

Biomes Run

Volume ovunque

1	Introduzione.....	4
1.1	Informazioni sul progetto.....	4
1.2	Abstract.....	4
1.3	Scopo.....	4
2	Analisi.....	5
2.1	Analisi del dominio.....	5
2.2	Analisi e specifica dei requisiti.....	5
2.3	Use case.....	9
2.4	Pianificazione.....	10
2.5	Analisi dei mezzi.....	11
2.5.1	Software.....	11
2.5.2	Hardware.....	11
3	Progettazione.....	12
3.1	Design dell'architettura del sistema.....	12
3.2	Design dei dati e database.....	12
3.3	Design delle interfacce.....	13
3.4	Design procedurale.....	20
4	Implementazione.....	20
4.1	Prima persona.....	20
5	Test.....	23
5.1	Protocollo di test.....	23
5.2	Risultati test.....	27
5.3	Mancanze/limitazioni conosciute.....	27
6	Consuntivo.....	27
7	Conclusioni.....	27
7.1	Sviluppi futuri.....	27
7.2	Considerazioni personali.....	27
8	Glossario.....	27
9	Bibliografia.....	27
9.1	Bibliografia per articoli di riviste.....	27
9.2	Bibliografia per libri.....	27
9.3	Sitografia.....	27
10	Allegati.....	27

Sommario

1	Introduzione.....	4
1.1	Informazioni sul progetto.....	4
1.2	Abstract.....	4
1.3	Scopo.....	4
2	Analisi.....	5
2.1	Analisi del dominio.....	5
2.2	Analisi e specifica dei requisiti.....	5
2.3	Use case.....	9
2.4	Pianificazione.....	10
2.5	Analisi dei mezzi.....	11
2.5.1	Software.....	11
2.5.2	Hardware.....	11
3	Progettazione.....	12
3.1	Design dell'architettura del sistema.....	12
3.2	Design dei dati e database.....	12
3.3	Design delle interfacce.....	13
3.4	Design procedurale.....	20
4	Implementazione.....	20
5	Test.....	23
5.1	Protocollo di test.....	23
5.2	Risultati test.....	27

5.3	Mancanze/limitazioni conosciute	27
6	Consuntivo.....	27
7	Conclusioni	27
7.1	Sviluppi futuri	27
7.2	Considerazioni personali	27
8	Glossario	27
9	Bibliografia.....	27
9.1	Bibliografia per articoli di riviste:	27
9.2	Bibliografia per libri	27
9.3	Sitografia	27
10	Allegati	27

Indice delle figure

Figura 1 - Use Case.....	9
Figura 2 - Gantt Iniziale.....	10
Figura 3 - Tabella classifica.....	12
Figura 4 - Schermata iniziale.....	13
Figura 5 - Schermata di pausa	14
Figura 6 - Schermata di vittoria	15
Figura 7 - Livello uno	16
Figura 8 - Livello due	16
Figura 9 - Livello tre	17
Figura 10 - Schermata perdita chiave falsa	18
Figura 11 - Schermata perdita vite	19
Figura 12 - Pavimento Grezzo.....	20
Figura 13 - Personaggio Grezzo	20
Figura 14 - Script Prima Persona	21
Figura 15 - Script Movimento	22

1 Introduzione

1.1 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.

1.2 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.

1.3 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.

2 Analisi

2.1 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.

2.2 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 sì 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

