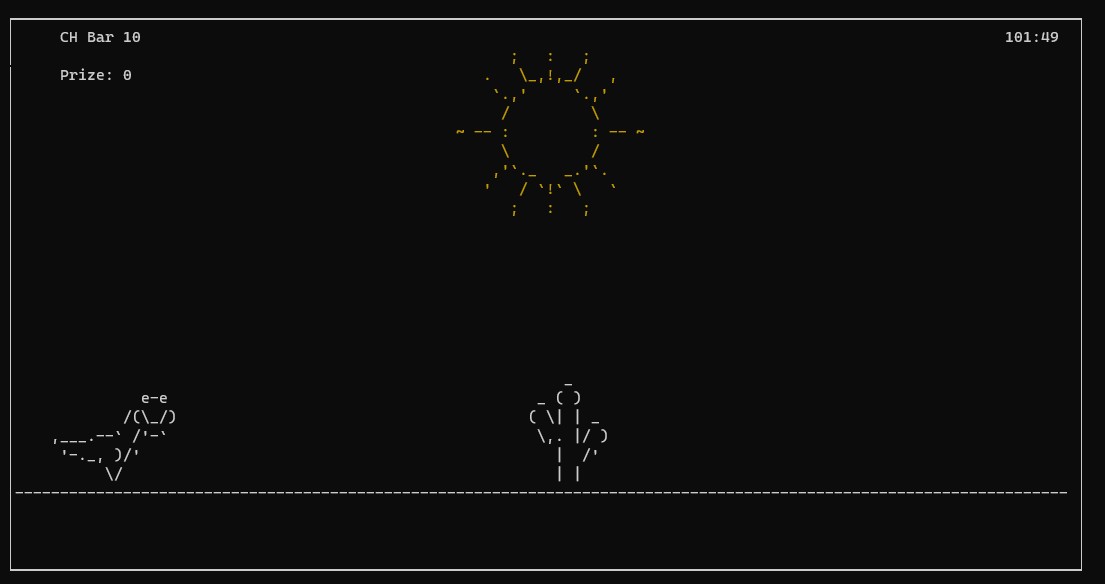
**Software System Architecture**

*Chrome Dinosaur*



*INTRODUZIONE AL DOCUMENTO*

In questo documento illustreremo:

* la struttura del progetto (quali file, cosa contengono in generale)
* Requisiti software
* organizzazione dei seguenti file (quali funzioni) e loro descrizione testuale
* Struttura delle funzioni e del flusso nel codice
* Esecuzione codice lato utente
* spiegazione del flusso di esecuzione delle funzioni più complesse
* Credits

*STRUTTURA DEL PROGETTO*

File principali:

*Appearance.c* : contiene la forma del dinosauro nelle sue varie forme che puo’ assumere in base ai comandi che vengono impartiti; le forme vengono create usando simboli presenti sulla tastiera come punteggiatura, parentesi e altri simboli disponibili su C.

Il codice presente in *Appearance.c* e le funzioni al suo interno possono essere richiamate inserendo nel file che necessita di queste parti di codice la libreria associata di nome *Appearance.h,* senza la quale le funzioni non sarebbero richiamabili se non nel file stesso.

*Game.c* : permette al gioco di girare, proseguire, e ne “detta i tempi”, anch’esso possiede una libreria associata chiamata *Game.h* verra’ poi utilizzata nel *main.c.*

*Menu.c* : serve a creare appunto un menu’ dove ci saranno i dati del high score e dove si potra’ attendere prima di iniziare la partita. Anche questo file possiede una libreria associata *Menu.h* per poter richiamare le funzioni al suo interno da altri file.

*highScore.txt* : semplice file di testo dove viene memorizzato il punteggio piu’ alto raggiunto e viene modificato se viene superato.

*Main.c :* e’ il file principale da cui parte tutto il programma, contiene infatti tutte le librerie dei file sopracitati, ed e’ il punto di partenza dove vengono richiamate funzioni da tutti i file a cascata.

*REQUISITI DEL SISTEMA*

per il funzionamento del seguente codice ci sono dei requisiti da rispettare come:

SO

Windows,MAC OS, Linux

Processore

Pentium a 90 MHz o più veloce

Memoria

16 MB

Storage

3 GB available space

Un IDE (codelite, VSC…) in grado di compilare il codice

Le varie librerie che servono ad aggiungere le operazioni utili al codice

Il prompt dei comandi utile a visionare il risultato finale

*ORGANIZZAZIONE DEI FILE*

Appearance.c : Sono presenti 11 funzioni, con combinazioni di caratteri, organizzate in modo da creare, forme e scritte, verranno poi richiamate per realizzare l’interfaccia.

