CHROME DINOSAUR

Immagine che contiene neve, LEGO, esterni, giocattolo

Descrizione generata automaticamenteSoftware System Architect

INTRODUZIONE

*Nel documento che segue si andrà a spiegare il funzionamento del gioco dinosaur Chrome, ovvero la parte del codice C, tutti i diversi file presenti nel repository e delle varie funzioni utilizzate per creare il gioco.*

STRUTTURA GENERALE DEL PROGETTO

Appearance.c

*Dentro a questo file vengono scritte tutte le funzioni con le figure che appaiono nel gioco ad esempio nella prima parte del codice vediamo una funzione “void dinosaur1” con dentro 5 “mvprintw” che vanno a simboleggiare il corpo del dinosauro che apparirà in gioco, e così via per le altre figure. (In seguito il file verrà descritto più dettagliatamente)*

*Infine, tutte le funzioni scritte dentro questo file saranno richiamate dentro il file “appearance.h”. (librerie che verranno utilizzate in altri file .c)*

Menu.c

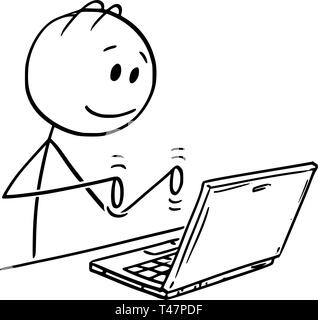
*In questo file viene scritto il codice dei diversi menu presenti nel gioco, come quello iniziale in cui vengono spiegati brevemente i tasti e i dati da inserire per cominciare a giocare, ma anche il menu dopo aver perso in cui viene chiesto se ricominciare oppure uscire dal gioco*

Game.c

*In questo file viene scritto il codice di come funziona il gioco, ovvero a quanti punti vengono dati i premi oppure quando la velocità deve aumentare o ancora quando fare il giorno o quando fare la notte e così via. (In seguito il file verrà descritto più dettagliatamente)*

*Infine, le funzioni “startEngine” e “showdinosaur” scritte dentro questo file saranno richiamate dentro il file “game.h”. (librerie che verranno utilizzate in altri file .c)*

REQUISITI SOFTWARE



*Per poter utilizzare dinosaur Chrome sul proprio dispositivo c’è bisogno di installare alcune cose:*

1. *Installare la libreria ncurses che servirà per disegnare nella shell*

*. sudo apt-get install libncurses5-dev libncursesw5-dev*

1. *Installare doxygen*

*. sudo apt-get install doxygen*

1. *Installare git*

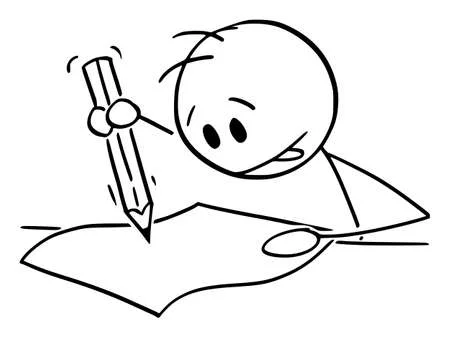
*. sudo apt-get install git-all*

1. *Installare gcc*

*. sudo apt-get install build-essential*

DESCRIZIONE DEI FILE

APPEARANCE.C



Void dinosaur1

*Nella prima funzione troviamo 5 “mvprintw” che rappresentano il primo stato del dinosauro*

e-e

/(\\_/)

,\_\_.—’ /'-‘

'-.\_, )/'

\/

Void dinosaur2

*Nella seconda funzione troviamo altri 5 “mvprintw” che rappresentano il secondo stato del dinosauro*

e-e

/(\\_/)

,\_\_.—’ /'-‘

'-.\_, )/'

/\

*Come si può osservare nei due stati cambia la posizione delle gambe così che successivamente durante il gioco il dinosauro sarà più dinamico muovendole*

Void cactus1

*Nella terza funzione troviamo il codice per la rappresentazione del cactus con 6 “mvprintw”*

\_

\_ ( )

