Pac-Man

-game-

Autor: Chirca Virginia-Mădălina, 331AA

An: 2021-2022

Cuprins

1.Introducere 3
2. Prezentare a suportului tehnic 3
3. Prezentare tehnica a etapei de realizare/implementare 4
3.1 Framework 4
3.2 Biblioteci 4
3.3 Implementare joc 5
4. Prezentare mod de utilizare 8

1.Introducere

Pac-Man marchează istoria jocurilor video printr-o schimbare de perpectivă în rândul jocurilor video și se remarcă printe toate cele de aventură spațială extrem de populare precum: Galaxian/Galaga, Asteroids și clasicul Space Invaders. Potrivit sursei [1], Pac-Man este jocul care i-a adus pe jucători cu picioarele pe pământ și, mai precis, i-a prins într-un labirint.

Anul 1980 a fost cel în care producătorul japonez de jocuri arcade Namco Limited a realizat și a prezentat lumii Pac-Man. Potrivit sursei [2], încercând să se îndepărteze de ideea unui joc violent, care implică doborârea inamicilor prin lovituri/împușcaturi, designerul principal al jocului, Iwatani Tohru, a creat un concept de joc prietenos, în care jucătorul reprezentat de o sferă galbenă parcurge un labirint populat de puncte colorate(mâncare).

Pornind de la la acest concept, am încercat realizarea unei variante de joc Pac-Man, urmărind ideea de bază a jocului original și adăugând funcționaliți noi, care să pună în dificultate jucătorul în atingerea obiectivului de a termina jocul cu un scor cât mai mare.

Astfel, obiectivele principale ale implementării se identifică ca fiind următoarele:

- Afișarea și derularea jocului într-un mod atractiv și plăcut vizual pentru jucători;
- Asigurarea unei înțelegeri rapide și facile a modului de deplasare a jucătorului în labirint;
- Stârnirea interesului jucătorului;
- Asigurarea unei afișări corecte ale punctajului în consolă corelat la numărul de cercuri de mâncare consumate.

2. Prezentare a suportului tehnic

Provocarea principală a jocului este evitarea întâlnirii cu cele patru fantome din labirint, care urmăresc jucătorul și, în cazul unei coliziuni, îi decrementeaza numărul de vieți. Cele patru fantome au culori diferite: Blinky (roșu), Inky (albastru deschis), Pinky (roz) și Clyde (portocaliu) și prezintă o strategie de atac diferită. Singura salvare în confruntarea fantomelor o reprezintă pastilele speciale, care îi oferă jucătorului puterea de „a mânca” inamicii o scurtă perioadă de timp. Odată ce o fantomă este înghițită, aceasta dispare, iar ochii ei se întorc spre cușca fantomei și se reformează pentru a lupta din nou. Odată cu parcurgerea labirintului și a „mâncării” de pe traseu, scorul jucătorului crește, iar jocul se încheie doar atunci cand numărul vieților ajunge să fie zero.

După cum este precizat și în sursa [2], Pac-Man a devenit rapid o senzație internațională, cu peste 100.000 de console vândute doar în Statele Unite, clasificându-l ca fiind cel mai de succes

joc arcade din istorie. Când pasionații acestui joc au aflat că fantomele se mișcau folosind un tipar, și-au făcut un scop din aflarea rutei precise pe care Pac-Man să o urmeze în evitarea inamicilor. Cu toate acestea, această aparentă predictibilitate a fost compensată de numărul mare de niveluri (256), care a adăugat o complexitate imensă căutării rutei perfecte. (În 1999, un rezident din Florida a câștigat în sfârșit această distincție, obținând 3.333.360 de puncte într-o sesiune de joc de șase ore.)

3. Prezentare tehnica a etapei de realizare/implementare

3.1 Framework

Jocul a fost realizat prin folosirea framework-ului de la sursa [3], fiind scris în limbajul C++, framework care, după cum este prezentat în [4], oferă toate funcționalitățile de bază ale unui motor grafic minimal, precum:

- Fereastra de desenare având la bază un context **OpenGL 3.3+**, OpenGL este considerat în principal un API (Application Programming Interface) care oferă un set mare de funcții pe care le putem folosi pentru a manipula grafice și imagini [5];
- Suport pentru încărcarea de modele 3D (cunoscute și ca **3D meshes**);
- Suport pentru încărcarea de imagini pentru texturarea modelelor 3D;
- Suport pentru definirea și încărcarea de shadere OpenGL.

