

Escape from GOP

Prodotto da :

Giambi Nico

Polla Yari

Indice

1) Librerie usate

1.1) ctime.h

1.2) Qt

2) Classi utilizzate

3) Fasi di Gioco

3.1) Schermata iniziale

3.2) Selezione del numero dei giocatori

3.3) Assegnazione nomi e Avatar

3.4) Lancio dei dadi

4) Adattamento per Android

1) Librerie Usate

1.1) <ctime.h>

La libreria ctime svolge un ruolo fondamentale nella creazione di oggetti randomici. Nel nostro caso è stata utilizzata nella creazione del campo da gioco, generato casualmente ad ogni partita, per definire le proprietà delle caselle.

Funzioni usate :

<srand(TIME(NULL))>

<int example = rand()%range>

(con range che rappresenta il valore massimo dell’intervallo di randomizzazione)

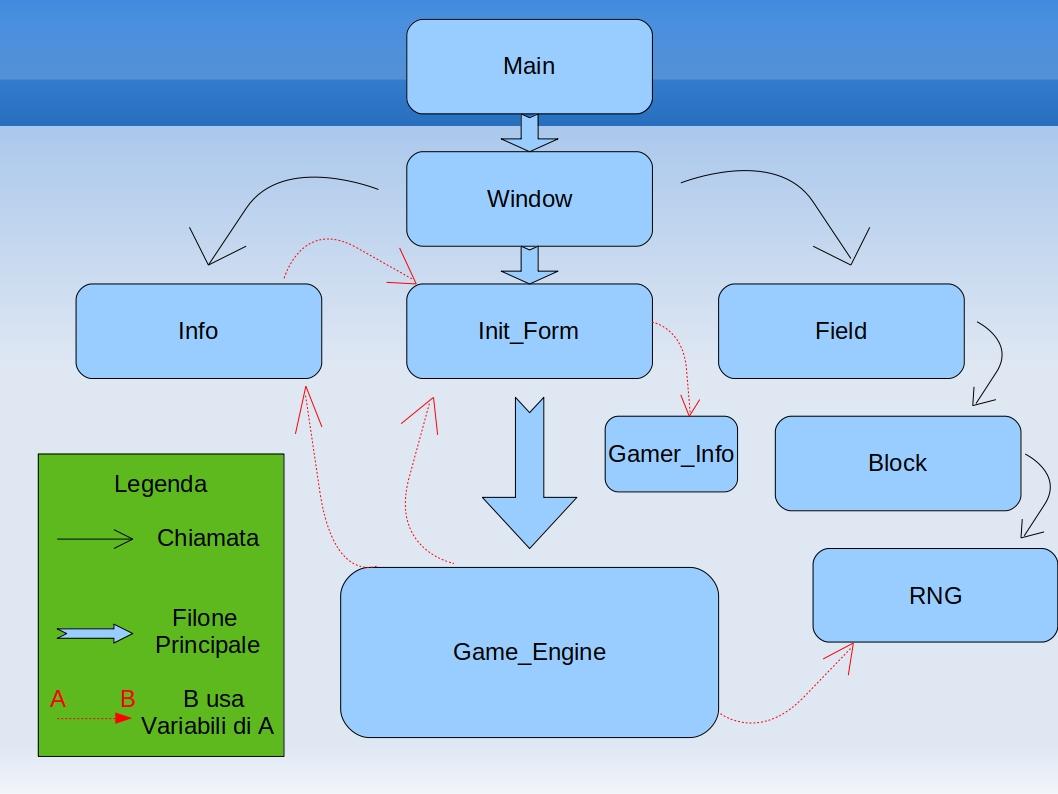
1.2) Qt

Il framework Qt è un insieme di librerie molto utilizzate nello sviluppo di applicazioni grafiche per PC, dispositivi mobili ed embedded.

Il framework in questione può essere utilizzato come estensione del C++ per realizzare delle interfacce grafiche di diverso tipo; esiste tuttavia un’implementazione per Python. Oltre a costituire un ambiente grafico, le Qt, offrono un complesso sottosistema di programmazione orientato agli oggetti con molte proprietà, da citare:

* Possibilità di connettere oggetti diversi attraverso **segnali** e **slots**;
* Utilizzo di alcuni tipi proprietari, con le relative funzioni di casting;
* Possibilità di compilare il codice per piattaforme diverse, modificando solo alcune piccole parti del progetto stesso.

