

Documento

Architetturale

*Chrome*

*Dinosaur*

*Chrome - Dinosau**r*

**Indice:**

* Spiegazione generale della struttura del progetto;
* Requisiti software, compilazione ed esecuzione;
* Per ogni file, spiegazione dell'organizzazione (quali funzioni) e loro descrizione testuale;
* Struttura delle funzioni e del flusso nel codice;
* Esecuzione codice lato utente ;
* Spiegazione del flusso di esecuzione indicando cosa fanno le varie sottosezioni e perché;
* Credits;

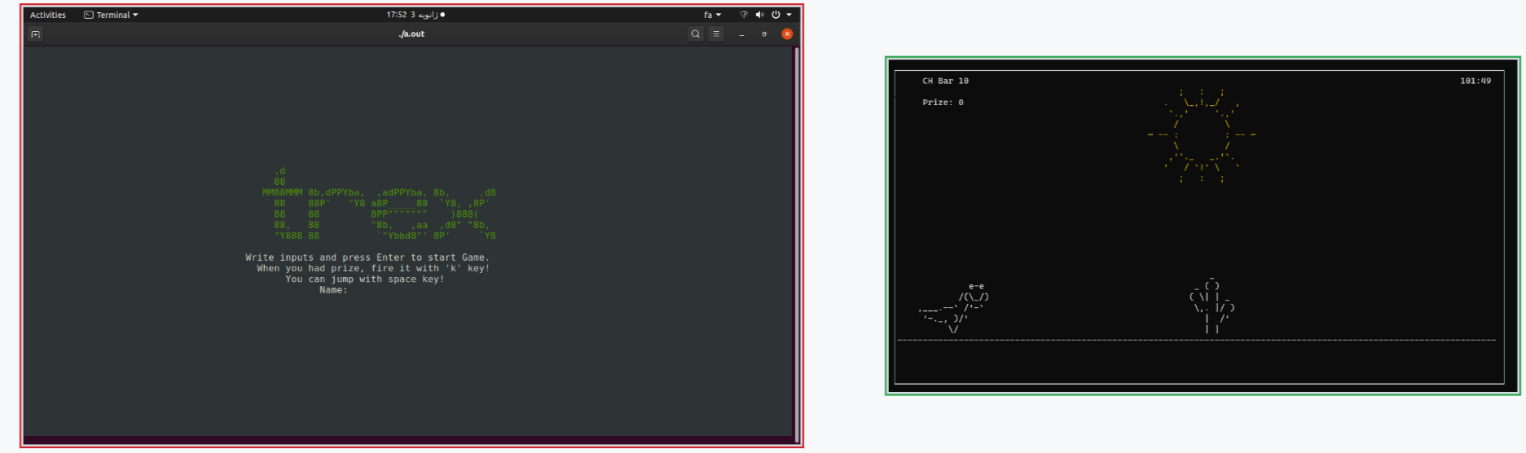
*Chrome - Dinosaur*

**Struttura generale del gioco:**

Il gioco contiene i seguenti file:

* Images:

Questo file contiene queste 2 mmagini



* .gitignore:

Questo file contiene:

dino.exe

\*.o

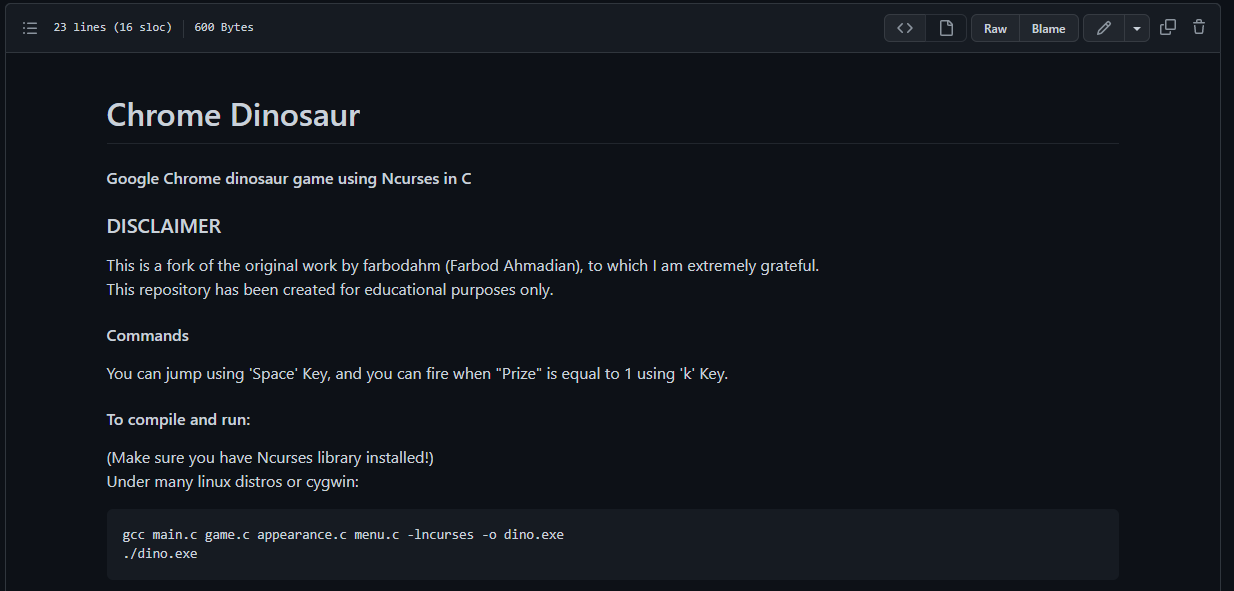
* LICENSE:

