



UNIVERSITÀ DEGLI STUDI ROMA TRE

Dipartimento di Ingegneria
Corso di Laurea in Ingegneria Informatica

Tesi Di Laurea

Analisi e progettazione di un

Laureando

Emanuele Pietropaolo

Matricola 513489

Relatore

Prof. Franco Milicchio

Anno Accademico 2021/2022

Ringraziamenti

Grazie a tutti

Abstract

In questa tesi verrà trattato il progetto Metarace, un crypto-game di competizioni equestri fruibile nel metaverso. Lo scopo di questo progetto è quello di creare una realtà virtuale immersiva in cui gli utenti possono diventare proprietari di cavalli digitali, iscriverli a competizioni equestri ed assistervi attraverso i propri avatar. La proprietà dei cavalli sarà garantita da un NFT, un token non fungibile basato sulla tecnologia blockchain che andrà a costituire il suo *certificato di autenticità*. Il gioco, infatti, si baserà sul paradigma *Play-to-earn*: gli oggetti digitali di gioco che l'utente otterrà in caso di vittoria saranno accompagnati dai propri NFTs. Metarace andrà ad inserirsi in un contesto più ampio di applicazioni con le quali comporrà il metaverso.

Introduzione

Il metaverso è un ambiente virtuale in cui l'utente ha la possibilità di creare e gestire identità digitali, creare asset e scambiarli con altri utenti. Esso sta dando vita ad un modo nuovo con cui le persone possono interagire e socializzare. In questa realtà, Metarace vuole essere per l'utente che ne usufruisce una possibilità di interagire con altri utenti all'interno del metaverso di Ringmaster. Metarace, infatti, è un gioco di gare equestri in 3D sviluppato con Unreal Engine 5. In questo gioco ogni utente con il proprio cavallo ha la possibilità di gareggiare contro i cavalli degli altri utenti, ognuno dei quali con i propri metadati. Ogni cavallo è associato ad un NFT (*Non-Fungible Token*), il quale costituisce il suo *certificato di proprietà*. Questo permette ai giocatori di scambiare i cavalli posseduti o anche di utilizzarli al di fuori di Metarace. La proprietà digitale è garantita dall'NFT in quanto questa nuova forma di criptovaluta si basa sulla tecnologia blockchain. Infatti, questa tecnologia permette la creazione di un database condiviso con la caratteristica principale di essere immutabile, cioè garantisce che i dati registrati su di essa non possano essere modificati in alcun modo.

Il gioco sfrutta un'architettura client-server e una metodologia di sviluppo MVC (*Model-View-Controller*). Lo svolgersi della gara è gestito da un server che decide, grazie ad un algoritmo, i tempi di ciascun cavallo partecipante sia sulla base dei propri metadati (come la velocità massima, l'accelerazione e la resistenza) sia su fattori casuali in modo che il risultato non sia prestabilito. Il client e il server comunicano tra loro con lo standard Json. Le comunicazioni passano attraverso una WebSocket e viene usato un paradigma di programmazione ad eventi.

Infine, molti degli asset presenti nel gioco sono stati creati con il software di modellazione 3D Blender.

Indice

Abstract	iii
Introduzione	iv
Indice	v
Elenco delle figure	vii
1 Background tecnologico	1
1.1 Il Metaverso	1
1.1.1 Storia del metaverso	1
1.1.2 Stato dell'arte	1
1.1.3 Prospettive future	1
1.2 Il motore grafico Unreal Engine 5	1
1.3 Il software di modellazione Blender	1
2 Fase di Analisi	2
2.1 Analisi di mercato	2
2.2 Analisi del software	2
2.2.1 Modello delle classi di dominio	2
2.2.2 Modello degli eventi	2
3 Fase di Progettazione	3
3.1 Scelte di progetto guidate dalla flessibilità	3
3.2 Architettura Client - Server	3

3.3	Programmazione ad Eventi (EDP)	3
3.4	Model - View - Controller (MVC)	3
4	Fase di Implementazione	4
4.1	Cosa ho fatto con Unreal Engine 5	4
4.2	Cosa ho fatto con Blender	4
4.3	Cosa ho fatto in C++	4
5	Risultati	5
5.1	Demo di gioco	5
5.2	Interazione utente	5
5.3	Design	5
5.4	Esempi	5
	Conclusioni e sviluppi futuri	6
	Bibliografia	7

Elenco delle figure

Capitolo 1

Background tecnologico

1.1 Il Metaverso

1.1.1 Storia del metaverso

1.1.2 Stato dell'arte

1.1.3 Prospettive future

1.2 Il motore grafico Unreal Engine 5

1.3 Il software di modellazione Blender

Capitolo 2

Fase di Analisi

Prova di testo di capitolo. Anche l'opera di [MS05] andrebbe riscritta interamente.

2.1 Analisi di mercato

2.2 Analisi del software

2.2.1 Modello delle classi di dominio

2.2.2 Modello degli eventi

Ancora del testo. Come si afferma in [JS96] molto lavoro deve ancora essere fatto.

Capitolo 3

Fase di Progettazione

3.1 Scelte di progetto guidate dalla flessibilità

3.2 Architettura Client - Server

Come visto in 4.1

3.3 Programmazione ad Eventi (EDP)

3.4 Model - View - Controller (MVC)

Capitolo 4

Fase di Implementazione

4.1 Cosa ho fatto con Unreal Engine 5

4.2 Cosa ho fatto con Blender

4.3 Cosa ho fatto in C++

Capitolo 5

Risultati

5.1 Demo di gioco

5.2 Interazione utente

5.3 Design

5.4 Esempi

Conclusioni e sviluppi futuri

La tesi è finita

Bibliografia

- [JS96] Trevor H. Jones and Il-Yeol Song. Analysis of binary/ternary cardinality combinations in entity-relationship modeling. *Data Knowledge Engineering*, 19(1):39–64, 1996.
- [MS05] M. Mealling and R. Salz. A universally unique identifier (uuid) urn namespace. *RFC 4122*, 2005.