.

Google Sketchup 8

Google Sketchup 8 ha numerosi strumenti interessanti e utili ai fini di una modellazione 3D delle forme e degli oggetti in generale. La schermata principale si presenta come segue:



La figura statica posta al centro del sistema di assi viene indicata con il nome di Stacy, ed è utile per sviluppare dei modelli confrontando le proporzioni di Stacy con quelle del modello che stiamo creando. Sketchup fa uso di tre assi cartesiani, quali l'asse verde e l'asse rosso per le linee su di un piano, e l'asse blu per delle linee verticali.

Strumento Lines

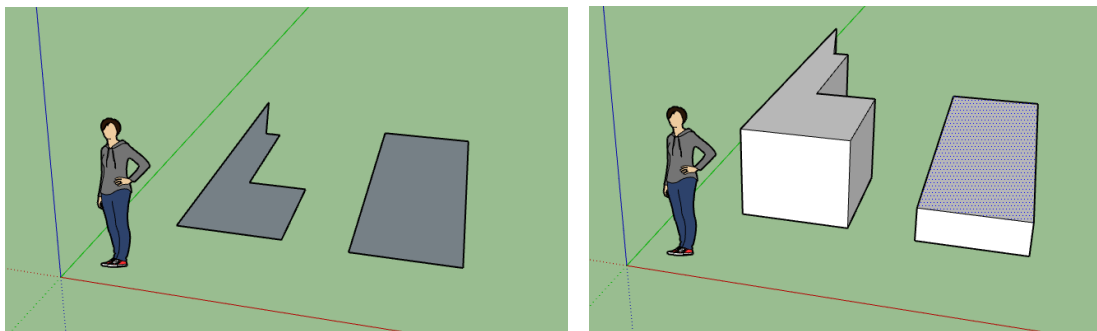
Uno degli strumenti più importanti è lo strumento *Lines*, indicato con una matita in alto a sinistra. Grazie allo strumento *Lines* è possibile creare delle linee fissando un punto di inizio e uno di fine, regolando anche l'inclinazione della stessa secondo i tre assi principali indicati da Sketchup.

Strumento Shapes

Altro importante strumento è lo strumento *Shapes*, che presenta l'icona del rettangolo. È possibile andare a cambiare la forma che voglio costruire andando semplicemente a cliccare sulla freccetta a lato del pulsante, e scegliendo il poligono che si desidera generare.

Strumento Pull

Uno strumento fondamentale per la creazione di modelli 3D, è lo strumento *Pull-up*. Grazie ad esso, è possibile creare delle forme geometriche tridimensionali partendo dal modello 2D della stessa forma. Quello che si fa è creare una forma 2D che intendiamo trasformare, e poi, grazie a questo strumento, dopo la pressione del mouse sulla faccia che abbiamo rappresentato effettuare l'estrusione nella direzione perpendicolare a quella del piano che contiene la figura. Vediamo qualche esempio di forme create con questa modalità.



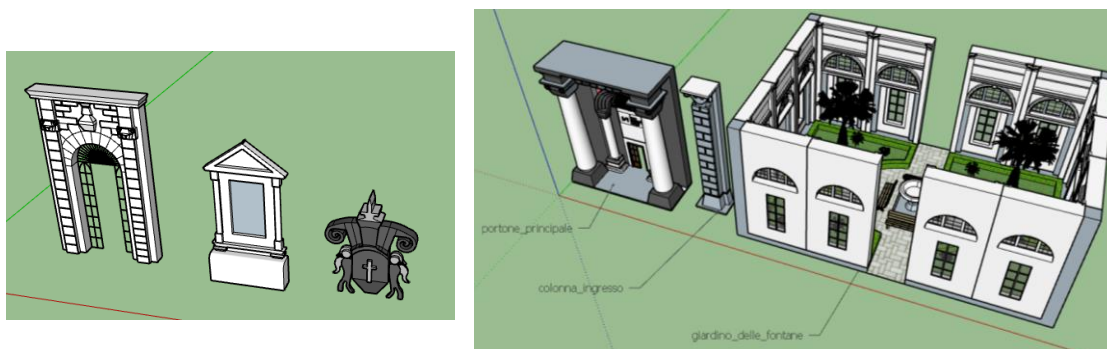
Altri strumenti utili

Raggruppando insieme gli altri strumenti che sono risultati utili ai fini dello sviluppo del modello tridimensionale del museo, ritroviamo lo strumento *Move*, per spostare una certa entità lungo una direzione, che sia essa una figura 2D, una linea, o un intero oggetto, lo strumento *Offset*, per la creazione di una forma che abbia le stesse proporzioni della forma di partenza, all'interno della forma stessa, o anche lo strumento *Scale*, che permette di modificare le proporzioni di un oggetto.