2.3 Use case

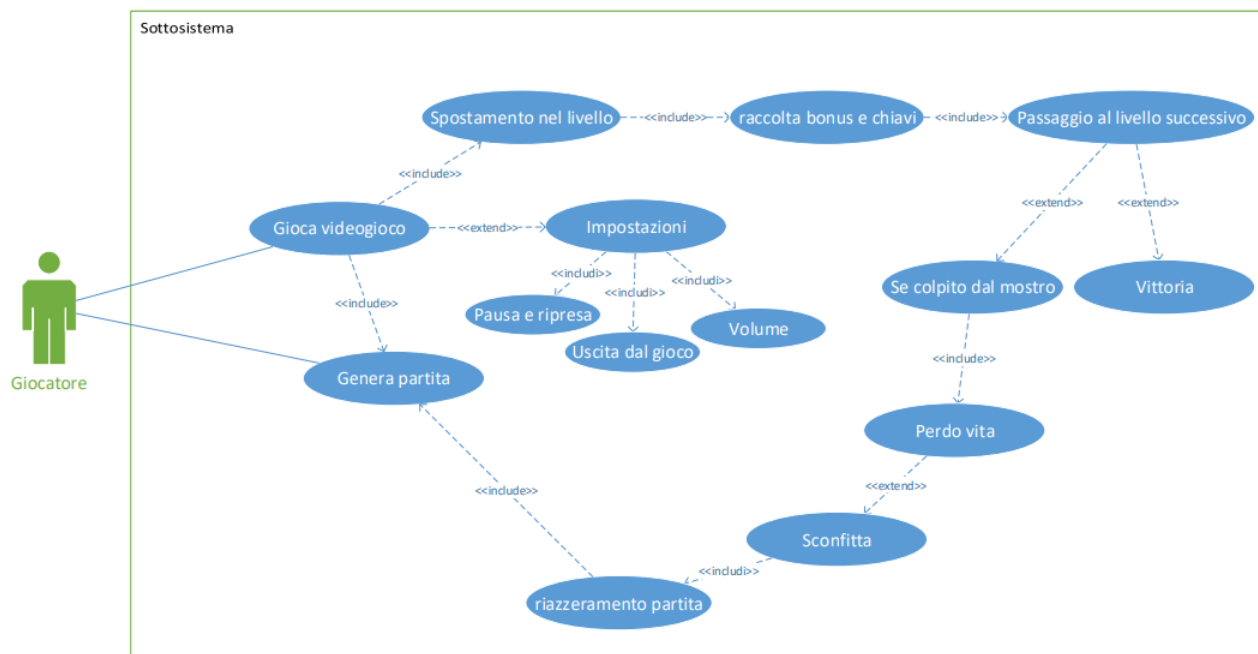


Figura 1 - Use Case

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 tutti gli oggetti e le chiavi.
- Giocando al videogioco esso genera una nuova partita.
- Esso può utilizzare la schermata di impostazioni contenenti:
 - Regolazione volume.
 - Pausa e ripresa durante il gioco.
 - Tasto per uscire dal gioco

