Sommario

[Introduzione 2](#_Toc171534369)

[Ambientazione e terreno 2](#_Toc171534370)

[Interazioni 2](#_Toc171534371)

[User Interface 2](#_Toc171534372)

[Narrazione 2](#_Toc171534373)

[Aree di gioco 3](#_Toc171534374)

[Menu Iniziale (scena introduttiva) 3](#_Toc171534375)

[Punto di partenza (scena di gioco) 4](#_Toc171534376)

[Staccionata 4](#_Toc171534377)

[Automobile 4](#_Toc171534378)

[Fuse Box 5](#_Toc171534379)

[Percorso 5](#_Toc171534380)

[Roccia con ombra del mostro 5](#_Toc171534381)

[Attacco del mostro 5](#_Toc171534382)

# Introduzione

La seguente relazione descrive il mio lavoro su questa applicazione VR. Nelle sezioni iniziali vengono mostrati gli aspetti introduttivi e più generali, nelle sezioni successive invece vengono descritte tutte le sezioni di gioco in sequenza. Ciascuna sezione di gioco ha un’introduzione che è sufficiente per avere un’idea generale del contenuto e di cosa bisogna fare per proseguire. A seguire ciascuna introduzione c’è una descrizione più dettagliata dei contenuti, dove mostro ad alto livello il mio lavoro evitando di entrare sempre negli aspetti tecnici e descrivendoli solo dove li ritengo necessario. Questi paragrafi includono anche i riferimenti agli script e agli asset creati, modificati o importati.

# Ambientazione e terreno

L’applicazione VR è un gioco a tema horror, l’ambientazione è notturna, con poca luce. Le interazioni cercano quindi di mantenere a grandi linee questo tema.

Il terreno è stato creato tramite il tool interno di Unity e utilizzando i packages “[Conifers [BOTD]](https://assetstore.unity.com/packages/3d/vegetation/trees/conifers-botd-142076)” per i modelli degli alberi e dell’erba, e “[Grass And Flowers Pack 1](https://assetstore.unity.com/packages/2d/textures-materials/nature/grass-and-flowers-pack-1-17100)” per ulteriori modelli dell’erba.

Ho quindi modellato l’altezza del terreno per rispecchiare la mia idea di gioco, utilizzando questi asset per creare una strada sterrata (utilizzando diversi materials di terreno, anche modificati, per renderlo meno piatto e banale) circondata da una fitta foresta.

Per rendere l’ambiente notturno e cupo, ho aggiunto uno skybox appropriato (riducendo la quantità di luce proveniente da esso), della nebbia e dei suoni in loop per simulare un clima di vento, una foresta abbastanza silenziosa, senza vita.

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

# Interazioni

Nelle seguenti interazioni faccio riferimento ai comandi per mouse e tastiera, dato che questi potrebbero essere mappati in modi diversi in base al visore che viene utilizzato.

Il movimento nel gioco è fatto tramite i pulsanti “WASD”, quindi in modo semplice e continuo. È inclusa un’area di gioco in una delle ultime sezioni dove è necessario utilizzare il metodo di teleport con i raycast (DA VERIFICARE POI QUANDO LO FACCIO).

È stato aggiunto un sistema di collisioni per l’utente che utilizza il componente “Character Controller” per limitare i movimenti dell’utente se necessario, ma soprattutto per supportare una verticalità che avrei voluto dare al gioco. Di conseguenza, se il terreno presenta una piccola salita o discesa, il controller gestirà automaticamente la posizione dell’utente, evitando che questo passi attraverso agli oggetti di gioco.

Per quanto riguarda le interazioni vengono utilizzate quelle di tipo “Direct grab” (tramite pulsante G di qualsiasi mano) per la maggior parte, per esempio per aprire uno sportello, muovere una leva, lanciare qualcosa. Vengono utilizzate anche quelle tramite ray interactor per interagire con le varie UI (quella nella scena iniziale e quella dell’User Interface) e per il movimento dei quadri all’interno della casa (DA VERIFICARE POI QUANDO LO FACCIO).