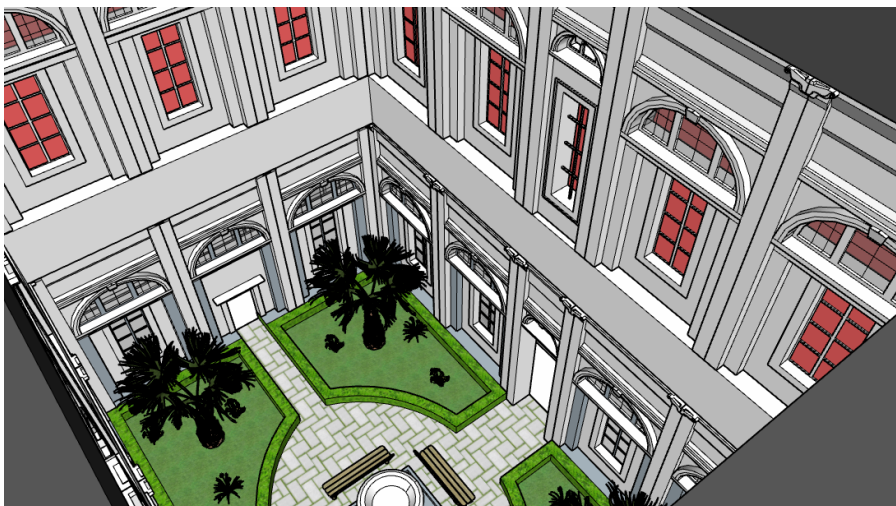
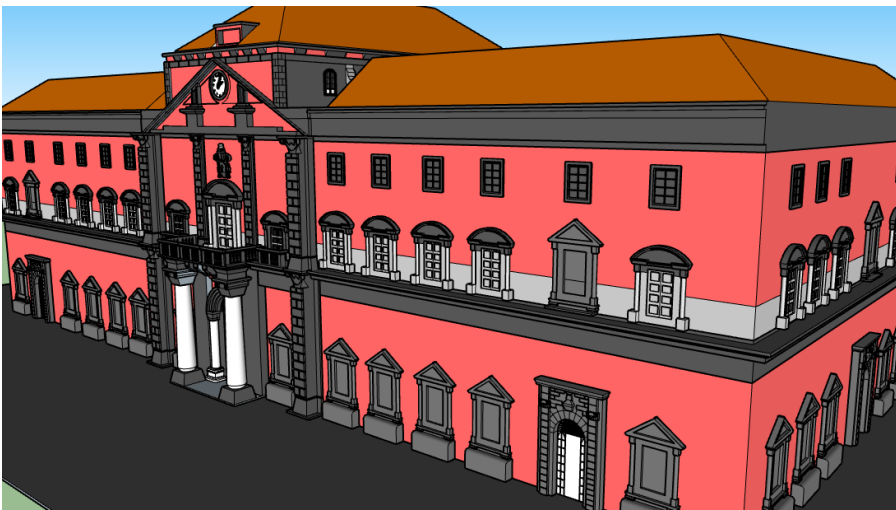
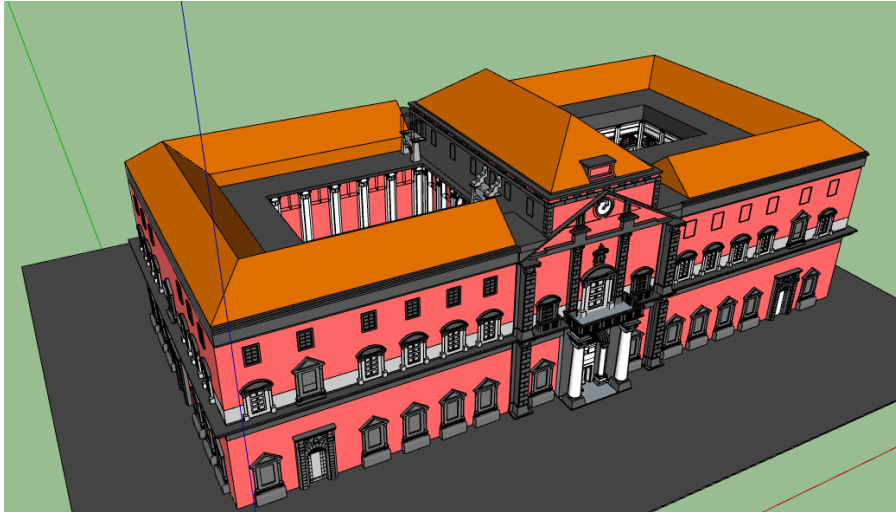
Modellazione Museo Archeologico

Utilizzando gli strumenti che mette a disposizione Sketchup, è stato possibile formalizzare le architetture del museo utilizzando una fotografia della piantina reale del museo, importata nel progetto stesso, tracciando delle linee lungo i contorni del museo presenti sulla cartina. Una volta create le tracce per le mura del museo, utilizzando lo strumento *Pull*, si è creato il museo vero e proprio, dando altezza alla piantina appena composta. Il tutto è stato ripetuto per i vari piani.

Per modellare la facciata e l'esterno in generale del museo, si è ricorso alla semplice inventiva, cercando di creare le forme corrette per avere delle finestre, delle colonne, un entrata e tutti gli altri elementi architettonici che compongono il museo quanto più simile a quelli visibile dal vivo, facendosi aiutare anche dalla comoda funzionalità *Street View* di Google Maps, per poter avere un modello fedele da riprodurre, oltre a fotografie reperite sul web circa gli interni e i due giardini principali. Di seguito vengono riportati alcuni componenti creati per l'estetica del museo.



Grazie a svariati tentativi di riprodurre il vero, e con un po' di ore di lavoro, si è riuscito ad ottenere una rappresentazione tridimensionale molto fedele del museo, che ha destato non poco stupore tra i componenti del gruppo stesso, e di cui sono presentate delle immagini in seguito.



