CIRCUIT F1

Student: Simina Dan-Marius

Proiect Prelucrare Grafică

Universitatea Tehnică din Cluj-Napoca

Cuprins

1. PREZENTAREA TEMEI

Tema aleasă pentru acest proiect constă în conceperea și realizarea unui circuit de Formula 1, urmată de animarea unei mașini care se deplasează pe traseul creat. Proiectul presupune integrarea elementelor de design grafic pentru conturarea pistei, precum și implementarea unui sistem de animație care să simuleze mișcarea vehiculului pe circuit într-un mod cât mai natural și dinamic.

Traseul va include detalii specifice, cum ar fi viraje, linii drepte și marcaje de delimitare, pentru a reproduce cât mai fidel un circuit profesionist. Mașina animată va fi programată să urmeze un algoritm predefinit de navigare pe traseu.

2. SCENARIUL

2.1. DESCRIEREA SCENEI ȘI A OBIECTELOR

Scena ilustrează un circuit de Formula 1 care se desfășoară în jurul unui lac, într-un peisaj ce include elemente naturale precum copaci ce mărginesc traseul. Circuitul este dotat cu patru stâlpi de iluminare amplasați strategic pentru a asigura vizibilitatea optimă în orice condiții, iar pe linia de start-sosire se regăsesc trei garaje.

Elementul central al scenei este o mașină de Formula 1 aparținând echipei Scuderia Ferrari, reprezentată în culorile sale emblematice.

Sursele de lumină includ obiecte de tip cubelight.

2.2. FUNCȚIONALITĂȚI

- Deplasarea camerei folosind tastatura și mouse-ul
- Reglarea luminii ambientale
- Reglarea intensității ceții
- Activarea dezactivarea animației de ploaie
- Mutarea camerei astfel încât să urmărească mașina
- Schimbarea între modurile de vizualizare

3. DETALII DE IMPLEMENTARE

3.1. FUNCȚII ȘI ALGORTMI

ALGORITMUL DE ANIMAȚIE AL MAȘINII:

1. Funcția initPoints():
- Inițializează 18 puncte cu poziții și unghiuri specifice
- Aceste puncte definesc un traseu pentru mașină
- Fiecare punct are o coordonată 3D (x, y, z) și un unghi de rotație

2. Funcția catmullRomInterpolation():
- Implementează interpolarea Catmull-Rom pentru 4 puncte
- Această interpolare creează o curbă netedă care trece prin punctele date
- Folosește parametrul t (între 0 și 1) pentru a calcula poziții intermediare pe curbă
- Formula folosește polynomiale cubice pentru a calcula poziția interpolată

3. Funcția interpolateAngle():
- Interpolează linear între două unghiuri (a0 și a1)
- Are grijă să aleagă cea mai scurtă cale între unghiuri
- Face corecții când diferența între unghiuri trece peste PI sau -PI

4. Funcția moveCar():
- Calculează următoarea poziție a mașinii pe traseu
- Folosește 4 puncte consecutive pentru interpolarea Catmull-Rom
- Actualizează poziția (carPos) și unghiul (carAngle) mașinii
- Când ajunge la ultimul punct intermediar, trece la următorul segment

În esență, acest cod implementează mișcarea fluidă a unei mașini de-a lungul unui traseu predefinit, folosind interpolare pentru a crea o traiectorie netedă între punctele de control. Mașina nu sare direct din punct în punct, ci se deplasează fluid între ele, atât ca poziție cât și ca orientare.

3.2. MODELUL GRAFIC

SHADERELE FOLOSITE:

- Shader-ele de bază (basic.vert / basic.frag)
- Shader-ele pentru Skybox (skyboxShader.vert / skyboxShader.frag)
- Shader-ele pentru cubul de lumină de culoare albă (lightCube.vert / lightCube.frag)

- Shader-ele pentru cubul de lumină de culoare roșie, din spatele mașinii (rearLightCube.vert / rearLightCube.frag)
- Shader-ele pentru depth mapping (depthShader.vert / depthShader.frag)
- Shader-ele pentru screen quad pentru vizualizarea shadow map-ului (screenQuad.vert / screenQuad.frag)
- Shader-ele pentru animația de ploaie (rain.vert / rain.frag)

