

GARA GRA i LABIRINTI

Configurazione e breve guida al gioco

Configurazione

Files

A regolare gran parte degli aspetti del gioco vi sono una serie di files di configurazione.

Tali files sono contenuti nella cartella `assets` (contenuta nella root dell'applicazione) e sono denotati dall'estensione `.cfg`

File di configurazione necessari

config.cfg File di configurazione principale, definisce:

- Dimensione dello schermo
- File di configurazione dei livelli
- File di configurazione dei bonus
- Font
- Effetti sonori
- Personaggio dell'utente

levels/A/A.cfg Almeno un file di configurazione per un livello, definisce:

- Nome del livello
- Aspetto di mura e corridoi
- Complessità
- Mappe
- Sottofondo musicale
- Avversari

Formato dei files

Di seguito la specifica BNF del formato dei files di configurazione.

```

{String Ch}      = {Printable} - ["]

DecimalLiteral   = [0123456789]{digit}*
FloatLiteral     = {Digit}* '.' {Digit}+

StringLiteral    = '"' {String Ch}* '"'

{IdTokenH}       = {letter} + [_]
{IdTokenT}       = {IdTokenH} + {digit}
Identifier       = {IdTokenH}{IdTokenT}*

<Size Value> ::= '[' DecimalLiteral ',' DecimalLiteral ']'
<Size Values> ::= ',' <Size Value> <Size Values> |
<Size Values list> ::= <Size Value> <Size Values>

<Rectangle Value> ::= '[' DecimalLiteral ',' DecimalLiteral ','
                        DecimalLiteral ',' DecimalLiteral ']'
<Rectangle Values> ::= ',' <Rectangle Value> <Rectangle Values> |
<Rectangle Values list> ::= <Rectangle Value> <Rectangle Values>

```

```

<Integer Value> ::= DecimallLiteral
<Integer Values> ::= ',' <Integer Value> <Integer Values> |
<Integer Values list> ::= <Integer Value> <Integer Values>

<Float Value> ::= FloatLiteral | DecimallLiteral
<Float Values> ::= ',' <Float Value> <Float Values> |
<Float Values list> ::= <Float Value> <Float Values>

<String Value> ::= StringLiteral
<String Values> ::= ',' <String Value> <String Values> |
<String Values list> ::= <String Value> <String Values>

<Size Declaration> ::= 'size' Identifier '=' <Size Value>
<Rectangle Declaration> ::= 'rectangle' Identifier '=' <Rectangle Value>

<Integer Declaration> ::= 'int' Identifier '=' <Integer Value>
<Float Declaration> ::= 'float' Identifier '=' <Float Value>

<String Declaration> ::= 'string' Identifier '=' <String Value>

<List of sizes Declaration> ::= 'list' '[' 'size' ']'
                                Identifier '=' <Size Values list>
<List of rectangles Declaration> ::= 'list' '[' 'rectangle' ']'
                                Identifier '=' <Rectangle Values list>
<List of ints Declaration> ::= 'list' '[' 'int' ']'
                                Identifier '=' <Integer Values list>
<List of floats Declaration> ::= 'list' '[' 'float' ']'
                                Identifier '=' <Float Values list>
<List of strings Declaration> ::= 'list' '[' 'string' ']'
                                Identifier '=' <String Values list>

<Declaration> ::= <Size Declaration>
                | <Rectangle Declaration>
                | <String Declaration>
                | <Integer Declaration>
                | <Float Declaration>
                | <List of sizes Declaration>
                | <List of rectangles Declaration>
                | <List of ints Declaration>
                | <List of floats Declaration>
                | <List of strings Declaration>
                | <Dictionary Declaration>

<Dictionary entries> ::= ',' <Declaration> <Dictionary entries> |
<Dictionary entries list> ::= <Declaration> <Dictionary entries> |

<Dictionary Declaration> ::= 'dictionary' Identifier '=' '{' <Dictionary entries list> '}'

<Declarations> ::= <Declaration> ';' <Declarations> |

```

Personaggio

I personaggi sono descritti da una collezione di proprietà definite all'interno di una variabile di tipo *dictionary*. Il personaggio controllato dall'utente è descritto dalla variabile `dictionary user` del file `config.cfg`. I personaggi avversari sono descritti nei files di configurazioni di ogni livello (come indicato in seguito).