Esportazione del modello

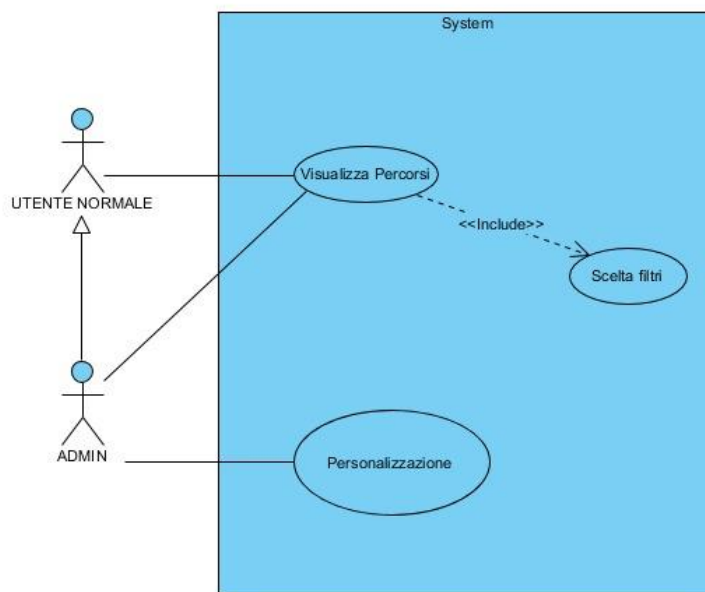
Per poter utilizzare il modello su Unity 3D, si è fatto ricorso alla funzione di esportazione del modello di Sketchup. Basta infatti fare *File -> Export -> 3D Model...* per ottenere un menù di salvataggio del file esportato. Da qui, settate le giuste opzioni come spuntare l'opzione di *triangulate all face* e le altre per esportare correttamente le texture create

utilizzando Sketchup su Unity. Fatto questo si può procedere all'export, salvando il file con l'estensione *fbx* . Fatto ciò, il modello è pronto per essere utilizzato su Unity 3D.

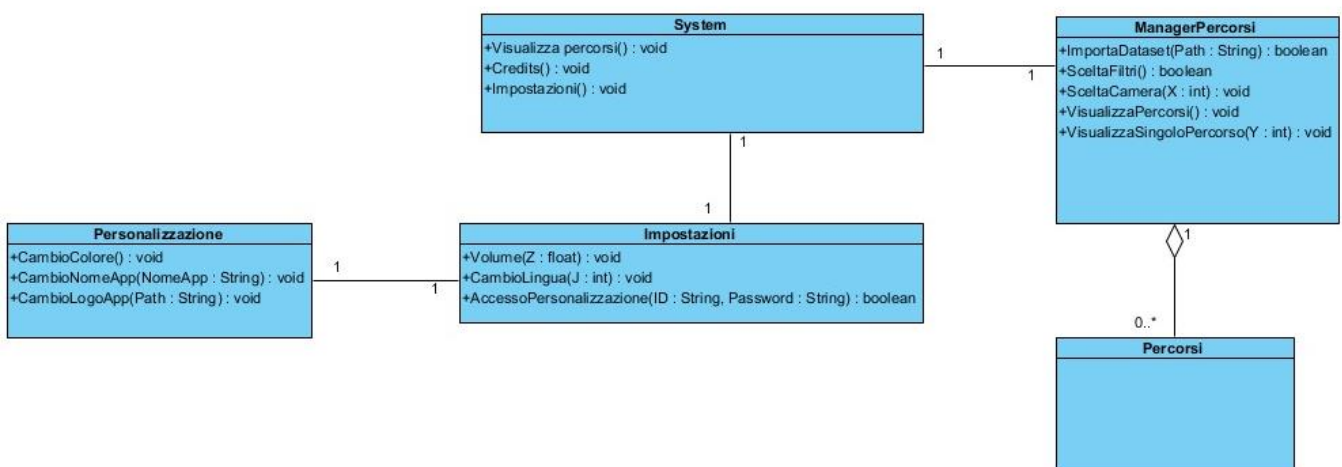
DIAGRAMMI UML

Per renderci il lavoro più facile si è pensato ad implementare dei diagrammi che aiutassero la stesura del codice, in modo da avere un progetto ben chiaro ai fini del suo sviluppo. Di seguito vengono riportati delle schermate contenenti i suddetti diagrammi. Per una migliore visualizzazione si invita l'apertura delle schermate allegate.

