

### Elaborato finale: gioco

Si realizzi un programma che permette di disputare partite a un gioco simile a Candy Crush.

Il fine del gioco è ottenere il massimo punteggio possibile entro un numero massimo di mosse. All'inizio di una partita, il giocatore vede una scacchiera di 10x10 caselle, riempita in modo quasi casuale da 100 valori, scelti tra tre lettere o numeri fissi (per esempio le lettere A, B, C). La configurazione iniziale deve essere tale da non presentare tre simboli uguali adiacenti lungo la direzione verticale, orizzontale e diagonale.

All'inizio della partita, il punteggio del giocatore è pari a 0.

A ogni turno, il giocatore indica le coordinate di una casella di partenza e il valore di una direzione, scelta tra alto, basso, destra, sinistra, diagonale alto-destra, diagonale alto-sinistra, diagonale basso-destra, diagonale basso-sinistra. Si lascia libertà di scegliere i tasti con cui il giocatore può indicare una delle sei possibilità. La casella di destinazione è quella ottenuta muovendosi di un passo nella direzione indicata. Una volta che la casella di destinazione è stata identificata, il programma effettua uno scambio di simboli tra le due caselle, in maniera tale che il simbolo prima presente nella casella di partenza sia ora nella casella di destinazione, e viceversa.

Se uno scambio porta lo stesso simbolo ad essere presente in tre (o più) caselle adiacenti, il gruppo di simboli viene *consumato*, cioè sostituito con spazi bianchi. Si noti che tale controllo deve essere effettuato per ognuna delle sei direzioni possibili (in altre parole, è necessario considerare l'adiacenza lungo le direzioni orizzontale, verticale e le due diagonali).

La presenza di spazi bianchi determina la caduta verso il basso dei simboli presenti nelle caselle superiori, come se questi fossero soggetti ad un fenomeno di gravità; è possibile che gli spostamenti dovuti alla gravità portino a un *effetto a catena*, dove altri gruppi di tre (o più) simboli vengano consumati a causa della loro adiacenza.

Alla fine del ciclo di cadute verso il basso e di consumi:

- Sia  $N$  (da 0 a 3) il numero di simboli distinti consumati.
- Sia  $S_i$  l' $i$ -esimo simbolo consumato (A, B, o C)
- Sia  $N_i$  il numero di elementi consumati per il simbolo  $S_i$
- Sia  $w_i$  il peso associato all' $i$ -esimo simbolo, che ne definisce il valore (a discrezione del programmatore)
- Il punteggio  $P(S_i)$  di un simbolo consumato  $S_i$  è calcolato come:

$$P(S_i) = N_i w_i$$

- Il punteggio del giocatore è incrementato di un importo pari alla **somma pesata** dei punteggi dei simboli consumati  $P(S_i)$ :

$$P = \sum_{i=1}^N P(S_i)$$

**Esempio.**

Supponiamo che i simboli siano A, B e C e le caselle consumate contenessero  $[A A A B B B B]$ . Allora, il numero di simboli consumati è  $N = 2$ .

Se  $S_1 = A$  e  $S_2 = B$  allora,  $N_1 = 3$  (numero di A consumate) e  $N_2 = 4$  (numero di B consumate).

Supponiamo che il programmatore abbia deciso che il simbolo  $S_1 = A$  valga  $w_1 = 1$  e che il simbolo  $S_2 = B$  valga  $w_2 = 3$ . Allora:

$$\begin{aligned}P(S_1) &= 3 \cdot 1 = 3 \\P(S_2) &= 4 \cdot 3 = 12\end{aligned}$$

Il punteggio del giocatore è quindi incrementato di:

$$P = 3 + 12 = 15$$

Dopo il completamento di una mossa, il programma riempie le caselle bianche con nuovi simboli, avendo cura di evitare che un inserimento casuale di un simbolo porti alla creazione di gruppi di tre (o più) simboli uguali adiacenti. Nel caso in cui questo non sia possibile scegliere una configurazione che garantisca l'assenza di esplosioni tra i nuovi simboli inseriti.

Al termine del gioco, il programma fornisce all'utente il punteggio realizzato.

Il programma consente l'accesso a diversi giocatori, identificati mediante un nome, e fornisce un comando per vedere la lista dei giocatori, con il massimo punteggio da loro realizzato, ordinata in modo decrescente in base al punteggio.

In aggiunta alla modalità di gioco interattiva, dove il giocatore specifica di volta in volta la prossima mossa da eseguire e tutte le informazioni sono visualizzate a video, il programma permette anche una modalità di gioco *sequenziale*. Quando eseguito in tale modalità, il programma accetta in ingresso due file testuali: il primo contiene la configurazione iniziale della scacchiera e il secondo la sequenza delle mosse del giocatore. Il programma produce un terzo file di testo contenente la configurazione finale prodotta dall'applicazione delle mosse. Il formato di salvataggio delle informazioni nel file di testo è il seguente:

- La configurazione della scacchiera è rappresentata da una sequenza di righe separate da “;”. Una riga è una sequenza di simboli separati da “,”.
- Una mossa è una sequenza di tre numeri interi: le coordinate della casella di partenza e il numero che rappresenta la direzione (1=alto, 2=basso, 3=destra, 4=sinistra, 5=diagonale alto-destra, 6=diagonale alto-sinistra, 7=diagonale basso-destra, 8=diagonale basso-sinistra).
- Le mosse nella sequenza sono separate da “;”

**FUNZIONI OPZIONALI**

1. Emettere suoni diversi in corrispondenza di ogni azione di gioco rilevante.
2. Realizzare una funzione che simuli in modo intelligente le scelte del giocatore.

3. Mediante le librerie grafiche di QT, realizzare una versione del gioco con interfaccia a finestre.