

Aetherlum Survivor¹

Lorenzo Cleti n°364138

¹Progetto per: Università degli Studi di Perugia - Professor
Luca Grilli



Chapter 1

Indice

1. INDICE
2. DESCRIZIONE DEL PROBLEMA
 - 2.1 *Concetto alla base dell'applicazione*
 - 2.2 *Riferimenti a giochi preesistenti*
3. SPECIFICA DEI REQUISITI
 - 3.1 *Funzionali*
 - 3.2 *Non Funzionali*
 - 3.3 *Di Gioco*
4. PROGETTO
 - 4.1 *Architettura del Sistema Software*
 - 4.2 *Descrizione dei Moduli*
 - 4.1 Model
 - 4.2 View
 - 4.3 Controller
 - 4.3 *Problematiche Ricontrate e Osservazioni*
5. APPENDICE
6. BIBLIOGRAFIA

Chapter 2

Descrizione del Problema

2.1 *Concetto alla base dell'applicazione*

Obiettivo di questo progetto è quello di creare un'applicazione grafica ispirata al genere "*bullet-hell*". Il giocatore controlla un personaggio che deve sopravvivere a ondate incessanti di nemici statici e/o in movimento. Eliminare i nemici garantisce *Punti Esperienza* che permettono di migliorare alcune caratteristiche del personaggio, rendendolo così più potente. In aggiunta, durante il gioco sono generati casualmente ulteriori *Potenziamenti*. Il personaggio si può difendere in 2 modi: spostarsi evitando così il contatto con i nemici e/o ucciderli tramite il lancio automatico di incantesimi. L'obiettivo non è quello di uccidere tutti i nemici, che continuano ad apparire ininterrottamente fino al *Game Over*, bensì quello di sopravvivere più a lungo possibile.

2.2 *Riferimento a giochi preesistenti*

Nella scelta del progetto sono stati presi come riferimento i giochi *Magic Survival* e *Vampire Survivors*, il cui concetto e idea generale sono stati di ispirazione per la progettazione e creazione di **Aetherlum Survivor**

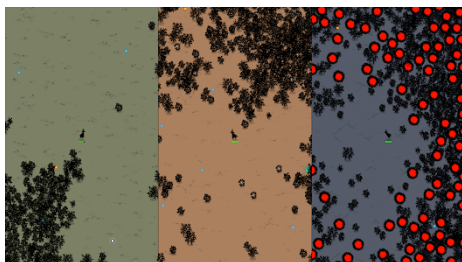


Figure 2.1: Alcune schermate giustapposte di Magic Survival

Chapter 3

Specifica dei Requisiti

Suddividiamo per praticità i requisiti nelle seguenti sezioni.

3.1 *Funzionali*

Sono elencati i requisiti che hanno un diretto riscontro da parte del giocatore.

- audio e suoni che possono essere disattivati
- difficoltà variabile tramite la possibilità di modificare il numero massimo di nemici contemporaneamente presenti
- possibilità di mettere in pausa la partita

3.2 *Non Funzionali*

Sono elencati i requisiti di gioco ignoti all'utente.

- architettura 'Model-View-Controller'
- applicazione leggera
- controllo di collisioni tra molteplici entità: personaggio, proiettili - nemici, potenziamenti

3.3 *Di Gioco*

Sono elencati i requisiti più strettamente legati alle meccaniche di gioco.

- più di uno scenario presente
- meccaniche di potenziamento del personaggio

Chapter 4

Progetto

Chapter 5

Appendice

Chapter 6

Bibliografia