Home sweet Home

# Introduzione

Home sweet Home è un gioco scritto in linguaggio java. Il gioco è diviso in livelli e il suo scopo è quello di acquisire più punti possibili.

Il giocatore dovrà difendere la sua casa dagli Zombi. La difficoltà aumenta ad ogni livello. Si acquistano punti colpendo gli Zombi e superando più livelli possibili. Il gioco terminerà solo quando gli Zombi riusciranno a distruggere completamente la porta della casa.

I creatori di Home sweet Home sono: Simone Massaro e Alessandro Spisso.

# Istruzioni

* Per muoversi usare: W ↑ S ↓ A ← D →
* Per utilizzare il martello contro gli zombi o riparare la porta usare: ENTER (l’azione verrà effettuata solo al rilascio del pulsante)
* Per iniziare il gioco: SPACE

# Composizione del gioco

Il gioco è diviso principalmente in quatro parti: le classi riguardanti l’interfaccia utente, le classi riguardanti la gestione del gioco, MODEL e TEST.

### L’interfaccia utente

Le classi BackgroundView, GameOverView, GamePauseView, PlayerView e ZombieView possono essere considerate solo dei “disegni” utilizzate da altre classi.

La classe GameView, con estensione Panel, svolge un ruolo fondamentale. La sua funzione è quella di “disegnare” le classi precedenti con la funzione paint. Inoltre ha un’implementazione Runnable per utilizzare un Thread con lo scopo di dare animazione al gioco, attraverso vari repaint.

### La gestione del gioco

Le classi che hanno un ruolo fondamentale nella gestione del gioco sono:

* GameSettings: utilizzata all’inizio del gioco, per dare la possibilità al Giocatore di selezionare la dimensione della finestra e il nome utente
* GameWindow: ha un’estensione JFrame, per creare una finestra e ha l’implementazione KeyListener, per ricevere i comandi
* GameLevel: controlla il passaggio del gioco ad un livello successivo

## Model (player e zombie trattati sotto)

* GameScores: gestisce il punteggio e lo stampa in un file al termine del gioco

## Test

### Le classi riguardanti il Player

Le classi che riguardano il Player sono:

* PlayerController con estensione PlayerView (disegno del Player). La sua funzione è controllare e animare il player.
* PlayerModel contiene le variabili del Player (coordinate, vita, forza, velocità).

### Le classi riguardanti gli Zombi

Le classi che riguardano gli Zombi sono:

* ZombieNodeModel con estensione ZombieView, è un nodo che possiede le variabili del singolo Zombi (coordinate, vita, forza, velocità).
* ZombiesController ha un l’implementazione Runnable, la sua funzione è di creare e gestire una lista di ZombieNodeModel con un Thread. Inoltre dà l’input di fine e di inizio livello.