**Documentație Proiect – „Samurai vs Ninja”**

Samurai vs Ninja – Joc tip Pac-Man realizat cu SDL2

**Descriere generală**

„**Samurai vs Ninja**” este un joc 2D inspirat de clasicul „Pac-Man”, dar reinterpretat într-un stil japonez medieval. Jucătorul preia rolul unui **samurai** care trebuie să colecteze **bile de energie** într-un labirint plin de pericole. Pe parcurs, este urmărit de mai mulți **demoni ninja**. Obiectivul principal este de a **colecta toate bilele** fără a fi atins de inamici. Dacă samuraiul este prins, jocul se termină cu înfrângere.

Proiectul a fost dezvoltat în **limbajul C**, utilizând biblioteca grafică **SDL2** și extensiile sale pentru imagini (SDL\_image), sunet (SDL\_mixer) și text (SDL\_ttf). Aceste librării facilitează dezvoltarea jocurilor 2D oferind un mod rapid de a încărca și redenda imagini, gestiona sunete și afișa informații pe ecran.

**Obiective urmărite în dezvoltare:**

* Învățarea manipulării graficii și animațiilor 2D în C.
* Utilizarea evenimentelor și a tastaturii în SDL.
* Implementarea unui AI simplu pentru inamici (urmărirea jucătorului).
* Integrarea sunetelor și a muzicii în gameplay.

**Tehnologii și resurse utilizate**

|  |  |
| --- | --- |
| **Tehnologie** | **Rol in proiect** |
| C | Limbajul de programare principal |
| SDL2 | Biblioteca pentru grafica, ferestre si input |
| SDL\_image | Pentru incarcarea imaginilor PNG |
| SDL\_mixer | Pentru redarea de sunete si muzica |
| SDL\_ttf | Pentru afisarea textului pe ecran |

**Structura generală a codului**

Codul este împărțit în mai multe secțiuni esențiale, fiecare având un rol clar în funcționarea jocului:

**Definirea constantelor și simbolurilor grafice**

#define WIDTH 40

#define HEIGHT 20

#define PACMAN 'S'

#define WALL '#'

#define FOOD '.'

#define EMPTY ' '

#define DEMON 'N'

* WIDTH și HEIGHT: definesc dimensiunea hărții (40 lățime × 20 înălțime).
* Simbolurile (PACMAN, WALL, etc.) reprezintă caracterele din matricea board, folosită ca hartă a jocului.
* Se utilizează aceste simboluri pentru a crea logică ușor de citit și gestionat în randarea pe ecran.

**Variabile globale și structuri**

int res, score, pacman\_x, pacman\_y, food, curr;

char board[HEIGHT][WIDTH];

SDL\_Window\* window;

SDL\_Renderer\* renderer;

SDL\_Texture\* textures[7];

Mix\_Chunk\* sound\_eat, \*sound\_lose, \*sound\_win;

Mix\_Music\* bg\_music;

TTF\_Font\* font;

int frame;

**Explicație detaliată:**

* res, score: rețin rezultatul jocului și scorul curent.
* pacman\_x, pacman\_y: poziția actuală a samuraiului pe hartă.
* food: numărul total de bile de mâncare rămase.
* board[HEIGHT][WIDTH]: matricea hărții jocului.
* textures[]: vector care stochează texturi încărcate (imagini PNG pentru personaje, bile, fundal).
* Mix\_Chunk și Mix\_Music: gestiune efecte sonore și muzică de fundal.
* font: fontul pentru scor și ecranul final.
* frame: contor folosit pentru a decide când se mișcă demonii (la fiecare 5 frame-uri).

**Funcția initialize()**

Această funcție creează harta jocului, inițializând:

* **Ziduri exterioare** pe margini (WALL).
* **Ziduri interioare aleatoare** (cu probabilitate 25%) pentru a crea un labirint.
* **Bile de mâncare** pe pătratele libere, păstrate cu simbolul FOOD.
* **Poziția samuraiului** (jucătorul), marcată cu PACMAN.
* **Poziția inamicilor** (demonii), marcați cu DEMON.

Aceasta este cheia pentru generarea dinamică a unui labirint.

**Funcția draw()**

Este responsabilă cu **afișarea hărții** și a scorului pe ecran. Se parcurge întreaga matrice board, iar pentru fiecare caracter se selectează textura potrivită:

* WALL → piatră
* FOOD → bilă
* EMPTY → iarbă
* PACMAN → samurai (animat)
* DEMON → ninja (animat)

Se folosește o **alternanță de texturi** pentru animația samuraiului și demonilor (cu frame % 2).

La final, scorul este randat în colțul din stânga sus cu ajutorul SDL\_ttf.

**Control jucător (move\_pacman())**

Funcția interpretează comenzile de la tastatură (WASD):

* Verifică dacă deplasarea este validă (nu în perete).
* Dacă ajunge pe o poziție cu FOOD, crește scorul, se elimină mâncarea și se redă sunetul.
* Dacă ajunge pe DEMON, se consideră că jucătorul a pierdut.
* Dacă nu mai există FOOD, jocul se încheie cu victorie.

**Mișcarea inamicilor (move\_demons\_towards\_pacman())**

O dată la 5 cadre, demonii se deplasează spre samurai:

* Caută în vecinătate direcția de deplasare optimă (care reduce distanța față de jucător).
* Dacă pot ajunge pe poziția jucătorului, jocul se termină cu înfrângere.
* Mișcarea este simplificată (fără algoritmi de pathfinding avansați).

**Ecrane de început și final (show\_start\_screen() și show\_end\_screen())**

* show\_start\_screen() afișează o imagine statică de start, cu instrucțiuni: „Apasă ENTER pentru a începe”.
* show\_end\_screen() afișează mesajul „AI CÂȘTIGAT!” sau „AI PIERDUT!” și așteaptă ESC pentru închidere.

**Controlul utilizatorului**

|  |  |
| --- | --- |
| **Tasta** | **Functie** |
| W | Deplasare in sus |
| A | Deplasare la stanga |
| S | Deplasare in jos |
| D | Deplasare in dreapta |
| Q | Iesire fortata din joc |
| Enter | Incepe jocul |
| ESC | Inchide jocul dupa terminare |

**Resurse media necesare**

**Imagini:**

* ***grass.png, stone.png, ball.png* –** fundal, perete și mâncare
* ***samurai1.png, samurai2.png* –** animație samurai
* ***ninja1.png, ninja2.png* –** animație demon
* ***startscreen.png* –** ecran de start

**Sunete:**

* ***eat.wav* –** când samuraiul mănâncă
* ***hurt.wav* –** când e prins de demon
* ***win.wav* –** la victorie
* ***bgmusic.ogg* –** muzică de fundal

**Fonturi:**

* ***OpenSans-Regular.ttf*** – font pentru afișarea scorului și a mesajelor

**Concluzii**

Acest proiect demonstrează:

* Capacitatea de a construi un joc 2D de la zero folosind SDL2.
* Implementarea unor mecanici clasice de joc într-un stil original.
* Utilizarea eficientă a resurselor multimedia (imagini, sunet, fonturi).
* Organizarea logicii unui joc: hartă, coliziuni, AI, input, ecrane.