



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

GOP – Gioco dell’Oca Pazza

Organizzazione del codice

Workgroup

Martina Sosto, Andrea Vicenzi, Sara Vorabbi

Obiettivi

Riuscire a ricreare il gioco da tavolo “Gioco dell’Oca” attraverso le tecniche di programmazione apprese durante il corso di Programmazione in modo completo e funzionale. Sperimentare le tecniche viste a lezione su scala più grande, in direzione di ciò che succede nel mondo reale e collaborare nella realizzazione del progetto.

Tema

La scelta è ricaduta sul tema piratesco, apprezzato da tutti i membri del team, in quanto ritenuto spiritoso, divertente e ormai molto diffuso e popolare nell’immaginario collettivo. Inoltre, è stato pensato in modo tale da rendere l’esperienza di gioco più avventurosa e avvincente.

Grafica

Per la realizzazione della parte grafica, non sono state utilizzate ulteriori librerie o tecnologie esterne, in quanto abbiamo ritenuto sufficiente l’utilizzo di quelle apprese a lezione e di alcune che ci hanno permesso di interagire con il sistema Windows. Il gioco viene dunque visualizzato da terminale e anche le interazioni avvengono tramite esso.

Caratteristiche

Giocatori:

Da 2 a 6, a ogni giocatore viene assegnato un nome e un carattere che userà come pedina durante lo svolgimento del gioco

Tabellone:

Dinamico, il numero delle caselle varia pseudo-casualmente ad ogni partita con un minimo di 35 e un massimo di 65. Le caselle sono di 11 tipologie diverse.

Caselle:

- Casella vuota

- Tira ancora i dadi
- Vai avanti di N caselle
- Pesca una carta
- Vai indietro di 5 passi
- Tira i dadi, se il risultato è 5 per subirai conseguenze, altrimenti pesca una carta
- Vai alla casella N
- Resta fermo 2 turni
- Il giocatore ne raggiunge un altro (scelto casuale)
- Scambia la posizione con un altro giocatore (scelto casuale)
- Torna all'inizio:
 1. Torna alla casella 0 se il giocatore si trova in una posizione <20
 2. Torna alla casella 20 se il giocatore si trova in una posizione >20

Carte:

Il mazzo è composto da 42 carte, e include 8 carte per ognuno dei 6 oggetti che i giocatori dovranno raccogliere durante la partita.

1. Rum
2. Polvere da sparo
3. Archibugio
4. Bussola
5. Pappagallo
6. Doblone

Le carte vengono pescate sia alla fine di ogni turno sia quando il giocatore finisce sulle apposite caselle "pesca carta". Ogni volta che se ne pesca una, il mazzo viene decrementato e, nel caso le carte finiscano, questo viene rigenerato automaticamente e casualmente.

I giocatori collezionano le carte raccolte durante il gioco in un inventario creato come array di interi, nel quale ogni oggetto corrisponde ad un indice dell'array. Ogni intero dentro all'array rappresenta il numero di carte di quel tipo possedute dal giocatore.

Gli oggetti collezionati attiveranno due diversi effetti speciali:

1. 6 carte dello stesso tipo ti permetterà di mandare tutti gli altri giocatori in stato di fermo per un turno
2. Una carta per ogni tipo ti permetterà di mandare tutti gli altri giocatori in dietro di 3 caselle

Svolgimento del gioco

Inizio della partita:

All'inizio del gioco, dopo la comparsa della schermata iniziale, si ha la possibilità di scegliere il numero dei giocatori, i rispettivi nomi e la propria pedina che si muoverà sul tabellone. Vengono inoltre decisi l'ordine di gioco, il numero delle caselle e viene creato il mazzo delle carte (tutto in

modo casuale). Prima dell'inizio della partita viene mostrata una schermata in cui si spiegano le regole e le dinamiche fondamentali del gioco.

Partita:

Durante la partita verranno mostrati il tabellone e le pedine di tutti i giocatori, le informazioni più importanti riguardanti i giocatori. Premendo determinati tasti, i giocatori avranno la possibilità di uscire dal gioco o visualizzare una pagina contenente maggiori dettagli riguardo a turni e carte collezionate durante il corso del gioco.

Turno:

Al lancio dei dadi segue l'avanzamento delle pedine nel tabellone e l'esecuzione delle operazioni secondarie, tra cui l'esecuzione degli effetti delle caselle e delle eventuali carte pescate. Alla fine di proprio turno il giocatore pescherà una carta.

Fine del gioco:

Il primo giocatore a tagliare il traguardo sarà il vincitore, portando il gioco alla sua conclusione.