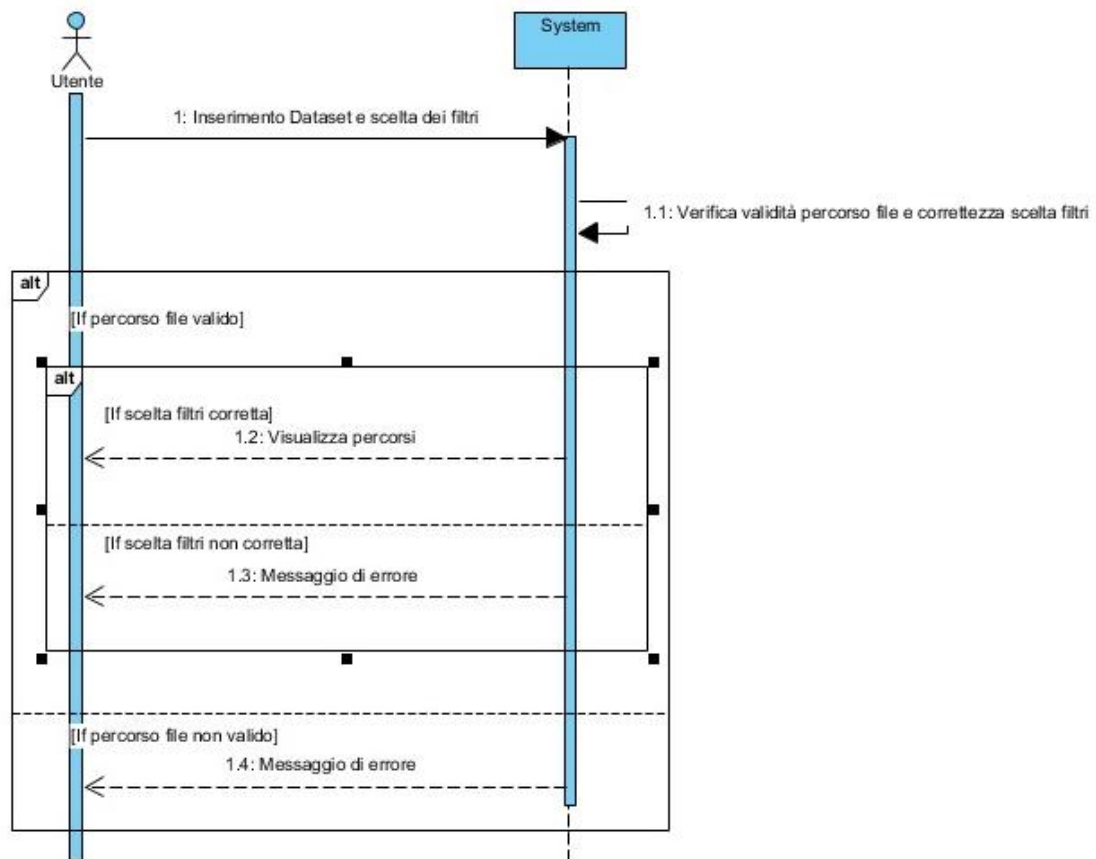
Use Case Diagram



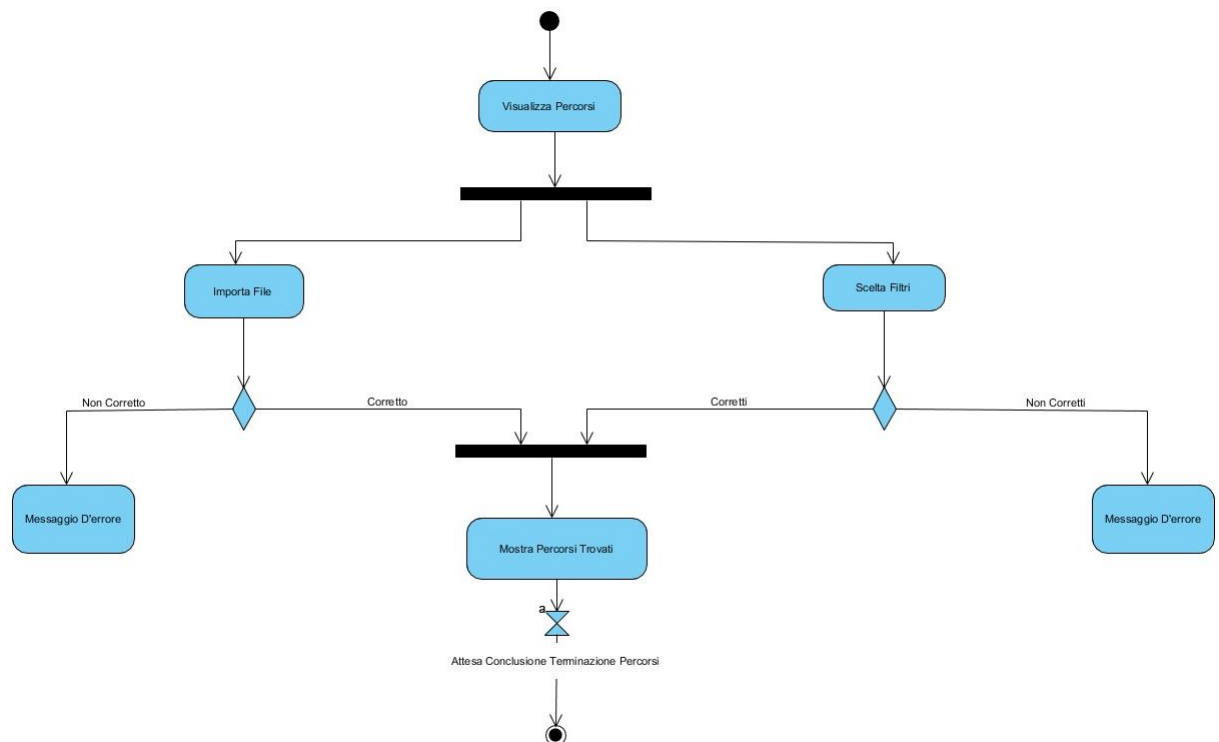
Class Diagram



Sequence Diagram di analisi per la visualizzazione dei percorsi



Activity Diagram per la visualizzazione dei percorsi



CREAZIONE DEL MENÙ INIZIALE

Al fine di rendere l'applicazione il più user-friendly possibile, e poter permettere all'utente di andare a modificare qualche parametro prima di accedere alla vera e propria applicazione, si è pensato di implementare in Unity 3D un menù iniziale.

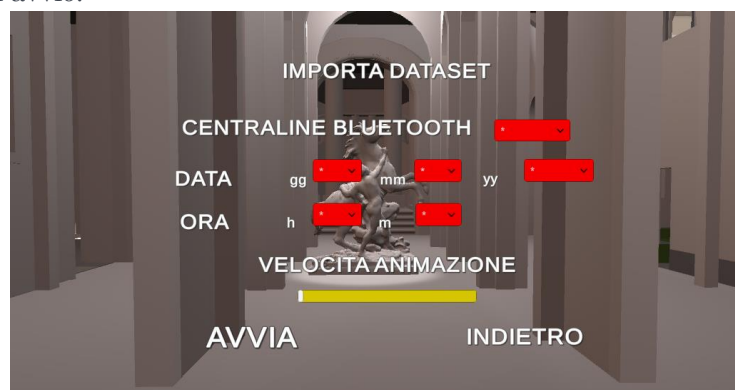
Struttura del menù

Per dotare la scena di una composizione grafica adeguata, si è pensato di porre alle spalle del menù il modello tridimensionale della struttura creata in precedenza. Di seguito si riporta una schermata esemplificativa del menù stesso.



