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

(gioco in generale, ambientazione, tipo horror, orientato sulle interazioni, guidato, introduzione alla struttura della relazione, gli asset importati sono linkati sul nome del package)

# Ambientazione e terreno

(anche skybox, fog, audio)

# Interazioni

(movimento (e collisioni), grab (asset mani), ray)

## User Interface

(come aprirla, come funziona)

# Narrazione

(audio e testo)

# 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, 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

Introduzione

Qua poi aggiungere un’introduzione sul progetto, come è ambientato etc……

Per lo sviluppo del progetto mi sono concentrato sulla simulazione all’interno di Unity tramite mouse e tastiera, dato che non possiedo un visore, e di conseguenza l’uso di un visore potrebbe non funzionare oppure creare problemi che sono fuori dal mio controllo.

Character controller

Dato che l’idea iniziale del mio progetto conteneva una verticalità, ho aggiungo il componente “Character Controller” all’XR Rig, per fare in modo che i movimenti dell’utente seguano la gravità (salendo e scendendo automaticamente quando il terreno presenta dei cambi di altezza) e le collisioni degli oggetti presenti nella scena (evitando che l’utente possa passare attraverso ai muri).

Per fare questo è stato necessario aggiungere anche un “Character Controller Driver” sempre nell’XR Rig, a cui però deve essere associato un locomotion provider di movimento. Però per il movimento vengono utilizzati i comandi da tastiera (pulsanti w a s d), e non sono riuscito a trovare un modo di collegarli al driver. Di conseguenza ho aggiunto un “Continuous Mode Provider” al locomotion system di modo da collegare indirettamente i movimenti dell’utente al driver.

Questo metodo ha raggiunto il risultato previsto (in quanto a gravità e rispetto delle collisioni) ma presenta dei piccoli “scatti” che sono probabilmente dovuti alla simulazione dei comandi tramite tastiera (se provo ad andare attraverso un muro vengo fermato, venendo “teletrasportato” indietro, al posto di essere semplicemente fermato). Sono comunque riuscito ad agire sui parametri del character controller per ridurre questo fenomeno, ma non è sparito. Posso però supporre che con una corretta configurazione di un headset e con dei movement provider da usare tramite i controller del visore, questo problema sarebbe risolto.