Proprietà

string `sprites` — Utente/Avversari

Path della bitmap contenente il personaggio.

rectangle `rect` — Utente/Avversari

Riquadro della bitmap riguardante il personaggio.

float `speed` — [default: 2] — Utente/Avversari

Velocità del personaggio.

int `ignores_collisions` — [default: 0] — Utente/Avversari

Specifica se il personaggio subisce (0) o meno (1) gli effetti di uno scontro con un'altro personaggio.

int `randomize_position` — [default: 0] — Utente/Avversari

Se 1 il personaggio è posizionato in una cella casuale della mappa.

float `alpha` — [default: 1] — Utente/Avversari

Di norma utilizzato per i bonus, livello di trasparenza del personaggio.

int `breaks_walls` — [default: 0] — Utente

Specifica se il personaggio è in grado (1) o meno (0) di abbattere le mura.

size `visibile_area_size` — [default: $[\infty, \infty]$] — Utente

Dimensione del riquadro, centrato nella posizione dell'utente, di mappa visibile.

int `show_shortest_path_to_exit` — [default: 0] — Utente

Di norma utilizzato per i bonus, se 1 viene evidenziato sulla mappa il percorso più breve dalla posizione dell'utente all'uscita.

int `lives` — [default: 3] — Utente

Numero di vite iniziali dell'utente. Minimo: 1. Massimo: 255.

size `chase_rect_size` — [default: [2, 2]] — Avversari

Dimensione del riquadro di inseguimento centrato nella posizione del personaggio.

size `exit_search_rect_size` — [default: [2, 2]] — Avversari

Dimensione del riquadro di ricerca dell'uscita centrato nella posizione del personaggio.

int `chase_user` — [default: 1] — Avversari

Determina il comportamento dell'avversario, se 1 l'avversario preferisce inseguire l'utente piuttosto che cercare l'uscita, viceversa se 0.

string `path_finding_method` — [default: BFS] — Avversari

Metodo di ricerca dei percorsi utilizzato del personaggio, tra:

- BFS
- Dijkstra
- A*

string `chasing_method` — [default: ``] — Avversari

Metodo di inseguimento, tra:

default → l'avversario cerca di raggiungere la cella in cui si trova l'utente.

predict position → l'avversario cerca di predire la posizione dell'utente tenendo in considerazione la sua posizione e la sua direzione e calcolando la cella più lontana raggiungibile da quest'ultimo.

trap → l'avversario *collabora* con un altro avversario e cerca di tendere un'imboscata al personaggio inseguito.

Esempio

```
dictionary enemy1 = {
    string      sprites          = "assets/enemies.bmp",
    rectangle   rect             = [0, 0, 72, 128],
    string      path_finding_method = "BFS",
    size        exit_search_rect_size = [2, 2],
    size        chase_rect_size    = [3, 3],
    int         chase_user         = 1,
    string      chasing_method     = "predict position",
    float       speed              = 3,
    int         ignores_collisions = 1
};
```

config.cfg: file di configurazione principale

Contiene informazioni fondamentali per l'avvio dell'applicazione. Di seguito le opzioni specificabili (in *corsivo* quelle obbligatorie).

Proprietà

size screen

Dimensione, in pixel, della finestra principale.

string font

Path della bitmap del font da utilizzare per le operazioni di scrittura.

string powerups

Path del file di configurazione dei bonus.

string levels_path

Path della cartella principale dei livelli

list[string] levels

Lista dei nomi dei file di configurazione dei livelli, in modo tale che questi si trovino in *levels_path/nome/nome.cfg*.

dictionary sfx

Raccolta dei percorsi degli effetti sonori del gioco:

intro → Nel menù principale

gameover → In caso di sconfitta

won → In caso di vittoria

wall → L'utente ha abbattuto un muro

crash → Scontro tra due personaggi

dictionary user

Configurazione del personaggio dell'utente.