Per le mani ho utilizzato il modello di Ocolous che ho trovato su internet, che comprende anche alcune animazioni (per quando viene premuto il pulsante “grab” oppure “pinch”), per rendere ancora più esplicita l’interazione. Ho poi implementato uno script “HandOverColor” che modifica il materiale delle mani a runtime, cambiandolo con un colore leggermente diverso quando la proprietà “Hover” è attiva, questo per rendere esplicita e semplice l’interazione con i grab interactors.

I ray interactor hanno una lunghezza dinamica, vengono allungati se necessario quando puntano ad un oggetto con cui si può interagire. È anche stato impostato un cambio di colore verso il blu quando si sta mirando ad un oggetto interactable.

Vengono utilizzate le “Interaction Layer Masks” per filtrare quali oggetti è possibile interagire con il metodo Direct e quali con il metodo Raycast.

Ovviamente le interazioni sono state create usando il package di Unity “XR Interactions Toolkit”, importandolo e settandolo da zero.

## User Interface

È possibile mostrare o nascondere l’user interface in qualsiasi momento, tramite il pulsante B (Primary Button) della mano sinistra. L’UI include un titolo e una descrizione che mostrano informazioni sulla sezione di gioco raggiunta, con dei suggerimenti su come proseguire. È presente anche un pulsante X che permette di chiuderla tramite ray interactor volendo, ma è possibile chiuderla semplicemente premendo B di nuovo.

Le schede mostrate al suo interno cambiano quindi dinamicamente, attivate da funzioni e da trigger zones, di modo da mostrare sempre informazioni sulla sezione di gioco attuale.

Ho quindi scritto lo script “CanvasVisibility” per rendere l’UI non visibile ad inizio del gioco (impostarla come invisibile manualmente dalla gerarchia dava problemi), lo script “InputButtonPressed” per mostrare l’UI quando viene premuto il pulsante, e lo script “ChangeCanvasInstruction” per passare automaticamente alla prossima scheda, grazie ad un numero incrementato internamente allo script.

# Narrazione

Oltre alla user interface, ho voluto implementare un sistema di narrazione, con l’obiettivo di rendere più chiari gli obiettivi di gioco all’utente e aumentare l’immersione di gioco (evitando quindi che l’utente debba aprire l’UI di continuo per capire come proseguire).

Durante il gioco verranno riprodotti automaticamente degli audio di narrazione che ho creato registrando la mia voce e utilizzando il tool AI di [ElevenLabs](https://elevenlabs.io/). Ho preferito questa opzione rispetto a quella text to speech perché registrando la mia voce, riuscivo a dare più emozione all’AI che quindi sembrava meno robotica. Il risultato è ovviamente non perfetto, sarebbe molto meglio avere un attore capace, ma è comunque un tipo di implementazione che ho voluto provare ad utilizzare.

Questi audio sono riprodotti automaticamente all’ingresso di trigger zones, oppure in specifiche funzioni che vengono eseguite al completamento o durante eventi di gioco specifici.

Ho scritto anche la funzione “PlayTwoAudiosInARow” nel caso dovessi assegnare ad una singola trigger zone la riproduzione di due audio di fila.

Le trigger zones sono generalizzate tramite lo script “Trigger” e il prefab “Trigger Zone” che ho creato.

Tutti gli audio di narrazione sono contenuti nella sezione “Narration Audios” della gerarchia.

# Aree di gioco

Nelle seguenti sezioni saranno descritte tutte le aree di gioco in sequenza, dalla prima incontrata all’ultima. Per avere un’idea generale dei contenuti è sufficiente leggere la parte introduttiva di ciascuna sezione, dopo la quale saranno riportati in maggiore dettaglio i contenuti della sezione, descrivendo il lavoro fatto senza entrare troppo in dettagli tecnici a meno che sia ritenuto necessario. In ogni caso saranno presenti tutti i riferimenti ai GameObject e agli Script creati, modificati e utilizzati all’interno del progetto.