( \| | \_

\,. |/ )

| /'

| |

Void cactus2

*Nella quarta funzione troviamo il codice per la rappresentazione del cactus con l’utilizzo di 7 “mvprintw” e anche grazie all’utilizzo del “init\_pair”, “attroff” e “attron” abbiamo dato al cactus la colorazione verde*

\_

( ) \_

\_| |/ )

( \ /'

\ |

| |

| |

Void sun

*Nella quinta funzione troviamo il codice per la rappresentazione del sole con l’utilizzo di 9 “mvprintw” e anche grazie all’utilizzo del “attroff” e “attron” abbiamo dato al sole la colorazione gialla*

. \\_,!,\_/ ,

`.,' `.,'

, " / \

~ -- : : -- ~

\ /

,'`.\_ \_.'`.

' / `!` \ `

; : ;

Void moon

*Nella sesta funzione troviamo il codice per la rappresentazione della luna con l’utilizzo di 9 “mvprintw” e anche grazie all’utilizzo del “attroff” e “attron” abbiamo dato alla luna la colorazione blu*

\_..\_

.' .-'`

/ /

| |

\ \

'.\_'-.\_

```

Void showLoss

*Nella settima funzione troviamo il codice per la rappresentazione della schermata che apparirà in caso di sconfitta con l’utilizzo di 10 “mvprintw”*

88

88

88

88 ,adPPYba, ,adPPYba, ,adPPYba,

88 a8" "8a I8[ "" I8[ ""

88 8b d8 `"Y8ba, `"Y8ba,

88 "8a, ,a8" aa ]8I aa ]8I

88 `"YbbdP"' `"YbbdP"' `"YbbdP"'

Press 'r' to play again!

Or 'q' to exit from game.

Void showTrex

*Nell’ottava funzione troviamo il codice per la rappresentazione della schermata che apparirà nel menu principale con l’utilizzo di 7 “mvprintw”*

,d

88

MM88MMM 8b,dPPYba, ,adPPYba, 8b, ,d8

88 88P' "Y8 a8P\_\_\_\_\_88 `Y8, ,8P'

88 88 8PP""""""" )888(

88, 88 "8b, ,aa ,d8" "8b,

"Y888 88 `"Ybbd8"'8P' `Y8

Void clearCactus1, Void cleardinosaurDown e cleardinosaurUp

*Infine nella nona, decima e undicesima funzione troviamo il codice per cancellare le figure durante il gioco, infatti nella prima troviamo come cancellare i cactus nella seconda come cancellare il dinosauro mentre nell’ultimo come cancellare il dinosauro quando salta*

**DESCRIZIONE DEI FILE**

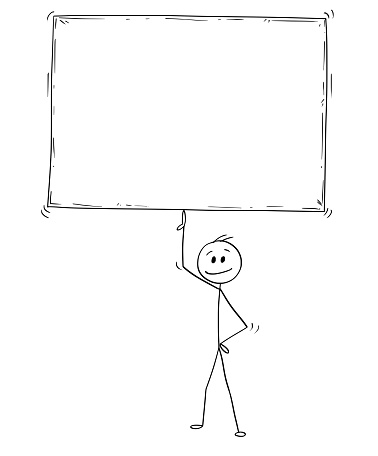
APPEARANCE.H



