

# Relazione Progetto OOP “Dimension Holiday”

Lorenzo Prati, Elvis Perlika  
Emanuele Dajko, Alessandra Versari

April 9, 2023

# Contents

<b>1</b>	<b>Analisi</b>	<b>2</b>
1.1	Requisiti . . . . .	2
1.2	Analisi e modello del dominio . . . . .	3
<b>2</b>	<b>Design</b>	<b>5</b>
2.1	Design dettagliato . . . . .	5
<b>3</b>	<b>Sviluppo</b>	<b>6</b>
3.1	Testing automatizzato . . . . .	6
3.2	Metodologia di lavoro . . . . .	6
3.3	Note di sviluppo . . . . .	6
<b>4</b>	<b>Commenti finali</b>	<b>7</b>
4.1	Autovalutazione e lavori futuri . . . . .	7
4.2	Difficoltà incontrate e commenti per i docenti . . . . .	7
<b>A</b>	<b>Guida utente</b>	<b>8</b>
<b>B</b>	<b>Esercitazioni di laboratorio</b>	<b>9</b>

# Chapter 1

## Analisi

### 1.1 Requisiti

Il gruppo si pone l'obiettivo di realizzare un videogioco roguelike *Dimension Holiday* vagamente ispirato a giochi famosi come *Hades* oppure *The Binding of Isaac*. Il giocatore controllerà un personaggio che è stato trasportato in un altro mondo dove dovrà esplorare un dungeon e affrontare un boss per tornare alla propria dimensione.

#### Elementi funzionali

- Per attraversare tutto il dungeon il giocatore dovrà passare da stanza in stanza, eliminando tutti i nemici presenti. Per farlo potrà usare la spada o lanciare proiettili energetici. Giocatore e nemici hanno delle vite (espresse in cuori) e se il giocatore perde tutti i cuori e' *Game over* e si deve ricominciare da capo
- Una volta che avrà ucciso tutti i nemici della stanza corrente compariranno dei reward casuali (cuori, monete ecc.) e il portale che ti trasporterà nella prossima stanza.
- Dopo un certo numero di stanze, comparirà una stanza shop dove sarà possibile effettuare acquisti usando le monete creando una piccola progressione
- Il gioco si conclude quando il giocatore sconfigge un nemico speciale chiamato *Boss*

## Elementi non funzionali

- Ci si pone l'obiettivo di creare un'architettura del software modulare ed espandibile ad aggiunte future, come l'aggiunta di nuovi nemici, mappe e oggetti.

## 1.2 Analisi e modello del dominio

Il giocatore potrà muoversi nelle 4 direzioni su, sinistra, giù, destra tramite i tasti, rispettivamente, W, A, S, D ed effettuare due tipi di attacchi:

- *meele*, ovvero ravvicinato, usando una spada e tramite il tasto sinistro del mouse
- dalla distanza, usando un proiettile energetico e tramite il tasto destro del mouse

Il gioco dovrà essere in grado di presentare al **giocatore** una serie di stanze dove affrontare dei **nemici**. Questi potranno avere diversi comportamenti e diverse tipologie di **attacco**. Il giocatore dovrà stare attento ad evitare gli attacchi dei nemici per non perdere cuori e allo stesso tempo non entrarci in contatto, cosa che comporterà ulteriore danno. Saranno presenti degli **oggetti** raccogliibili (cuori, monete ecc.) che dovranno applicare degli **effetti** alle entità con cui entrano in contatto, ad esempio l'incremento della vita oppure l'incremento della valuta posseduta dal giocatore. Lo shop sarà gestito *in game*, nel senso che non comparirà un'interfaccia grafica che permetterà al giocatore di scegliere i potenziamenti, ma il giocatore dovrà interagire dinamicamente con degli oggetti presenti nella mappa per acquistarli. Anche gli attacchi applicheranno degli effetti con le entità con cui entrano in contatto, come ad esempio la perdita di cuori. Il **mondo** di gioco (*dungeon*) sarà composto di una serie di stanze. Tramite l'interazione con un oggetto portale, il giocatore sarà trasportato alla stanza successiva senza possibilità di tornare indietro. All'interno delle stanze sono presenti dei muri, che bloccano il passaggio al giocatore. Esistono tre tipi di stanze: normale, shop, boss. Nella stanza normale compariranno dei nemici, in numero e tipo variabile in base al momento della partita, nella stanza shop invece compariranno gli oggetti rappresentati i potenziamenti acquistabili, e nella stanza boss comparirà solo il nemico boss. Le stanze normali saranno intercambiate randomicamente tra un pool prescelto di mappe create a mano, mentre le stanze shop e boss saranno uniche.

