Biomes Run

Sommario

[1 Introduzione 6](#_Toc122696485)

[1.1 Informazioni sul progetto 6](#_Toc122696486)

[1.2 Abstract 6](#_Toc122696487)

[1.3 Scopo 6](#_Toc122696488)

[2 Analisi 7](#_Toc122696489)

[2.1 Analisi del dominio 7](#_Toc122696490)

[2.2 Analisi e specifica dei requisiti 7](#_Toc122696491)

[2.3 Use case 11](#_Toc122696492)

[2.4 Pianificazione 12](#_Toc122696493)

[2.5 Analisi dei mezzi 13](#_Toc122696494)

[2.5.1 Software 13](#_Toc122696495)

[2.5.2 Hardware 13](#_Toc122696496)

[3 Progettazione 14](#_Toc122696497)

[3.1 Design dell’architettura del sistema 15](#_Toc122696498)

[3.2 Design dei dati e database 16](#_Toc122696499)

[3.3 Design delle interfacce 17](#_Toc122696500)

[3.4 Design procedurale 24](#_Toc122696501)

[4 Implementazione 25](#_Toc122696502)

[4.1 Costruzione labirinto 25](#_Toc122696503)

[4.2 Costruzione 3 livelli in Unity 27](#_Toc122696504)

[4.3 Decorazione Labirinto 28](#_Toc122696505)

[4.3.1 Modelli 3D 28](#_Toc122696506)

[4.3.2 Decorazione labirinto 29](#_Toc122696507)

[4.3.3 Albero cartelle 31](#_Toc122696508)

[4.3.4 Script Rotatore 31](#_Toc122696509)

[4.4 Prima persona 32](#_Toc122696510)

[4.4.1 Script telecamera 33](#_Toc122696511)

[4.4.2 Script prima persona personaggio 35](#_Toc122696512)

[4.5 Vite personaggio 38](#_Toc122696513)

[4.6 Programmazione mostro 40](#_Toc122696514)

[4.7 Programmazione schermate 41](#_Toc122696515)

[4.7.1 Schermata 0 (Benvenuto) 41](#_Toc122696516)

[4.7.2 Schermata 1 Descrizione Livello 1 42](#_Toc122696517)

[4.7.3 Schermata 2 Gioco Livello 1 42](#_Toc122696518)

[4.7.4 Schermata 3 (Secondo livello) 43](#_Toc122696519)

[4.7.5 Schermata 4 Gioco Livello 2 43](#_Toc122696520)

[4.7.6 Schermata 5 (Livello3) 44](#_Toc122696521)

[4.7.7 Schermata 6 Gioco Livello 3 44](#_Toc122696522)

[4.7.8 Schermata 7 (Vittoria) 45](#_Toc122696523)

[4.7.9 Schermata 8 (Chiave falsa) 45](#_Toc122696524)

[4.7.10 Schermata 9 (Vite finite) 46](#_Toc122696525)

[4.7.11 Schermata 10 (Pausa) 47](#_Toc122696526)

[4.7.12 Script gestione Schermate 49](#_Toc122696527)

[4.8 Script Gemme livelli 50](#_Toc122696528)

[4.9 Script Gestione apertura porte livelli 54](#_Toc122696529)

[4.10 Script gestione velocità stelle 56](#_Toc122696530)

[5 Test 57](#_Toc122696531)

[5.1 Protocollo di test 57](#_Toc122696532)

[5.2 Risultati test 60](#_Toc122696533)

[5.3 Mancanze/limitazioni conosciute 61](#_Toc122696534)

[6 Consuntivo 62](#_Toc122696535)

[7 Conclusioni 63](#_Toc122696536)

[7.1 Sviluppi futuri 63](#_Toc122696537)

[7.2 Considerazioni personali 63](#_Toc122696538)

[8 Glossario 63](#_Toc122696539)

[9 Bibliografia 64](#_Toc122696540)

[9.1 Sitografia 64](#_Toc122696541)

Indice delle figure

[Figura 1 - Use Case 11](#_Toc122696542)

[Figura 2 - Gantt Iniziale 12](#_Toc122696543)

[Figura 3 - Diagramma delle classi 15](#_Toc122696544)

[Figura 4 - Tabella classifica 16](file:///E:\01%20I3BC\M306%20+%20Progetti\BiomesRun\3_Documentazione%20(word%20e%20pdf)\SimoneRiva\Documentazone_Riva_Chiarani.docx#_Toc122696545)

[Figura 5 - Schermata iniziale 17](#_Toc122696546)

[Figura 6 - Schermata di pausa 18](#_Toc122696547)

[Figura 7 - Schermata di vittoria 19](#_Toc122696548)

[Figura 8 - Livello uno 20](#_Toc122696549)

[Figura 9 - Livello due 20](#_Toc122696550)

[Figura 10 - Livello tre 21](#_Toc122696551)

[Figura 11 - Schermata perdita chiave falsa 22](#_Toc122696552)

[Figura 12 - Schermata perdita vite 23](#_Toc122696553)

[Figura 13 - Diagramma UML 24](#_Toc122696554)

[Figura 14 - Empty Object 25](#_Toc122696555)

[Figura 15 - Costruzione muri 25](#_Toc122696556)

[Figura 16 – Inizio costruzione labirinto 26](#_Toc122696557)

[Figura 17 - Layout livello 3 26](file:///E:\01%20I3BC\M306%20+%20Progetti\BiomesRun\3_Documentazione%20(word%20e%20pdf)\SimoneRiva\Documentazone_Riva_Chiarani.docx#_Toc122696558)

[Figura 18 - Layout livello 2 26](file:///E:\01%20I3BC\M306%20+%20Progetti\BiomesRun\3_Documentazione%20(word%20e%20pdf)\SimoneRiva\Documentazone_Riva_Chiarani.docx#_Toc122696559)

[Figura 19 – Layout livello 1 26](file:///E:\01%20I3BC\M306%20+%20Progetti\BiomesRun\3_Documentazione%20(word%20e%20pdf)\SimoneRiva\Documentazone_Riva_Chiarani.docx#_Toc122696560)

[Figura 20 - Gerarchia modelli 28](file:///E:\01%20I3BC\M306%20+%20Progetti\BiomesRun\3_Documentazione%20(word%20e%20pdf)\SimoneRiva\Documentazone_Riva_Chiarani.docx#_Toc122696561)

[Figura 21 - Modelli 3D 28](#_Toc122696562)

[Figura 22 - Decorazione livello1 29](#_Toc122696563)

[Figura 23 - Decorazione livello 2 29](file:///E:\01%20I3BC\M306%20+%20Progetti\BiomesRun\3_Documentazione%20(word%20e%20pdf)\SimoneRiva\Documentazone_Riva_Chiarani.docx#_Toc122696564)

[Figura 24 - Decorazione livello3 30](file:///E:\01%20I3BC\M306%20+%20Progetti\BiomesRun\3_Documentazione%20(word%20e%20pdf)\SimoneRiva\Documentazone_Riva_Chiarani.docx#_Toc122696565)

[Figura 25 - Ordine oggetti 31](file:///E:\01%20I3BC\M306%20+%20Progetti\BiomesRun\3_Documentazione%20(word%20e%20pdf)\SimoneRiva\Documentazone_Riva_Chiarani.docx#_Toc122696566)

[Figura 26 - Script rotatore oggetti 31](#_Toc122696567)

[Figura 27 - Chiave con script rotazione 32](#_Toc122696568)

[Figura 28 - Pavimento Grezzo 32](#_Toc122696569)

[Figura 29 - Personaggio Grezzo 33](#_Toc122696570)

[Figura 30 - Settaggio posizione camera 33](#_Toc122696571)

[Figura 31 - movimento telecamera 34](#_Toc122696572)

[Figura 32 - Script completo movimento + funzionalità aggiuntive 36](#_Toc122696573)

[Figura 33 - Inserimento immagine vite 38](#_Toc122696574)

[Figura 34 - Script vite 38](#_Toc122696575)

[Figura 35 - Inserimento immagini nell'array 39](#_Toc122696576)

[Figura 36 - Bake del mostro 40](#_Toc122696577)

[Figura 37 - tag del mostro 40](#_Toc122696578)

[Figura 38 - script mostro 40](#_Toc122696579)

[Figura 39 - Ordine scene 41](#_Toc122696580)

[Figura 40 - Schermata 0 (Benvenuto) 41](#_Toc122696581)

[Figura 41 - controllo nome 41](#_Toc122696582)

[Figura 42 - Schermata 1 (Descrizione Livello 1) 42](#_Toc122696583)

[Figura 43 - Schermata 2 (Schermata Gioco Livello 1) 42](#_Toc122696584)

[Figura 44 - Schermata 3 (Livello 2) 43](#_Toc122696585)

[Figura 45 - Schermata 4 (Schermata Gioco Livello 2) 43](#_Toc122696586)

[Figura 46 - Schermata 5 (Livello 3) 44](#_Toc122696587)

[Figura 47 - Schermata 6 (Gioco Livello 3) 44](#_Toc122696588)

[Figura 48 - Schermata 7 (Vittoria) 45](#_Toc122696589)

[Figura 49 - Schermata 8 (Chiave falsa) 45](#_Toc122696590)

[Figura 50 - Schermata 9 (Vite finite) 46](#_Toc122696591)

[Figura 51 - Schermata 10 (Pausa) 47](#_Toc122696592)

[Figura 52 - Script di gestione del volume 47](#_Toc122696593)

[Figura 53 - Script di gestione della schermata di pausa 48](#_Toc122696594)

[Figura 54 - Script gestione schermate 49](#_Toc122696595)

[Figura 55 - Script gemme livello1 50](#_Toc122696596)

[Figura 56 - Script gemme livello2 51](#_Toc122696597)

[Figura 57 - Script gemme livello3 52](#_Toc122696598)

[Figura 58 - Script porte livello1 54](#_Toc122696599)

[Figura 59 - Script porte livello2 54](#_Toc122696600)

[Figura 60 - Script porte livello3 55](#_Toc122696601)

[Figura 61 - Script gestione stelle 56](#_Toc122696602)

[Figura 62 - Gantt Consuntivo 62](#_Toc122696603)

# Introduzione

## Informazioni sul progetto

For this project, we decided to make a game programmed with Unity.

This project was born because we like programming video games and we wanted to program one with Unity.

At the moment there are already games similar to the one we would like to create, but we would like to create a completely different version from those already present online.

in short: this work is mainly focused on entertaining users thanks to our passion for programming video games.

## Abstract

For this project, basic knowledge of Unity will be needed.

Following the requirements set by us, for this project it was essential to organize ourselves to divide the various parts of the project in order to be able to restrict the time and stay with the deliveries.

Various objects will be implemented in the game, each with a specific task, which will lead the player to think while still having fun.

The game offers various levels, each with different characteristics that will not make the player bored during his gaming experience.

**Results**

This game is made from scratch with a graphics engine (Unity). Not having followed a real module related to it but a simple two-day course, it will be a new thing for us too.

## Scopo

Lo scopo del progetto è principalmente riuscire a far divertire il maggior numero di persone con il nostro videogioco perché ci piace far divertire la gente, inoltre abbiamo deciso questo progetto per poter aumentare le nostre conoscenze con il linguaggio C# usando il motore grafico Unity.

# Analisi

## Analisi del dominio

Il gioco verrà utilizzato principalmente dai ragazzi, essendo un gioco progettato appositamente per loro, perché il nostro gioco ha come scopo di uscire da un labirinto nel minor tempo possibile.  
Inoltre consente ai ragazzi di liberare la mente e svagarsi restando comunque attivi mentalmente attraverso un gioco tranquillo e di ragionamento.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Costruzione labirinto |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Costruire la base del labirinto |
| **Sotto requisiti** | |
| **001** | Analisi e progettazione completate |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Prima persona |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | La visuale dell’utente sarà in prima persona |
| **Sotto requisiti** | |
| **001** | Il personaggio deve essere completato |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Movimento personaggio |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | L’utente deve avere la possibilità di spostarsi all’interno del labirinto |
| **Sotto requisiti** | |
| **001** | Il layout del gioco deve essere completato in maniera grezza (Livelli) |
| **002** | Il personaggio deve essere completato |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | **Decorazione labirinto** |
| **Priorità** | **2** |
| **Versione** | **1.0** |
| **Note** | **Decorare il labirinto grezzo, far si che ci siano i 3 biomi dei 3 livelli ed aggiungere delle canzoncine spaventose e posizionare tutti gli oggetti 3d** |
| **Sotto requisiti** | |
| **001** | **Labirinto grezzo completo** |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Programmazione mostri |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Nei livelli ci saranno dei mostri che cercheranno di fermare il giocatore |
| **Sotto requisiti** | |
| **001** | Il layout livelli deve essere completato |
| **002** | Il personaggio deve essere completato |

|  |  |
| --- | --- |
| **ID: REQ-06** | |
| **Nome** | Vite personaggio |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | L’utente dispone di tre vite durante il gioco |
| **Sotto requisiti** | |
| **001** | I mostri devono essere completati |
| **002** | Il personaggio deve essere completato |

|  |  |
| --- | --- |
| **ID: REQ-07** | |
| **Nome** | **Funzionamento schermata iniziale** |
| **Priorità** | **3** |
| **Versione** | **1.0** |
| **Note** | **Rendere funzionante la schermata iniziale** |
| **Sotto requisiti** | |
| **001** | **Design schermata iniziale completato** |
| **002** | **Funzionamento gioco completo** |

|  |  |
| --- | --- |
| **ID: REQ-08** | |
| **Nome** | Funzionamento schermata impostazioni |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | Rendere funzionante la schermata iniziale |
| **Sotto requisiti** | |
| **001** | Design schermata impostazioni completato |
| **002** | Funzionamento gioco completo |

|  |  |
| --- | --- |
| **ID: REQ-09** | |
| **Nome** | Database |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | Creare un database il quale contiene i migliori tempi dei giocatori ed i loro nickname |
| **Sotto requisiti** | |
| **001** | Design schermata vincita e perdita |
| **002** | Funzionamento gioco completo |
| **003** | Funzionamento schermata impostazioni e iniziale |

|  |  |
| --- | --- |
| **ID: REQ-10** | |
| **Nome** | Funzionamento schermata vincita (classifica) |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | Quando il giocatore vince la partita compare la schermata di vincita la quale contiene una classifica dei migliori tempi (DB) |
| **Sotto requisiti** | |
| **001** | Database funzionante |
| **002** | Gioco completo |

## Use case

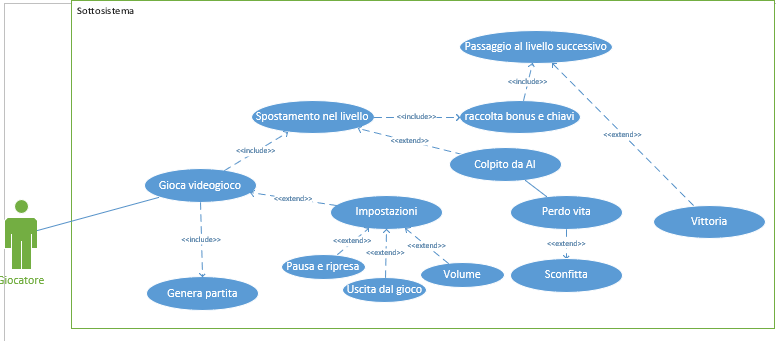


Figura - Use Case

All’interno dell’applicativo c’è solo un utente che è il giocatore:

