

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Corso di
Paradigmi di Programmazione e Sviluppo
Docenti: Mirko Viroli, Roberto Casadei

TBOI

The Binding of Isaac

Autori:
Matteo Brocca · 1005681
Alan Mancini · 1005481
Federico Mazzini · XXXXXXXX

Settembre 2021

Indice

1	Introduzione	2
2	Processo di sviluppo adottato	3
2.1	Meeting	3
2.1.1	Sprint planning	3
2.1.2	Daily Scrum	3
2.1.3	Sprint review	3
2.1.4	Sprint retrospective	3
2.2	Divisione dei Task	3
2.3	Revisione dei Task	3
2.4	Tool	3
3	Requisiti	4
3.1	Requisiti di business	4
3.2	Requisiti utente	4
3.3	Requisiti funzionali	4
3.3.1	Requisiti funzionali 1.0	4
3.3.2	Requisiti funzionali 2.0	5
3.4	Requisiti non funzionali	6
3.5	Requisiti di implementazione	6
4	Design architetturale	7
5	Design di dettaglio	8
6	Implementazione	9
7	Retrospettiva	10

1 Introduzione

Realizzazione di ...

2 Processo di sviluppo adottato

2.1 Meeting

2.1.1 Sprint planning

2.1.2 Daily Scrum

2.1.3 Sprint review

2.1.4 Sprint retrospective

2.2 Divisione dei Task

2.3 Revisione dei Task

2.4 Tool

3 Requisiti

3.1 Requisiti di business

- Creazione di un gioco di genere Roguelike su piattaforma JVM
- Possibilità di gioco su browser

3.2 Requisiti utente

Matteo Brocca in questo progetto ricopre il ruolo di esperto di dominio e committente. E' un appassionato di giochi Roguelike ma essendo troppo bravo li ha finiti tutti. Da qui l'idea di crearne uno nuovo per lui.

1. L'utente può avviare una nuova partita
 - 1.1 All'inizio di ogni partita l'utente partirà con caratteristiche e punti vita base, all'interno di una mappa composta da stanze definite in modo procedurale
2. L'utente potrà controllare un personaggio all'interno della mappa per
 - 2.1 spostarsi all'interno della mappa
 - 2.2 sparare ai nemici
 - 2.3 raccogliere elementi utili all'aumento delle sue caratteristiche

3.3 Requisiti funzionali

3.3.1 Requisiti funzionali 1.0

1. Generazione in modo casuale di una mappa 2D formata da più stanze
2. Ogni stanza è generata in modo casuale e può essere di diversa tipologia:
 - 2.1 stanza con 1 oggetto
 - 2.2 stanza con nemici ed elementi bloccanti
 - 2.3 stanza con boss (1 sola nella mappa)
3. Personaggio con caratteristiche e punti vita
4. Controllo del personaggio tramite movimento e sparo
5. Diversi tipi di nemici con diverse tipologie di movimento: casuale o intelligente
6. Diversi tipi di oggetti per potenziare le caratteristiche del personaggio
7. Menù di gioco

3.3.2 Requisiti funzionali 2.0

1. Da un menu di gioco, deve essere possibile avviare una nuova partita
2. Generazione di una mappa 2D in maniera procedurale
 - 2.1 Una mappa è formata da più stanze
 - 2.1.1 Una stanza è di diverse tipologie
 - 2.1.1.1 Stanza con 1 oggetto
 - 2.1.1.2 Stanza con nemici ed elementi bloccanti
 - 2.1.1.3 Stanza con boss (1 sola nella mappa)
 - 2.1.2 Una stanza ha due porte
 - 2.1.2.1 La porta permette di entrare all'interno di una stanza
 - 2.1.2.2 La porta permette di uscire da una stanza
 - 2.1.2.3 All'entrata del giocatore nella stanza, le porte si chiudono
 - 2.1.2.4 Le porte di una stanza si aprono solamente quando il giocatore ha eliminato tutti i nemici all'interno
3. Personaggio con caratteristiche e punti vita controllato dall'utente
 - 3.1 Il personaggio è controllato dall'utente
 - 3.1.1 L'utente può muovere il personaggio nelle quattro direzioni principali (sopra, sotto, destra, sinistra)
 - 3.1.2 L'utente può eseguire uno o più "shot" verso una delle quattro direzioni principali (sopra, sotto, destra, sinistra)
 - 3.2 Il personaggio può infliggere danno ai nemici
 - 3.2.1 Il personaggio infligge danno ai nemici quando uno "shot" colpisce un nemico
 - 3.3 Il personaggio perde punti vita
 - 3.3.1 Il personaggio è danneggiato, e i suoi punti vita diminuiscono, quando è toccato da un nemico
 - 3.4 Il personaggio può raccogliere oggetti che modificano le sue caratteristiche
4. Presenza di nemici all'interno delle stanze
 - 4.1 I nemici possono avere caratteristiche diverse
 - 4.1.1 Un nemico può avere 2 tipologie di movimento
 - 4.1.1.1 Nemico con movimenti casuali all'interno di una stanza
 - 4.1.1.2 Nemico con movimento in direzione del giocatore

- 4.1.2 Un nemico può infliggere danno diverso in base alla sua tipologia
- 4.2 Un nemico è presente in una e una sola stanza, da cui non può uscire
- 4.3 Un nemico è danneggiato, e i suoi punti vita diminuiscono, quando è colpito da uno shot del giocatore
- 5. Presenza di oggetti all'interno delle stanze
 - 5.1 Un oggetto è un artefatto che modifica le caratteristiche di un personaggio
 - 5.2 Gli oggetti sono di diversi tipi e sono disposti in modo casuale all'interno della mappa

3.4 Requisiti non funzionali

- 1. Fluidità di gioco
 - 1.1 Il gioco non deve presentare lag o inestetismi marcati che rendano la user experience non ottimale
- 2. Bilanciare numero e caratteristiche dei nemici per ottenere un livello di difficoltà di gioco medio
 - 2.1 Il gioco non deve essere troppo facile
 - 2.1.1 Presenza di pochi nemici all'interno di una stanza
 - 2.1.2 Presenza di oggetti che potenzino le caratteristiche del giocatore rendendo il conflitto giocatore-nemici impari
 - 2.2 Il gioco non deve essere troppo difficile
 - 2.2.1 Presenza di troppi nemici all'interno di una stanza
 - 2.2.2 Presenza di oggetti che non potenzino o riducano le caratteristiche del giocatore rendendo il conflitto giocatore-nemici impari

3.5 Requisiti di implementazione

Implementazione: requirement posti agli sviluppatori. Danno indicazioni sul lavoro da fare. Possono comprendere: tipi di strumenti utilizzati, modi di sviluppo, documentazione fornita.

4 Design architetturale

Design architetturale (architettura complessiva, descrizione di pattern architetturali usati, componenti del sistema distribuito, scelte tecnologiche cruciali ai fini architetturali – corredato da pochi ma efficaci diagrammi)

5 Design di dettaglio

Design di dettaglio (scelte rilevanti, pattern di progettazione, organizzazione del codice
– corredato da pochi ma efficaci diagrammi)

6 Implementazione

Implementazione (per ogni studente, una sotto-sezione descrittiva di cosa fatto/co-fatto e con chi, e descrizione di aspetti implementativi importanti non già presenti nel design)

7 Retrospettiva

Restrospettiva (descrizione finale dettagliata dell'andamento dello sviluppo, del backlog, delle iterazioni; commenti finali)

Riferimenti bibliografici