File di esempio

```

size          screen      = [1024, 768];

string        levels_path = "assets/levels";
list[string]  levels      = "A", "B", "C", "D", "E", "F";

string        powerups    = "assets/powerups.cfg";

string        font        = "assets/font.bmp";

dictionary sfx = {
    string      intro      = "assets/audio/intro.mod",
    string      gameover   = "assets/audio/gameover.wav",
    string      won        = "assets/audio/won.wav",
    string      wall       = "assets/audio/wall.wav",
    string      crash      = "assets/audio/crash.wav"
};

dictionary user = {
    string      sprites    = "assets/user.bmp",
    rectangle   rect       = [0, 0, 72, 128],
    float       speed      = 3
};

```

powerups.cfg: file di configurazione dei bonus

Raccoglie le definizioni dei bonus che possono comparire sulla mappa nel corso del gioco. Le definizioni sono organizzate in una variabile *dictionary* per ogni tipo di bonus e una di tipo *list*, `list[string] powerups` avente per elementi i nomi dei bonus (nomi delle variabili *dictionary*).

Proprietà di un bonus***dictionary picker***

Descrive gli effetti che il bonus ha sul personaggio che lo raccoglie (e quindi utilizza).

La proprietà può essere composta da 2 elementi:

dictionary user — effetti del bonus se chi raccoglie è l'**utente**

Le proprietà specificabili sono analoghe a quelle utilizzabili in fase di descrizione dei personaggi.

dictionary enemy — effetti del bonus se chi raccoglie è un **avversario**

Le proprietà specificabili sono analoghe a quelle utilizzabili in fase di descrizione dei personaggi.

dictionary characters

Descrive gli effetti che il bonus ha su tutti i personaggi eccetto chi lo raccoglie. Anche in questo caso le proprietà specificabili sono analoghe a quelle utilizzabili in fase di descrizione dei personaggi.

size effects_rect_size

Dimensione del riquadro, centrato nella posizione dell'utente che utilizza il bonus, all'interno del quale far registrare gli effetti.

float appearance_probability

Probabilità ([0.0, 1.0]) che in una delle celle destinate ai bonus compaia questo tipo di bonus.

int duration

Durata, in secondi, degli effetti del bonus.

int timeout

Numero di secondi prima della scomparsa del bonus.

int limit

Numero massimo di volte che un bonus può essere inserito in una mappa.

string trigger

Tasto che l'utente deve premere per attivare il bonus. Se non è dichiarato, il bonus è attivato automaticamente nel momento in cui è raccolto.

string audio

Traccia audio da riprodurre quando è utilizzato il bonus.

string texture_file

Path della bitmap contenente la rappresentazione grafica del bonus.

rectangle texture_rect

Riquadro della bitmap contenete il bonus.

File di esempio

```
dictionary RandomPosition = {
    dictionary picker = {
        dictionary user = {
            int    randomize_position = 1,
            float  alpha = 0.
        },
        dictionary enemy = {
            int    randomize_position = 1,
            float  alpha = 0.
        }
    },

    float    appearance_probability = 0.05,
    int      limit = 2,
    string    texture_file = "assets/powerups.bmp",
    rectangle texture_rect = [0, 96, 24, 32]
};

dictionary T1 = {
    dictionary picker = {
        dictionary user = {
            float speed = 2.
        }
    },

    float    appearance_probability = 0.05,
    int      limit = 2,
    int      timeout = 10,
    string    texture_file = "assets/powerups.bmp",
    rectangle texture_rect = [0, 192, 24, 32]
};

list[string] powerups = "RandomPosition", "T1";
```

levels/nome/nome.cfg: file di configurazione di un livello

Ogni file di configurazione di un livello contiene informazioni riguardo: le caratteristiche del livello, le mappe che lo compongono e gli avversari che lo caratterizzano.