Questo file contiene: il Copyright, il Preambolo, i Termini e le condizioni e come applicarli ai nuovi programmi.

*Chrome - Dinosaur*

* README.md:

Questo file contiene un disclaimer, i vari comandi del gioco:



* appearance.c

Questo file contiene semplicemente tutti i simboli e lettere utilizzate per creare le seguenti immagini che compariranno su schermo

* appearance.h:

Questo file serve per applicare le cose fatte su appereance.c

* game.c:

Questo file contiene una parte di codice del gioco, è presente qui infatti il cuore del programma, ovvero la parte più importante del gioco.

* game.h:

Abbiamo del codice necessario per il gioco.

* highScore.txt:

Contiene il punteggio più alto raggiungibile.

* main.c:

Questo file contiene il codice del gioco con le varie chiamate alle numerose funzioni.

* menu.c:

Qua sono presenti le parti di codice che costituiscono il menu iniziale.

* menu.h:

Anche qui sono presenti delle parti necessarie per il menu.

* minunit.h:

go

**Requisiti software per compilazione ed esecuzione**

Per poter compilare ed eseguire il “gioco” bisogna essere sicuri di aver scaricato la libreria ncurses, e per avviare il gioco bisogna fare una fork su github sul proprio account, successivamente clonare il gioco su una repository all’interno del proprio pc, modificare la riga 26 del file menu.c e mettere prima di “Game” “the”, e  poi basta compilare il gioco e avviarlo e ci si può giocare.

**Descrizione di ogni file**

Il primo file che troviamo è:

**appearance.c**:

Questo file è quello che crea il dinosauro, i cactus, il sole, la luna e che fa apparire le scritte quando perdi, per ogni momento del gioco vi è una funzione che rappresenta ciò che viene chiamato dal main.

**game.c:**

Questo è file principale quello che fa funzionare il gioco, all’interno di esso vi sono la funzione che velocizza il gioco, quella che sceglie quale dinosauro stampare, quella che assegna i premi al giocatore , e anche il “motore” del gioco ovvero quella funzione che è alla base del gioco, quella chiama il resto delle funzioni, scegli se è giorno o notte, fa vedere i dati e fa il resto dei calcoli affinché il gioco funzioni.

Dopo di esse vi è una sezione che fa i test prima di giocare  per assicurarsi che il gioco funzioni.

**main.c:**

Qui vi è il main da cui parte tutto che avvia i test che a loro volta se andranno bene avvieranno il motore del gioco.

**menu.c:**

Questo file è quello che si occupa del menu di gioco ovvero tiene lo score più alto, ti permette di ricominciare o di chiudere il gioco o anche di salvare.

**Come sono suddivise le diverse funzioni e quali altre contengono**:

* Run\_all\_tests
* Startmenu(initpair,attron,showtrex,attroff,mvprintw,getstr,noecho,startengine);
* endGame(init\_pair,nodelay,attron,showloss,mvprintw,getch,attroff,startengine)
* dinasaur1
* dinosaur2
* cactus1
* cactus2
* sun
* moon
* showloss
* showtrex
* clearcactus1
* cleardinosaurdown
* cleardinosaurup
* checkgame
* computertime
* showdinosaur(dinosaur1, dinosaur2)
* computerprize
* startengine(getmaxx,init\_pair,getch,moon,attron,sun,mvprintw,computerprize,clearcactus1,box,cleardinosaurup,showdinosaur,cleardinosaurdown,endgame)

**ESECUZIONE CODICE UTENTE**

Generalmente il percorso è sempre molto simile come ogni volta che avviamo il gioco dall’inizio, prima

verranno fatti dei controlli partendo sempre dalla funzione “run\_all\_tests” poi verrà eseguito startmenu

per costruire sullo schermo il menu e poi inizierà il vero e proprio gioco tramite le varie funzioni grafiche

come dinosaur1,cactus1,sun ecc….

**SPIEGAZIONE FUNZIONI COMPLESSE:**

Nel codice di questo gioco sono presenti numerose funzioni, da quelle che generano i vari elementi estetici, come il sole, la luna, il dinosauro stesso e i vari ostacoli.