Come si evince dalla schermata, il menù presenta diverse voci:

- **AVVIA:** nella sezione “avvia”, l'utente viene indirizzato su una nuova schermata, dove è possibile selezionare una serie di parametri, come scegliere una specifica centralina che deve essere contenuta nei percorsi importati, o selezionare un filtro per la data e l'ora in cui è avvenuta la registrazione del percorso. Inoltre, in questa sezione avviene il caricamento del dataset tramite l'esplorazione risorse, selezionando il file da cui caricare i percorsi, oltre ad uno slider per selezionare la velocità di visualizzazione dei percorsi stessi. Di seguito vi è una dimostrazione della struttura dell'area di avvio.



- **IMPOSTAZIONI:** la sezione “impostazioni” è forse la più importante. In questa parte del menù è possibile andare a modificare la lingua del nostro applicativo, settare il livello desiderato di volume, ed entrare in una sezione di personalizzazione ulteriore (lato back-end dell'applicazione), che prevede però l'accesso tramite un username e una password, che si è pensato di settare di default con “admin” e “admin”. Nell'area di personalizzazione, è consentito modificare parametri come il nome da assegnare all'applicazione, o il logo che essa deve presentare in fase di sviluppo. Come logo si è pensato a settarne uno di default. Inoltre, nella sottosezione dedicata ai colori, è possibile settare diversi tipi di colori applicati a diversi oggetti dell'applicazione come il colore del testo, dei pulsanti e dei menù a tendina per la scelta dei filtri. Le modifiche effettuate in ogni sessione, vengono salvate tramite un metodo, *PlayerPrefs*, che prevede il salvataggio di un valore utilizzando una chiave, detta *key*. Di seguito riportiamo alcune schermate delle sezioni che vanno a comporre la scheda “impostazioni”.



- **CREDITS:** in quest'area sono presenti i riferimenti ai creatori dell'applicazione e al docente che ha commissionato la stessa.
- **ESCI:** cliccando su pulsante, banalmente si va a chiudere l'applicazione.

Per l'area delle impostazioni che prevede l'aggiunta di filtri, si è pensato di fare sì che all'utente non sia permesso l'inserimento di formati di data non validi. Infatti, qualora egli voglia selezionare una data inesistente, o non congrua con il formato indicato nel dataset, viene mostrato a video un opportuno popup, che suggerisce di modificare le scelte fatte. Così come il popup per la scelta dei filtri, è stato implementato un ulteriore schermata di errore circa l'inserimento di credenziali non valide per l'accesso alla sezione di personalizzazione, e per dare conferma dell'uscita dall'applicazione stessa. Vediamo adesso qualche particolare tipo di implementazione di alcune funzionalità utilizzate nel menù.

Cambiamento della lingua

Per effettuare il cambio della lingua, si è pensato di implementare la traduzione utilizzando un meccanismo basato sulla sostituzione delle parole incluse nei vari elementi del menù, etichettando quest'ultime con delle parole chiave, e andando a creare un file di corrispondenze per tutte le parole che si desidera tradurre. Il file in questione è stato inserito nella risorse del progetto stesso. Per effettuare la traduzione, si va a pescare all'interno del file che ha l'etichetta indicata per la lingua selezionata, e viene effettuata un cambiamento dei parametri dell'editor stesso a tempo di esecuzione, selezionando una specifica lingua per esso. Il file di testo per effettuare il cambio di lingua sarà quindi corredato della parola chiave di quel particolare testo, e della sua traduzione nella specifica lingua selezionata tramite menù.

Settaggio dei colori

Per effettuare un cambiamento di colori, si deve disporre delle credenziali per effettuare l'accesso. Una volta entrati nella sezione di modifica, viene visualizzata una palette di colori per ogni genere di colore che compare nel menù. Scegliendo il colore del testo, si vanno a modificare le label, il colore dei testi dei bottoni, e il colore dei testi contenuti nei menù a tendina per la selezione dei filtri. Modificando il colore del testo evidenziato, vado a modificare anche il colore che visualizzo durante il passaggio con il mouse sul pulsante stesso, mentre per cambiare il colore di sfondo dei menu a tendina, devo modificare il colore specifico. La modifica dei colori è concettualmente semplice: cliccato il pulsante relativo ad uno specifico colore, tramite script vengono modificati i colori specifici per ogni oggetto dell'interfaccia, nelle modalità già indicate. Anche per i colori avviene il salvataggio dei

```
public void CaricaColori(){
    string temp = PlayerPrefs.GetString("ColoreTesto");
    if (temp.Equals("blu")){
        ColoreTestiBlu();
    }
    else if (temp.Equals("giallo")){
        ColoreTestiGiallo();
    }
    else if (temp.Equals("rosso")){
        ColoreTestiRosso();
    }
    else if (temp.Equals("bianco")){
        ColoreTestiBianco();
    }
    else if (temp.Equals("nero")){
        ColoreTestiNero();
    }
    else if (temp.Equals("verde")){
        ColoreTestiVerde();
    }
    else{
        //null
    }
}

public void ColoreTestiBlu()
{
    int count = ElencoTesti.Count;
    for (int i = 0; i < count; i++)
    {
        ElencoTesti[i].color = Color.blue;
    }

    int conta = ElencoImmaginiSliderFill.Count;
    for(int j = 0; j < conta; j++)
    {
        ElencoImmaginiSliderFill[j].color = Color.blue;
    }

    PlayerPrefs.SetString("ColoreTesto", "blu");
}
```

settaggi, tramite il metodo *PlayerPrefs*. Di seguito vediamo un esempio di script per il settaggio di un colore, in questo caso il colore del testo blu.