La difficoltà sarà gestita in modo tale che proseguendo nel dungeon risulti più difficile il gioco, ad esempio facendo comparire più nemici nelle stanze

oppure nemici piu' forti. Questo aumento della difficoltà sara' compensato dai potenziamenti che il giocatore potra' acquistare nello shop, che comparira' dopo un numero costante di stanze normali superate.

Una delle maggiori difficoltà consistera' nella creazione di un architettura che permetta la gestione sia di diversi tipi di nemici (zombie, shooter, boss ecc.), ognuno con un proprio comportamento, sia di diversi attacchi utilizzabili sia dal giocatore che dai nemici (proiettili, attacchi meelee). Inoltre, si cerchera' di realizzare un sistema di combattimento *real-time* quanto piu' possibile fluido e responsivo e un'alternanza di mappe e generazione dei nemici in modo tale da far sembrare ogni partita diversa.

Dato il monte ore previsto, si rimanda al futuro una gestione accurata delle performance del gioco.

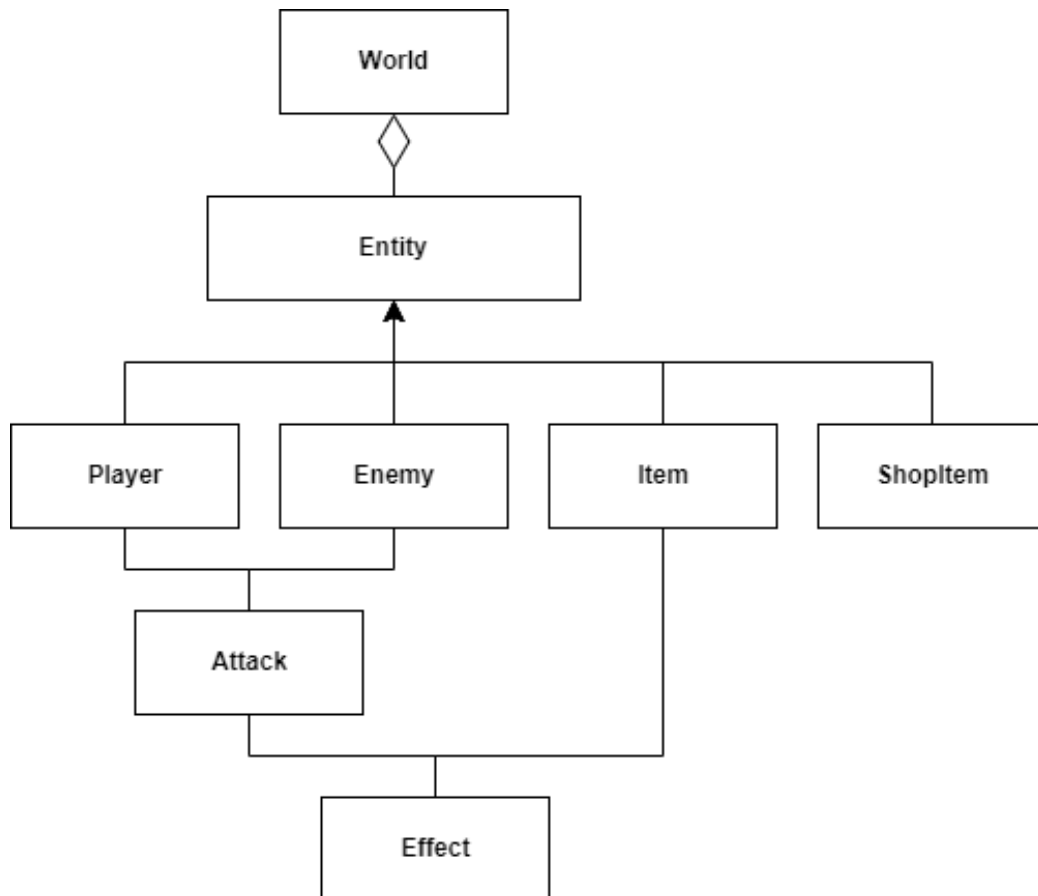


Figure 1.1: Diagramma uml concettuale rappresentante le varie entita' presenti nel dominio del gioco

# Chapter 2

## Design

### 2.1 Design dettagliato

# Chapter 3

## Sviluppo

3.1 Testing automatizzato

3.2 Metodologia di lavoro

3.3 Note di sviluppo

## Chapter 4

### Commenti finali

4.1 Autovalutazione e lavori futuri

4.2 Difficoltà incontrate e commenti per i docenti



# Appendix A

## Guida utente

## Appendix B

### Esercitazioni di laboratorio

# Bibliography