Scelte implementative

Oggetti Sviluppati

È stato deciso di modularizzare completamente il gioco attraverso l'utilizzo della programmazione a oggetti, ovvero con la creazione di classi che interagiscono tra di loro. In questo modo si è potuto creare un progetto facilmente modificabile e leggibile, poiché questo tipo di programmazione permette di gestire singolarmente i singoli oggetti e apportare modifiche o aggiunte in modo più semplice e intuitivo.

Gop

Classe principale, la sua chiamata fa iniziare il gioco, attraverso la creazione degli array dei giocatori, delle classi, delle carte

- `l_giocatori[sei]` array che punta agli oggetti della classe Giocatori
- `l_caselle[sessantacinque]` – array che punta agli oggetti della classe caselle
- `mazzo[quarantadue]` - array che punta agli oggetti della classe
- `num_giocatori` – numero totale dei giocatori per partita
- `num_caselle` – numero totale delle caselle per partita
- `num_carte` – numero delle carte che verrà decrementato
- `round` – variabile che viene incrementata ogni volta che tutti i giocatori hanno eseguito il proprio turno
- `next_turno` – variabile che tiene conto di chi gioca durante ogni round

Giocatori – campi

- `nome[venti]` – nome del giocatore
- `numero` – intero che indica l'ordine di inserimento dei giocatori
- `turno` – intero che indica l'ordine di gioco
- `pedina` – char utilizzato nel tabellone
- `posizione` – intero che indica la casella attuale su cui si trova il giocatore

- inventario – array in cui vengono “collezionati” gli oggetti trovati pescando le carte. Ad ogni indice corrisponde un oggetto e lo stesso oggetto può essere collezionato molteplici volte. Se l’oggetto viene collezionato più di sei volte o il giocatore colleziona tutti i sei oggetti prima del termine della partita, attiva delle proprietà speciali che lo avvantaggiano nel gioco

Caselle - campi

- Numero – numero della casella
- Effetto – effetto della casella che viene scelto con una diversa probabilità a seconda della tipologia

Carte - campi

- Numero – numero della carta nel mazzo
- Oggetto – numero che l’oggetto corrispondente alla carta (varia da 1 a 6)

Software e metodi di condivisione codice

CLion

Abbiamo scelto un IDE comune a tutti per lo sviluppo del progetto per evitare problemi di compatibilità. La scelta è ricaduta su CLion per le sue caratteristiche e funzionalità che ci hanno aiutato durante lo sviluppo del codice. Il suo utilizzo è stato possibile grazie all’account dell’università.

Dropbox

Per la condivisione dei file e del codice durante la fase di sviluppo abbiamo scelto di utilizzare Dropbox come archiviazione generale, inserendo in cartelle compresse le parti di codice. Ad ogni aggiornamento venivano inserite le nuove versioni del codice, così facendo al termine del lavoro abbiamo potuto monitorare l’intera evoluzione del codice.

Codeshare

Editor di testo open-source online, utilizzato per scambio di codice e programmazione simultanea.

Report Incontri

26/01/2018- Brainstorming sulla realizzazione del progetto.

Presenti: Martina Sosto, Andrea Vicenzi, Sara Vorabbi

23/02/2018 - Decisione della distribuzione dei ruoli e finalità del progetto, inizio stesura dei documenti.

Presenti: Martina Sosto, Andrea Vicenzi, Sara Vorabbi

26/02/2018 - Ricerca per l'elaborazione grafica del progetto.

Presenti: Martina Sosto, Andrea Vicenzi

02/03/2018 - Meeting aggiornamento svolgimento progetto.

Presenti: Martina Sosto, Andrea Vicenzi, Sara Vorabbi

03/03/2018 - Scambio di idee riguardanti la creazione grafica (e non) del tabellone.

Presenti: Martina Sosto, Andrea Vicenzi

09/03/2018 - Scambio di idee riguardanti l'organizzazione dei file e delle classi.

Presenti: Martina Sosto, Sara Vorabbi

12/03/2018 - Avanzamento creazione grafica del tabellone e ricerca sullo sviluppo riguardanti le carte e caselle.

Presenti: Martina Sosto, Andrea Vicenzi

05/06/2018 - Controllo stato avanzamento progetto.

Presenti: Martina Sosto, Andrea Vicenzi, Sara Vorabbi

06/06/2018 - Creazione funzione per i turni ed eventuali funzioni relative.

Presenti: Martina Sosto, Sara Vorabbi

07/06/2018 - Correzione di eventuali bug e creazione di funzioni riguardanti le gestioni del turno.

Presenti: Martina Sosto, Sara Vorabbi

08/06/2018 - Sviluppo classe carte, risoluzione di bug grafici e implementativi, aggiornamento della documentazione del progetto.