* Esso può giocare al videogioco, ha come scopo quello di scappare dal labirinto nel minor tempo possibile prendendo tuti gli oggetti e le chiavi.
* Giocando al videogioco esso genera una nuova partita.
* Esso può utilizzare la schermata di impostazioni contenti:
  + Regolazione volume.
  + Pausa e ripresa durante il gioco.
  + Tasto per uscire dal gioc

## Pianificazione

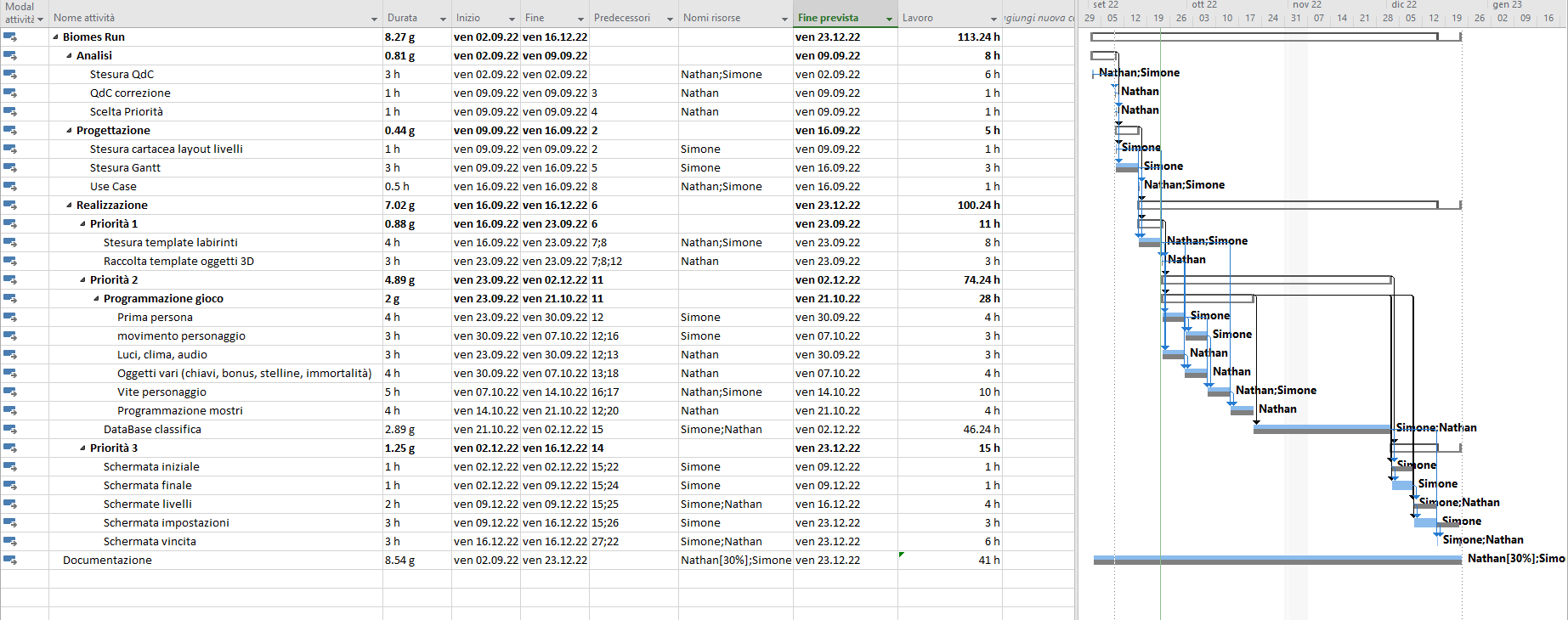


Figura - Gantt Iniziale

La durata del nostro progetto, come si puó vedere nell’immagine qui sopra, dura una settimana in meno rispetto alla data di consegna. Questo è perché ho deciso di inserire un margine di errore nel caso dovessi avere problemi oppure dovessi essere indietro rispetto al Gantt. Nel mio Gantt sono riuscito a ricavare varie attività con 29 righe da svolgere, inoltre ho ricavato 3 macrocategorie con 3 sottocategorie che sono:

1. **Analisi:** All’interno di questa macrocategoria ci sono le parti contenenti la pianificazione teorica del progetto con il Quaderno dei compiti e la sua relativa correzione e la scelta delle varie priorità delle parti del progetto.
2. **Progettazione:** All’interno di questa macrocategoria c’è l’inizio della vera e propria progettazione del progetto utilizzando un Diagramma di Gantt ed una Use Case, inoltre abbiamo iniziato a stendere i layout in maniera cartacea del vari livelli.
3. **Realizzazione:** Questa parte riguarda la vera e propria realizzazione del progetto, l’ho suddiviso in tre categorie:
   1. **Priorità 1:** In questa categoria c’è l’inizio della realizzazione dei livelli creandoli in Unity ed iniziando a cercare i vari oggetti per la loro decorazione.
   2. **Priorità 2:** In questa categoria andremo a creare il DataBase per la classifica ed è presente una sottocategoria per la programmazione vera e propria del gioco.
      1. **Programmazione gioco:**  in questa micro-categoria sono presenti tutte le parti per la programmazione del videogioco, dalla prima persona, alla programmazione dei mostri rendendo il gioco quasi completo.
   3. **Priorità 3:** In questa categoria sono presenti tutte le schermate relative al gioco che sono:
      1. **Schermata benvenuto:** Questa schermata è la schermata iniziale che da il benvenuto nel gioco all’utente
      2. **Schermata impostazioni:** Questa schermata è la schermata che permette all’utente di mettere il gioco in pausa, inoltre permette all’utente di regolare il volume, di uscire dal gioco e di riprenderlo.
      3. **Schermata livelli:** Queste schermate appaiono all’inizio di ogni livello, esse comprendono una breve descrizione del livello e di quello che si deve fare per passarlo, inoltre comprende anche una descrizione dei pericoli che aspettano l’utente nello svolgersi del livello.
      4. **Schermata vincita:** Questa schermata è la schermata finale, quando l’utente riesce a scappare dall’ultimo livello appare questa schermata contenente le congratulazioni per la vincita dell’utente.

## Analisi dei mezzi

### Software

* **Visual studio 2022**
* **Unity 2022.1**

### Hardware

* 2 PC identici: Le componenti dei PC che verranno utilizzati per lo sviluppo dell’applicativo sono:
  + I7-9700 @ 3.00GHz
  + RAM 32GB
  + SSD 512GB
  + NVIDIA GeForce RTX 2060

# Progettazione

## Design dell’architettura del sistema

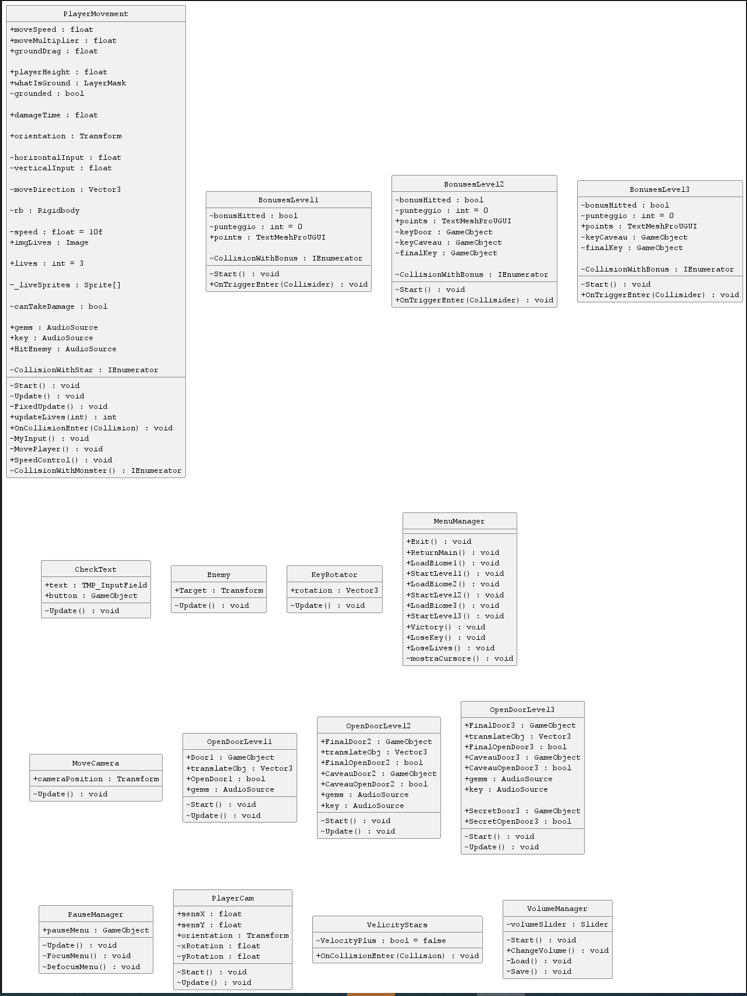


Figura - Diagramma delle classi

## Design dei dati e database

Figura - Tabella classifica

ID -> È una chiave AUTO\_INCREMENT con la quale distinguiamo i player

nickname -> È il nome utente del giocatore

time -> È il tempo che ha impiegato il giocatore per finire i 3 livelli

place -> È il posizionamento fatto ordinato in base al time (desc)

## Design delle interfacce

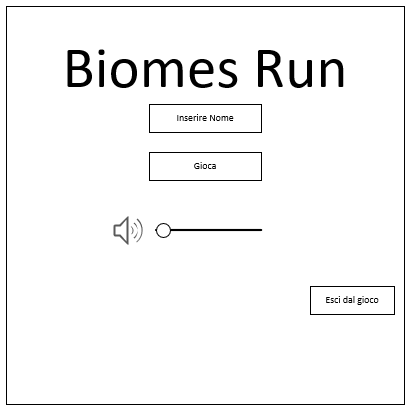


Figura - Schermata iniziale

La prima schermata che appare appena si apre il gioco è quella di benvenuto, essa è composta da:

* Il titolo del gioco.
* Un textbox dove l’utente dovrà inserire il suo nome che servirà per essere memorizzato nel DataBase per la classifica.
* Un bottone “Gioca” che permetterà all’utente di iniziare la partita.
* Un bottone “Esci dal gioco” che permetterà all’utente di uscire dal gioco.
* Uno slider che permetterà di alzare a abbassare il volume.

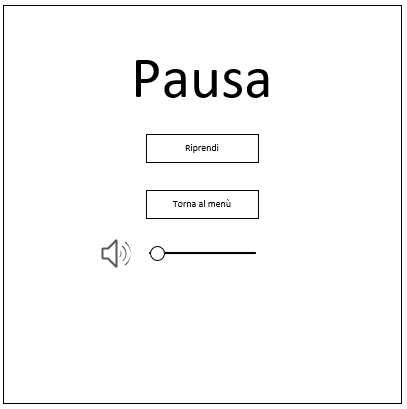


Figura - Schermata di pausa

La schermata di pausa potrà essere accessibile durante il gameplay, l’utente potrà aprirla utilizzando il tasto “esc” e si aprirà questa schermata con:

* Un bottone “Riprendi” che riprenderà il gioco.
* Un bottone “Esci dal gioco” che permetterà all’utente di uscire dal gioco.
* Uno slider che permetterà di alzare a abbassare il volume.



Figura - Schermata di vittoria

La schermata di vittoria apparirà all’utente quando riuscirà a completare anche l’ultimo livello, essa comprende:

* Il tempo impiegato per terminare il gioco
* La relativa posizione in classifica
* Un bottone “Rigioca” per ricominciare il gioco
* Un bottone “Torna al menù” per tornare alla schermata iniziale



Figura - Livello uno

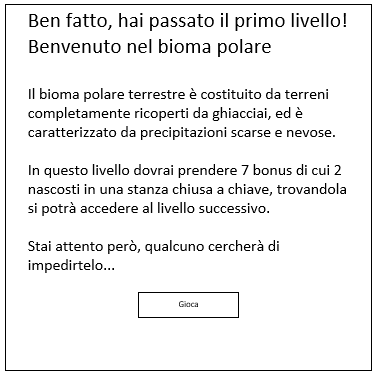


Figura - Livello due

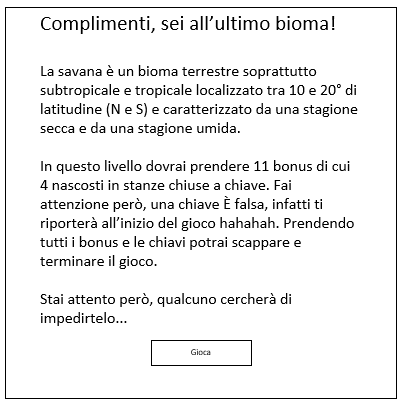


Figura - Livello tre

Le schermate dei tre livelli apparranno sempre prima dell’inizio del relativo livello e contengono:

* Una descrizione del relativo bioma.
* La spiegazione di quello che l’utente dovrà fare per accedere al livello successivo
* Un bottone “Gioca” per iniziare il livello

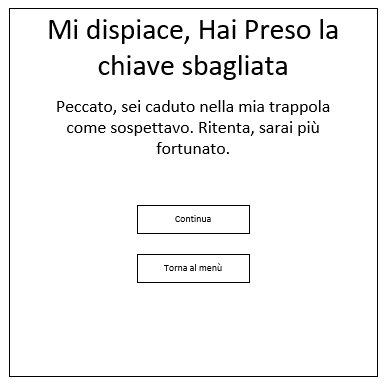


Figura - Schermata perdita chiave falsa

Questa schermata appare all’utente se nell’ultimo livello dovesse prendere la chiave sbagliata, essa contiene due bottoni:

* Il bottone “Continua” fa ricominciare il gioco all’utente
* Il bottone “Torna al menù” riporta l’utente alla schermata iniziale



Figura - Schermata perdita vite

Questa schermata appare all’utente se dovesse perdere tutte le vite per via de mostri. Essa contiene due bottoni:

* Il bottone “Rigioca” fa ricominciare il gioco all’utente
* Il bottone “Torna al menù” riporta l’utente alla schermata iniziale