Per effettuare il salvataggio tramite *PlayerPrefs*, vengono utilizzati in combinata due metodi diversi: il metodo *GetString* ci permette di dare un valore all'etichetta utilizzata per i colori, che in questo caso abbiamo chiamato "*ColoreTesto*". Utilizzando questa etichetta, per ogni colore andiamo a dare il valore della chiave corrispondente ad uno specifico colore alla stessa, utilizzando il metodo *SetString*, richiamato per ogni funzione di cambio del colore. Nell'esempio in questione, l'etichetta "*ColoreTesto*" viene associata al valore della chiave "blu". Ciò viene fatto per ogni cambio di colore, e per ogni colore che si è deciso poter essere settabile da applicazione.

Splash Screen

In abbinata al menù, si è implementato una animazione iniziale che si avvia in automatico, la quale consente di effettuare tutte le inizializzazioni del caso. Essa consiste in un breve testo che in dissolvenza scompare introducendo la scena principale con il modello tridimensionale del museo e una indicazione che ci invita a cliccare la barra spaziatrice per proseguire. Dopo aver cliccato la barra, viene avviata la scena contenente il menù descritto in precedenza. Vediamo una schermata esemplificativa.



Volume e musica

Per creare un sottofondo musicale alla nostra applicazione, è stato scelto un brano tratto dal programma televisivo Super Quark; il brano in questione è la celebre composizione di J.S. Bach "aria sulla IV corda". Passando all'aspetto implementativo, per generare un suono si è utilizzato il componente *AudioMixer*, che è possibile inserire anche tramite Unity 3D stesso: una volta selezionato il file da riprodurre, viene settato il valore dell'audio tramite la funzione *SetFloat*, passando a quest'ultima il valore e il parametro da modificare tramite uno slider, anch'esso presente nella scheda delle impostazioni. Vediamo le righe di codice per il settaggio dinamico del volume.

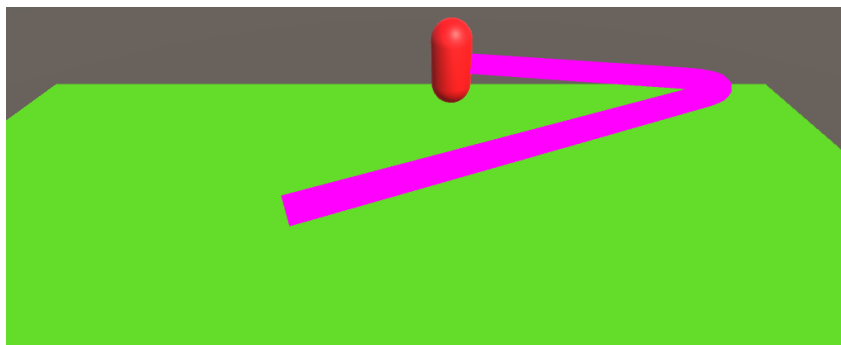
```
public class scriptAudio : MonoBehaviour
{
    public AudioManager audioMixer;

    public void SetVolume(float volume)
    {
        audioMixer.SetFloat("volume", volume);
    }
}
```

Per ottenere informazioni aggiuntive circa le implementazioni di singole funzionalità, o di altro non specificato nel presente documento, si invita alla lettura delle righe di codice presenti tra gli script generati per lo sviluppo dell'applicazione.

VISUALIZZAZIONE 3D DEI PERCORSI

In accordo con le dinamiche del problema, si è reso possibile una implementazione per la visualizzazione dei percorsi, utilizzando apposite funzionalità che mette a disposizione Unity 3D. Lo scopo che si è cercato di ottenere è quello di mostrare i percorsi come se l'utente che fa uso dell'applicazione sia immerso del tutto in essa, andando a visualizzare i percorsi come delle linee dinamiche che si vengono man mano a creare sul pavimento del modello 3D. Per fare ciò è utilizzata la funzionalità di Navigazione di Unity, utilizzando l'applicativo *NavMesh* per definire le zone percorribili e per stabilire i punti chiave lungo i quali il percorso deve svilupparsi. Per la definizione dei punti vengono utilizzati dei punti tridimensionali, detti *Waypoint*, che definiranno la posizione delle varie centraline all'interno del modello. Per capire meglio di cosa si parla, vediamo un esempio esterno di visualizzazione di un tragitto di un oggetto in una scena di prova.



L'intento è quello di creare delle linee utilizzando dei punti chiave, attraverso i quali un oggetto crea una sorta di scia.

Definizione dei WayPoint

Per dare una definizione di centralina bluetooth nella nostra applicazione, siamo ricorsi all'utilizzo di un oggetto invisibile, che come detto si è definito come *WayPoint*. Per specificare al motore grafico quali fossero le centraline X, I, ecc. si è ricorso all'utilizzo dei tag, etichette da assegnare ad ogni oggetto che fa le veci di una centralina, e che ci permette di obbligare facilmente, tramite uno stralcio di codice opportuno, il passaggio delle linee per quell'oggetto, qualora quel percorso richiedesse ciò.