Proprietà

string name

Nome del livello, univoco, utilizzato internamente. Se non è specificato, al livello è assegnato un nome generato in maniera casuale.

string background_texture

Path alla bitmap contenente le immagini da utilizzare per i percorsi delle mappe del livello.

rectangle background

Riquadro della bitmap contenente il percorso del livello in questione.

string wall_texture

Path alla bitmap contenente le immagini da utilizzare per le mura delle mappe del livello.

rectangle wall

Riquadro della bitmap contenente le mura del livello in questione.

float complexity

Grado di complessità del livello ([0.0, 1.0]). **Maggiore** è il grado di complessità:

- **minori** saranno i bonus inseriti;
- **maggiore** sarà l'attesa tra due inserimenti di bonus;
- **maggiori e più vari** i percorsi a percorrenza più lenta/veloce;

string audio

Path della traccia audio da riprodurre come sottofondo musicale del livello.

list[string] maps

Lista di stringhe che definisce i livelli in ordine di apparizione. Ogni stringa può essere:

nome_mappa.cfg il percorso di un file di configurazione di una mappa (relativo alla cartella del livello)

random per generare una mappa casuale

random perfect per generare una mappa casuale contenente un labirinto perfetto (cioè senza cicli).

list[string] enemies

Lista dei nomi delle variabili che descrivono gli avversari.

dictionary enemyN

Dizionario che definisce un'avversario dichiarando le proprietà dei personaggi.

levels/nome/map.cfg: definizione di una mappa

Proprietà

float powerup_probability

Probabilità che una cella possa contenere un bonus (1.0 = tutte le celle contengono un bonus).

float powerups_time

Tempo di attesa tra due inserimenti di bonus.

int powerups_limit

Numero massimo di bonus che possono comparire in totale. $\simeq \infty$ se non specificato.

size size







Dimensione in celle della mappa, è necessario che la mappa sia sufficientemente grande da contenere tutte le celle definite.

string map

Rappresentazione testuale della mappa. I caratteri validi sono:

- `#` — un muro
- `[1 - 9]` — un percorso percorribile con una velocità diversa da quella standard (5)
- `" "` (*spazio, senza apici*) — rappresenta un percorso
- `S` — cella di inizio mappa
- `E` — cella di fine mappa
- `P` — cella destinata ad un bonus
- `V` — cella destinata ad un avversario

Mappa dei tasti

| | |
|---|--|
|  | Audio on/off |
|  | Pausa on/off |
|  | Chiude il gioco |
|  | Attiva un bonus. Il tasto assegnato ad ogni bonus è visualizzato sotto la voce <i>Power-UPs</i> nella schermata principale del gioco |
|  | Controllano i movimenti del personaggio e la selezione delle opzioni nel menù principale |
|  | Selezione di un'opzione nel menù principale |

Utilizzo

Menu principale

Eseguita l'applicazione ci si trova nel menù iniziale attraverso il quale è possibile decidere se cominciare a giocare, regolare le impostazioni o chiudere il gioco.

Gioca Avvia il gioco mostrando la prima mappa del primo livello specificato nel file di configurazione.

Disattiva/Attiva audio Permette di regolare lo stato dell'audio sia per il menù che per il resto del gioco.

Esci Chiude il gioco.



Figura 1: Schermata di gioco

Livelli

Il gioco è caratterizzato da una sequenza di livelli che, nell'ordine in cui sono definiti, devono essere completati dall'utente. Ogni livello, a sua volta, è caratterizzato da un diverso numero e tipo di mappe (labirinti) e di avversari. I personaggi sono liberi di muoversi, rimanendo nei corridoi nelle mappe e passare da una mappa all'altra attraverso un percorso, che può essere attraversato in una sola direzione (a seconda dell'ordine di comparizione, dalla mappa precedente ad una mappa successiva).

Scopo: vittoria e sconfitta

Vittoria Perché l'utente vinca è necessario che completi tutti i livelli prima di ogni avversario e con almeno 1 vita rimanente.

