

# 3DClouds Report

Andrea Valota

Maggio 2023

## 1 Descrizione dell'applicazione

Il gioco creato rispecchia i requisiti da voi forniti e, seguendo le vostre indicazioni, tutte le funzionalità sono state create utilizzando C++ ad eccezione della UI. Tutti i punti dei requisiti sono stati soddisfatti, compresi quelli presenti nella sezione "Extra Points".

Nella cartella "Content/ThirdPersonCPP/Maps" sono presenti i due livelli utilizzati. Il "MainMenuLevel" è utilizzato per la creazione del Main Menu spiegato in seguito, mentre il livello "ThirdPersonExampleMap" è stato modificato e contiene il livello principale. Per far sì che il menu venga visualizzato correttamente durante la simulazione nell'editor è necessario iniziare a giocare avendo aperto il livello relativo al menu. Nella cartella "Content/HUD" sono presenti le GameMode e i blueprint utilizzati per la gestione dei menu e dell'HUD. Nella cartella "Content/ReusableStructures" sono presenti varie sottocartelle che contengono dei blueprint di tutti gli oggetti sviluppati (torrette, bottoni, indicatori fluttuanti, porte). I blueprint presenti in queste cartelle sono stati creati solamente per rendere più facile la gestione e il posizionamento delle singole strutture ma come da indicazioni tutte le funzionalità alla base di ogni oggetto sono state codificate e gestite attraverso le classi in C++. Tutte le classi presenti nella cartella "C++Classes/Clouds3D\_Project" sono state sviluppate in C++ per gestire i vari oggetti e le funzionalità richieste. Tutte le classi sono state create e sviluppate da zero tranne la classe per la gestione del personaggio, che ho modificato a partire da quella creata dall'engine per il template "ThirdPerson".

Il gioco presenta un Main Menu che permette di caricare il livello, visualizzare i crediti o chiudere il gioco. Il livello presenta una zona di spawn in cui il giocatore trova 2 bottoni con cui interagire. Il primo bottone apre la porta che permette di accedere alla prima stanza e permette al giocatore di capire in modo immediato l'uso dei pulsanti presenti anche nelle stanze successive. Il secondo bottone invece permette di attivare la "God Mode" rendendo quindi il giocatore immune ai danni delle torrette. Questi sono gli unici due pulsanti a cui non è stata aggiunta la sfera fluttuante poiché sono gli unici elementi presenti nella zona di spawn e presentano scritte che rendono chiaro al giocatore le funzioni di questi pulsanti. In tutte le altre stanze gli indicatori fluttuanti sono presenti. Gli indicatori rossi indicano le torrette mentre gli indicatori verdi identificano i

pulsanti.

La prima stanza presenta 2 torrette, un pulsante che spegne le torrette e apre la porta verso la zona successiva e delle strutture che permettono al giocatore di mettersi al riparo dalle torrette.

Nella seconda stanza sono presenti 3 torrette (2 torrette standard e una che spara un proiettile fisico con traiettoria parabolica) e 2 pulsanti. Uno dei due pulsanti è bloccato dalla presenza di due torrette posizionate di fronte ad esso. In questo caso il pulsante accessibile permette di spegnere le torrette, rendendo quindi possibile accedere al secondo pulsante che permette di aprire la porta verso l'ultima stanza.

La terza stanza presenta 4 torrette (3 a proiettile fisico e una standard) e un solo pulsante. In questa stanza alcune strutture sono state posizionate in modo da permettere al giocatore di raggiungere una sezione sopraelevata dai cui può arrivare al pulsante. In questa stanza la funzione delle torrette è quella di andare a ostacolare il giocatore della sezione di platforming e di punirlo in caso di caduta.

Sono presenti dei checkpoint all'inizio di ogni stanza in modo che il giocatore non debba ripercorrere tutte le stanze già completate.

Per quanto riguarda la UI, sono presenti come HUD: un timer (che si resetta quando il livello viene ricaricato o se il giocatore muore nella prima stanza), una barra degli HP, e un contatore dei punti (il giocatore ottiene 100 punti ogni volta che un pulsante viene attivato e ne perde 10 ogni volta che viene colpito da una torretta). Inoltre, è presente un menu di pausa da cui sono possibili 3 operazioni: riprendere la partita, fare un restart del livello e chiudere il gioco. Vari parametri come la salute massima del giocatore, i danni inflitti da ogni singola torretta, il range delle torrette ecc. sono stati esposti nell'editor in modo da poterli modificare facilmente durante lo sviluppo.

## 1.1 Comandi

I pulsanti sono:

- W,A,S,D sono utilizzati per muovere il personaggio,
- la SPACEBAR per saltare,
- il mouse per muovere la visuale,
- F per interagire con gli oggetti che si trovano di fronte al personaggio,
- P per mettere il gioco in pausa.

## 2 Report tempistiche e problemi affrontati

In questa sezione riporto le tempistiche impiegate (indicative) per lo sviluppo di ogni funzionalità e i principali problemi che ho riscontrato.

Funzionalità implementata	Tempo
Porta, bottone, meccanismo di apertura della porta	4 ore
Torretta standard, sistema di spegnimento delle torrette, gestione salute, danno e morte del personaggio	6 ore
HUD con timer, barra degli HP e punteggio	4 ore
Menu di pausa	2 ore
Piazzamento oggetti, creazione delle stanze e feedback visivo per pressione dei bottoni	4 ore
Torrette con proiettile fisico	3 ore
Indicatori fluttuanti	1 ora
Sistema checkpoint	1 ora
Main Menu	4 ore
Testing e bugfix generali	2 ore

Table 1: Tempistiche

Essendo questo il mio primo progetto in Unreal Engine i principali problemi che ho affrontato sono stati relativi alla ricerca e comprensione dei meccanismi di Unreal. Buona parte del tempo impiegato nelle prime parti dello sviluppo è stato impiegato a cercare quali librerie Unreal mettesse a disposizione per la gestione dei vari sistemi, comprendere le funzionalità delle strutture da utilizzare (in particolare gli Actor). Inoltre, avendo già sviluppato alcuni progetti con Unity3D ho avuto difficoltà nell'implementazione di alcune funzionalità come i menu poiché i due engine gestiscono l'implementazione di questi sistemi in modo differente e ho quindi dovuto imparare come Unreal gestisca i vari livelli e le game mode. Ho incontrato inoltre alcuni problemi nella creazione della UI in quanto la programmazione mediante blueprint era per me completamente nuova ed ha quindi richiesto una prima parte di ricerca per comprenderne al meglio il funzionamento. In conclusione, i principali problemi riscontrati sono stati dovuti alla mia poca conoscenza di Unreal e delle sue librerie ma lo sviluppo in C++ e la gestione di tutte le strutture dati si è rivelata molto semplice dopo aver effettuato le dovute ricerche sull'engine.