## Design procedurale

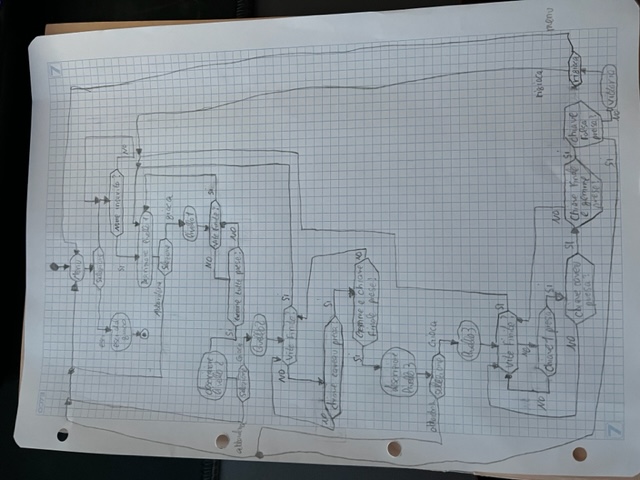


Figura - Diagramma UML

# Implementazione

## Costruzione labirinto

Per avere ordine nella barra della gerarchia creerò man mano durante la costruzione del labirinto degli empty object nei quali dividerò tutti i modelli 3d. Per esempio: ho creato un empty object chiamato Walls nel.

quale metterò tutti i muri

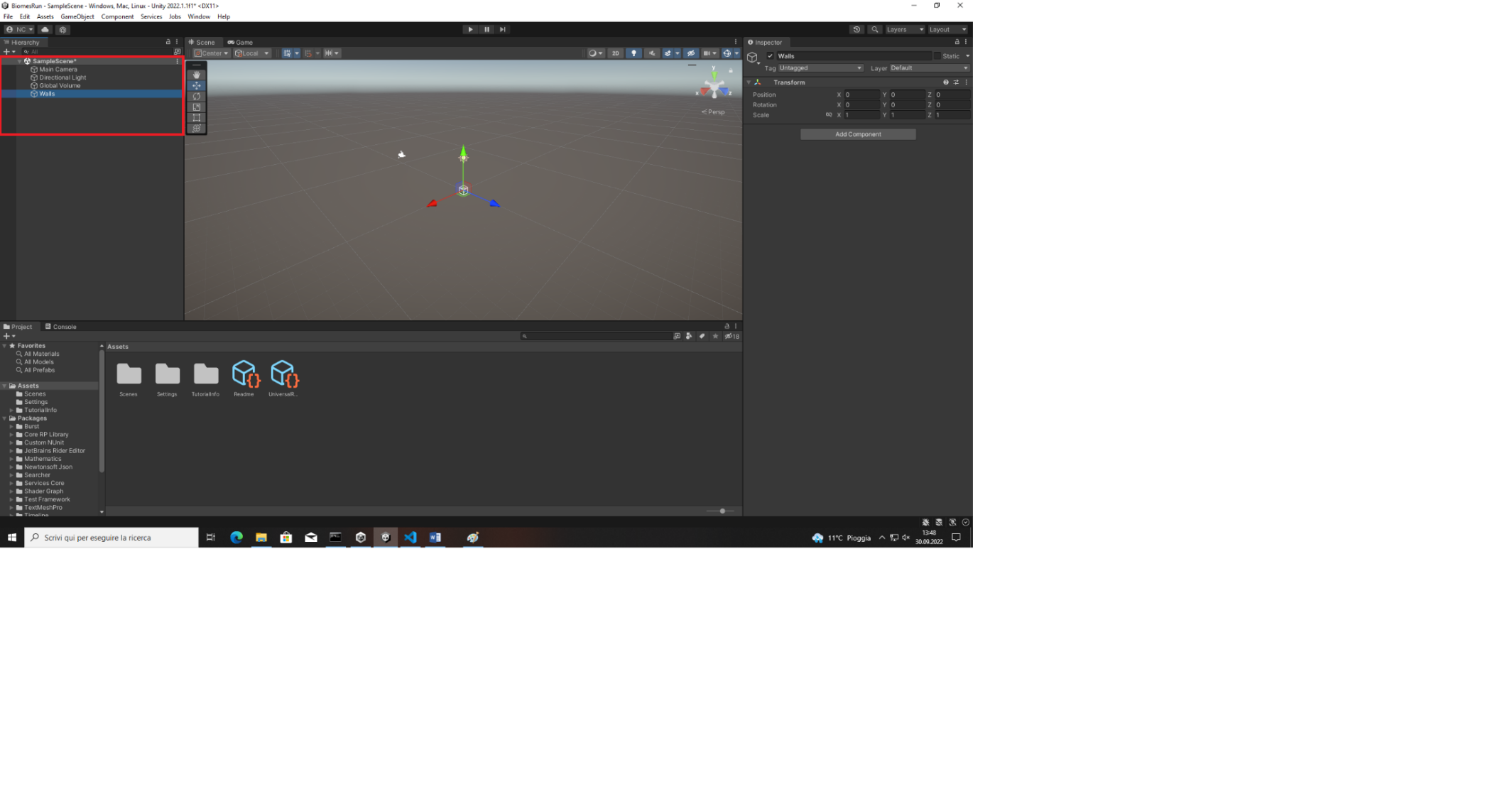


Figura - Empty Object

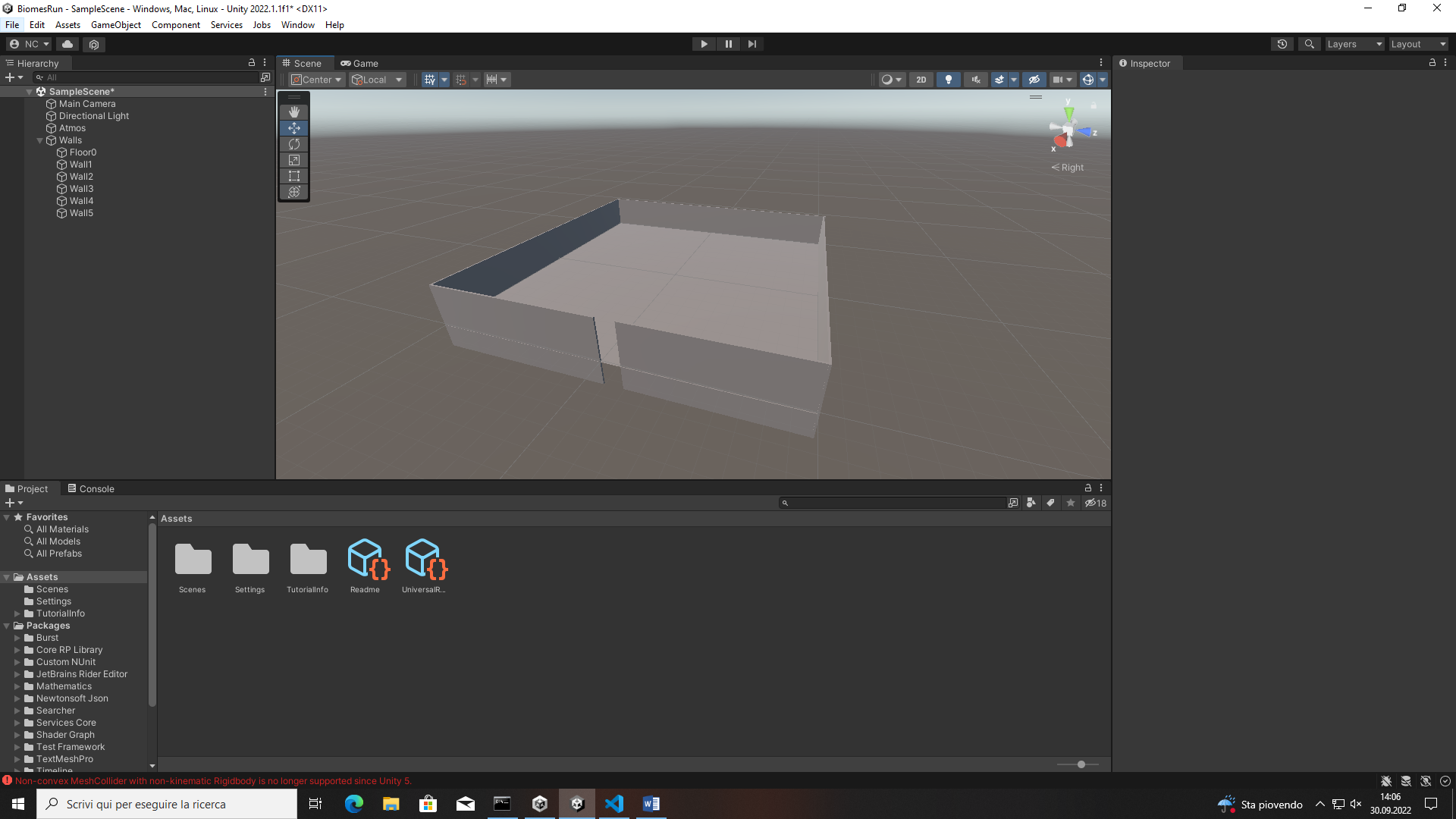
Ho creato i muri e il piano per costruire la base del labirinto. Nella barra della gerarchia si vede che i muri sono nell’ empty object Walls.

Figura - Costruzione muri

#### Layout labirinto dei 3 livelli

Prima di creare i labirinti direttamente su Unity, ho sviluppato casualmente il loro layout con Paint.

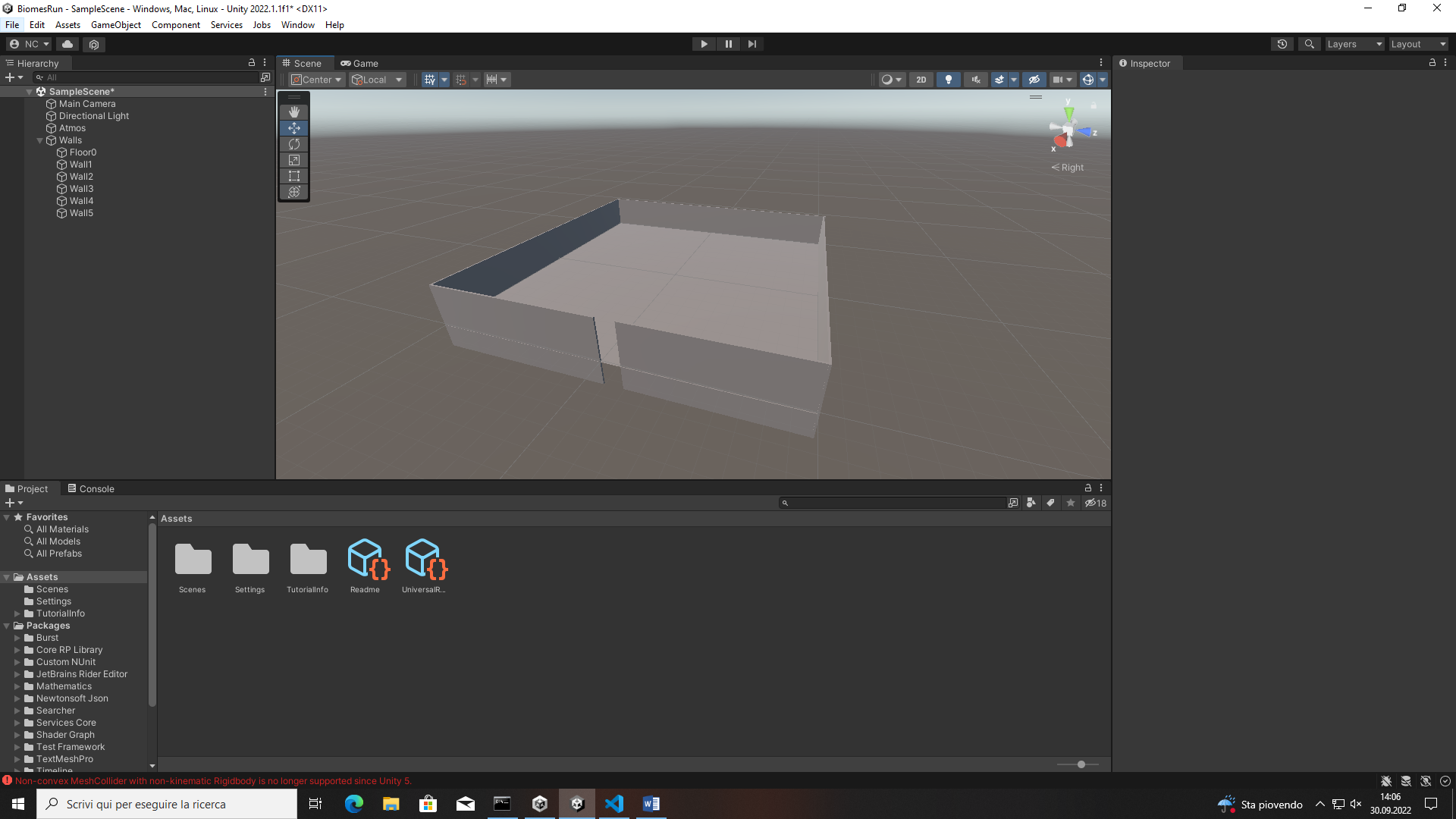


Figura – Inizio costruzione labirinto

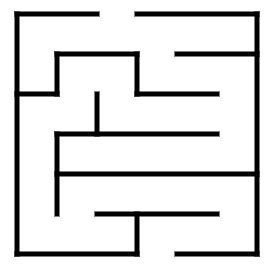
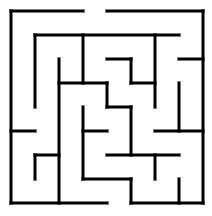
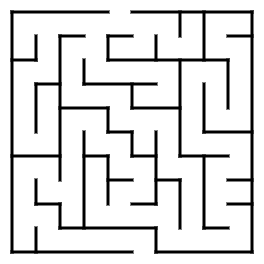


Figura - Layout livello 3

Figura - Layout livello 2

Figura – Layout livello 1

## Costruzione 3 livelli in Unity

Ho fatto il layout dei 3 labirinti seguendo più o meno i layout creati in Unity.