*Dopo aver scritto tutte le funzioni dentro al file appearance.c andiamo a richiamarle tutte dentro al file omonimo .h per creare una libreria che richiameremo successivamente in altri file*

**DESCRIZIONE DEI FILE**

MENU.C

****

*Partendo dall’ inizio andiamo a vedere le librerie usate per questo file, possiamo vedere che sono state richiamate 3 librerie di quelle create da noi ovvero: appearance.h (pag. 4-5-6-7), menu.h (pag. 8-9) e game.h (pag. 10-11-12-13).*

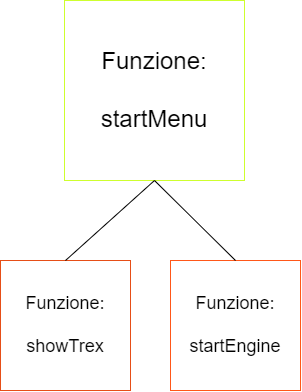
Void startMenu

*Nella prima funzione vediamo il codice per come impostare il menu principale e per gestire il file per il maggior punteggio, come prima cosa inizializziamo una struct e il punteggio maggiore, in seguito apriamo un file con il nome “highScore.txt” e con una “fopen” ci entriamo in modalità di lettura e con una ”fscanf” andiamo a leggere il punteggio maggiore.*

*In seguito andiamo a richiamare una funzione presente nella libreria “appearance.h” ovvero “showTrex” (riferimento a pag. 5) per stampare a video la scritta del menu principale e grazie ad “attron” e “attroff” gli diamo il colore verde.*

*In seguito, utilizziamo le “mvprintw” per richiedere all’utente diversi parametri come nome, cognome ed età che saranno inseriti nella “getstr”.*

*Infine, richiamiamo la funzione della libreria “game.h” ossia “startEngine” (riferimento a pag. 10) ovvero il nucleo del gioco.*

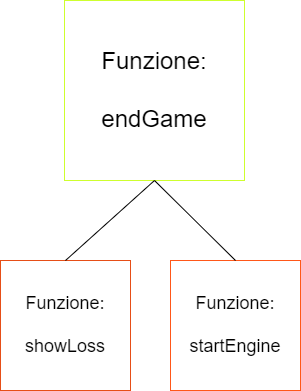


Void endGame

*Nella seconda funzione troviamo un “init\_pair” che servirà in seguito per utilizzare il colore rosso, successivamente troviamo un “if” che se il punteggio attuale è maggiore del punteggio più alto si entra dentro la condizione e si apre il file “highscore.txt” in modalità scrittura per cambiarne il valore.*

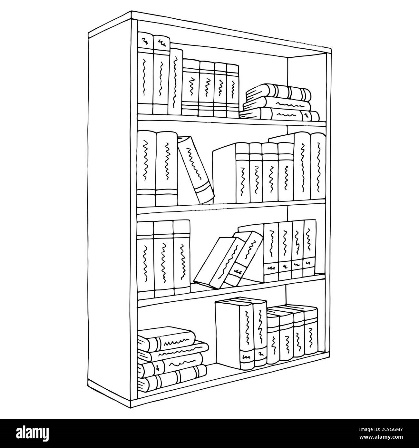
*In seguito, andiamo a richiamare una funzione presente dentro la libreria “appearance.h” ovvero “showLoss” (riferimento a pag. 5) per stampare a video la scritta della sconfitta e grazie ad “attron” e “attroff” gli diamo il colore rosso.*