Sconfitta La sconfitta può avvenire in 2 casi:

1. L'utente termina le vite a disposizione
2. Un avversario termina un livello per primo

Utente

Il personaggio può essere controllato, tramite i tasti direzionali, per far sì che cambi direzione. Nel momento in cui il personaggio comincia a muoversi o cambia direzione continuerà il movimento fino a scontrarsi con un muro o un avversario.



Figura 2: Personaggio dell'utente

Utente – Avversario Lo scopo dell'utente, oltre a raggiungere la fine della mappa prima degli avversari, è anche quello di evitare di scontrarsi con questi ultimi. In caso di collisione, i personaggi coinvolti vengono spostati in celle casuali della mappa e all'utente è sottratta 1 vita.

Vite Sono indicate dai cuori in basso a destra nella schermata del gioco. Quando terminano, l'utente è stato sconfitto e termina anche il gioco. È possibile ottenere vite aggiuntive attraverso i bonus (se configurati in questo senso).

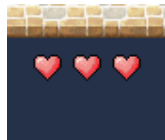


Figura 3: Vite rimanenti

Mappa

Di base la mappa è costituita da 2 tipi di celle: celle di tipo corridoio, attraverso le quali i personaggi si muovono e tentano di raggiungere l'uscita, e celle di tipo muro che fungono da ostacolo. Alcune di queste celle però hanno funzioni particolari.

Cella di fine mappa È una cella speciale che consente il passaggio da una mappa all'altra (4). Tale cella è evidenziata da una sfumatura in nero.



Figura 4: Cella di fine mappa

Celle comunicanti Quando il personaggio arriva in corrispondenza di una cella di tipo corridoio che si trova su uno dei lati della mappa e sul lato opposto è presente un'altra cella di tipo percorso, il personaggio continua il suo movimento spostandosi in quella cella (5, celle sul lato sinistro e destro).



Figura 5: Cella comunicanti

Celle con costo diverso Oltre alle normali celle di tipo corridoio e mura, è possibile che nella mappa siano presenti celle corridoio percorribili più/meno velocemente dai personaggi.

Le celle di tipo corridoio a percorrenza veloce (6) sono di un colore più chiaro (tendente al bianco) rispetto a quelle normali.



Figura 6: Cella corridoio a percorrenza veloce

Le celle di tipo corridoio a percorrenza lenta (7) sono di un colore più scuro (tendente al nero) rispetto a quelle normali.



Figura 7: Cella corridoio a percorrenza lenta

Avversari

In ogni livello sono presenti un certo numero di avversari con diverse proprietà che hanno lo scopo di competere ed ostacolare l'utente.

Alcuni avversari tenderanno ad inseguire l'utente, altri a cercare l'uscita della mappa.

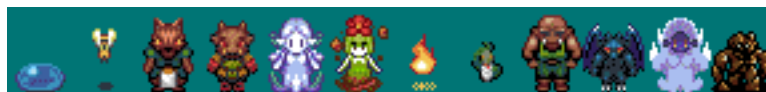


Figura 8: Personaggi avversari

Bonus

Ad intervalli di tempo regolari, in determinate celle della mappa, possono comparire degli oggetti che i personaggi, posizionandosi nella cella che li contiene, possono raccogliere e utilizzare per modificare temporaneamente alcune delle loro proprietà. A seconda dei valori assegnati alle proprietà, un bonus può favorire o sfavorire un personaggio.



Figura 9: Bonus

Raccolto un bonus, se nel file di configurazione non è specificato alcun tasto associato, questo viene attivato immediatamente. In caso sia specificato un tasto, invece, il bonus è aggiunto alla lista dei bonus dell'utente. In figura 10, ad esempio, è stato raccolto un bonus *Speed* attivabile tramite la pressione del tasto **S** e della durata di 10 secondi.

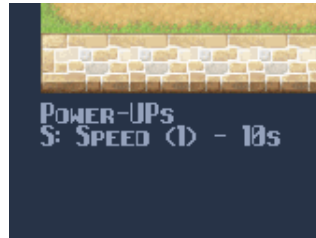


Figura 10: Lista dei bonus