2) Classi usate



Le classi che costituiscono la parte di gioco sono **Window, Init\_Form e Game\_Engine e RNG**.

Window: E’ la prima classe chiamata dal main, e rappresenta la finestra di selezione del numero dei giocatori.

---------------------------------------------------------------------------------

Init\_Form: Crea una lista di **n** giocatori (dove **n** è il numero di giocatori inserito in Window) tutti uguali, poi permette di modificare nomi e avatar dei singoli giocatori.

**Funzioni principali:**

* **Lists\_Creator() --**→ Dopo che Window ha preso in input dall’utente il numero (gamers\_counter) dei giocatori, questa funzione crea una matrice con nome e avatar (quello standard) dei giocatori.
* **Gamer\_Properties() ---→** Serve a visualizzare a schermo la matrice con nomi e avatar dei giocatori e i vari pulsanti per modificarli.
* **Edit\_Gamer() ---→** Permette di scegliere dalla cartella **“../multimedia/avatars”** presente in **“GOP”** un’immagine (.png o .jpg) da associare a un giocatore. E’ possibile cercare in tutto il PC utilizzato per giocare un’immagine non presente all’interno della cartella iniziale , da selezionare come avatar.

---------------------------------------------------------------------------------

Game\_Engine: E’ il vero e proprio motore di gioco. Contiene tutte le funzioni necessarie a far avanzare o retrocedere i giocatori, attivare gli effetti speciali delle caselle e così via.

**Funzioni principali:**

* **Init\_Deck() ---→** Crea un array di 40 **int** che simboleggiano l’effetto delle carte.
* **Random\_Deck()** **---→** Mescola il deck grazie a una variabile d’appoggio e ad un ciclo for.
* **Gamers\_Seq() ---→** Si avvia dopo aver cliccato **“roll”**. Stabilisce qual’è il giocatore che deve svolgere il proprio turno (tenendo in conto i turni bonus e gli stop) e lancia il dado.
* **Turno() ---→** Aggiorna la posizione del giocatore che si è spostato, ne sposta l’avatar ed emette il suono **“Quack”**. Poi, se il giocatore finisce in una casella con effetto, viene chiamata la funzione **Effect.**
* **Effect (int block\_type) ---→** Usa **block\_type** (che rappresenta il tipo di casella su cui è finito il giocatore dopo il suo ultimo spostamento) per effettuare uno **switch/case** che sceglierà quale effetto applicare al giocatore in base alla casella su cui è finito. Poi fa ripartire **Turno()** se c’è un effetto da applicare, altrimenti fa ripartire **Gamers\_Seq()**.

---------------------------------------------------------------------------------

RNG: E’ il Random Number Generator. Come dice il nome, contiene tutte le funzioni utili a generare un numero randomicamente sfruttando <ctime.h>. Viene usata ad esempio per generare il numero di caselle che compongono il campo e gli spostamenti casuali dati dagli effetti delle caselle.

**Funzioni principali:**

* **roll() ---→** Genera un numero randomico tra **1** e **6**.
* **general\_rng(int** min**, int** max**) ---→** Genera un numero randomico tra **min** e **max**.
* **blocks\_rng() ---→** Genera un numero randomico tra **40** e **76** (che indicherà la lunghezza del tabellone).
* **type\_rng() ---→** Serve a generare la proprietà di ogni singola casella, facendo in modo di non creare due caselle Rosse consecutive (le caselle rosse fanno retrocedere, quindi mettendole in serie potremmo incappare in un tratto di tabellone insuperabile).

---------------------------------------------------------------------------------

Mentre invece Field, Block e Info costituiscono la parte grafica

Block: E’ la classe che gestisce le caselle del tabellone.

**Funzioni principali:**

* **Block\_Type() ---→** Assegna i colori alle caselle in base al numero ad esse assegnato in precedenza.
* **Block\_Position() ---→** Ritorna le cordinate del blocco utilizzate da **Field** per disegnare il campo di gioco.

---------------------------------------------------------------------------------

Field: E’ la classe che si occupa di disegnare il campo. Prende le informazioni da Block per colorare le caselle.

**Funzioni principali:**

* **draw\_blocks() ---→** Sceglie la posizione in cui deve essere disegnata la casella successiva, poi richiama **Recursive\_Drawer()**
* **Recursive\_Drawer() ---→** Disegna il campo in modo ricorsivo partendo dalla casella centrale fino a formare una spirale.
* **Draw\_White\_Blocks (…) ---→** Disegna insieme alle caselle anche dei piccoli quadrati bianchi che contengono il numero della casella.
* **add\_papera() ---→** Richiama la funzione **Draw\_Papera** (che disegna l’avatar sul campo) e ne determina la posizione in cui farla apparire.

