

Progetto MiniLaska

Gruppo: 59

Membri: Giorgio Basile, Tommaso Palmisano, Lorenzo Fontana.

Struttura del progetto

Lo sviluppo del progetto si è svolto in due fasi principali. La prima fase ha visto la creazione della struttura di base del progetto ideando l'interfaccia grafica, la struttura iniziale del progetto e la divisione del lavoro. Nella seconda fase il gruppo si è concentrato sullo sviluppo delle funzionalità più avanzate del gioco e soprattutto sulla modalità Player vs PC.

La creazione del gioco è incominciata con il realizzare gli assets di gioco, ossia la struttura pedina e la scacchiera dove posizionarla, quest'ultima era stata inizialmente implementata con una grafica minimal, successivamente ha subito un Restyling completo per renderla più gradevole e chiara.

La matrice che lavora alla base è una matrice formata da array di array e ogni cella ha i suoi campi.

Una volta create le basi si è passato alla realizzazione della modalità player contro player, creando le funzioni che permettono il movimento, la mangiata, la promozione ed i vari controlli per far sì che le regole del gioco siano rispettate, la funzione di controllo per il movimento si accerta che un giocatore possa muovere le pedine solamente in caselle libere o nel caso di una possibile mangiata, la funzione di movimento sfrutterà poi il controllo per effettuare l'effettivo spostamento o mangiata della pedina nel caso sia possibile.

Per rispettare completamente le regole di miniLaska abbiamo implementato la funzione obbligamangiata, che come dice il nome, obbliga alla mangiata una pedina che ne ha la possibilità, per consentire una maggiore libertà al/i giocatore/i questa parte del regolamento può essere disabilitata nelle impostazioni di gioco.

Le strutture che abbiamo creato a doc per il nostro scopo sono state 2 pedina e pos.

La prima va a comporre la matrice di gioco che identifica ogni casella se è presente una pedina nera/bianca, o se è vuota oppure se è un muro. Inoltre se è una pedina in quella cella verranno salvati colore, altezza e se è promossa o meno.

La struttura posizione viene utilizzata nella funzione bot per mantenersi in un array di pos tutte le r e c di tutte le pedine che si possono spostare.

Terminata la realizzazione della modalità 1vs1 tra due giocatori si è passati alla creazione del bot permettendo a un'utente di giocare contro il computer.

Per la realizzazione del bot abbiamo pensato a un algoritmo iterativo che va a vedere tutte le pedine possibili del nero, numero ottenuto con la funzione conta, e attraverso la funzione controllo si va a salvare soltanto quelle posizioni che hanno come valore ≥ 1 , così va a scartare qualsiasi pedina non spostabile, dopo si seleziona la pedina con il valore più alto (i valori corrispondono a 1 = movimento, 2 = mangiata, 3 = promozione) e si va a passare quella posizione alla funzione mangiata la quale proverà a spostare la pedina selezionata in "orologio", provando tutte le possibili vie finché non riesce a spostare la pedina.

Nonostante il tempo che abbiamo investito per poter debuggare e ottimizzare il più possibile questa funzione ci sono dei casi in cui va lo stesso in segmentation fault, non permettendo la fine della partita.

Per facilitare l'esperienza utente e per renderlo il più gradevole possibile è stata aggiunta una schermata con il menù iniziale.

Il menù è dotato di una schermata del titolo realizzato in ascii art, di impostazioni di gioco dalle quali è possibile modificare le regole di gioco a seconda delle preferenze dell'utente (come specificato sopra) e di una scelta tra le modalità di gioco player vs player e player vs PC, infine è possibile uscire dal gioco.

Organizzazione del lavoro

Il lavoro è stato svolto per la maggior parte a distanza ad eccezione di un primo incontro introduttivo in presenza. Gli incontri del gruppo sono stati utilizzati per confrontarsi e scambiarsi idee sull'andamento del lavoro, per poi dividersi i compiti di volta in volta in modo da gestire insieme il carico di lavoro.

Per poter sviluppare le funzioni di base del gioco e per accordarsi su cosa sviluppare nei momenti più avanzati del progetto i primi incontri si sono svolti in un intervallo di circa una settimana, stabilite le funzioni da realizzare gli incontri sono stati dedicati al risolvere i problemi che si potessero presentare.

Inoltre abbiamo utilizzato un file.h chiamato "lib.h" dove oltre inserire le dichiarazioni dei tipi strutturati matrice e pos abbiamo messo all'interno tutte le firme di funzione e cosa facevano, questo file è stato utile ogni tal volta che scambiavamo il progetto tra di noi si aveva un recap di cosa quella persona aveva fatto.

Difficoltà incontrate

La realizzazione del bot è stata difficoltosa, non solo dal punto di vista concettuale, ma anche dal punto di vista pratico, infatti l'ideazione e la scrittura del bot sono state le attività che hanno richiesto la maggiore attenzione da parte del gruppo oltre ad aver richiesto più tempo rispetto alla scrittura delle altre funzioni.

Un'altra difficoltà si è presentata con la funzione controllo, questa funzione viene utilizzata sia dal bot, che dall'obbligo di mangiata che per tutte le meccaniche di gioco principali. Per questo motivo la funzione ha presentato molti bug ed è stata modificata e corretta numerose volte.

La gran parte delle difficoltà minori sono state prevalentemente al livello concettuale.

L'approccio alla grafica da implementare ha portato ad una scelta più minimale per evitare di creare confusione nell'utente e per rendere l'osservazione della scacchiera più chiara possibile.

La scelta di fornire all'utente la possibilità di abilitare oppure no l'opzione dell'obbligo di mangiata rientra sempre nell'idea di rendere il più confortevole possibile l'esperienza di gioco, permette inoltre di rendere più varie le partite potendo modificare le impostazioni e così diversificare le partite tra loro.

L'ultima difficoltà l'abbiamo riscontrata scambiandoci il codice, infatti è stato sviluppato su Linux e Windows, abbiamo notato che il comando system(clear) non è compatibile con Windows, quindi il programma potrebbe perdere alcune funzionalità cambiando ambiente di sviluppo o sistema operativo.