*Infine, troviamo un altro “if” che se viene schiacciata la lettera “r” il gioco ricomincerà grazie al richiamo della funzione “startEngine” altrimenti se viene schiacciata la lettera “q” si uscirà dal gioco*



**DESCRIZIONE DEI FILE**

MENU.H



*Le due funzioni scritte nel file “menu.c” andremo a richiamarle nella libreria “menu.h”, inoltre dentro a questo file andremo ad impostare la struct “user” con la lunghezza massima di stringa del nome, cognome ed età.*

DESCRIZIONE DEI FILE

GAME.C

**Immagine che contiene testo, arma

Descrizione generata automaticamente**

*Partendo dall’ inizio andiamo a vedere le librerie usate per questo file, possiamo vedere che sono state richiamate 4 librerie di quelle create da noi ovvero: appearance.h (pag. 4-5-6-7), menu.h (pag. 8-9), game.h (pag. 10-11-12-13) e minunit.h*

int checkGame

*In questa funzione possiamo osservare il codice per cui capire se il gioco è terminato, ovvero far capire al programma se l’utente ha perso*

int computeTime

*Nella seconda funzione vediamo il codice per come far aumentare la velocità del gioco, troviamo un “if” in cui se il “delayTime” è maggiore di 250000 allora bisogna farlo diminuire di 1000 e così via, finché non arriva ad essere il “delayTime” minore di 250000 ma maggiore di 20000 a quel punto bisognerà farlo diminuire di 600 invece che di 1000, infine se il “delayTime” sarà minore di 2000 bisognerà farlo diminuire di 200 invece che di 600.*

*Alla fine di questa funzione il valore restituito sarà il “delayTime”.*

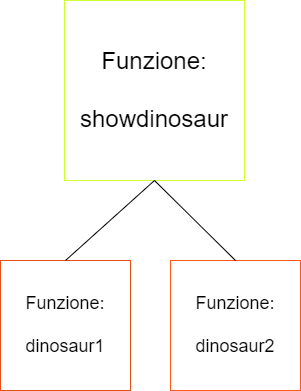
*(delayTime (di base) = 300000)*

void showdinosaur

*Nella terza funzione vediamo il codice per come far muovere in due modi differenti il dinosauro, come prima cosa verrà inizializzato un counter = 0, in seguito è posto un “if” in cui se il*

*counter = 0 il dinosauro avrà lo stato della funzione “dinosaur1” presente dentro la libreria “appearance.h” (pag. 3) e il counter verrà incrementato di 1, in seguito la funzione verrà ripetuta in loop ma in questo caso il valore del counter sarà diverso da 0 quindi entrerà nel else e il dinosauro avrà lo stato della funzione “dinosaur2” presente dentro la libreria “appearance.h” (pag. 3) e il counter verrà decrementato di 1.*