Creazione delle linee

Per poter comodamente rappresentare i percorsi in un ambiente 3D, si è ricorso come detto all'utilizzo del *NavMesh*. Per dare una interpretazione grafica al movimento di un oggetto all'interno di un ambiente creato proprio utilizzando il *NavMesh*, si è pensato di aggiungere ad ogni oggetto invisibile una coda, che in Unity è interpretata come *Trail Render*. Questo componente permette ad un oggetto in movimento di lasciare una traccia durante il suo tragitto, traccia che può avere forma, colore e intensità variabile.

Vista l'enorme mole di dati che l'applicazione dovrà gestire, si è implementato un opportuno script che permette di generare dei colori casuali per ogni percorso, il quale consta nell'associazione di una nuova colorazione alla scia dell'oggetto invisibile, colorazione che si fa a comporre settando dei valori casuali per i parametri R, G e B del materiale associato a quella colorazione. Vediamo allora come viene effettuato questo settaggio tramite codice.

```

ListaGameObject[conta].AddComponent<TrailRenderer>();
TrailRenderer trail = ListaGameObject[conta].GetComponent<TrailRenderer>();
trail.time = 1000000000;
trail.startWidth = 0.05f;
trail.material.color = new Color(Random.value, Random.value, Random.value, 1.0f);

```

Come si può vedere, durante il ciclo per lo scorrimento del vettore di percorsi filtrati, che si è preferito chiamare *ListaGameObject*, viene aggiunta una coda ad ogni oggetto presente nella lista, viene settato un tempo di visibilità della coda pari a un valore relativamente alto, per evitare che essa scompaia dopo un numero finito di secondi, viene settato lo spessore della stessa, e viene inizializzato il nuovo colore per ogni oggetto.

Creazione degli oggetti

Per far sì che le linee siano associate a degli oggetti fisici, e siccome ciò che si desiderava ottenere era una linea che non avesse un oggetto fisico visibile come guida, si è pensato di andare a deselezionare l'opzione di *MeshRenderer* da un oggetto utilizzato per effettuare la generazione degli oggetti che vanno a creare la linea. Ciò ha permesso di rendere l'oggetto stesso invisibile, ma ugualmente abile a rilasciare una coda per la creazione visiva del percorso stesso. Utilizzando poi la funzione *Istanziare*, viene generato dinamicamente un numero di oggetti pari al numero di percorsi filtrati, facendo quindi in modo che anche tutti i figli dell'oggetto utilizzato per la creazione del percorso godano delle suddette proprietà di invisibilità, nelle stesse modalità dell'oggetto padre.

Ovviamente si è previsto che gli oggetti, quali statue, e altri oggetti presenti nel museo, oltre alle zone non accessibili di esso, non facciano parte della suddetta zona percorribile, andando a spuntare l'apposita opzione. Le funzionalità del *NavMesh* non richiedono particolari settaggi tramite codice, quindi il tutto è stato realizzato tramite l'editor di Unity stesso.

Per gestire la visualizzazione multipla dei percorsi, si è pensato di generare diverse tipologie di agenti, che siano più o meno alti a seconda della quantità di percorsi già generati. Infatti, se sono stati già generati un numero di percorsi sufficiente a rendere il terreno una vera e propria tavolozza, viene prevista, per i percorsi successivi, la creazione di agenti con un'altezza maggiore, in modo da visualizzare le linee su più livelli.

Oltre a ciò, implementando una barra di scorrimento per il settaggio di una velocità per la generazione dei percorsi, è sorto il problema della generazione multipla di quest'ultimi. Infatti, qualora l'utente finale volesse mostrare tutti i percorsi presenti nel file dataset, sarebbe stato graficamente confusionario implementare semplicemente la generazione contemporanea delle linee. Ciò che si è implementato è una sorta di timer che detti il tempo di creazione degli oggetti, in grado di aumentare o diminuire la velocità degli oggetti invisibili a seconda se essi debbano compiere una tratta breve o no per raggiungere un altro punto chiave. Infatti, per raggiungere ogni punto c'è bisogno di calcolare una proporzione tra tempo totale e distanza tra le varie centraline che quel percorso coinvolge: per esempio, se ho solo due centraline e la prima è molto più vicina della seconda, significa che l'oggetto dovrà arrivare al primo punto in un tempo più breve di quanto egli stesso ci metta per raggiungere in un secondo momento l'altro punto.

Navigation Mesh

La *Navigation Mesh*, letteralmente **maglia di navigazione**, è una struttura dati astratta, appartenente al campo dell'intelligenza artificiale di Unity stesso, che può essere applicata ad una qualsiasi mesh al fine di produrre attraverso un algoritmo di ricerca un particolare

