Documentatie

Snake project

**1.Descriere Generală**

Acest proiect reprezintă o implementare mai complexa a jocului clasic Snake, folosind biblioteca grafică SDL2 în limbajul C. Scopul jocului este de a controla un șarpe care se deplasează pe ecran, colectând diverse tipuri de mâncare și evitând coliziunile. Jocul introduce elemente dinamice precum pereți temporari, fructe rare și bonusuri de scor, aducând un plus de dificultate și varietate.

**2.Funcționalități**

**Controlul Șarpelui:**  
 Mișcarea șarpelui este realizată prin tastele W, A, S, D.

**Generarea Aleatoare a Mâncării:**  
 Mâncarea apare în poziții aleatorii pe ecran, fără a se suprapune cu șarpele sau pereții.

**Tipuri de Mâncare:**

**Fructe normale** – cresc dimensiunea șarpelui și oferă 1 punct.

**Fructe care micșorează** – reduc lungimea șarpelui.

**Fructe rare** – oferă +5 puncte, dar nu influențează dimensiunea.

**Creșterea și micșorarea șarpelui:**  
 Șarpele își modifică dimensiunea în funcție de tipul fructului consumat.

**Coliziuni:**

-Coliziunea cu marginea ferestrei sau cu propriul corp duce la terminarea jocului.

-Coliziunea cu pereți temporari este, de asemenea, fatală.

**Scor și High Score:**

Scorul curent este afișat pe ecran.

La finalul fiecărui joc, scorul maxim (high score) este salvat și afișat automat.

**Pereți Temporari:**  
 Apar și dispar periodic, adăugând dificultate suplimentară jocului.

**Bonusuri Speciale de Scor:**

Dacă jucătorul mănâncă 3 fructe în mai puțin de 5 secunde, primește +3 puncte bonus.

Fiecare fruct rar oferă automat +5 puncte.

**Interfață Grafică cu SDL2:**

Fundal verde și elemente desenate în timp real (șarpe, mâncare, pereți, text).

Fonturi personalizate cu SDL\_ttf.

1. **Biblioteci Incluse și Define-uri**

**include <SDL2/SDL.h> -** operațiile grafice și de input/output din joc

**#include <SDL2/SDL\_ttf.h> -** încărcarea și afișarea fonturilor (TTF).

**#include <stdio.h> -** Biblioteca standard C pentru input/output

**#include <stdlib.h> -** Biblioteca standard C pentru funcții utilitare, cum ar fi alocarea memoriei, conversii și generare numere aleatoare

**#include <windows.h> -** pentru a interacționa direct cu sistemul de operare Windows.

**#include <conio.h> -** funcții pentru input/output în consolă

**#include <time.h> -** gestionarea timpului și a funcțiilor legate de timp

**#define rows 20 -** numărul de **rânduri**

**#define cols 30 -** numărul de **coloane**

**#define MAX\_SIZE 256 -** dimensiunea maximă a șarpelui

**#define foods 10 -** numărul maxim de **alimente**

**#define MAX\_TEMP\_WALLS 20 -** numărul maxim de **pereți temporar**

**#define WINDOW\_WIDTH 600 -** Lățimea ferestrei grafice

**#define WINDOW\_HEIGHT 400 -** Înălțimea ferestrei grafice

**#define TILE\_SIZE 20 -** Dimensiunea fiecărei celule

**4.Funcții**

void clear\_screen(); - Șterge ecranul

void build\_board(); - Construiește tabla de joc

void print\_board(); - Afișează tabla de joc în consolă

void draw\_snake(); - Desenează șarpele pe ecran, cu capul și corpul în culori diferit

void move\_snake(int deltaX, int deltaY); - Mută șarpele în direcția indicată (deltaX, deltaY). Capul avansează și corpul îl urmează.

void read\_keyboard(); - Citește intrarea de la tastatură (W, A, S, D) pentru a schimba direcția șarpelui

void draw\_food(); -Desenează toate tipurile de mâncare pe tabla de joc (măr, simbol $ pentru puncte bonus, bombă).void setup\_food(); - Generează poziții aleatorii pentru mâncare pe tabla de joc, evitând șarpele.