Game.c : richiama diverse librerie standard oltre che le librerie interne del progetto, quali: *Appearance.h, Game.h, menu.h e minuit.h.*

Contiene poi 9 funzioni che servono rispettivamente a:

1. checkGame: Controlla se il gioco e’ finito in quel ciclo di aggiornamento.
2. computerTime: Rende il gioco piu’ veloce e ne detta i tempi
3. showdinosaur: Verifica quale tipo di dinosauro deve essere stampato a schermo
4. computerPrize: conta i punti per l’high score.
5. startEngine: riceve gli input, e manda gli output di conseguenza, una volta richiamata, il gioco parte, chiama la fine del gioco una volta terminata l’esecuzione.
6. test\_foo, test\_bar, all\_tests, run\_all\_tests: test per verificare che il gioco funzioni correttamente.

Menu.c : richiama le librerie *menu.h*, *game.h* e *appearance*, e contiene 2 funzioni:

1. startMenu : crea il menu’ da mostrare all’apertura del gioco, dove vengono richiesti, nome, cognome ed eta’.
2. endGame : crea il menu’ da mostrare alla fine della partita con i punteggi.

Main.c : richiama le librerie *menu.h e minuit.h* oltre ad altre librerie standard. Contiene solo la funzione main che lancia il gioco.

*STRUTTURA DEL FLUSSO DEL CODICE*

File di partenza:

main.c

↓

Richiama delle funzioni presenti in:

menu.c e minuit.h

↓

In menu.c vengono chiamate a sua volta funzioni da:

game.c e appearance.c

↓

Anche game.c richiama funzioni dagli altri file, menu.h, appearance.h, minuit.h.

↓

appearance.c invece non richiama nessuna funzione.