2.4 Pianificazione

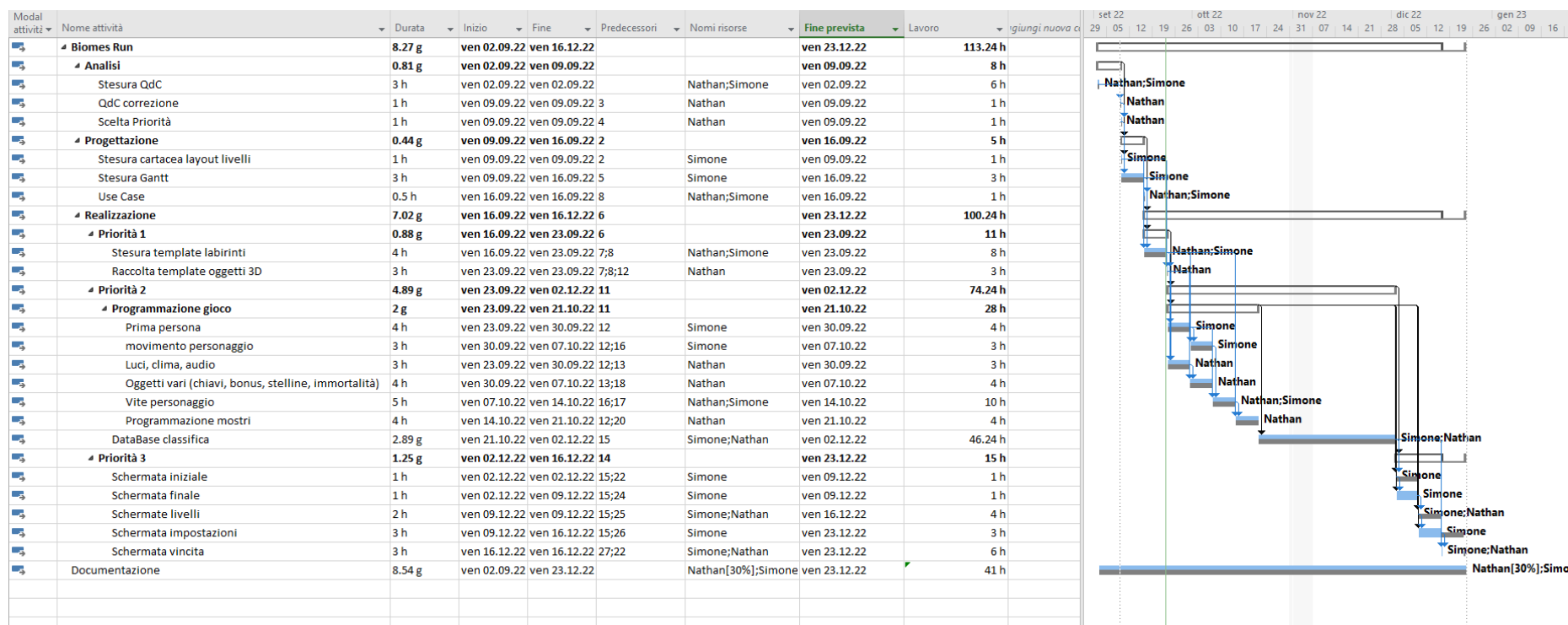


Figura 2 - Gantt Iniziale

La durata del nostro progetto, come si può 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 dei vari livelli.
3. **Realizzazione:** Questa parte riguarda la vera e propria realizzazione del progetto, l'ho suddiviso in tre categorie:
 - a. **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.
 - b. **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.
 - i. **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.
 - c. **Priorità 3:** In questa categoria sono presenti tutte le schermate relative al gioco che sono:
 - i. **Schermata benvenuto:** Questa schermata è la schermata iniziale che dà il benvenuto nel gioco all'utente
 - ii. **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.
 - iii. **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.
 - iv. **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.

2.5 Analisi dei mezzi

2.5.1 Software

- Visual studio 2022
- Unity 2022.1

2.5.2 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

3 Progettazione

3.1 Design dell'architettura del sistema

SQL lite in Unity

3.2 Design dei dati e database

LeaderBoard	
ID	int
nickname	varchar(45)
time	int
place	int

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)

Figura 3 - Tabella classifica

3.3 Design delle interfacce

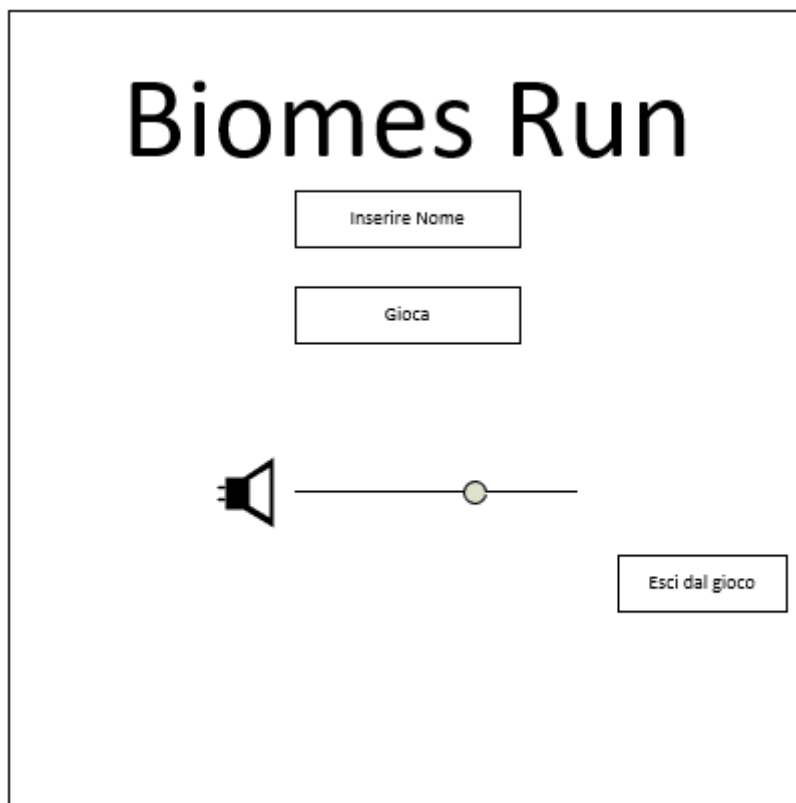


Figura 4 - Schermata iniziale

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
- Uno slider per il volume della musica del gioco
- Un bottone "Esci dal gioco" che permetterà all'utente di uscire dal gioco.