De asemenea, potrivit aceleiași surse [7], pe langă funcționalitățile de bază, framework-ul implementează un model generic pentru scrierea de aplicații OpenGL. Astfel, sunt oferite următoarele aspecte:

- Control pentru fereastra de afișare;
- Management pentru input de la tastatură și mouse;
- Cameră de vizualizare cu input predefinit pentru a ușura deplasarea și vizualizarea scenei;
- Model arhitectural al unei aplicații simple OpenGL, bazat pe toate aspectele prezentate.

3.2 Biblioteci

În cadrul framework-ului sunt folosite următoarele biblioteci:

GLFW

- Oferă suportul de bază pentru API-ul OpenGL precum context, fereastră, input, etc

GLEW

- Asigură suportul pentru extensiile de OpenGL suportate de procesorul grafic

GLM

- Funcționalități matematice bazate pe specificațiile limbajului GLSL (shadere OpenGL)

ASSIMP

- Oferă suport pentru încărcarea de modele și scene 3D

STB

- Oferă suport pentru încărcare/decodare de imagini JPG, PNG, TGA, BMP, PSD, etc.

3.3 Implementare joc

Implementarea jocului cu toate funcțiile, clasele folosite se află în "gfx-framework-master\src\lab_m1\PacMan".

În fișierul "**object2D-pacman.h**" sunt declarate funcțiile pentru crearea obiectelor folosite pentru realizarea spațiului de joc, toate fiind de tip "mesh", obiect definit prin vârfuri și indici:

- CreateCircle – creare elipsa pentru realizarea jucătorului, mâncării, folosit și la fantome;
- CreateRectangle - creare dreptunghiuri pentru realizarea pereților labirintntului, folosit și la fantome, mâncare și jucător pentru realizarea coliziunilor;
- CreateTriangle - creare triunghi pentru realizarea gurii jucătorului;
- Heart – creare formă inimă pentru afișarea vieții

În fișierul "**transform2D-pacman.h**" sunt definite funcțiile pentru calculul matricilor de translație, rotație și scalare, folosind biblioteca GLM, o bibliotecă implementată cu matrici în formă coloană.

În fișierul "pacman.h" sunt declarate variabilele folosite pentru pozițiile obiectelor randate, cât și alte variabile auxiliare. Tot aici sunt declarate toate funcțiile folosite în crearea jocului.

Generarea labirintului sub formă matriceală, conform algortimului sursei [4]**.**

Dimensiunea labirintului este de 9 linii si 18 coloane.

- ResetGrid(int grid[162]) - marcarea tuurot zonelor ca fiind nevizitate;
- IndexGrid(int x, int y) - convertire pereche de indecși într-un singur indexș
- int Bounds(int x, int y) - verificarea limitelor;
- void Visit(int x, int y, int grid[162]) - în funcție de punctele cardinale, se marcheaza zonele vizitate.

În funcția de Init() s-au inițializat toate variabilele folosite, s-au creat obiectele folosite la realizarea jocului, s-a inițializat labirintul și s-a făcut o parcurgere pentru reținerea pozițiilor de randare a fantomelor și a mâncărurilor speciale.

În cadrul jocului, obiectele sunt create în spațiul logic (0, 0) - (16, 9).
În funcția "OnInputUpdate(float deltaTime, int mods)", am realizat deplasarea jucătorului în labirint: stânga-dreapta, sus-jos, cu tastele A-D, respectiv W-S.

Implementare coliziuni

Toate coliziunile sunt de tip dreptunghi-dreptunghi, pentru obiectele de tip circle (jucător, mâncare), am randat în spatele lor, dreptunghiuri negre pentru care am reținut pozițiile în verificare coliziunilor.