*Tutto ciò verrà ripetuto in loop per far sembrare che le gambe del dinosauro si muovano.*

**

void computePrize

*Nella quarta funzione vediamo il codice per dare il prize all’utente che sta giocando, possiamo subito vedere 3 “if” che chiedono tre valori per poter dare il prize*

1. *se il punteggio attuale è maggiore di 20, se il punteggio attuale è minore di 40 e se il usedprize è uguale a 0, se tutti e tre i sono veri allora il programma darà il prize uguale a 1.*
2. *se il punteggio attuale è maggiore di 60, se il punteggio attuale è minore di 80 e se il usedprize è uguale o minore di 1, se tutti e tre i sono veri allora il programma darà il prize uguale a 1.*
3. *se il punteggio attuale è maggiore di 100, se il punteggio attuale è minore di 120 e se il usedprize è uguale o minore di 2, se tutti e tre i sono veri allora il programma darà il prize uguale a 1.*
4. *In tutti gli altri casi il valore del prize sarà uguale a 0.*

void startEngine

*Nella quinta funzione troviamo tutto il codice per far funzionare il gioco, possiamo vedere che abbiamo inizializzato la maggior parte dei dati che ci serviranno in seguito all’inizio del file come, ad esempio, il delayTime=300000 oppure il fire messo a False ecc…*

*Successivamente se il valore di “gameStatus” è uguale a 1 possiamo entrare dentro al while e vediamo subito la funzione “getc()” che serve per ricevere i dati in ascii presi dalla tastiera, in seguito vediamo un “if” che se il punteggio attuale diviso per 50 modulato a 2 lascia un resto diverso da 0 farà vedere sullo schermo la luna grazie alla funzione “moon” presa dalla libreria “appearance.h” altrimenti mostrerà il sole grazie alla funzione “sun” presa dalla libreria “appearance.h”.*

*Continuando nel codice troviamo due “mvprintw” che mostrano sullo schermo il nome, cognome, età, punteggio massimo e punteggio attuale.*