## Menu Iniziale (scena introduttiva)

Immagine che contiene testo, schermata, aria aperta

Descrizione generata automaticamente

La prima scena contiene una copia dello stesso terreno, con alcune piccole modifiche per creare quest’area ristretta dove l’utente può muoversi.

All’interno si può trovare una semplice UI che contiene alcune istruzioni sulle interazioni e come simularle tramite mouse e tastiera (non ho aggiunto riferimenti per i visori dato che suppongo siano diversi per ciascun visore), e un pulsante “Play” che permette di cambiare scena utilizzando la mia funzione “SceneChanger”, iniziando il gioco.

## Punto di partenza (scena di gioco)

Immagine che contiene aria aperta, albero, Veicolo terrestre, veicolo

Descrizione generata automaticamente

L’utente si ritrova in una strada sterrata in una foresta. Da un lato c’è una staccionata che blocca la strada, dietro la quale c’è la sua auto con le luci accese, un vetro rotto e il cofano rialzato da cui esce fumo ad indicare che l’auto ha avuto un problema. Questo, quindi, fa capire all’utente che non deve interagire con l’auto stessa, ma deve seguire la strada, dal lato della staccionata in cui si ritrova l’utente.

Quando l’utente arriva in quest’area (automaticamente, visto che è il punto di partenza del gioco), vengono riprodotti due audio di narrazione introduttivi, per aiutare l’utente a capire in che situazione si ritrova.

L’UI riporterà le seguenti informazioni:

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

### Staccionata

La staccionata utilizza il modello del package “[RPG Medieval Props](https://assetstore.unity.com/packages/3d/props/rpg-medieval-props-demo-248681)” (uno dei pochi asset del pack compatibili con la built-in render pipeline che ho usato) ed è stata creata creata manualmente alternando 2 prefab diversi di staccionata, allungandone e inclinandone uno sul pezzo di staccionata “rotto”, per renderla un po’ più interessante.

Ho poi deciso di semplificare le collisioni con l’utente andando a sovrapporre un box collider semplice, per evitare che l’utente possa andare in quel lato della mappa dove c’è l’auto.

### Automobile

L’automobile è un modello importato dal package “[Low Poly Destructible 2 Cars no. 8](https://assetstore.unity.com/packages/3d/vehicles/land/low-poly-destructible-2-cars-no-8-45368)”, utilizzando gli asset disponibili per personalizzare la macchina (si possono cambiare gomme, paraurti, etc). La cosa importante è che l’auto di questo package non è un oggetto singolo ma è l’insieme delle sue parti, il che lo rende molto personalizzabile e, come da nome del package, distruttibile. È quindi semplice spostare alcune sotto-parti per (nel mio caso) tenere una portiera aperta, avere il cofano aperto, impostare una texture di vetro rotto ad uno dei vetri.

Le due sorgenti particle system per il fumo che esce dal motore sono state create manualmente da me, di modo che appunto assomigliassero a del fumo, seguendo anche le collisioni superiori con il cofano rialzato e la direzione del vento globale. Per creare un particle system realistico ho quindi lavorato sulle sezioni emission, shape, color over lifetime, size over lifetime, external forces, collision e render.

Sono anche state aggiunte manualmente due spot lights su ciascun faro anteriore, di modo da rendere la scena ancora più realistica (dato che la luce interagisce con la staccionata e gli alberi) ed aggiungere una sorgente di luce. Ho aggiunto anche due point lights e due effetti halo sui fari, per dare appunto l’impressione che i fari siano accesi. Altrimenti ci sarebbe una sorgente di luce proveniente dai fari che però sarebbero scuri, spenti.

È anche presenta una sorgente di audio 3d che riproduce un basso rumore dell’auto in idle, per aggiungere più atmosfera.

## Fuse Box

## Percorso

Suoni horror distanti

## Roccia con ombra del mostro

## Attacco del mostro