Legenda dei colori utilizzati:

* I buchi presenti in alto a sinistra dei labirinti segnano il punto di inizio di ogni livello
* La parete bucata in mezzo è la fine del labirinto, e quando si avrà varcato la porta c si ritroverà in alto a sinistra del livello successivo.
* I muri verdi rappresentano le porte quali si apriranno con le chiavi mentre il muro verde rappresenta una porta falsa, in quella stanza sarà presente la chiave falsa, la quale ti farà riiniziare il gioco

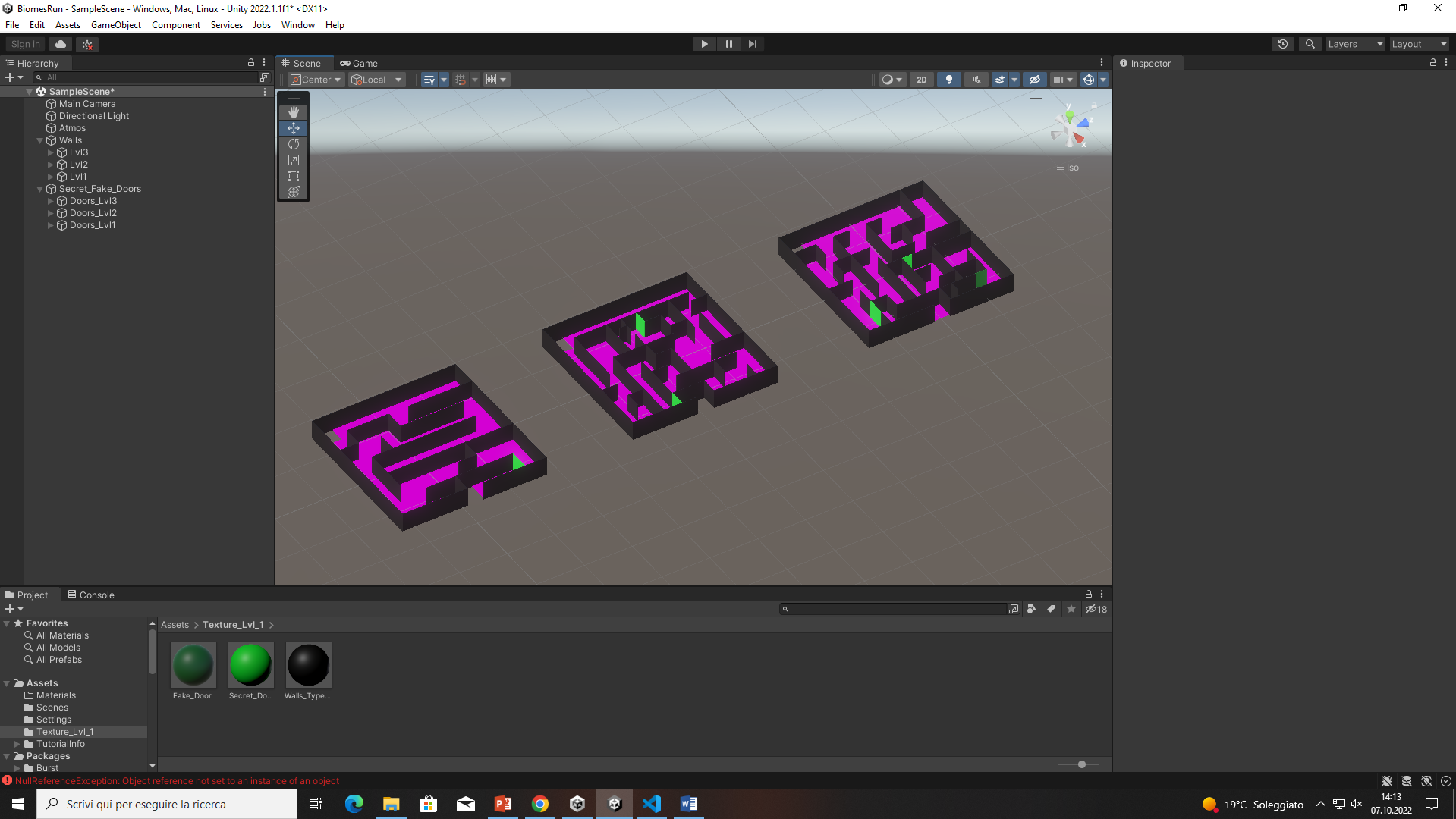


Figura 18 - Layout 3 labirinti

## Decorazione Labirinto

### Modelli 3D

Figura - Gerarchia modelli

Per decorare il labirinto ho scaricato dei modelli 3D e gli ho messi in una cartella che mi contiene tutti gli oggetti. Per avere ordine ho creato una struttura di cartelle in questo modo:

Livello\_nr/3D Models/\*.fbx

Ogni livello ha la propria cartella dei modelli 3D.

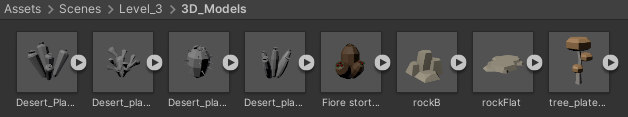


Figura - Modelli 3D

### Decorazione labirinto

Ho decorato i 3 labirinti utilizzando i modelli 3D scaricati. I 3 labirinti gli ho decorati in base ad un bioma

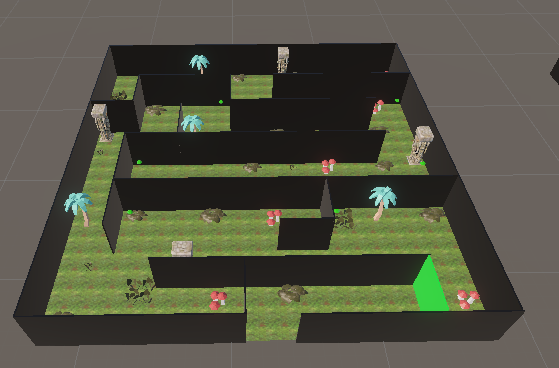
* Livello 1 è basato sul bioma della giungla 

Figura - Decorazione livello1

* Livello 2 è basato su un bioma polare

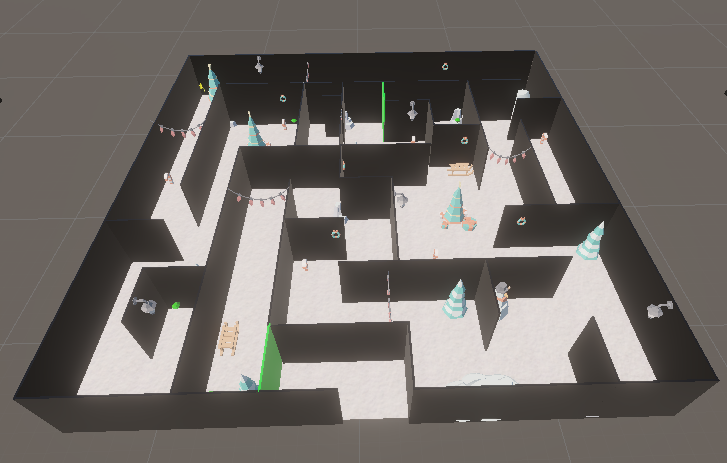


Figura - Decorazione livello 2

* Livello 3 è basato sul bioma della savana

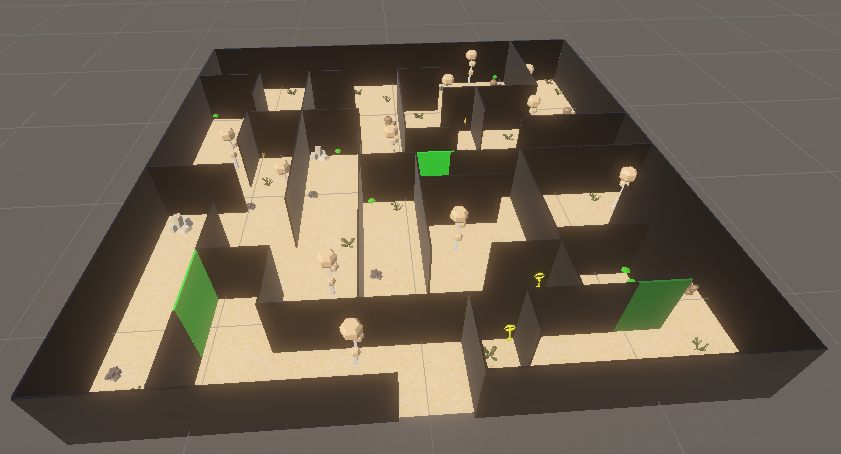


Figura - Decorazione livello3

### Albero cartelle

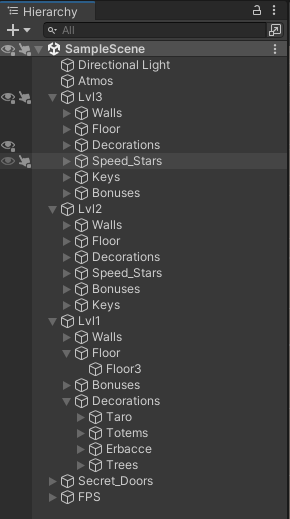
Alla fine della decorazione per avere ordine nei miei oggetti ho creato diversi empty object per smistare il tutto per livelli.

Figura - Ordine oggetti

### Script Rotatore

Chiavi, stelline e punti bonus gli ho fatti roteare su sé stessi utilizzando uno script. Ho creato la mia cartella degli Scripts in Assets e ho creato uno script chiamato KeyRotator, all’interno ho scritto questo codice:

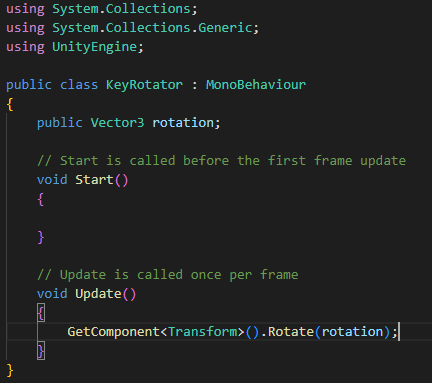


Figura - Script rotatore oggetti

E su Unity collego lo script agli oggetti, e nello script potrò scegliere la velocità e verso che verso farli girare

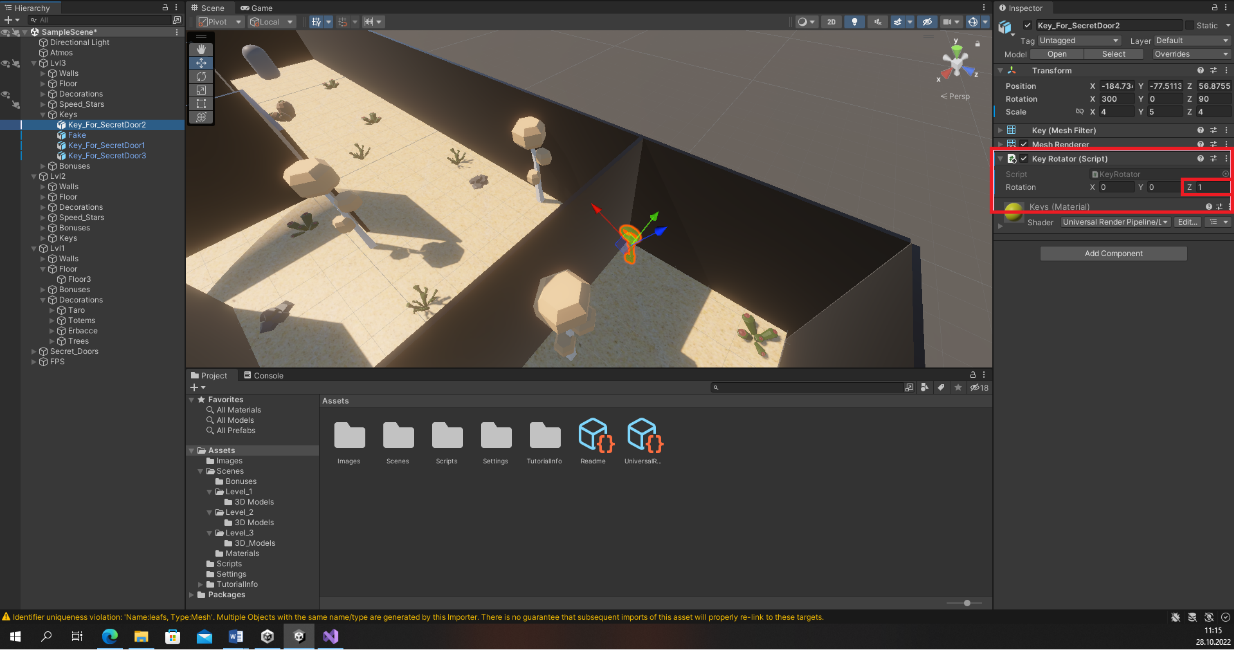


Figura - Chiave con script rotazione

## Prima persona

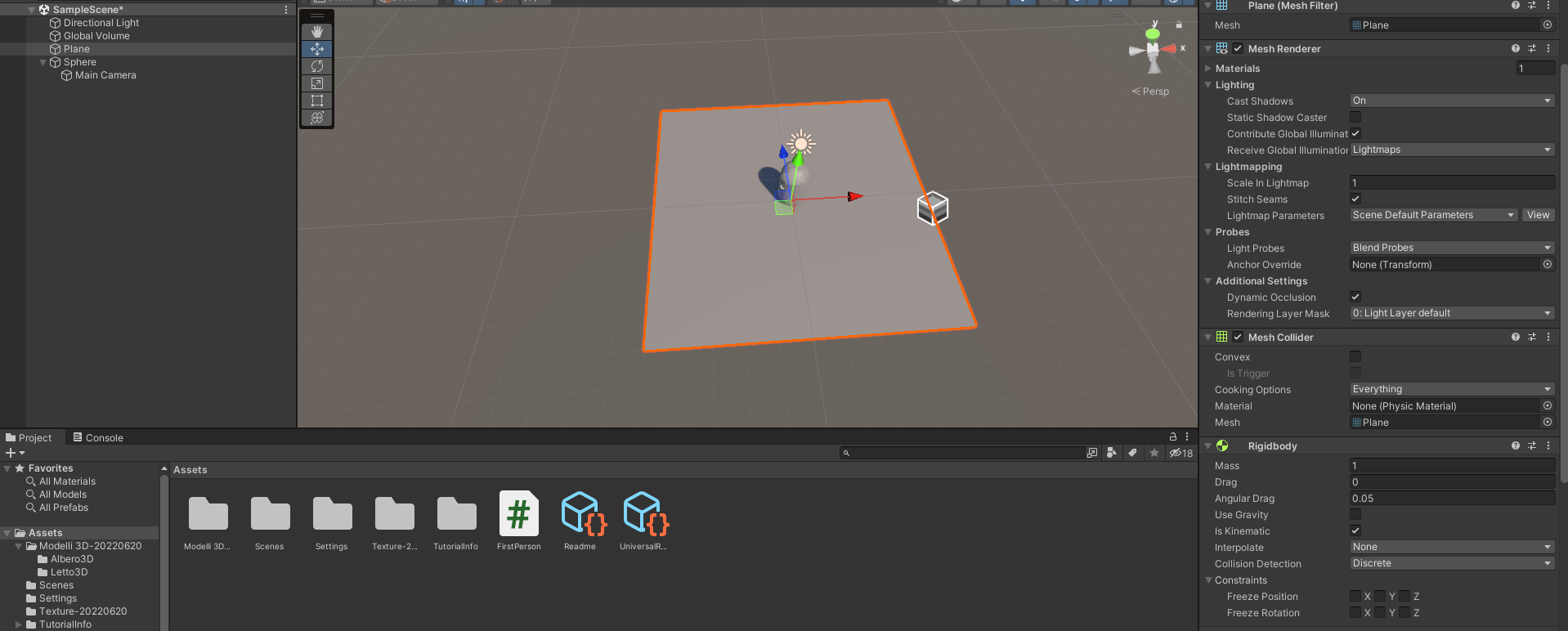
Per il movimento in prima persona ho creato un piano al quale ho aggiunto il componente Rigidbody, ho tolto la gravità e ‘ho reso statico per far si che non si sposti durante la partita

Figura - Pavimento Grezzo

Per introdurre la camera in prima persona ho creato un personaggio grezzo (Capsule) al quale ho bloccato la rotazione X e Z perché altrimenti si sarebbe spostato.