Ma c’è una funzione che permette il funzionamento del gioco ed è abbastanza complessa, è una procedura chiamata *StartEngine* scritta nel file *game.c.*

In questa possiamo notare numerose variabili che però permettono il funzionamento del gioco: ci sono dei passaggi che definiscono il colore dei cactus, del sole e della luna; altre operazioni permettono il susseguirsi del giorno e della notte; la possibilità di vedere il proprio nome scritto all’inizio e ovviamente quello che permette al dinosauro di saltare. Qui di seguito alcune parti della procedura *StartEngine.*

*// Qui inizia la procedura*

*void startEngine(int highScore, struct user firstUser) {*

*srand(time(NULL));*

*int x, y, diX=5, prize=0, usedPrize=0, score=0, delayTime = 300000*

*, gameStatus=1, cactusNum=0;*

*int maxX=getmaxx(stdscr);*

*x = maxX-20;*

*y = getmaxy(stdscr)-6;*

*int diY = y;*

*int arrowX=(diX+15), arrowY=(diY-3);*

*char userInput;*

*int jumping=-1;*

*bool fire=FALSE;*

*clear();*

*nodelay(stdscr, TRUE);*

*init\_pair(1,COLOR\_WHITE,COLOR\_BLACK);*

*init\_pair(4,COLOR\_BLUE,COLOR\_BLACK);*

*init\_pair(5,COLOR\_GREEN,COLOR\_BLACK);*

*init\_pair(6,COLOR\_YELLOW,COLOR\_BLACK);*

*while (gameStatus == 1) {*

*userInput = getch();*

*// Show day or night*

*if((score/50)%2 != 0) {*

*moon(10, (maxX/2)-10);*

*}*

*else {*

*attron(COLOR\_PAIR(1));*

*sun(10, (maxX/2)-10);*

*}*

*// clear arrow*

*if (fire) {*

*mvprintw(arrowY, arrowX-2, " ");*

*}*

*score++;*

*// Qua vengono mostrate a schermo le informzaioni inserite all’inizio*

*mvprintw(1, 6, "%s %s %s", firstUser.name, firstUser.lastName, firstUser.age);*

*mvprintw(1, getmaxx(stdscr)-9, "%d:%d", highScore, score);*

*// Use prize to destroy cactus*

*prize = computePrize(score, usedPrize);*

*mvprintw(3, 6, "Prize: %d    ", prize);*

*if (prize == 1) {*

*if (userInput == 'k') {*

*usedPrize++;*

*fire = TRUE;*

*}*

*}*

*if (fire) {*

*mvprintw(arrowY, arrowX, "\*");*

*arrowX += 2;*

*}*

*if ((x+4)-arrowX <= 1 && fire) {*

*clearCactus1(y, x-1);*

*mvprintw(arrowY, arrowX-2, " ");*

*x = getmaxx(stdscr)-20;*

*fire = FALSE;*

*arrowX = diX+15;*

*}*

*// ----------*

*box(stdscr, ACS\_VLINE, ACS\_HLINE);*

*//for clearing screen*

*cleardinosaurUp(diY, diX);*

*if (x <= 7) {*

*x = getmaxx(stdscr)-20;*

*cactusNum = rand() % 2;*

*}*

*// Se il comando del giocatore corrisponde allo spazio allora il dinosauro salterà.*

*if (userInput == ' ' && jumping<0) {*

*diY -= 7;*

*jumping = 3;*

*}*

*showdinosaur(diY, diX);*

*if (userInput == ' ') {*

*cleardinosaurDown(diY, diX);*

*}*

*if (x-diX <= 7) {*

*x -= 10;*

*}*

*if (cactusNum == 0) {*

*cactus1(y, x);*

*}*

*else {*

*cactus2(y, x);*

*}*

*if (x-diX <= 7) {*

*x += 10;*

*}*

*gameStatus = checkGame(y, x, diY, diX);*

*// Il dinosauro verrà portato all’inizio in caso di sconfitta*

*jumping--;*

*if (jumping==0) {*

*diY += 7;*

*}*

*mvhline(y+1, 1, '-', getmaxx(stdscr)-3);*

*refresh();*

*clearCactus1(y, x);*

*refresh();*

*usleep(delayTime);*

*x -= 7;*

*delayTime = computeTime(delayTime);*

*userInput = 'q';*

*}*

*endGame(score, highScore, diY, diX, firstUser);*

*attroff(COLOR\_PAIR(1));*

*}*

Come si può notare questa procedura è ricca di variabili ed è abbastanza complessa, però compone prevalentemente tutto il cuore del gioco chrome-dinosaur.

**CREDITS:**

Cordaro Matteo, Camandona Lorenzo, Sgarzi Mauro e Suteu Lorenzo.