

Matricola : 2087310

Corso : informatica - presenza M/Z

Nome : Mattia

Cognome : Pandolfi

Decisione di progettazione delle specifiche:

E' stato utilizzato Java Swing per la parte grafica

La partita completa comprende 6 livelli + 2 livelli boss, l'arena è una matrice che viene aggiornata generando tutti gli sprite quando si passa al prossimo livello.

la partita è stata ideata in modo tale da creare delle "scene", per passare dal menu al gioco.

I 5 nemici Implementati sono:

- Puopen, si muove avanti e indietro

- Denkyun, quando colpisce un muro cambia direzione

- StarNuts, ha la preferenza di andare a destra

- Cupper, in un momento random si teletrasporta in un punto a caso della mappa

- Kierun, in un momento random diventa invisibile

c'è una classe per ogni tipo nemico che estendono la classe Enemy (dove ci sono parti di codice in comune)

I boss implementati sono:

- Bigeron e Clown Mask,

anche le classi dei boss estendono la classe Enemy

I power sono:

- Aarmor: rende invincibili per tot tempo

- Extra Bomb: aggiunge una bomba

- Explosion Expander: il range del fuoco è piu grande

- Accelerator: aumenta velocità

- Bombberman: aggiunge una vita di massimo 9

- Maximum Explosion: mette il range massimo del fuoco

- Walkable bombs: puoi attraversare le bombe

- Carbonara: guadagna grande quantità di punti

sono una classe astratta e ogni tipo di classe power up è un'estensione della classe base PowerUp.

le animazioni sono state implementate in 2 modi diversi, per il player abbiamo usato una successione di sprites mentre per powerup e nemici abbiamo utilizzato i gif

possibilità di giocare con il mouse sviluppata creando un interfaccia mouselistener, selezionabile al menu iniziale.

in base dove si clicca sullo schermo con il tasto sinistro verrà stabilita la direzione del personaggio(cliccare sopra farà andare il bomberman sopra etc..) e con il tasto destro verrà rilasciata la bomba.

Sono state implementate 2 classi per la riproduzione di audio, una per le musiche e una per gli effetti sonori. la differenza sta che puoi riprodurre più effetti sonori contemporaneamente mentre la musica appena verrà caricata una nuova traccia fermerà quella precedente automaticamente.

La gestione dei profili è stata ideata nel seguente modo: ci sono tre file .dat che corrispondono a 3 profili possibili. giocando la partita si aggiorneranno i file in modo da salvare le statistiche anche quando il gioco viene chiuso.

Design Pattern adottati

MVC:

Pattern adottato per dividere tutte le classi in modo ordinato.

Nella parte controller ci saranno le classi che permetteranno all'user di interfacciarsi nel gioco, per esempio l'input dei tasti WASD modificheranno nella model del player le coordinate in cui si trova.

Nella parte del model saranno presenti le classi cui conterranno dati.

Nella parte della view saranno presenti le classi che gestiranno tutta la parte grafica

Observer Observable :

Utilizzato per aggiornare il giocatore e i nemici. Quando nella classe Player, Enemy etc.. avviene un cambiamento della posizione allora viene notificato alle loro rispettive classi nella parte grafica in tal modo di aggiornare la parte visiva da mostrare al giocatore

Singleton :

utilizzato per permettere di istanziare solo UNA partita di BomberMan

Uso degli Stream

gli stream sono stati usati per:

- + il caricamento del prossimo livello di gioco nella classe "Arena.java"
- + gestire il caricamento di nemici all'interno di una matrice (rappresentante un livello del gioco) e li aggiunge a una lista di oggetti nemici. individua le celle che rappresentano nemici specifici (in base al valore numerico) e crea oggetti nemici corrispondenti con le coordinate di partenza. Gli oggetti nemici vengono quindi aggiunti a una lista per essere gestiti nel gioco.