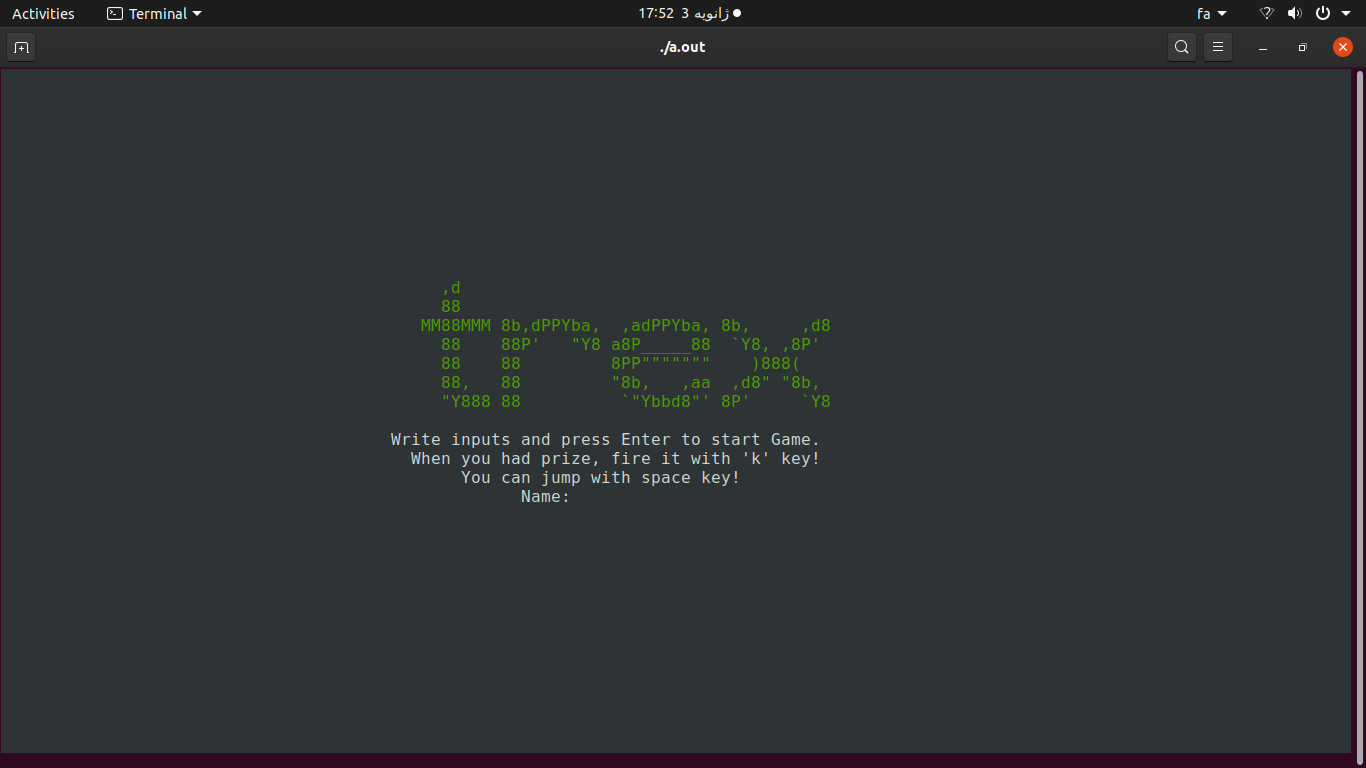
*ESECUZIONE CODICE LATO UTENTE*

appena avviato il codice, l'utente avrà davanti una schermata con scritto i vari comandi da usare cioè:

“k” per sparare

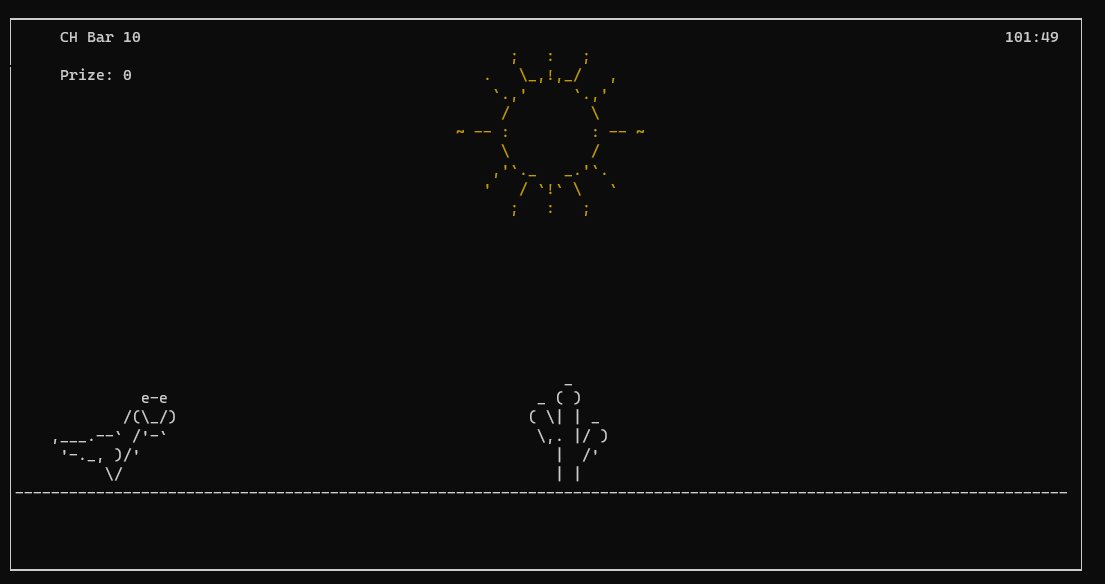
“space” per saltare

le istruzioni per poter entrare nel gioco e la possibilità di scrivere il proprio nome.



dopo aver seguito le varie istruzioni per entrare nel gioco, all'utente apparirà la seguente schermata di gioco dove possiamo trovare il nostro dinosauro contornato da un sole e i vari ostacoli che dovrà evitare per non perdere, e se notiamo bene in alto a sinistra

possiamo anche trovare il punteggio fatto in sessione



*FUNZIONI COMPLESSE*

*Game.c →* startEngine: riceve i dati dell’utente e l’high score, crea successivamente la pagina che puo’ essere sia giorno che notte, riceve gli input dall’utente, gestisce i movimenti del dinosauro che il programma deve mostrare in relazione all’input ricevuto.

*Menu.c →* startMenu: richiede all’utente nome, cognome e eta’, prende il valore di high score e li mostra, in attesa dell’input per far iniziare l’esecuzione del gioco.

*Menu.c →* endGame: prende il punteggio raggiunto, lo inserisce come high score se e’ maggiore dell’high score precedente, mostra inoltre questi dati all’utente.

*CREDITS*

*scritto da:*

*Giorgio Vacchiotti e Mario Di Benedetto*

*con la collaborazione di:*

*Matteo Benini e Clara Montanaro*