CheckCollisionPlayerObs() – verificare coliziune jucător-pereții labirintului

- În cazul existenței, jucătorul ocolește marginea peretelui.

CheckCollisionPlayerFood() – verificare coliziune jucător-mâncare

- În cazul existenței, mâncarea dispare și scorul crește.

CheckCollisionGhostObs(float posghostx,float posghosty) – verificare coliziune fantomă-pereții labirintului

- În cazul existenței, fantoma se deplaseaza în direcția opusă până la apariția unei noi coliziuni (cele patru fantome sunt create astfel încât să se deplaseze pe direcția orizontală atât cat le permite labirintul).

CheckCollisionGhostPlayer(float posghostx, float posghosty) – verificare coliziune jucător-fantomă
CheckCollisionSpecialPlayer()– verificare coliziune jucător-mâncare specială

Exemplu de implementare verificare coliziune jucător-pereții labirintului [6]

```
bool PacMan::CheckCollisionPlayerObs() // AABB - AABB collision
{
    // collision x-axis?
    bool collisionX = PosObstacle.x + 0.8 >= RectPos.x &&
        RectPos.x + 0.6 >= PosObstacle.x;
    // collision y-axis?
    bool collisionY = PosObstacle.y + 0.8 >= RectPos.y &&
        RectPos.y + 0.6 >= PosObstacle.y;
    // collision only if on both axes
    return collisionX && collisionY;
}
```

În funcția “Update(float deltaTimeSeconds)” am verificat, pentru început, existența unei coliziuni dintre jucator și oricare dintre cele patru fantome.

- In cazul în care jucătorul nu atinge niciuna dintre cele patru fantome, el este randat fără nicio modificare;
- In cazul în care jucătorul atinge una dintre fantome și nu este sub efectul de mâncare specială, el este readus pe poziția inițială din labirint și își pierde una dintre vieți;
- In cazul în care jucătorul atinge una dintre fantome și este sub efectul de mâncare special are puterea de a merge prin prima fantomă întâlnită, fără aș pierde din viață.

Tot in funcția “Update” am făcut parcurgerea labirintului cu setarea “i = 0.5;
j = -0.35;”, ca fiind pozițiile inițiale de unde se incepe randarea pereților pentru labirint în spațiul logic.
Fiecare perete se randează doar dacă în matricea-labirint este marcat cu “1” elementul respectiv, conform capturii de cod de mai jos.

```
if (grid[IndexGrid(x, y)] == 1)
{//randare obstacole/peretii labirintului
    obstacleMatrix = glm::mat4(1);
    obstacleMatrix *= visMatrix * transform2D::Translate(i, j);

    RenderMesh2D(meshes["obstacle"], shaders["VertexColor"], obstacleMatrix);
    PosObstacle = GetObstaclePosition(i, j);
```

În caz contrar, de marcare cu “0” a elementului, se începe randarea pentru cele patru fantome, pentru mâncare și pentru mâncarea specială, verificând aici existența coliziunilor dintre obiecte.

Functia RenderLives() este folosiă pentru randarea inimilor, care reprezintă viața jucatorului. În momentul în care numărul vieților atinge valoarea 0, jocul se încheie, fereastra de afișare a jocului se închide, iar în consolă va fi afișat scorul final al jucătorului.

4. Prezentare mod de utilizare

Jucătorul se deplasează în labirint: stânga-dreapta, sus-jos, folosind tastele A-D, respectiv W-S.
Deplasarea unei fantome este continuă pe direcția orizontală, cât îi este permis de pereții labirintului sau de ieșirea din labirint.

Pentru a strânge puncta adunate la scorul final, jucătorul trebuie să parcurgă o cât mai mare suprafață din labirint, adică să treacă peste cât mai mute cercuri de mâncare și să evite pe cât posibil intersecția cu o fantomă. Fiecare mâncare peste care se trece îi aduce un cumul de 100 de puncte. În caz negativ, își va pierde o viață și va fi repoziționat pe poziția sa inițială.

În cazul trecerii peste o mâncare specială, jucătorul primește șansa de a trece peste o fantomă fără a fi afectat de coliziunea cu aceasta. Fiecare mâncare specială peste care se trece îi aduce un cumul de 200 de puncte.

