**Progetto Mini Laska**

***Cos'è il progetto “Mini Laska”?***

Laska è un gioco da tavola formato da una scacchiera 7x7 e delle pedine che si muovono diagonalmente verso la sponda avversaria.   
Si utilizzano dei turni che permettono al giocatore di effettuare una sola mossa. Si può spostare la pedina diagonalmente di una cella (a patto che la cella sia libera) oppure si puo effettuare una sola mangiata. La mangiata avviene solo se si salta di due celle diagonalmente sopra una pedina del colore opposto.

Una pedina che raggiunge la sponda avversaria viene promossa dandole la possibilità di muoversi all’indietro.   
La partita finisce quando uno dei giocatori non ha pedine con mosse disponibili oppure non ha più pedine per poter giocare.

“Mini-Laska” è una variante del gioco originale “Lasca”.

Rispetto al gioco originale Mini-Laska prevede le seguenti limitazioni:

• si può mangiare/conquistare una sola volta per mossa;

• le torri possono essere alte al massimo 3 pedine, superato questo limite, la pedina più in basso viene rimossa dalla scacchiera.

***In cosa consiste il progetto?***

Il progetto consiste in una versione digitale del gioco spiegato di sopra. Ogni partita viene giocata contro un altro giocatore a turni. I comandi vengono inseriti tramite tastiera con il seguente format: righe-colonne senza spazi tra il dash e i numeri.  
Se le coordinate inserite non sono valide o non rispettano le regole del gioco, il programma stamperà sullo schermo l’errore/spiegazione commesso dal giocatore e chiederà di reinserire delle coordinate valide.

Il programma termina l’esecuzione se si seleziona il comando “Quit” dal menu principale oppure quando la partita finisce.

***Come lo abbiamo strutturato?***

Il programma inizia mostrando il menù, in cui può scegliere:

1. visualizzare le regole del gioco per poi tornare al menu,
2. scegliere la modalità di gioco, avente le seguenti opzioni:
   1. giocare contro un amico
   2. tornare al menu
3. terminare e chiudere il programma.

Una volta fatto partire il gioco inizierà il turno del PLAYER\_1, ovvero le pedine bianche, al quale verrà chiesto di mettere in input le coordinate della pedina che si vuole selezionare.

Inserendo delle coordinate valide verrà poi chiesto al giocatore di inserire le coordinate dove si vuole muovere la pedina. Se le coordinate sono valide la pedina verrà spostata alle coordinate indicate e il turno del PLAYER\_1 finirà e verrà passato al PLAYER\_2, ovvero le pedine nere. Il procedimento sopra indicato verrà ripetuto anche per il PLAYER\_2.  
Questo ciclo continua fino a che uno dei giocatori vince, ovvero quando uno dei giocatori non ha più mosse o pedine con cui giocare.

***Che funzione svolgono i vari file presenti nel progetto?***

* ***main.c*:** Vengono inizializzate le variabili necessarie per le coordinate e gli stati in cui si trova il programma. Contiene il ciclo principale con le chiamate alle funzioni necessarie per effettuare i controlli e le mosse.
* ***check\_move.c:*** contiene la funzione promotion() che gestisce la promozione delle pedine, e le funzioni *check\_selection(), mandatory\_selectable(), e check\_move()*, che controllano se le pedine sono effettivamente selezionabili , in che caselle si possono muovere e se hanno mangiate obbligatorie
* ***const.h:*** vengono definite le costanti usate in più parti del programma
* ***init.c:*** si trovano le funzioni che inizializzano le celle e le pedine
* ***list.c:*** si trovano le funzioni che gestiscono la creazione, il controllo, la modifica e l’eliminazione delle liste
* ***menu.c:*** si trovano le funzioni che gestiscono le opzioni del menu che si vede all’inizio del programma
* ***move.c:*** viene gestito lo spostamento e la mangiata delle pedine
* ***struct.h*** viene definita la struct *piece\_t* su cui il programma si basa
* ***pop\_array.c*** contiene la funzione pop\_front, che viene usata per rimuovere una pedina dalla cima di una torre quando questa viene mangiata
* ***print.c***sono contenute le funzioni necessarie per stampare la scacchiera e le pedine, ogni turno
* ***turno.c*** è contenuta la funzione *prossimo\_turno()*, che permette di cambiare il turno dei giocatori
* ***Doxyfile:*** file di configurazione per la nostra documentazione
* ***Documentazione Doxygen:*** la documentazione html è disponibile aprendo con il browser il file Doxygen\_Directory/html/index.html.

***Organizzazione del lavoro***

Abbiamo deciso di lavorare assieme tramite “Discord” per ragionare su come sviluppare il codice. Abbiamo suddiviso le tasks di alcuni lavori da fare per poi testarli assieme e sistemare eventuali errori/bugs.

**Dumitru Zotea**

main.c

pop\_array.c

init.c  
promotion()  
move()

check\_selection()

check\_move()  
mandatory\_selectable()

title()  
menu()const.h  
struct.h

**Filippo Zampiron**

main.c

print.c

turno.c

list.c

const.h

struct.h

Doxygen

**Aresù Sadeghi Gol**

main.c

menu.c

mandatory\_selectable()

const.h

struct.h

***Difficoltà riscontrate***

* All’inizio non sapevamo come fare la struct su cui abbiamo basato il progetto
* spesso abbiamo incontrato bug che abbiamo messo ore a risolvere, solo per trovare che la soluzione era semplice come un mistype.
* Abbiamo avuto un po’ di difficoltà a capire come fare le torri e inizialmente non sapevamo come organizzare le ore di lavoro sul progetto.