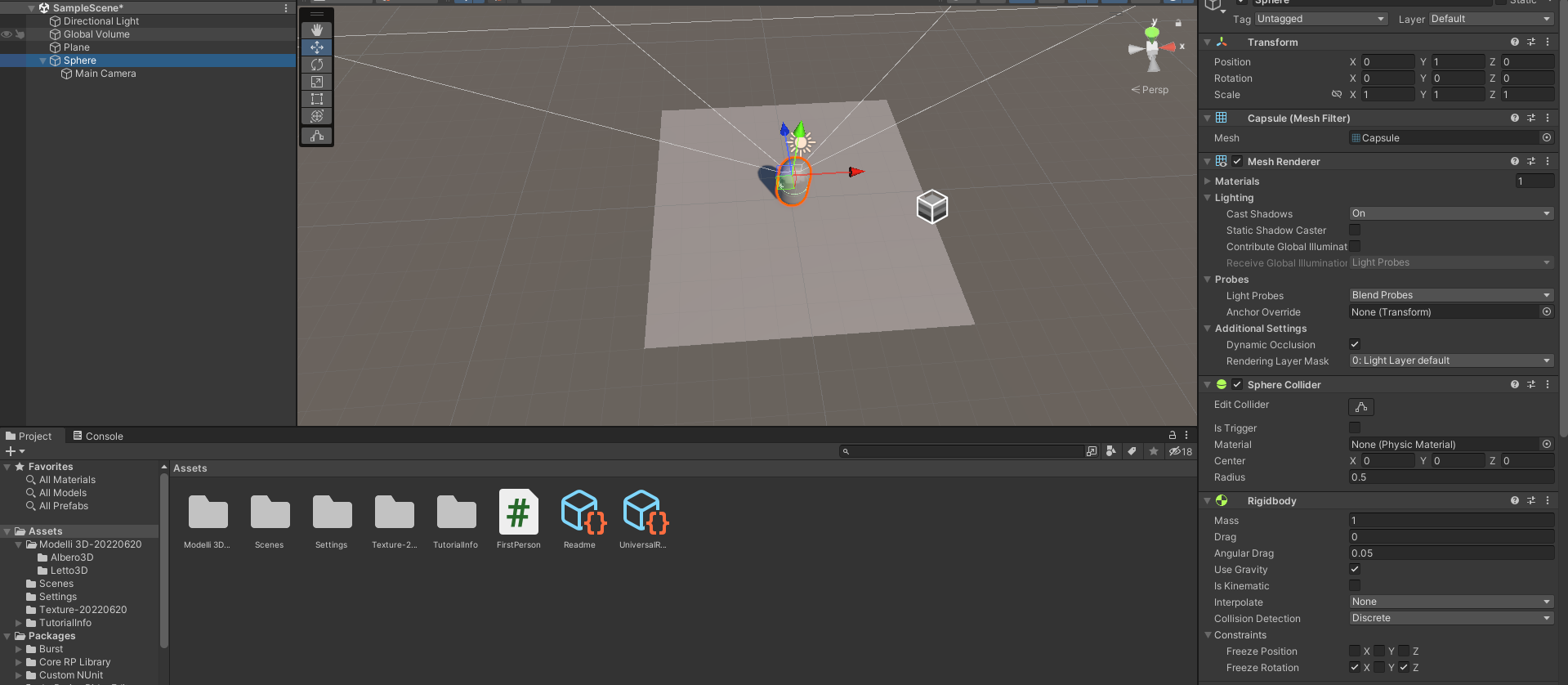


Figura - Personaggio Grezzo

### Script telecamera

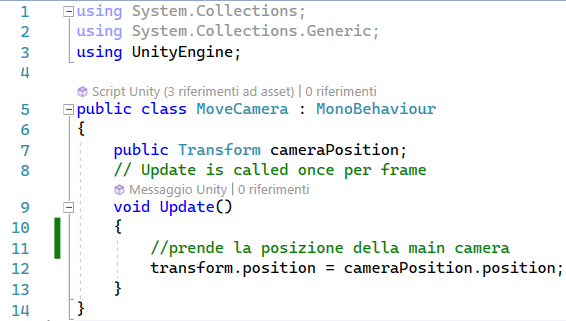


Figura - Settaggio posizione camera

Questo script permette di dare la posizione per il movimento della telecamera

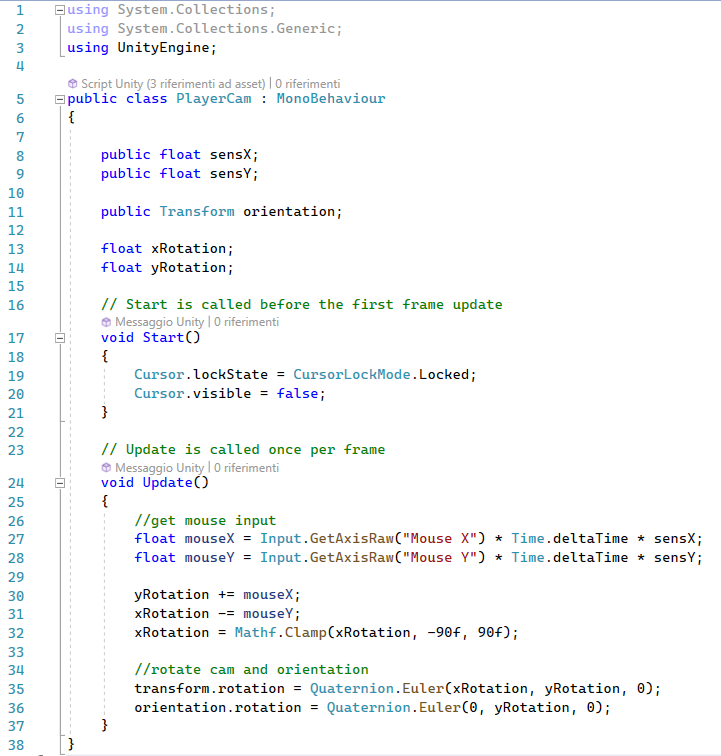
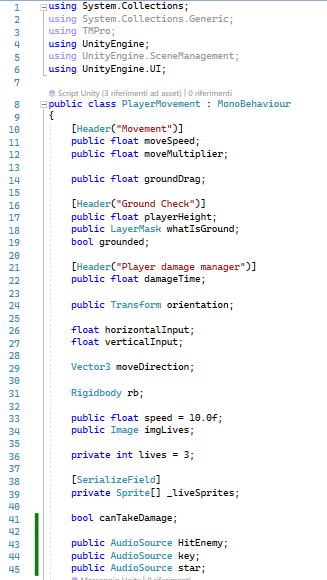


Figura - movimento telecamera

Questo script prende l’input del mouse del player e permette il movimento della telecamera con il mouse bloccando il cursore al centro dello schermo e rendendolo invisibile

### Script prima persona personaggio



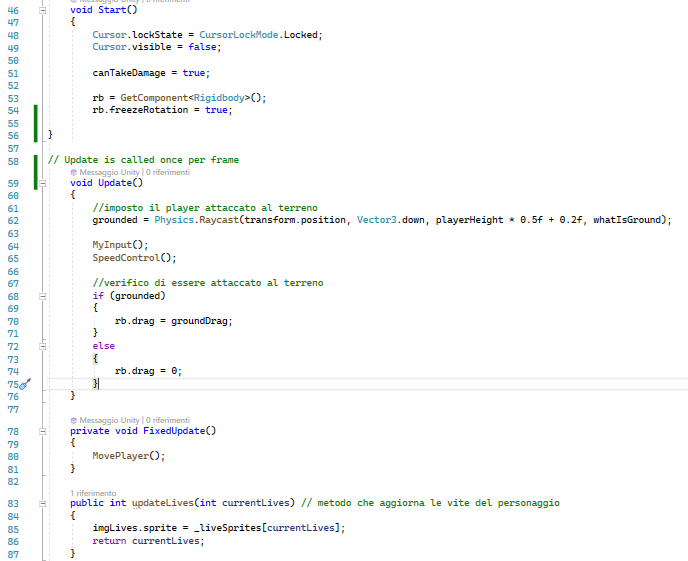


Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Figura - Script completo movimento + funzionalità aggiuntive

Questo script comprende tutto quello che concerne il movimento del player, inoltre comprende anche tutte le funzionalità per il passaggio dei livelli collidendo con i Portal, la perdita per colpa della chiave falsa, il counter delle vite con il caricamento della schermata di perdita se si dovessero perdere tutte, il controllo della velocità per fare in modo che sia omogenea e l’aumento di velocità se si collide con una stella e tutto quello che riguarda gli input del mouse per il movimento, inoltre comprende gli effetti audio per il collezionamento degli oggetti.

## Vite personaggio

Per creare le vite del personaggio ho utilizzato un sito che mi ha permesso di avere delle immagini con background trasparente, Il risultato è questo:

🡪  🡪  🡪 

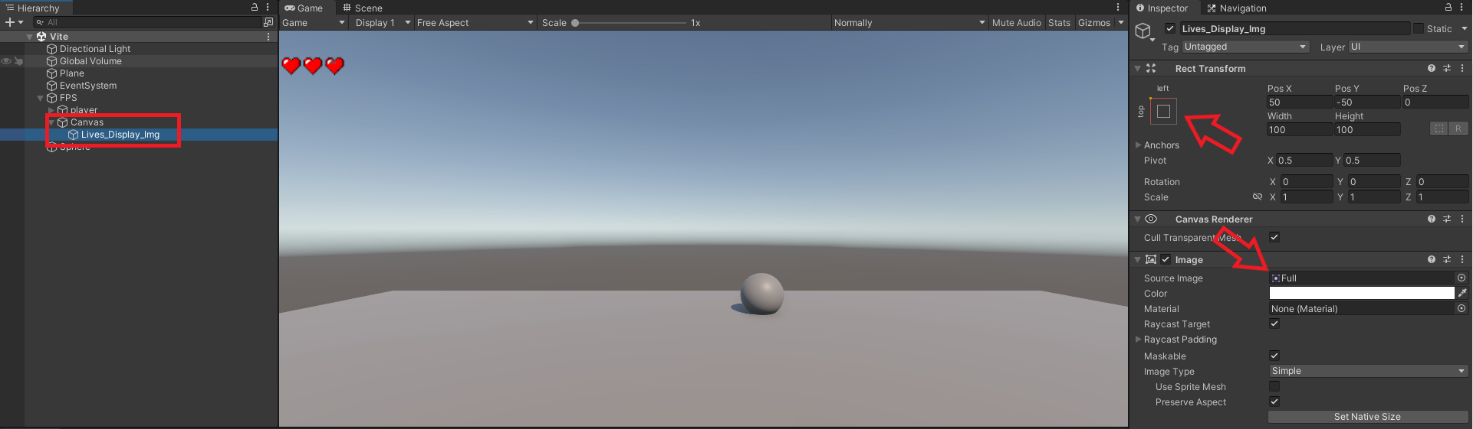
Per poter avere un player con delle vite ho dovuto creare all’interno del gameObject del player un canvas con all’interno un immagine contenente l’immagine delle tre vite: 

Figura - Inserimento immagine vite

Dopodiché ho creato uno script contenente un array di immagini dove ogni volta che il player viene colpito, incrementa una variabile che aumenta la posizione dell’array e l’immagine delle vite cambia. Dopo averlo creato ho inserito nella lista dello script le vite.

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Figura - Script vite

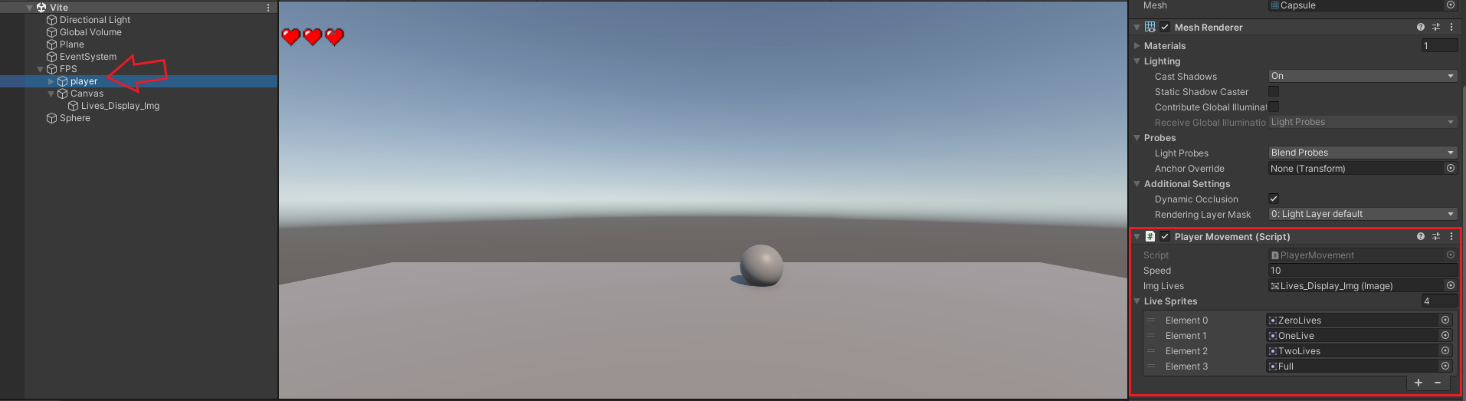


Figura - Inserimento immagini nell'array

Facendo così basta dare al nemico il tag “mostro” e il gioco è fatto. Se il player dovesse finire le vite, verrebbe caricata la schermata di perdita.

## Programmazione mostro

Per la programmazione dei mostri ho creato uno script che inseguisse il player. Inoltre, bisogna “dire” al nemico dove può passare grazie al bake, esso permette di stabilire le parti camminabili e non.