---------------------------------------------------------------------------------

Info: E’ la finestra visibile che contiene le informazioni riguardanti i singoli giocatori nella parte superiore, e il log degli spostamenti effettuati dai giocatori in quella inferiore. Il colore delle informazioni del log è lo stesso della casella che causa il movimento.

Abbiamo inoltre **gamer\_info.h**, che non è altro che una stuttura contenente i dati dei singoli giocatori.

3) Fasi di Gioco

3.1) Schermata iniziale

Appena avviato il gioco, verrà presentata una schermata con tre pulsanti (da cliccare con il mouse) che sono in ordine:

* **New Game** (Vedi sotto)
* **Credits** (Mostra i nomi degli sviluppatori del gioco)
* **Exit** (Esce dal gioco)

Cliccando su **New Game** verrà’ generato il campo in background, mentre gli utenti dovranno selezionare il numero dei giocatori.

3.2) Selezione del numero dei giocatori

Apparirà una finestra dove sarà possibile inserire il numero dei giocatori partecipanti alla partita. Si può giocare minimo in due e massimo in 2,147,483,647 (limite positivo dell’**int**). Il numero dei giocatori sarà modificabile sia da tastiera che da mouse cliccando sulle freccette accanto al numero.

3.3) Assegnazione nomi e Avatar

Un’altra finestra verrà mostrata, e al suo interno possiamo trovare sulla sinistra l’elenco dei giocatori, sui quali possiamo cliccare per accedere all’interfaccia del giocatore (posizionata sulla parte destra della finestra) dalla quale si potrà cambiare nome e avatar del giocatore. L’avatar sarà un immagine selezionata dall’utente presente nel dispositivo di gioco, che si muoverà poi sul tabellone per tenere traccia delle **“mosse”** del giocatore.

Cliccando poi su **finish**, verrà creata ufficialmente la lista dei giocatori, ordinati secondo l’utente durante l’assegnazione.

3.4) Lancio dei dadi

A questo punto ha inizio il gioco.

I giocatori saranno posizionati inizialmente sulla casella centrale, dalla quale si dovranno muovere verso l’esterno per “scappare” dal tabellone. Sulla destra dello schermo, a fianco del tabellone di gioco, ci saranno due finestre, affiancate verticalmente, ovvero:

* **Sopra**: sarà visibile l’elenco dei giocatori con i prorpi avatar e posizione sul tabellone.
* **Sotto**: sarà presente un log delle azioni svolte dai giocatori, con colori diversi in base al tipo di azione.
* **Inoltre**, in alto a destra ci sarà il pulsante **“Roll”**, grazie al quale sarà possibile (cliccando su di esso o usando la shortcut **Alt+R**) lanciare il dado, così da far muovere il giocatore in avanti sul tabellone.

Il turno di un giocatore inizia quando si lancia il dado, e termina quando questo cade su una casella vuota **(Nera)**.

Se il giocatore cade su una casella con effetto, questo verrà attivato e potrà essere di diversi tipi:

* **Avanza di n caselle; (Verde)**
* **Retrocedi di n caselle; (Rosso)**
* **Vai alla casella n; (Blu)**
* **Stai fermo un turno; (Giallo)**
* **Pesca una carta; (Magenta)**

Quando si cade nella casella Magenta, si potrà attivare uno degli effetti precedenti o alcuni supplementari, descritti nel mazzo di carte (creato all’interno della classe Game\_Engine) che possonoi essere:

* **Gioca un turno bonus;**
* **Avanza di 15 caselle;**
* **Avanza di 25 caselle;**
* **Torna al via;**

4) Adattamento per Android

**Escape from GOP** è giocabile anche in dispositivi Android, con solo alcune piccole differenze:

* Il dado del turno non sarà più lanciabile tramite il pulsante **“roll”** o premendo **“Alt+R”,**  ma basterà cliccare un qualsiasi punto della parte sinistra dello schermo, ovvero quella del tabellone.
* Sulla destra non ci sono le informazioni dei giocatori ma solo un log.
* Sarà possibile giocare in 2-4 giocatori.
* Non sarà possibile selezionare gli avatar, verranno assegnati direttamente ai giocatori (ovviamente tutti diversi).