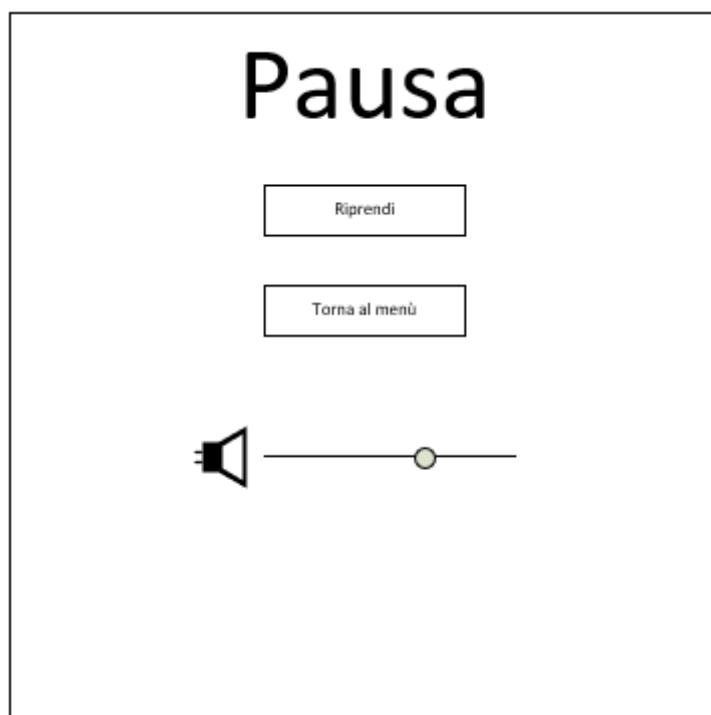


Figura 5 - Schermata di pausa

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 per il volume della musica del gioco