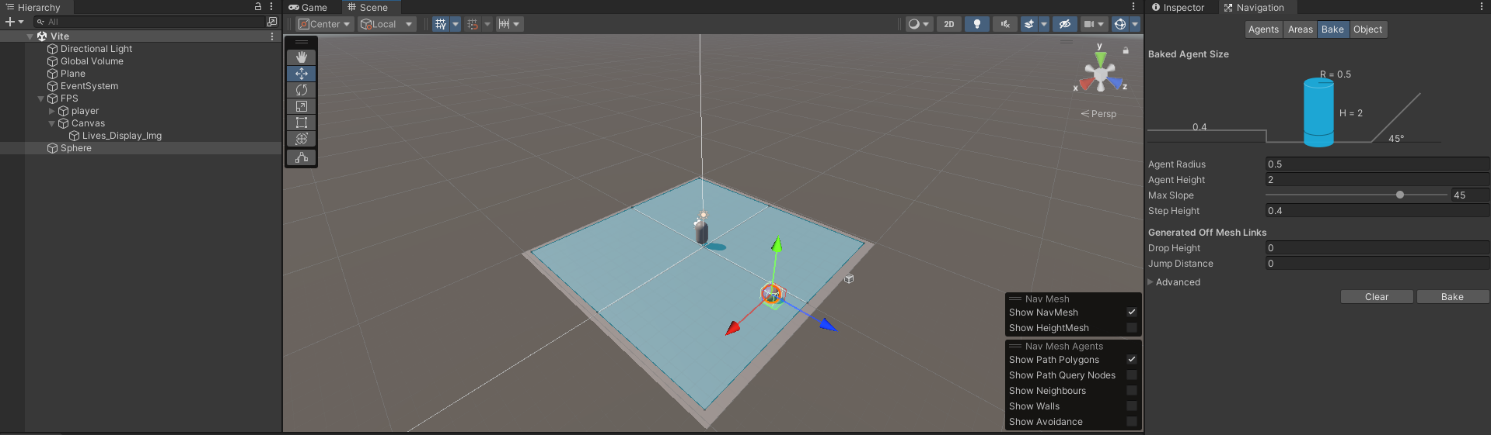


Figura - Bake del mostro

Il mostro deve avere il tag mostro per poter far funzionare il counter delle vite e l’inseguimento

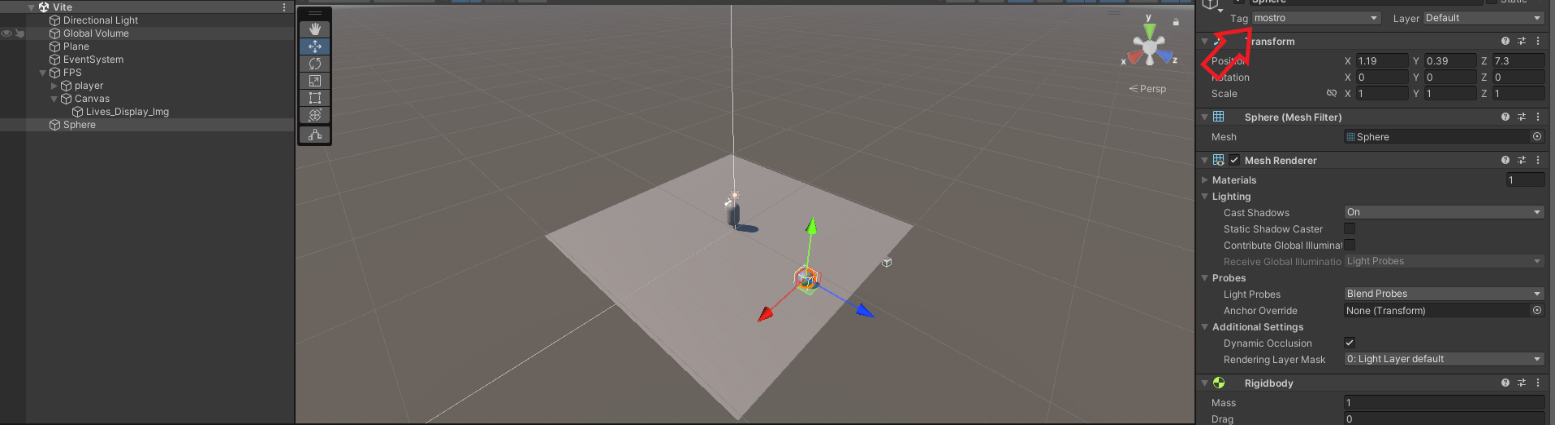


Figura - tag del mostro

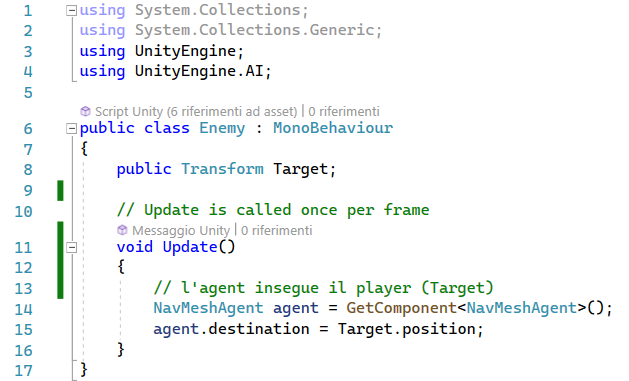


Figura - script mostro

## Programmazione schermate

Oltre alla schermata delle vite, ho creato diverse schermate:

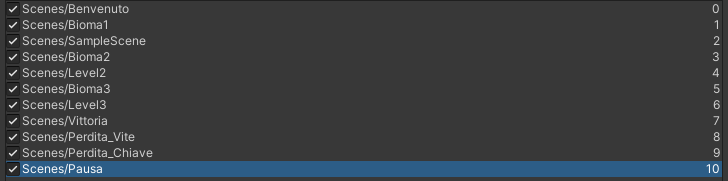


Figura - Ordine scene

### Schermata 0 (Benvenuto)

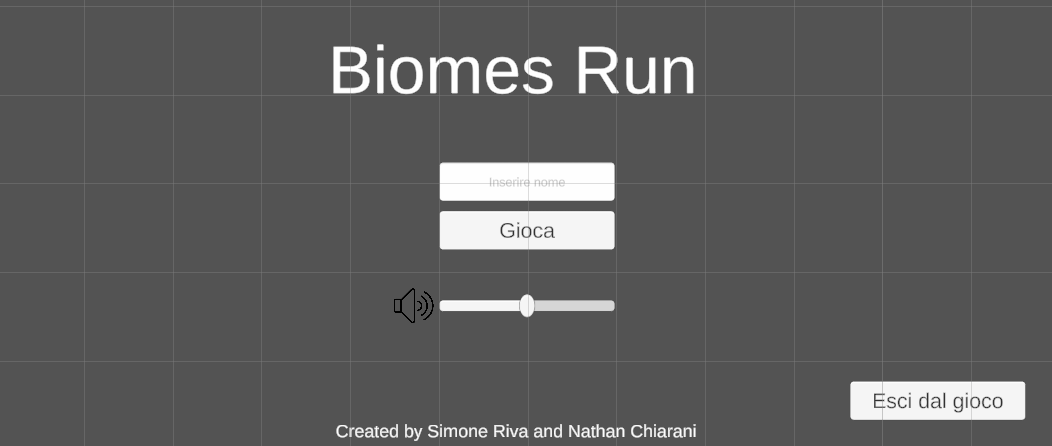


Figura - Schermata 0 (Benvenuto)

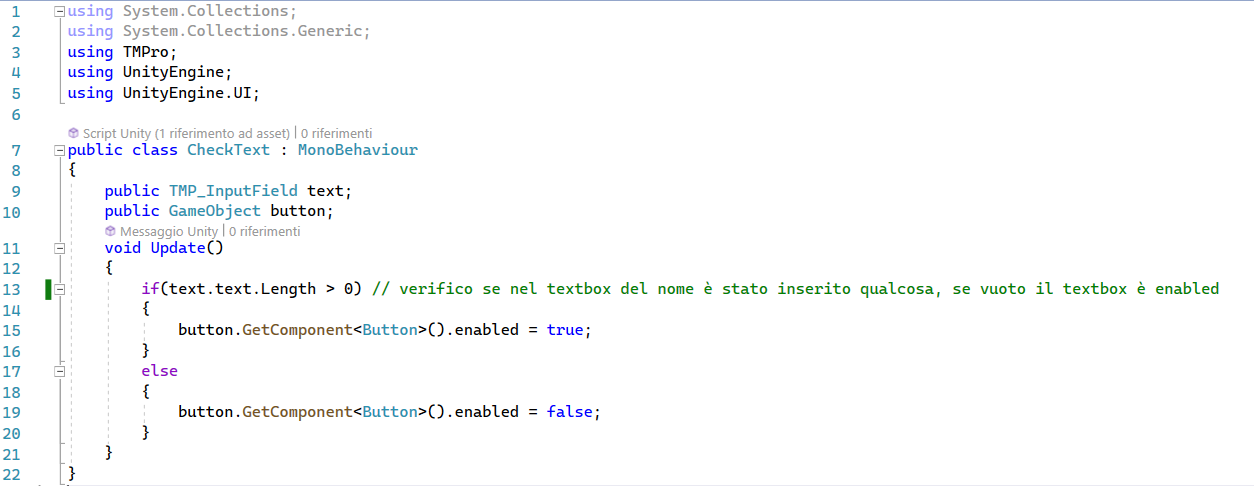
Per verificare che il nome sia stato inserito, utilizzo uno script di controllo:  


Figura - controllo nome

Se il nome non è inserito, il tasto gioca è enabled, altrimenti si può iniziare a giocare.

### Schermata 1 Descrizione Livello 1

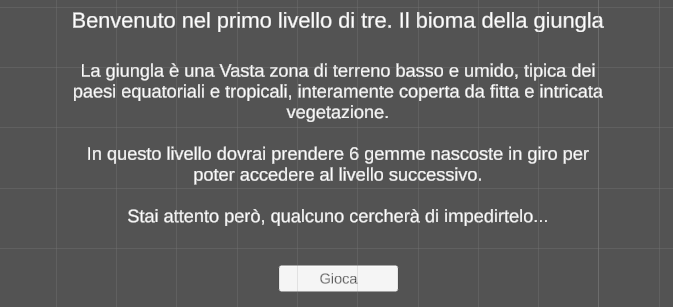


Figura - Schermata 1 (Descrizione Livello 1)

### Schermata 2 Gioco Livello 1

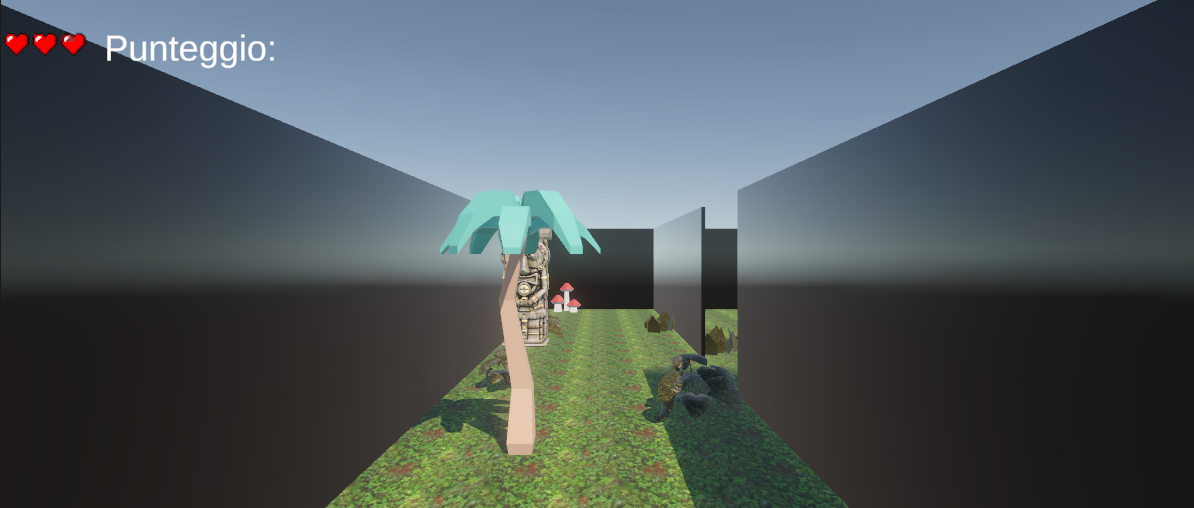


Figura - Schermata 2 (Schermata Gioco Livello 1)

### Schermata 3 (Secondo livello)

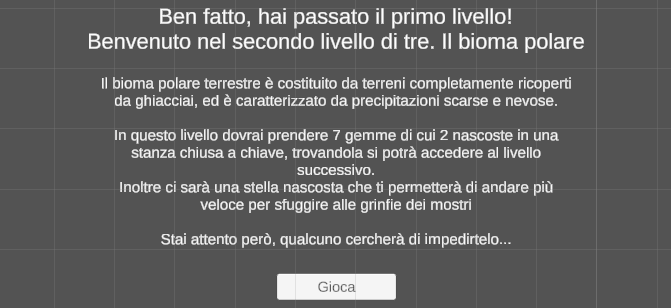


Figura - Schermata 3 (Livello 2)

### Schermata 4 Gioco Livello 2

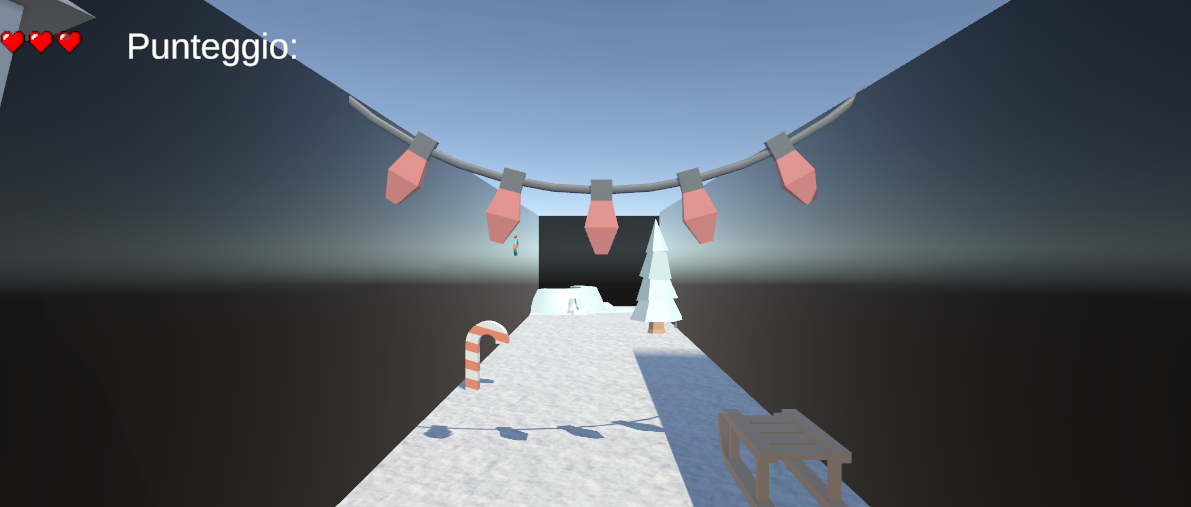


Figura - Schermata 4 (Schermata Gioco Livello 2)

### Schermata 5 (Livello3)

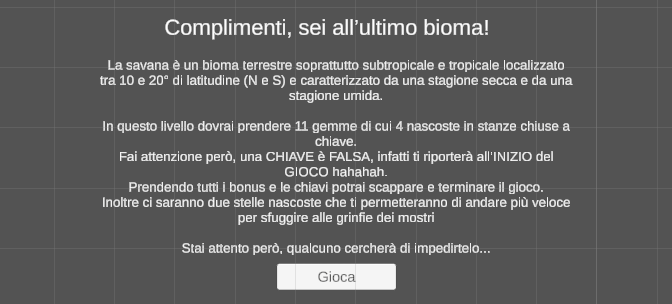


Figura - Schermata 5 (Livello 3)

