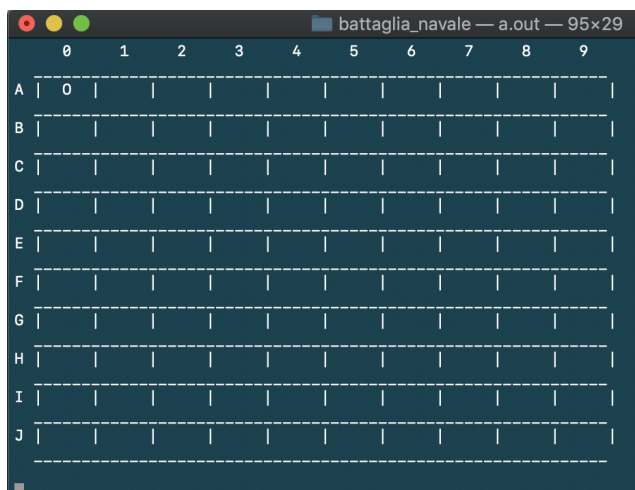


Relazione informatica

30 maggio 2019

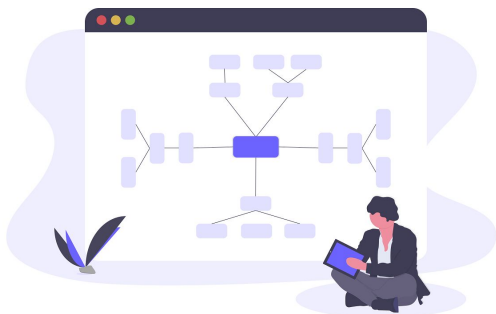
Panoramica del progetto finito

Quello mostrato nella foto che segue è il progetto completo e finito. Avevamo delle linee guida da rispettare, alcune obbligatorie ed alcune opzionali. Sono state completate tutte il prodotto è funzionante nella sua interezza.



Concept di base

Secondo me l'obiettivo principale di ogni progetto è tentare, dove possibile, di rendere il contenuto fruibile dalla maggior parte dei dispositivi, per questo il nostro programma è multiplatforma, grazie anche alla libreria ncurses.



Obbiettivi

Come precedentemente descritto, uno degli obiettivi primari è il multiplatform. Ovviamente il solo multiplatform non fa il prodotto, per questo ci siamo posti ulteriori obiettivi:

- Rispecchiare il gioco reale
- Creare un applicativo facile da usare
- Rendere il codice open-source



Dettagli

Il principio di funzionamento è molto semplice. Le librerie che vengono usate, escludendo quelle standard, sono:

- Ncurses per avere la possibilità di interagire in modo dettagliato con le azioni dell'utente
- windows.h / time.h /unistd.h scelte dal pre-compilatore in base al sistema operativo

Il gioco non è destinato ad avere una dimensione standard, perché la tabella viene creata ed auto-regolata in base alla lunghezza voluta dall'utente.

Il codice è stato gestito in funzioni per consentire un facile approccio durante modifiche successive.

L'utente per posizionare le navi muove un cursore con le freccette, non vengono usate le tipiche coordinate.

Le posizioni vengono salvate insieme ad altri valori all'interno di un vettore tridimensionale (in seguito spiegato nel dettaglio).

E' disponibile per l'utente la possibilità di giocare in multiplayer locale o contro un bot.

Le Funzioni

Ogni funzione ha un preciso scopo, ora verranno elencate le principali e spiegata la loro funzione:

- **Display()** → Prende un parametro in input, che servirà a determinare se eseguire o meno alcune azioni. Serve per mostrare a video la tabella di gioco aggiornata
- **A_navi()** → Prendendo in input un parametro di scelta, richiede all'utente di posizionare le sue navi, avvalendosi della funzione Tasti()
- **Tasti()** → Prende in input un parametro di scelta. Gestisce la parte grafica delle scelte e il movimento con le frecce del / dei cursori.
- **Scelta_cas()** → Non prende alcun parametro in input. Si occupa di richiedere al player la posizione dove vuole attaccare il nemico e passarla alla funzione verifica
- **Verifica()** → Prendendo in input la posizione ed altri parametri, determina se l'utente / bot ha colpito una nave nemica.
- **Menu()** → Gestisce i menu di scelta presenti nel gioco
- **Frase()** → Semplice funzione che avvalendosi della funzione Sleep_ms() stampa a schermo una stringa presa in input con una piccola animazione
- **Sleep_ms()** → Prendendo in input i millisecondi, gestisce gli sleep in base al sistema operativo



Vettore tridimensionale

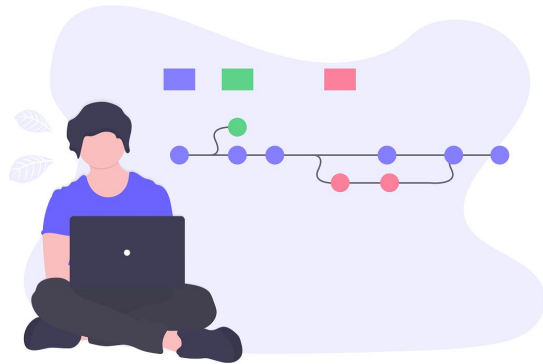
Viene usata per gestire la tabella del player, vengono immagazzinate le seguenti informazioni:

- Al primo indice è presente la posizione della riga
- Al secondo è presente la posizione della colonna
- Al terzo indice sono presenti informazioni come: il carattere della nave, il danno della nave, la sua lunghezza e un codice univoco legato alla sola nave

Sistema di versionamento e github

Come sistema di versionamento abbiamo usato git.

Visto che uno dei principali obiettivi è l'open-source il codice è stato reso pubblico su github



Difficoltà riscontrate

Nel suo complesso il codice è molto semplice ma essendo abbastanza lungo (circa 1600 righe) durante lo sviluppo sono stati commessi alcuni errori, corretti successivamente.

Il bug più grave che abbiamo affrontato è stato un problema grafico dovuto alla libreria ncurses.