*In seguito, troviamo la funzione “computePrize” descritta in precedenza e un “mvprintw” che stamperà a schermo il valore del prize continuando troviamo un if che se il premio è uguale a 1 e l’utente sta cliccando la lettera “k” usedPrize aumenterà di 1 e la variabile fire diventerà TRUE, con l’utilizzo di altri “if” e della funzione “clearCactus1” presa dalla libreria “appearance.h” si vedrà il prize utilizzato “\*” e il cactus che scomparirà.*

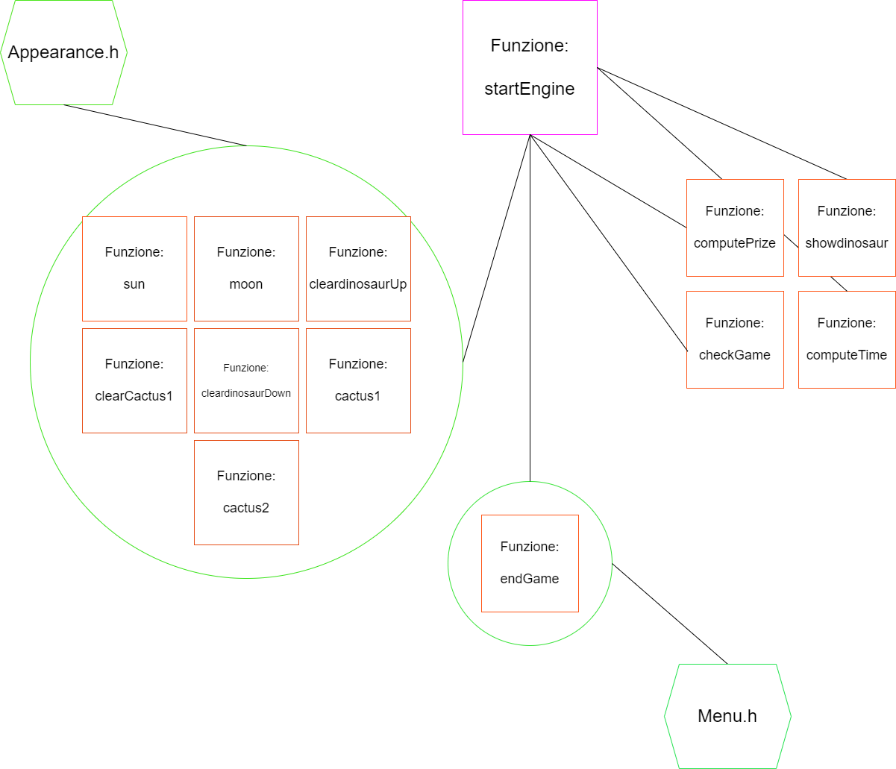
*Successivamente mettiamo una “rand” che restituisca due possibili valori 0 o 1 che ci serviranno più tardi, in seguito controlliamo se è stato premuto lo spazio dall’utente, se è stato premuto e la variabile jumping è minore di 0 allora andiamo ad assegnare a “jumping” il valore 3 e al “diY” il valore stesso – 7 e in seguito richiamiamo la funzione “showdinosaur” e in seguito la funzione “cleardinosaurDown”, altrimenti se è solo stato premuto spazio ma la variabile “jumping” è maggiore o uguale a 0 prima richiamiamo la funzione “showdinosaur” e poi richiamiamo la funzione “cleardinosaurDown”.*

*In seguito riprendiamo il valore generato dalla “rand” e se è uguale a 0 verrà generato il cactus base grazie alla chiamata della funzione “cactus1” altrimenti verrà generato il cactus verde generato grazie alla chiamata della funzione “cactus2”.*

*Infine, si controlla se si ha perso grazie alla funzione “checkGame” e si aumenta il “delayTime” grazie alla funzione “computeTime”*

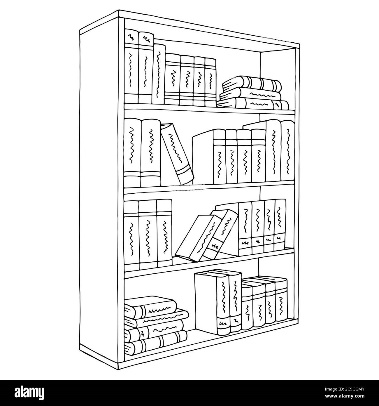
*Finito tutto ciò ricomincia dal while e se il “checkGame” ha restituito 0 si esce dal while e verrà richiamata la funzione “endGame”*

SCHEMA SULLA FUNZIONE STARTENGINE, CON TUTTE LE FUNZIONI UTILIZZATE E LE RISPETTIVE LIBRERIE

**

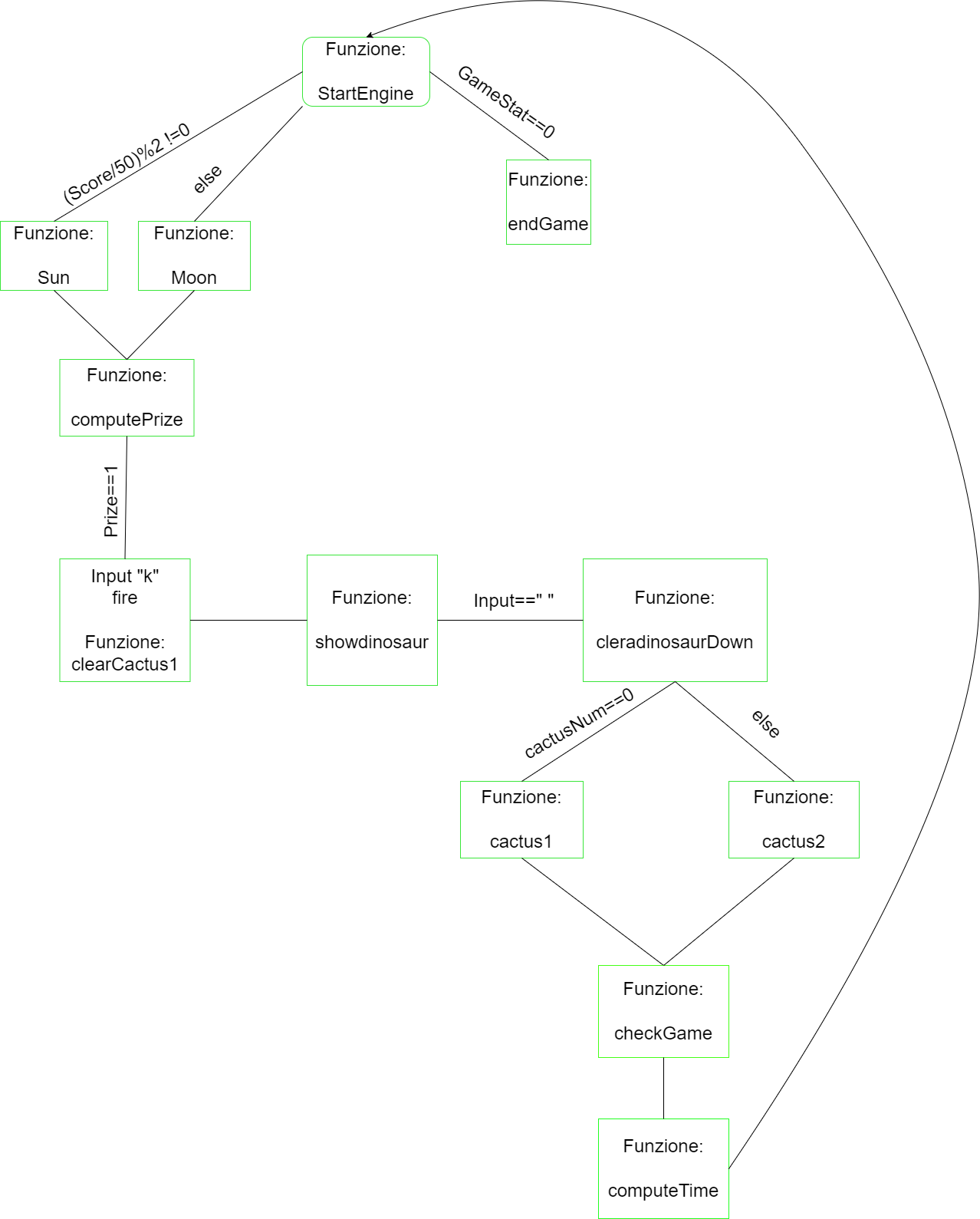
**DESCRIZIONE DEI FILE**

GAME.H



*Dentro al file “game.h” andremo a richiamare la funzione “showdinosaur” e la funzione “startEngine” presenti nel file “game.c”*

POSSIBILE PERCORSO DELL’UTENTE NEL CODICE



*Documento architetturale realizzato da:*

*Lorenzo Boggiatto, Marianny Javier, Alessandro Petrillo & Carlo Pognant*