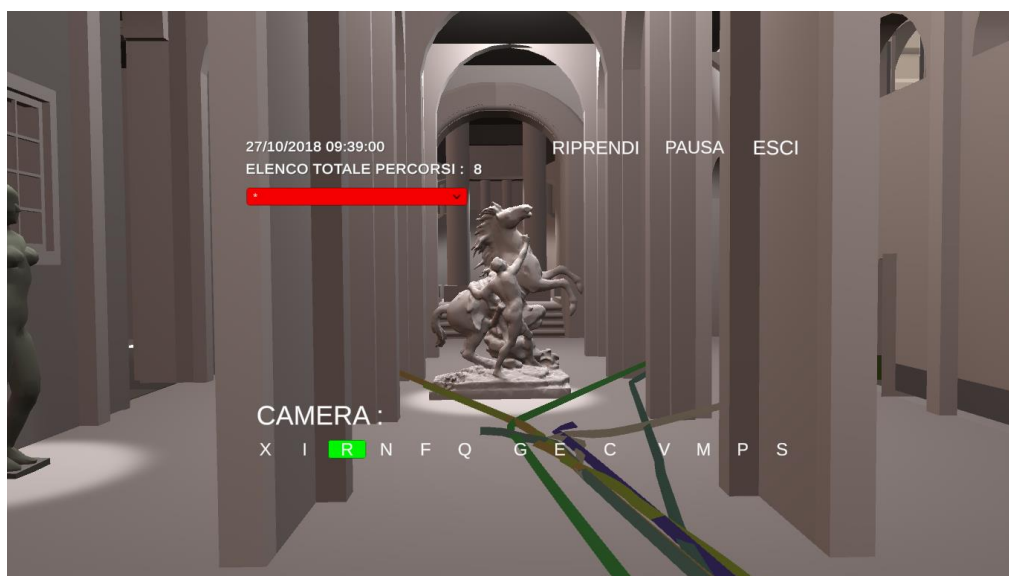
tipo di comportamento motorio all'interno di spazi notoriamente vasti. La funzione della NavMesh è quella di consentire al soggetto in questione di muoversi in relazione al contesto fisico che lo circonda, evitando collisioni e cercando di trovare quindi il percorso più semplice e realistico possibile in termini di scelte. Ciò però costa non pochi problemi, in quanto qualora si dovesse prevedere un percorso che richiede il passaggio per un WayPoint situato in zone impervie, l'agente tenderà sempre a prendere la strada più breve, anche se prevede il passaggio per zone inaccessibili. Ecco perché è stato importante definire le zone non percorribili, anche sfruttando stratagemmi come l'utilizzo di qualche oggetto invisibile che faccia da ostacolo invalicabile per l'agente.

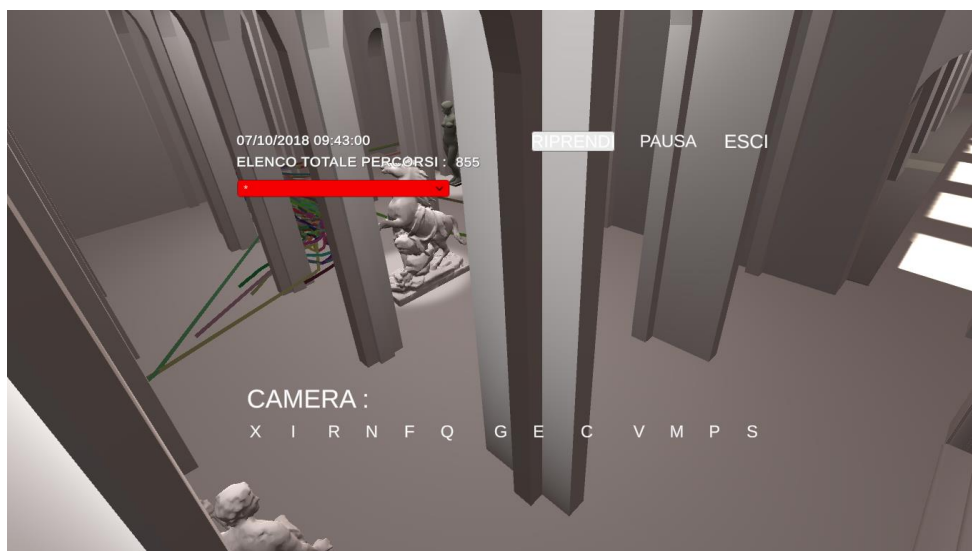
Modalità di visualizzazione

Dopo aver specificato tutte le componenti volte a rappresentare i percorsi stessi, ci si è posti il problema della visualizzazione multipla di essi, vista la potenziale grande dimensione del file di dataset dovuta alle centinaia di visitatori al giorno del museo stesso.

Si è quindi scelto di prevedere un nuovo menu grafico, secondo il quale è possibile, oltre a mettere in pausa o di riprodurre la generazione dei percorsi, scegliere di visualizzare un solo percorso alla volta, tra quelli già precedentemente filtrati. Selezionando un percorso specifico dall'apposito menù a tendina, viene mostrato un unico percorso, rendendo invisibili tutte le altre linee generate.

Inoltre, per poter visualizzare cosa accade nella prossimità delle centraline stesse, sono state previste delle telecamere di videosorveglianza, che è possibile attivare in mutua esclusione, sempre utilizzando il suddetto menù per la selezione dei percorsi. Viene quindi prevista una serie di bottoni, per l'attivazione della visuale del raggio d'azione della centralina specifica per quel pulsante. Vediamo la schermata di visualizzazione e qualche esempio di camera.







APPLICAZIONE WEB

Oltre alle modalità di avvio dell'applicazione tramite sistema operativo Windows, si è pensato che per essere più portabile possibile, l'applicazione avrebbe dovuto consentire anche una modalità di esecuzione tramite Web. A tal proposito viene utilizzato una piattaforma di clouding per effettuare caricamenti in rete di eseguibili, chiamata *SimmerIO*.

Durante lo sviluppo della modalità di esecuzione web, il codice base dell'applicazione non è stato modificato, ma si è dovuti ricorrere ad un accorgimento per il caricamento del dataset. Infatti, il caricamento non avviene mediante la navigazione attraverso i file di sistema della propria macchina, ma si è permesso la scrittura dell'URL di un file dataset, di tipo testuale.

A tal ragione si è pensato quindi di non andare ulteriormente ad inserire alla parte web dell'applicazione l'intera directory dell'intero progetto, ma viene allegata solo la funzionalità relativa alla nuova modalità di import del file testuale.

Per raggiungere l'applicativo tramite web, è sufficiente inserire il seguente link all'interno di un qualsiasi browser web.

<https://simmer.io/@Federik17/traccia1-sm->