Figura 6 - 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

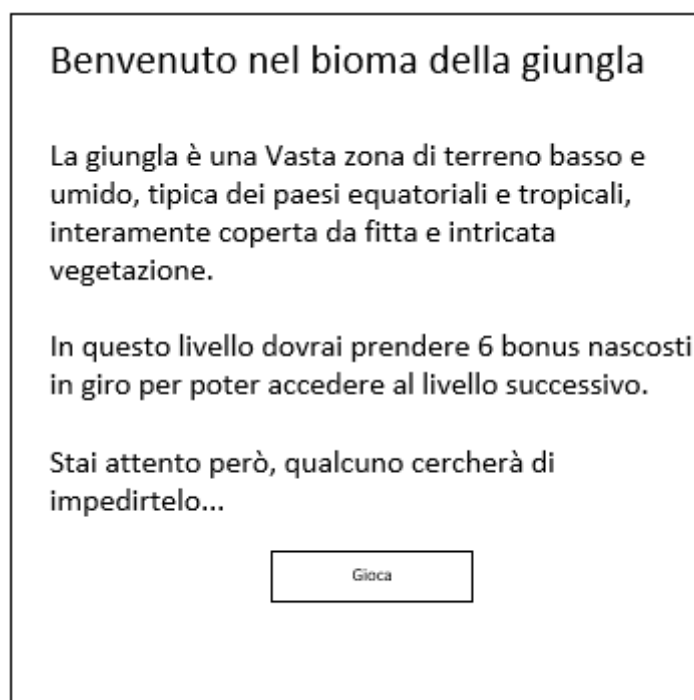


Figura 7 - Livello uno

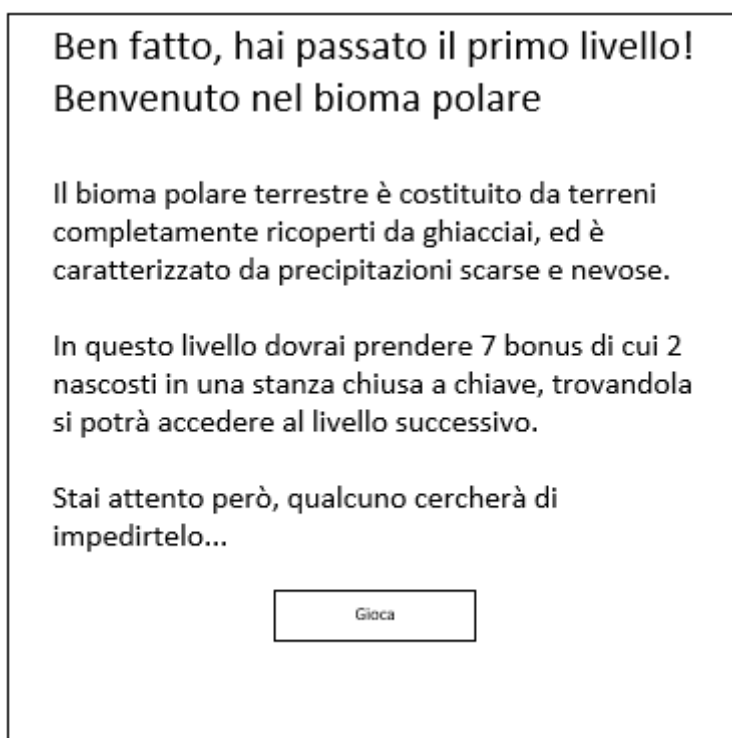


Figura 8 - Livello due

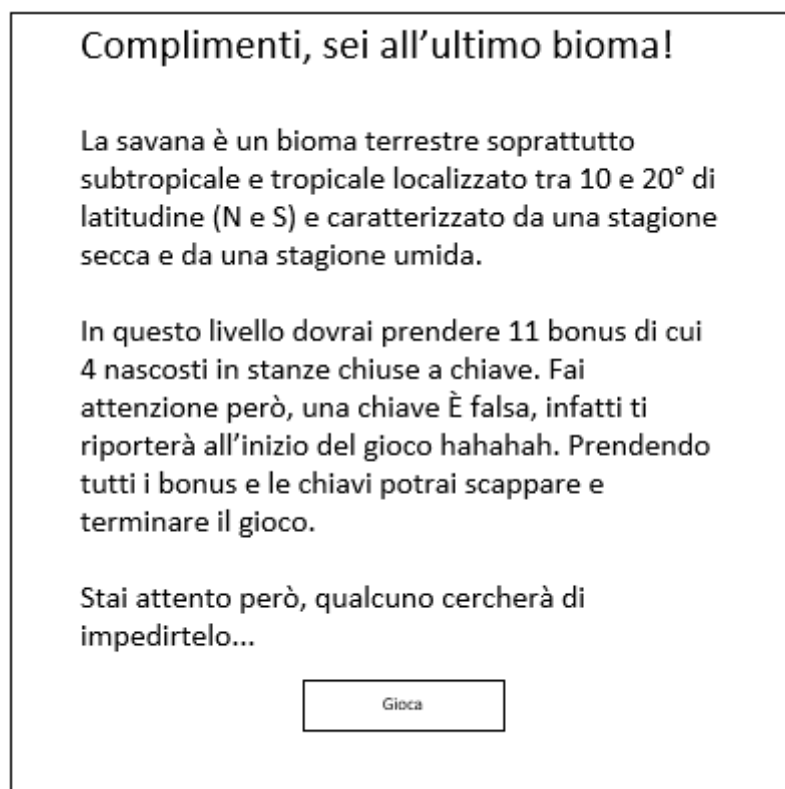


Figura 9 - Livello tre

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 10 - 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 11 - 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

3.4 Design procedurale

4 Implementazione

4.1 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 sì che non si sposti durante la partita

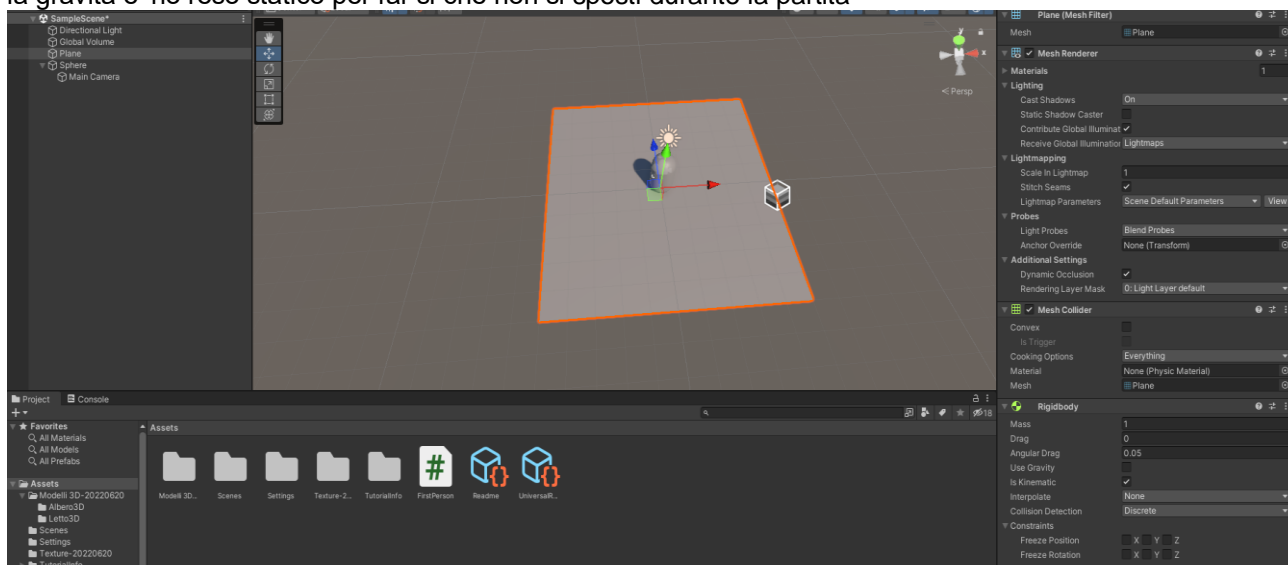


Figura 12 - Pavimento Grezzo

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.