### Schermata 6 Gioco Livello 3

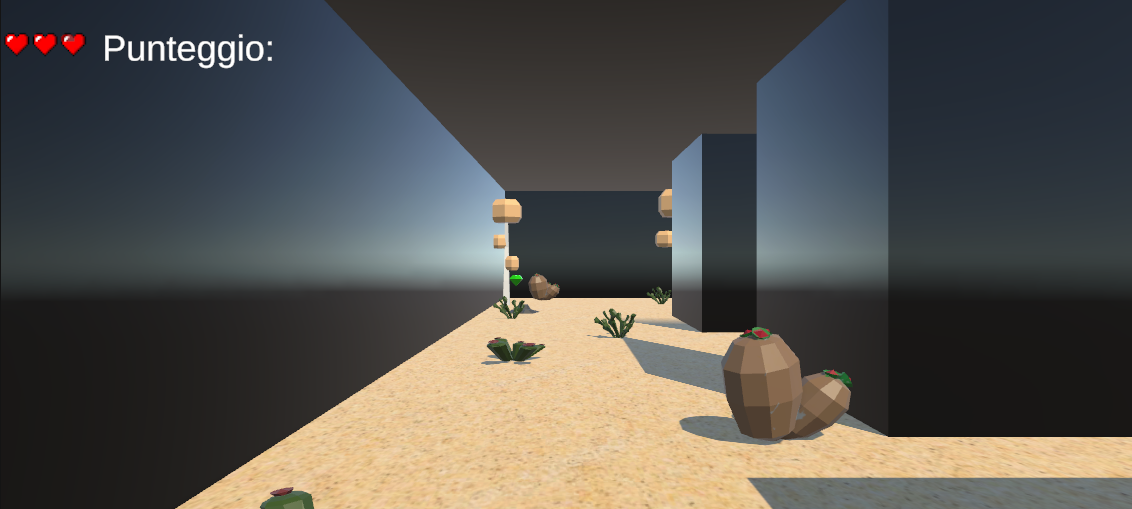


Figura - Schermata 6 (Gioco Livello 3)

### Schermata 7 (Vittoria)

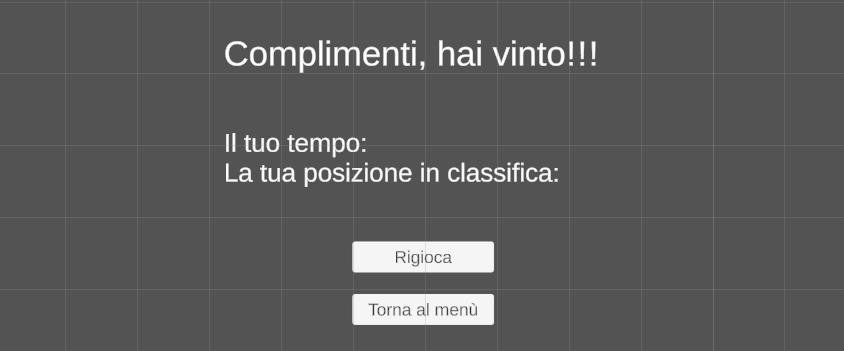


Figura - Schermata 7 (Vittoria)

### Schermata 8 (Chiave falsa)

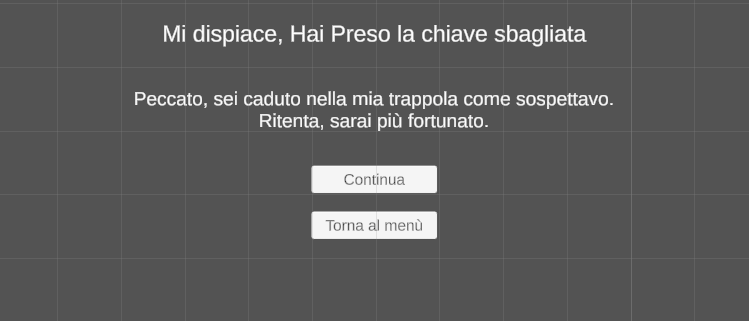


Figura - Schermata 8 (Chiave falsa)

### Schermata 9 (Vite finite)

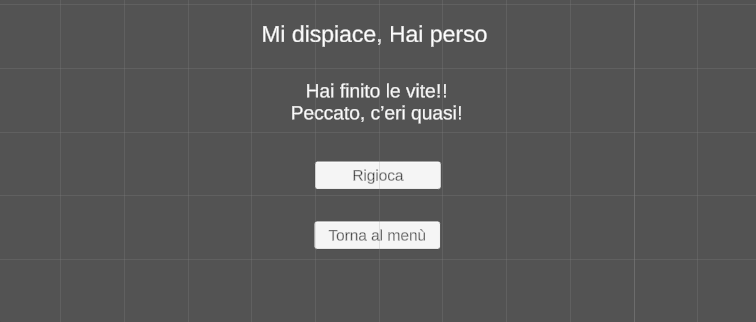


Figura - Schermata 9 (Vite finite)

### Schermata 10 (Pausa)



Figura - Schermata 10 (Pausa)

Per il volume regolabile dagli slider ho utilizzato il seguente script che imposta il volume, se non era stato impostato, lo imposta a 0.2:  


Figura - Script di gestione del volume

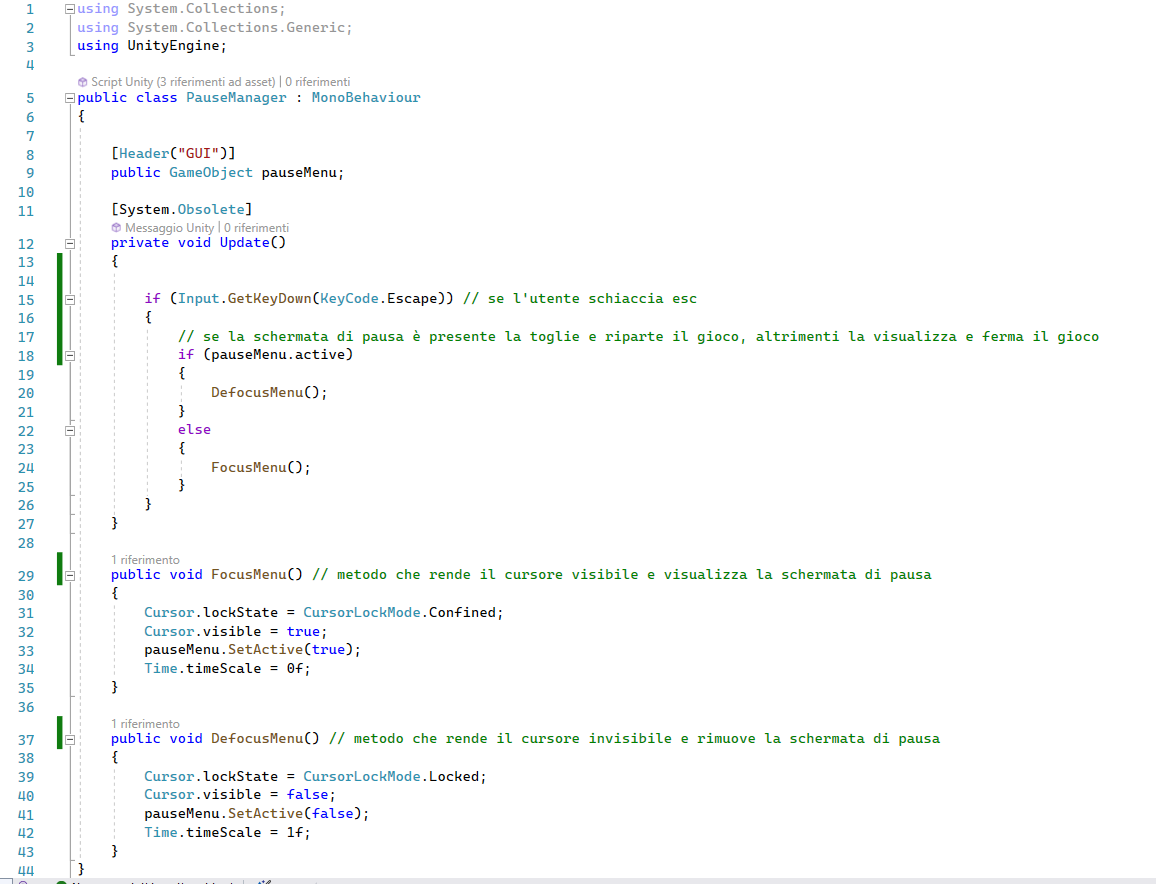
Per l’apparizione della schermata di pausa ho utilizzato il seguente script:  


Figura - Script di gestione della schermata di pausa

Questo script inizia quando l’utente preme “esc” e controlla se la schermata di pausa è in visualizzazione o meno, se l’utente la sta visualizzando, la rimuove e il gioco riparte, altrimenti la visualizza e ferma il gioco.

### Script gestione Schermate

Questo script ha il compito di gestire lo spostamento tra scene tramite metodi.

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

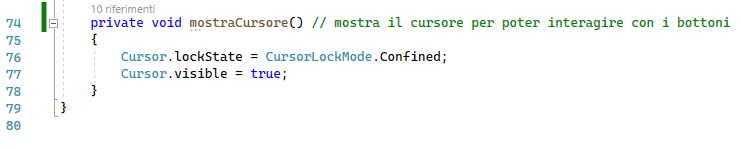


Figura - Script gestione schermate

## Script Gemme livelli

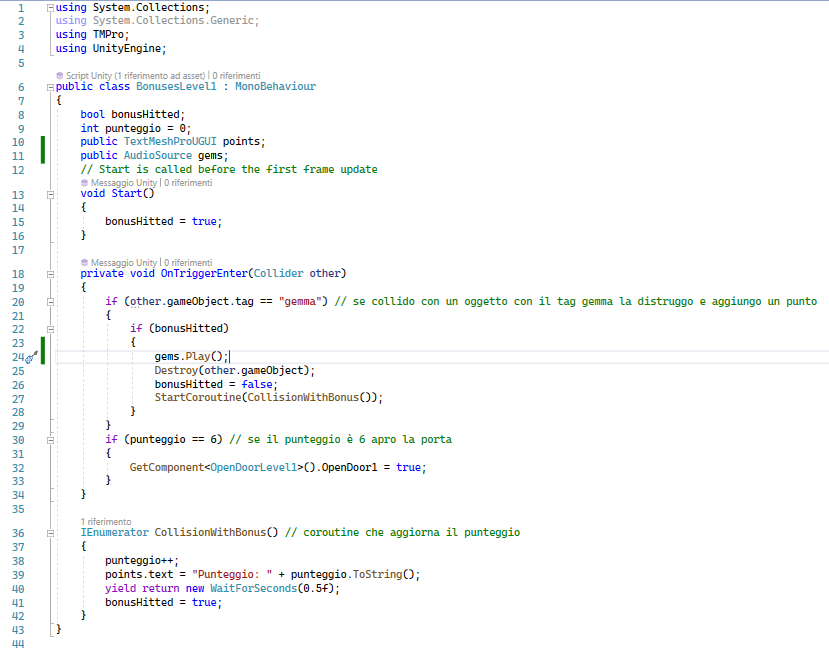


Figura - Script gemme livello1

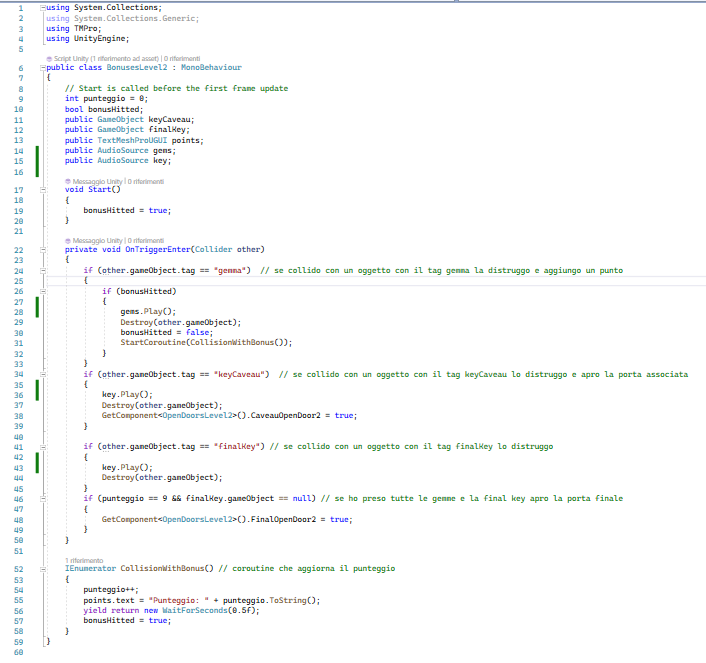
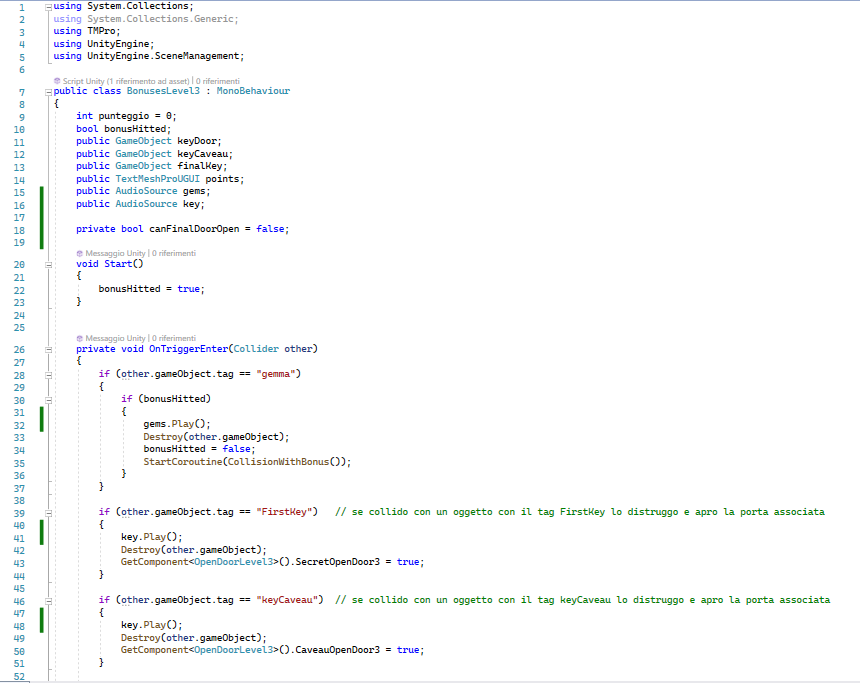


Figura - Script gemme livello2



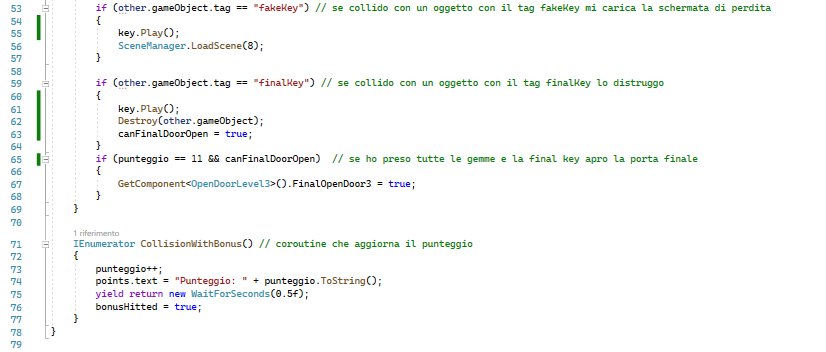


Figura - Script gemme livello3

Questi script vengono utilizzati per verificare che tutte le gemme appartenenti ai livelli siano state collezionate, inoltre verificano che le chiavi dei rispettivi livelli siano state prese e gestiscono lo spostamento in altre schermate tramite lo script MenuManager. Inoltre nello script del terzo livello viene anche gestita la schermata nel caso venga presa la chiave sbagliata. Inoltre comprende gli effetti audio per quando collezioni una gemma.