Când jucătorul își pierde toate viețile fereastra de afișare se închide, iar în consolă va fi afișat scorul său final.

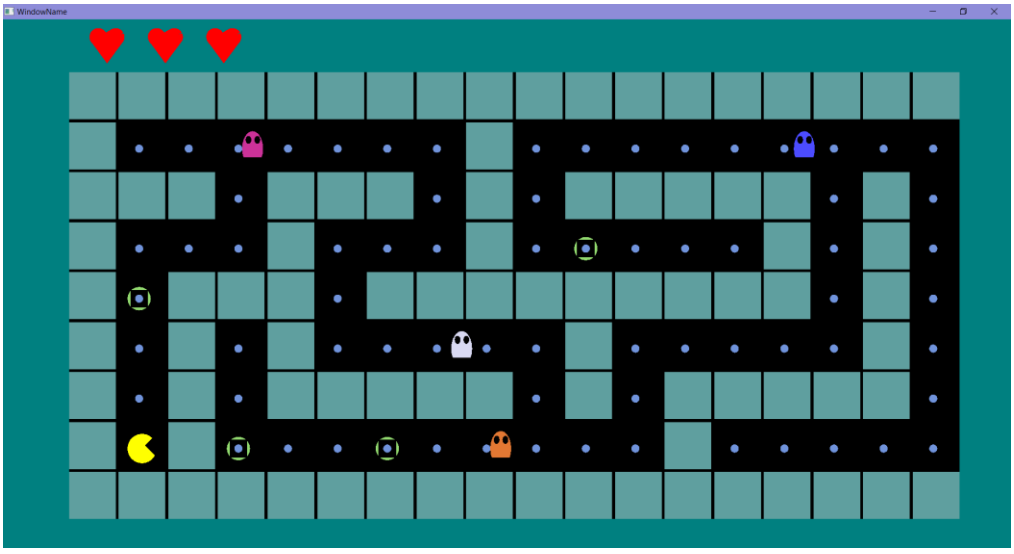

Figura1 – Afișarea ferestrei de joc cu jucătorul în poziția inițială

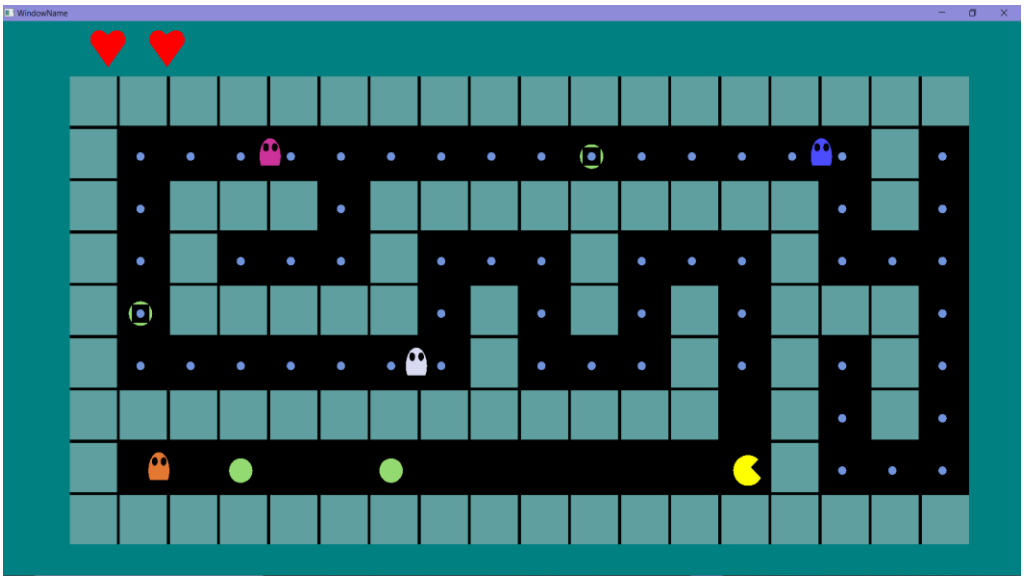

Figura2 – Afișarea ferestrei de joc cu jucătorul pe traseu și cu două vieți rămase