void setup\_snake(); - Inițializează șarpele cu o lungime și poziție de start.

void check\_self\_collision(); - Verifică dacă șarpele s-a lovit de corpul său.

void game\_rules(); - Controlează regulile jocului: detectează coliziuni, mâncarea consumată, actualizează scorul, regenera mâncarea, verifică coliziunea cu pereții și pereții temporari.

int is\_position\_on\_snake(int x, int y); - Verifică dacă poziția (x, y) este ocupată de șarpe (corp sau cap).

void setup\_temp\_walls(); - Inițializează pereții temporari (setați ca inactivi la start).

void spawn\_temp\_walls(); - Generează pereți temporari aleator pe hartă, fără să se suprapună cu șarpele.

void update\_temp\_walls(); - Actualizează pereții temporari

void load\_high\_score(); - Încarcă scorul maxim dintr-un fișier.

void save\_high\_score(); - Salvează scorul maxim în fișier dacă scorul curent e mai mare.

void afiseaza\_mesaj\_start(SDL\_Renderer\* renderer, TTF\_Font\* font, int screen\_w, int screen\_h); - Afișează un mesaj de start pe ecran.

void handle\_events(); - Procesează evenimente SDL (închidere fereastră, apăsare taste, mouse).

void draw\_circle(SDL\_Renderer \*renderer, int cx, int cy, int radius); - Desenează un cerc pixel cu pixel la poziția (cx, cy) cu raza dată.

void draw\_dollar\_sign(SDL\_Renderer\* renderer, int x, int y); - Desenează semnul $ (mâncare rară) la poziția (x, y).

void draw\_bomb(SDL\_Renderer\* renderer, int x, int y, int tile\_size) ; - Desenează o bombă la poziția (x, y).

void draw\_apple(SDL\_Renderer\* renderer, int x, int y); - Desenează un măr roșu la poziția (x, y).

void draw\_score(SDL\_Renderer\* renderer, TTF\_Font\* font, int score, int high\_score)- Deseneaza score-ul;

int init\_graphics(); - Inițializează fereastra și motorul SDL.

void draw\_background\_green(); - Desenează fundalul verde (pentru terenul de joc).

void close\_graphics(); - Eliberează resursele SDL și închide fereastra.

**5.Interfața grafică**

**Fereastra de joc** este împărțită într-un grid cu un număr fix de coloane și rânduri, pe baza cărora poziționăm elementele jocului (șarpele, mâncarea, pereții temporari etc.).

**Afișarea unui mesaj introductiv la începutul jocului**, care informează utilizatorul despre regulile de bază . Acest mesaj este afișat grafic pe ecran înainte de începerea efectivă a jocului.

**Desenarea șarpelui**:Capul șarpelui este desenat cu o culoare diferită (albastru închis).Corpul șarpelui este desenat cu o altă culoare (albastru deschis).Pentru cap, am desenat și un ochi mic (pătrat alb) .

**Desenarea mâncării**:Diferite tipuri de mâncare sunt reprezentate grafic prin forme specifice, cum ar fi măr (cu o funcție dedicată draw\_apple), simbolul dolarului (draw\_dollar\_sign), sau bombe (draw\_bomb).

**Desenarea pereților temporari** care apar și dispar după un anumit interval de timp.

**Controlul culorilor** și poziționarea precisă a elementelor grafice în fereastră prin funcțiile SDL: SDL\_SetRenderDrawColor, SDL\_RenderFillRect, SDL\_RenderDrawPoint.

**Gestionarea evenimentelor de intrare** de la tastatură prin funcția handle\_events(), care permite controlul șarpelui cu tastele W, A, S, D și oprirea jocului cu ESC.

**Actualizarea grafică a ecranului** pentru fiecare frame, astfel încât jocul să fie animat și să reacționeze în timp real la mișcările utilizatorului.