## Script Gestione apertura porte livelli

Immagine che contiene testo

Descrizione generata automaticamente

Figura - Script porte livello1

Immagine che contiene testo

Descrizione generata automaticamente

Figura - Script porte livello2



Figura - Script porte livello3

Questi script non fanno nient’altro che dare l’istruzione di far scivolare le porte grazie alla variabile di tipo Vector3 translateObj

## Script gestione velocità stelle

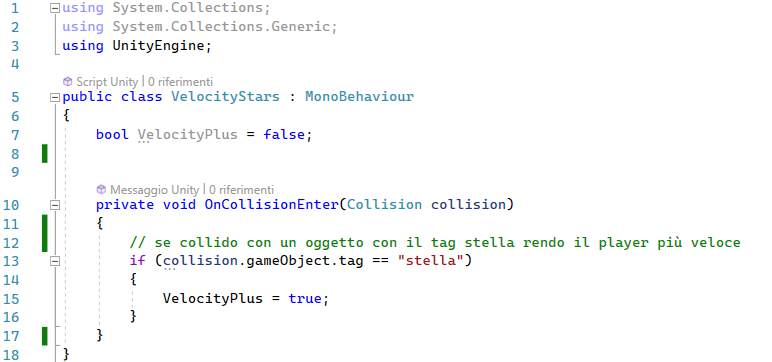


Figura - Script gestione stelle

Questo script entra in gioco quando avviene una collisione con un oggetto chiamato “stella”, dopo la collisione la velocità del giocatore aumenterà di 5 per 3 secondo grazie alla coroutine CollisionWithStar() impostata nella classe PlayerMovement.cs

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento** | TC-001  REQ-001 | **Nome:** | Costruzione labirinto |
| **Descrizione** | La base del gioco, La costruzione dei tre livelli | | |
| **Prerequisiti** | - Layout personaggio completato | | |
| **Procedura** | 1. Aprire il Unity 2. Disegnare i tre livelli | | |
| **Risultati attesi** | SI avrà la base del gioco completata | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento** | TC-002  REQ-002 | **Nome:** | Prima persona |
| **Descrizione** | Il movimento del personaggio è di 360° in orizzontale e 180° e l’utente sposterà la visuale con il mouse | | |
| **Prerequisiti** | - Layout personaggio completato | | |
| **Procedura** | 1. Aprire il gioco 2. Iniziale la partita 3. Muovere la visuale | | |
| **Risultati attesi** | Facendo partire il gioco, l’utente è in grado di cambiare la visuale del personaggio utilizzando il mouse | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento** | TC-003  REQ-003 | **Nome:** | Movimento personaggio |
| **Descrizione** | Durante la partita, l’utente può muovere il personaggio tramite “WASD” all’interno dei labirinti | | |
| **Prerequisiti** | - Labirinti completati  - Personaggio completato  - Visuale personaggio completata | | |
| **Procedura** | 1. Aprire gioco 2. Iniziare la partita 3. Muovere il personaggio | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento** | TC-004  REQ-004 | **Nome:** | Programmazione mostri |
| **Descrizione** | L’utente dovrà scappare dai guardiani dei biomi che avranno come obiettivo quello di far perdere tutte le vite all’utente | | |
| **Prerequisiti** | - Raccolta template mostri  - Labirinti completati  - Personaggio completato | | |
| **Procedura** | 1. Aprire gioco 2. Iniziare la partita 3. Aspettare l’arrivo dei mostri | | |
| **Risultati attesi** | I mostri dovranno inseguire l’utente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento** | TC-005  REQ-0055 | **Nome:** | Vite personaggio |
| **Descrizione** | Quando l’utente viene toccato da un mostro perde una vita, quando perde tutte le vite il gioco finisce. | | |
| **Prerequisiti** | - Personaggio completato  - Mostri completati | | |
| **Procedura** | 1. Aprire il gioco 2. Iniziare la partita 3. Farsi colpito dai mostri | | |
| **Risultati attesi** | I mostri dovranno inseguire l’utente e se lo toccano, esso perde una vita | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento** | TC-006  REQ-006 | **Nome:** | Funzionamento schermata iniziale |
| **Descrizione** | L’utente ha la possibilità di inserire il proprio nome e visualizzare la classifica dei tempi migliori. | | |
| **Prerequisiti** | - Design schermata iniziale completata  - Gioco completato | | |
| **Procedura** | 1. Aprire il gioco 2. Inserire un nome 3. Schiacciare il bottone classifica 4. Schiacciare il bottone gioca | | |
| **Risultati attesi** | L’utente potrà visualizzare la classifica cliccando sul pulsante classifica, inoltre potrà inserire il suo nome ed impostare il volume, infine l’utente potrà iniziare una partita cliccando sul tasto “Gioca”. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento** | TC-007  REQ-007 | **Nome:** | Funzionamento schermata impostazoni |
| **Descrizione** | L’utente ha la possibilità di aprire le impostazioni durante il corso della partita | | |
| **Prerequisiti** | - Design schermata impostazioni completata  - Gioco completato | | |
| **Procedura** | 1. Aprire il gioco 2. Aprire la schermata impostazione 3. Verificare i vari pulsanti | | |
| **Risultati attesi** | L’utente potrà visualizzare nella schermata impostazioni tramite che il tasto “ESC”, potrà riprendere la partita cliccando sul bottone riprendi, uscire dal gioco schiacciando sul bottone “Torna al menù” ed impostare il volume | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento** | TC-008  REQ-008 | **Nome:** | DataBase |
| **Descrizione** | Il DataBase contiene tutti i nomi dei giocatori e i loro relativi tempi che saranno inseriti nella classifica | | |
| **Prerequisiti** | - Schermata delle impostazioni e schermata iniziale finite  - Gioco completato | | |
| **Procedura** | 1. Aprire il gioco 2. Schiacciare sul pulsante “Classifica” 3. Vedere i posizionamenti | | |
| **Risultati attesi** | L’utente potrà visualizzare la classifica nella schermata iniziale con il tasto “Classifica” e vedere così il suo tempo effettivo. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento** | TC-09  REQ-09 | **Nome:** | Funzionamento schermata vincita |
| **Descrizione** | L’utente, quando finirà la partita, potrà vedere subito il suo tempo per completare il gioco. | | |
| **Prerequisiti** | - Database finito  - Gioco completato | | |
| **Procedura** | 1. Aprire il gioco 2. Finire il gioco 3. Controllare il tempo effettivo e la posizione | | |
| **Risultati attesi** | L’utente potrà visualizzare alla fine della partita il suo tempo e la sua posizione in classifica. | | |

## Risultati test

|  |  |  |  |
| --- | --- | --- | --- |
| Test  Case | Esito | Risultati | Data |
| TC-001 | Passato | Prima visualizzazione: parte grezza dei livelli ancora da decorare | 14.10.2022 |
| TC-002 | Passato | Il giocatore sarà in grado di non visualizzare se stesso e vedere intorno a se in prima persona | 18.11.2022 |
| TC-003 | Passato | Il giocatore sarà in grado di spostarsi nella mappa tramite i pulsanti WASD e le frecce della tastiera | 07.10.2022 |
| TC-004 | Passato | Il giocatore vedrà i mostri che lo rincorrono in giro per la mappa | 14.10.2022 |
| TC-005 | Passato | Il giocatore perderà vite se un mostro dovesse colpirlo | 21.10.2022 |
| TC-006 | Passato | Il giocatore sarà in grado di inserire un proprio nome, far partire il gioco, regolare il volume ed uscire dal gioco | 21.10.2022 |
| TC-007 | Passato | Il giocatore, premendo il tasto “esc” sarà in grado di riprendere la partita, tornare al menù e regolare il volume | 21.12.2022 |
| TC-008 | Non Passato | Purtroppo per la mancanza del mio collega durante più della metà del progetto non ho avuto abbastanza tempo per utilizzare il DataBase per la memorizzazione del tempo dei giocatori nel completamento del gioco | 23.12.2022 |
| TC-009 | Parzialmente  Passato | La schermata viene visualizzata e il giocatore può interagire correttamente con i pulsanti ma non può vedere ne il suo tempo ne la posizione in classifica | 23.12.2022 |

## Mancanze/limitazioni conosciute

La mancanza principale del nostro progetto è stata la mancanza del mio collega di progetto per malattia. Purtroppo, sono rimasto da solo a fine ottobre dovendo svolgere il resto del progetto senza il supporto del mio collega, i suoi lavori li ho dovuti svolgere io stesso non riuscendo a finire per poco l’intero progetto come stabilito nel QdC (mancanza DataBase).  
Con la presenza del mio collega sicuramente saremmo riusciti a terminare tutto ciò che ci eravamo prefissati a settembre.

# Consuntivo

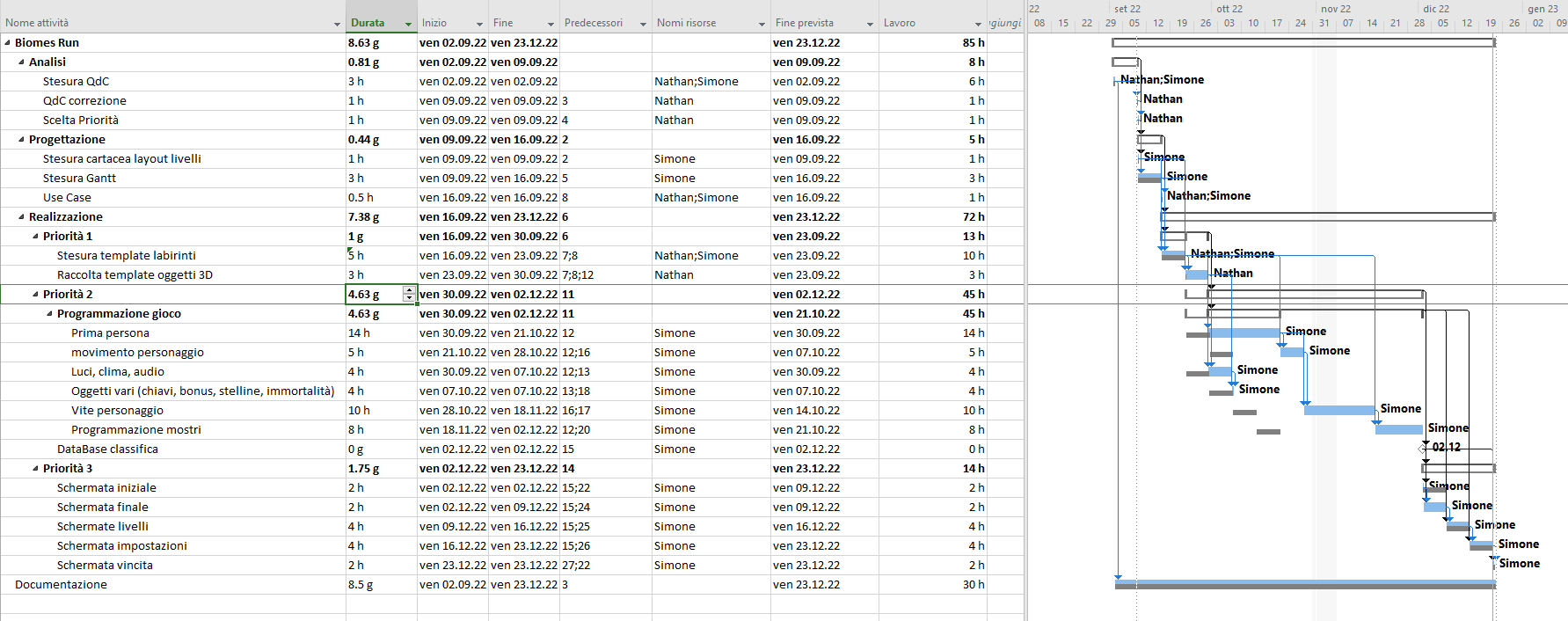


Figura - Gantt Consuntivo

# Conclusioni

## Sviluppi futuri

Questo progetto può essere migliorato in moltissimi modi aggiungendo tantissime funzionalità, come per esempio dei nuovi livelli più complessi, oppure con delle versioni differenti del gioco (a tempo, senza tempo, con nemici, senza nemici, dare possibilità al personaggio di difendersi dai nemici tramite superpoteri, nemici più cattivi e molte altre.

## Considerazioni personali

In conclusione, posso dire che questo progetto mi ha insegnato a programmare su Unity tramite C#, insegnandomi tutte le funzionalità di base unite alle funzionalità complesse. Inoltre, questo progetto mi ha insegnato ad essere molto ordinato e organizzato creando una vera e propria gerarchia di attività da svolgere durante le giornate lavorative. Inoltre, grazie a questo progetto ho potuto aumentare la mia capacità di lavorare in gruppo (fino a quando ho potuto)

# Glossario

|  |  |
| --- | --- |
| **Nome** | **Definizione** |
| **Array** | Insieme di elementi dello stesso tipo, identificati da un nome |
| **Assets** | Cartella contenente le parti principali del progetto (scene, modelli, oggetti 3D,…) |
| **Consuntivo** | risultati di un dato periodo di attività |
| **Database** | archivio di dati strutturato in tabelle contenenti attributi per la gestione e l'aggiornamento delle informazioni e la gestione di ricerche complesse |
| **Diagramma** | Disegno o rappresentazione schematica |
| **Empty Object** | Oggetto vuoto che andrà a contenere dei sotto oggetti in Unity (usato per ordinare le gerarchie di oggetti) |
| **Gantt** | Diagramma per pianificare un intero progetto e dividerlo in varie attività rendendolo più organizzato |
| **Hardware** | Parte fisica di un sistema |
| **Rigidbody** | Proprietà che rende il “corpo” dell’oggetto importato rigido in maniera da non passarci attraverso |
| **Script** | File che contiene righe di codice |
| **Sprite** | Immagine bidimensionale di gioco |
| **Use case** | Caso d’uso, contiene tutte le funzionalità di un progetto |

# Bibliografia

## Sitografia

* [https://kenney.nl/assets/nature-pack 16.09.2022](https://kenney.nl/assets/nature-pack%2016.09.2022)
* <https://www.piskelapp.com/> 14.10.2022
* <https://medium.com/codex/creating-a-lives-counter-ui-element-in-unity-9d48b485555e>11.11.2022
* <https://www.youtube.com/watch?v=f473C43s8nE> 16.12.2022
* <https://www.youtube.com/watch?v=yWCHaTwVblk> 21.12.2022
* <https://sketchfab.com/store/3d-models> 09.12.2022
* <https://assetstore.unity.com/> 09.12.2022