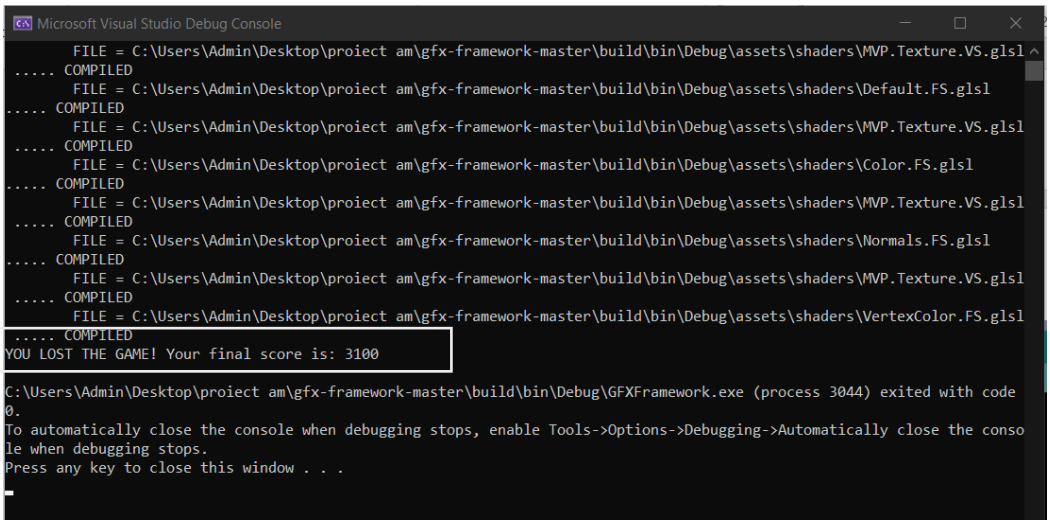

Figura3 – Afișarea consolei cu punctele acumulate de jucător

5. Concluzii

Revenind la obiectivele enumerate la începutul prezentării putem bifa:

- ✓ Afișarea și derularea jocului într-un mod atractiv și plăcut vizual pentru jucători prin culorile afișate și deplasarea obiectelor;
- ✓ Asigurarea unei înțelegeri rapide și facile a modului de deplasare a jucătorului în labirint prin introducerea a patru taste (W-A-S-D), responsabile de deplasarea jucătorului;
- ✓ Stârnirea interesului jucătorului prin obiectivul de a evita coliziunea cu fantomele și finaliza jocul cu un scor cât mai mare;
- ✓ Asigurarea unei afișări corecte ale punctajului în consolă prin corelarea lui cu numărul de cercuri de mâncare consumate(100 puncte mâncare normală, 200 puncte mâncare specială).

Având în vedere că, jocul implementat nu copiază în întregime jocul original de Pac-Man, unde, spre exemplu, fantomele urmăresc traseul jucătorului și dispar când acesta deține puterea specială, aș putea încheia menționând că jocul implementat este o variantă reinterpretată și particulară a celui popular și că este deschis îmbunătățirilor din punct de vedere al graficii și al interacțiunii cu jucătorul.

6. Referințe bibliografice

1.Thompson, T. et al. "An Evaluation of the Benefits of Look-Ahead in Pac-Man." 2008 IEEE Symposium On Computational Intelligence and Games, 15-18 Dec. 2008, pp. 310–315

2. Britannica, website, **https://www.britannica.com/topic/Pac-Man-1688279**

3. EDU, website, **https://www.cefns.nau.edu/~pek7/CS200/Project%209.pdf**

4. GitHub, website, **https://github.com/UPB-Graphics/gfx-framework**

5. LearnOpenGL, website, **https://learnopengl.com/Getting-started/OpenGL**

6. LearnOpenGL, website, **https://learnopengl.com/In-Practice/2D-Game/Collisions/Collision-detection**

7. Ocw, website, **https://ocw.cs.pub.ro/courses/egc/laboratoare/01**