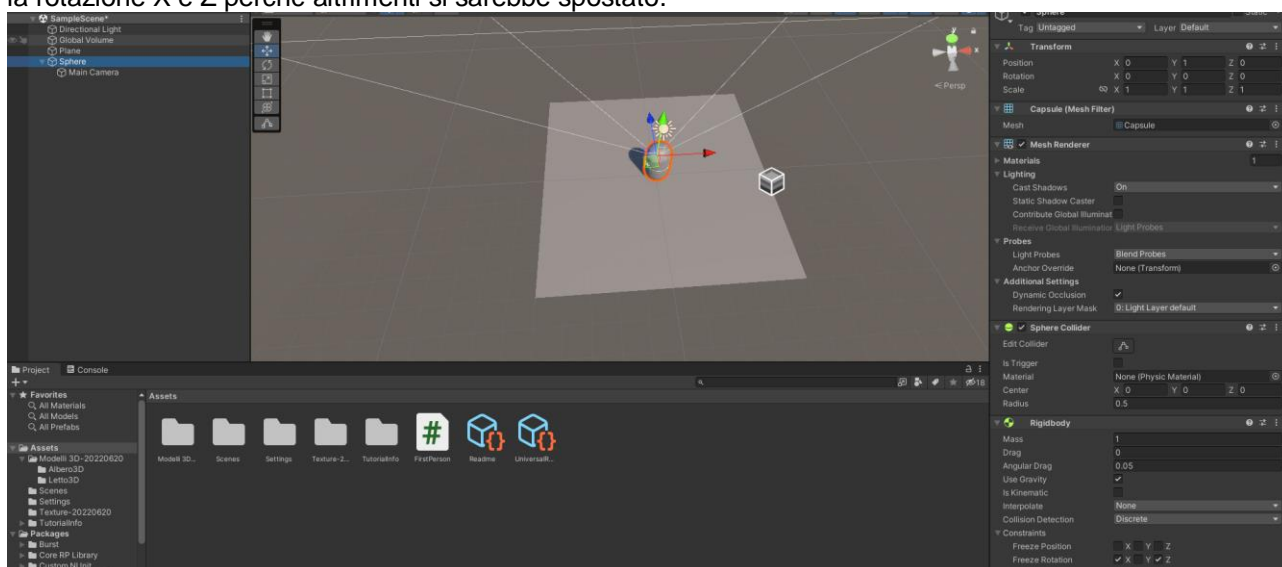
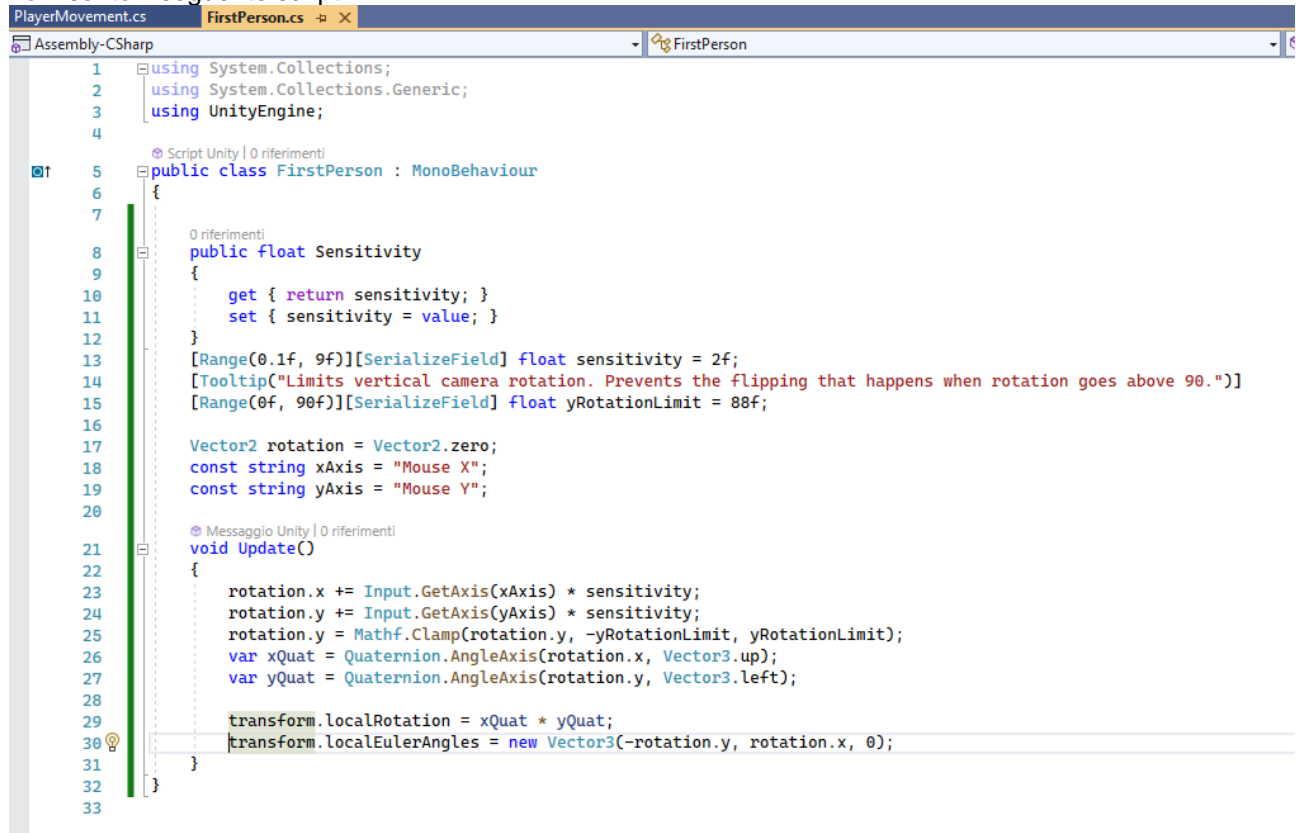


