

OOP-FARGOAL

Bulgarelli Marco, marco.bulgarelli@studio.unibo.it
Ravaioli Alessandro, alessandro.ravaioli@studio.unibo.it
Tassinari Sabrina, sabrina.tassinari@studio.unibo.it
Tramonti Daniele, daniele.tramonti@studio.unibo.it

15 febbraio 2025

Indice

1	Analisi	2
1.1	Descrizione e requisiti	2
1.1.1	Requisiti funzionali	2
1.1.2	Modello del dominio	3
2	Design	4
2.1	Architettura	4
2.2	Design dettagliato	4
2.2.1	Bulgarelli Marco	4
2.2.2	Ravaioli Alessandro	4
2.2.3	Tassinari Sabrina	4
2.2.4	Tramonti Daniele	4
3	Sviluppo	5
3.0.1	Testing automatizzato	5
3.0.2	Note di sviluppo	5
3.0.3	Tassinari Sabrina	5
3.0.4	Tramonti Daniele	5
4	Commenti finali	6
4.0.1	Autovalutazione e lavori futuri	6
4.0.2	Difficoltà incontrate e commenti per i docenti	6
A	Guida utente	7
B	Esercitazioni di laboratorio	8

Capitolo 1

Analisi

1.1 Descrizione e requisiti

Il software mira alla costruzione di un videogioco ispirato a “Sword of Fargoal”. Quest’ultimo è un “Dungeon Crawler Arcade”, ovvero basato sull’esplorazione di un labirinto a più piani, con lo scopo di riportare in superficie la Spada di Fargoal attraversando innumerevoli pericoli. La nostra versione cercherà di essere il più fedele possibile al gioco originale.

1.1.1 Requisiti funzionali

- Il giocatore si muoverà all’interno di una grande stanza che corrisponde ad un piano del Dungeon. La generazione di ogni piano (e dei suoi contenuti) dovrà essere casuale.
- Il piano è caratterizzato dalla presenza di mostri. Questi possono apparire in luoghi specifici o in punti casuali, aumentando gradualmente con il passare del tempo, rendendo l’ambiente sempre più pericoloso.
- All’interno del piano saranno presenti due tipologie di oggetti di cui il giocatore potrà usufruire: dei bauli, che possono contenere degli oggetti magici o delle trappole, oppure delle sacche di monete.
- Il sistema di progresso del personaggio è legato all’accumulo di esperienza, che contribuisce all’aumentare del suo livello. Questa può essere ottenuta sia combattendo contro i mostri che offrendo donazioni in oro ai templi.
- In ogni piano sarà presente almeno un tempio, dove è possibile donare oro per salire di livello, rigenerare punti ferita e, ogni volta che sono state donate 2000 monete, di essere curati completamente. Inoltre, i templi garantiscono temporanea inattaccabilità ed una rigenerazione accelerata.
- Una volta recuperata la spada, è necessario riportarla in superficie, ritornando al primo piano e salendo le scale per uscire dal labirinto. La sfida

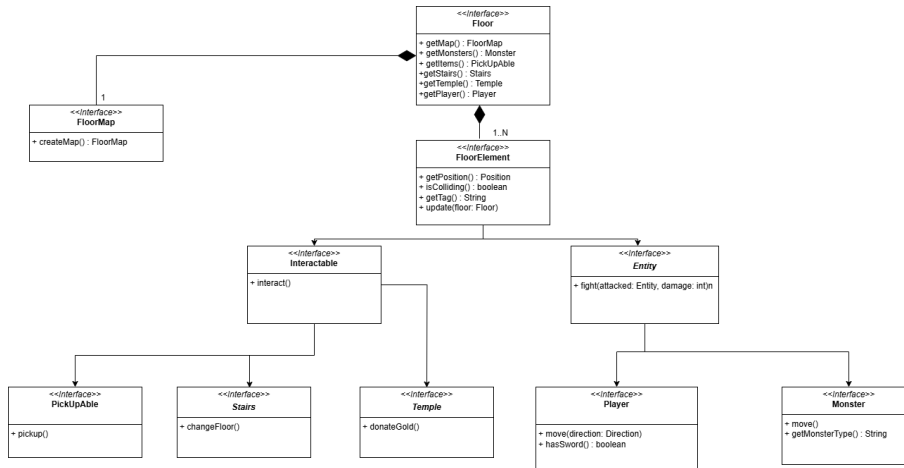


Figura 1.1: Schema UML dell’analisi del problema, con rappresentate le entità principali ed i rapporti fra di loro

sta nel fatto che, se il giocatore viene attaccato da un mostro, perde la spada, che ritornerà automaticamente nel piano in cui è stata trovata.

1.1.2 Modello del dominio

Il labirinto (*dungeon*) è formato da più piani (*Floor*). Per muoversi fra un piano e l’altro si utilizzeranno delle rampe di scale; in uno stesso piano saranno presenti più rampe per risalire o per scendere. Ogni volta che si entra in un nuovo piano, che sia antecedente o successivo, esso verrà generato casualmente. All’interno di ogni piano saranno presenti vari elementi (in 1.1 si chiamano *FloorElement*): alcuni hanno la caratteristica di essere interagibili (*Interactable*), mentre gli altri elementi sono delle entità (*Entity*), che si muovono all’interno del piano. I primi, che sono fissi nella mappa, sono l’insieme formato dalle ceste (che contengono oggetti magici), le scale (in 1.1 *Stairs*) e il tempio (in 1.1 *Temple*), che è unico all’interno del piano. Le entità, invece, sono l’insieme dei mostri (in 1.1 *Monster*) e del giocatore (in 1.1 *Player*); entrambi si muovono liberamente all’interno del piano e possono entrare in combattimento l’uno con l’altro.

Capitolo 2

Design

2.1 Architettura

2.2 Design dettagliato

2.2.1 Bulgarelli Marco

2.2.2 Ravaioli Alessandro

2.2.3 Tassinari Sabrina

2.2.4 Tramonti Daniele

Capitolo 3

Sviluppo

3.0.1 Testing automatizzato

3.0.2 Note di sviluppo

Bulgarelli Marco

-

Ravaioli Alessandro

-

3.0.3 Tassinari Sabrina

-

3.0.4 Tramonti Daniele

-

Capitolo 4

Commenti finali

4.0.1 Autovalutazione e lavori futuri

Bulgarelli Marco

Ravaoli Alessandro

Tassinari Sabrina

Tramonti Daniele

4.0.2 Difficoltà incontrate e commenti per i docenti

Appendice A

Guida utente

Appendice B

Esercitazioni di laboratorio