Presenti: Martina Sosto, Andrea Vicenzi, Sara Vorabbi

09/06/2018 – Risoluzione di bug funzioni carte, implementazione funzioni grafiche carte, aggiornamento della documentazione del progetto.

Presenti: Martina Sosto, Andrea Vicenzi, Sara Vorabbi

10/06/2018 – Risoluzione di bug, aggiornamento della documentazione del progetto.

Presenti: Andrea Vicenzi, Sara Vorabbi

Monte ore

Martina Sosto:

- 1 ½h Brainstorming sulla realizzazione del progetto
- 1h Decisione della distribuzione dei ruoli e finalità del progetto, inizio stesura dei documenti
- 1 ½h Ricerca per l'elaborazione grafica del progetto
- 2h Creazione grafiche e settaggi iniziali di gioco
- 1h Creazione codice riguardante assegnamento casuale turni e grafiche iniziali
- 1 ½h Avanzamento creazione grafica per il tabellone
- ½h Scambio di idee riguardanti l'organizzazione dei file e delle classi
- 3h Rivisita del codice riguardante le classi GOP e Giocatore, inserimento di metodi riguardanti la gestione del Giocatore
- 2 ½h Stampa del tabellone e correzione di eventuali bug riguardanti le singole caselle
- 2h Modifiche e integrazione del tabellone nella cartella di progetto originale
- 2h Realizzazione info status dei giocatori sotto il tabellone, ampliamento finestra di gioco (e annesse funzioni)
- 2h Creazione funzione per la gestione dei turni
- 3 ½h Perfezionamento gestione turni e eventuali bug
- 2h Correzione errori riguardanti: turni, dado, caselle, giocatori
- 3h Implementazione metodi classe Carta e correzione bug
- 2 ½h Metodi classe casce Carte + testing
- 4 ½h Creazione parte grafica riguardante le Carte + testing
- 4 h Integrato le funzioni delle carte nei turni e corretto vari bug + testing
- 5h Correzione errori riguardanti il tabellone, funzioni riguardanti le carte e testing generale

TOTALE ORE IMPIEGATE: 45h

Andrea Vicenzi:

- 1 ½h Brainstorming sulla realizzazione del progetto
- 1h Decisione della distribuzione dei ruoli e finalità del progetto, inizio stesura dei documenti
- 1 ½h Ricerca per l'elaborazione grafica del progetto
- 1 ½h Ricerca sullo sviluppo riguardanti le carte e caselle
- 1 ½h Studio tabellone e caselle
- 3 h Creazione algoritmo posizionamento pedine
- 1 ½h Studio tabellone e caselle
- 2 h Studio e creazione effetti carte

- 3 h Studio classe carta e funzioni annesse
- 3 h Creazione classe mazzo e funzioni dedicate
- 2 h Testing programma completo e rilevazione bug
- 1 h Modifiche e correzione file di testo
- 5½ Completamento e revisione finale dei documenti, correzione bug finali
- 1h Correzione bug finali

TOTALE ORE IMPIEGATE: 27½h

Sara Vorabbi:

- 1 ½h Brainstorming sulla realizzazione del progetto
- 1h Decisione della distribuzione dei ruoli e finalità del progetto, inizio stesura dei documenti
- ½h Scambio di idee riguardanti l'organizzazione dei file e delle classi
- 1h Ricerca per la divisione dei file in classi e successiva riorganizzazione del codice
- 1½h Creazione dado, algoritmo e grafica
- 1h Ricerche sulle librerie per grafica dado e numeri pseudo-casuali
- 2h Creazione metodi caselle
- 1h Studio della probabilità uscita caselle
- 1 ½h Creazione grafica caselle
- 2h Creazione funzione per la gestione dei turni
- 3 ½h Perfezionamento gestione turni ed eventuali bug
- 2h Correzione errori riguardanti: turni, dado, caselle, giocatori
- 1h Modifica dei documenti
- 4 ½h Creazione funzioni riguardanti le Carte + supporto
- 3h Modifica, stesura e correzione dei documenti
- 2h Creazioni funzioni grafiche
- 5h Correzione errori riguardanti il tabellone, funzioni riguardanti le carte e testing generale
- 5½ Completamento e revisione finale dei documenti, correzione bug finali
- 1h Correzione bug finali

TOTALE ORE IMPIEGATE: 39½h

Fonti

<http://www.messletters.com/en/big-text/>

<https://stackoverflow.com/>

<http://www.cplusplus.com/>

<https://www.google.it/>

<http://www.html.it/>

<https://msdn.microsoft.com>

Slide del corso di Programmazione