Figura 13 - Personaggio Grezzo

Inoltre per il movimento effettivo in prima persona ho implementato nella gerarchia della sfera la camera ed ho inserito il seguente script:



```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class FirstPerson : MonoBehaviour
6  {
7
8      public float Sensitivity
9      {
10         get { return sensitivity; }
11         set { sensitivity = value; }
12     }
13     [Range(0.1f, 9f)][SerializeField] float sensitivity = 2f;
14     [Tooltip("Limits vertical camera rotation. Prevents the flipping that happens when rotation goes above 90.")]
15     [Range(0f, 90f)][SerializeField] float yRotationLimit = 88f;
16
17     Vector2 rotation = Vector2.zero;
18     const string xAxis = "Mouse X";
19     const string yAxis = "Mouse Y";
20
21     void Update()
22     {
23         rotation.x += Input.GetAxis(xAxis) * sensitivity;
24         rotation.y += Input.GetAxis(yAxis) * sensitivity;
25         rotation.y = Mathf.Clamp(rotation.y, -yRotationLimit, yRotationLimit);
26         var xQuat = Quaternion.AngleAxis(rotation.x, Vector3.up);
27         var yQuat = Quaternion.AngleAxis(rotation.y, Vector3.left);
28
29         transform.localRotation = xQuat * yQuat;
30         transform.localEulerAngles = new Vector3(-rotation.y, rotation.x, 0);
31     }
32 }
33

```

Figura 14 - Script Prima Persona

Per lo spostamento del personaggio ho implementato nel personaggio grezzo questo script:

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class FirstPerson : MonoBehaviour
6  {
7      // Start is called before the first frame update
8      [SerializeField]
9      public float sensitivity = 5.0f;
10     [SerializeField]
11     public float smoothing = 2.0f;
12     // the character is the capsule
13     public GameObject character;
14     // get the incremental value of mouse moving
15     private Vector2 mouseLook;
16     // smooth the mouse moving
17     private Vector2 smoothV;
18
19     // Use this for initialization
20     void Start()
21     {
22         character = this.transform.parent.gameObject;
23     }
24
25     // Update is called once per frame
26     void Update()
27     {
28         // md is mouse delta
29         var md = new Vector2(Input.GetAxisRaw("Mouse X"), Input.GetAxisRaw("Mouse Y"));
30         md = Vector2.Scale(md, new Vector2(sensitivity * smoothing, sensitivity * smoothing));
31         // the interpolated float result between the two float values
32         smoothV.x = Mathf.Lerp(smoothV.x, md.x, 1f / smoothing);
33         smoothV.y = Mathf.Lerp(smoothV.y, md.y, 1f / smoothing);
34         // incrementally add to the camera look
35         mouseLook += smoothV;
36
37         // vector3.right means the x-axis
38         transform.localRotation = Quaternion.AngleAxis(-mouseLook.y, Vector3.right);
39         character.transform.localRotation = Quaternion.AngleAxis(mouseLook.x, character.transform.up);
40     }
41 }

```

Figura 15 - Script Movimento

Il risultato finale è il nostro personaggio che può muoversi in tutte le direzioni cambiando visuale, inoltre non può alzare e abbassare la visuale oltre i 90° per evitare una rotazione inutile.

-----gif movimento

Per la programmazione ho svolto questi passaggi:

Aggiungere

-immagini vite

-immagini impostazioni vite

-gerarchie

-script (vite, mostro)

5 Test

5.1 Protocollo di test

Test Case:	TC-002	Nome:	Prima persona
Riferimento	REQ-002		
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	<ol style="list-style-type: none"> 1. Aprire il gioco 2. Iniziare 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:	TC-003	Nome:	Movimento personaggio
Riferimento	REQ-003		
Descrizione	Durante la partita, l'utente può muovere il personaggio tramite "WASD" all'interno dei labirinti		
Prerequisiti	<ul style="list-style-type: none"> - Labirinti completati - Personaggio completato - Visuale personaggio completata 		
Procedura	<ol style="list-style-type: none"> 1. Aprire gioco 2. Iniziare la partita 3. Muovere il personaggio 		

Test Case:	TC-004	Nome:	Decorazione labirinto
Riferimento	REQ-004		
Descrizione	Raccolta di oggetti per la decorazione dei labirinti (bonus, chiavi, mostri, stelline, personaggio), inserimento luci e musiche		
Prerequisiti	- Livelli completati		
Procedura	<ol style="list-style-type: none"> 1. Aprire il gioco 2. Iniziare la partita 3. Muoversi nel labirinto 		

Risultati attesi

Alla fine si avranno i livelli completati definitivamente, con tutti gli interni completi.

Test Case:	TC-005	Nome:	Programmazione mostri
Riferimento	REQ-005		
Descrizione	L'utente dovrà scappare dai guardiani dei biomi che avranno come obiettivo quello di far perdere tutte le vite all'utente		
Prerequisiti	<ul style="list-style-type: none"> - Raccolta template mostri - Labirinti completati - Personaggio completato 		
Procedura	<ol style="list-style-type: none"> 1. Aprire gioco 2. Iniziare la partita 3. Aspettare l'arrivo dei mostri 		
Risultati attesi	I mostri dovranno inseguire l'utente		

Test Case:	TC-006	Nome:	Vite personaggio
Riferimento	REQ-006		
Descrizione	Quando l'utente viene toccato da un mostro perde una vita, quando perde tutte le vite il gioco finisce.		
Prerequisiti	<ul style="list-style-type: none"> - Personaggio completato - Mostri completati 		
Procedura	<ol style="list-style-type: none"> 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:	TC-007	Nome:	Funzionamento schermata iniziale
Riferimento	REQ-007		
Descrizione	L'utente ha la possibilità di inserire il proprio nome e visualizzare la classifica dei tempi migliori.		
Prerequisiti	<ul style="list-style-type: none"> - Design schermata iniziale completata - Gioco completato 		
Procedura	<ol style="list-style-type: none"> 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:	TC-008	Nome:	Funzionamento
Riferimento	REQ-008		schermata impostazioni
Descrizione	L'utente ha la possibilità di aprire le impostazioni durante il corso della partita		
Prerequisiti	<ul style="list-style-type: none"> - Design schermata impostazioni completata - Gioco completato 		
Procedura	<ol style="list-style-type: none"> 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:	TC-009	Nome:	DataBase
Riferimento	REQ-009		
Descrizione	Il DataBase contiene tutti i nomi dei giocatori e i loro relativi tempi che saranno inseriti nella classifica		
Prerequisiti	<ul style="list-style-type: none"> - Schermata delle impostazioni e schermata iniziale finite - Gioco completato 		
Procedura	<ol style="list-style-type: none"> 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:	TC-010	Nome:	Funzionamento
Riferimento	REQ-010		schermata vincita
Descrizione	L'utente, quando finirà la partita, potrà vedere subito il suo tempo per completare il gioco.		
Prerequisiti	<ul style="list-style-type: none"> - Database finito - Gioco completato 		
Procedura	<ol style="list-style-type: none"> 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.		

5.2 Risultati test

5.3 Mancanze/limitazioni conosciute

6 Consuntivo

7 Conclusioni

7.1 Sviluppi futuri

7.2 Considerazioni personali

8 Glossario

9 Bibliografia

9.1 Bibliografia per articoli di riviste:

9.2 Bibliografia per libri

9.3 Sitografia

10 Allegati