3.3. STRUCTURI DE DATE

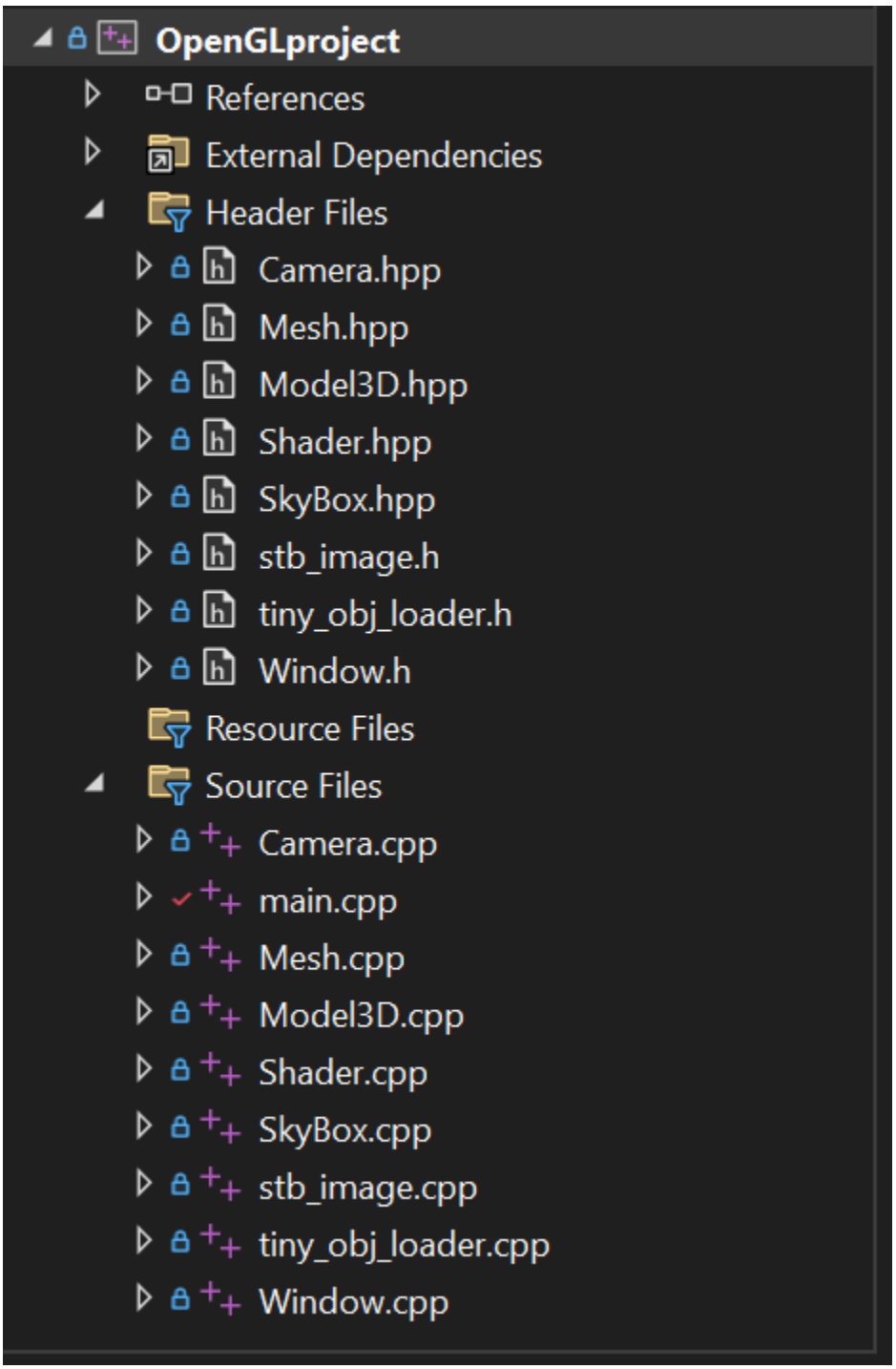
STRUCTURA DE DATE PENTRU POZIȚIA MAȘINII:

```
struct Point {
    glm::vec3 point;
    float angle;
};
```

STRUCTURA DE DATE PENTRU PICĂTURILE DE PLOAIE:

```
//rain
struct Raindrop {
    float x, y, z;
    float speed;
};
```

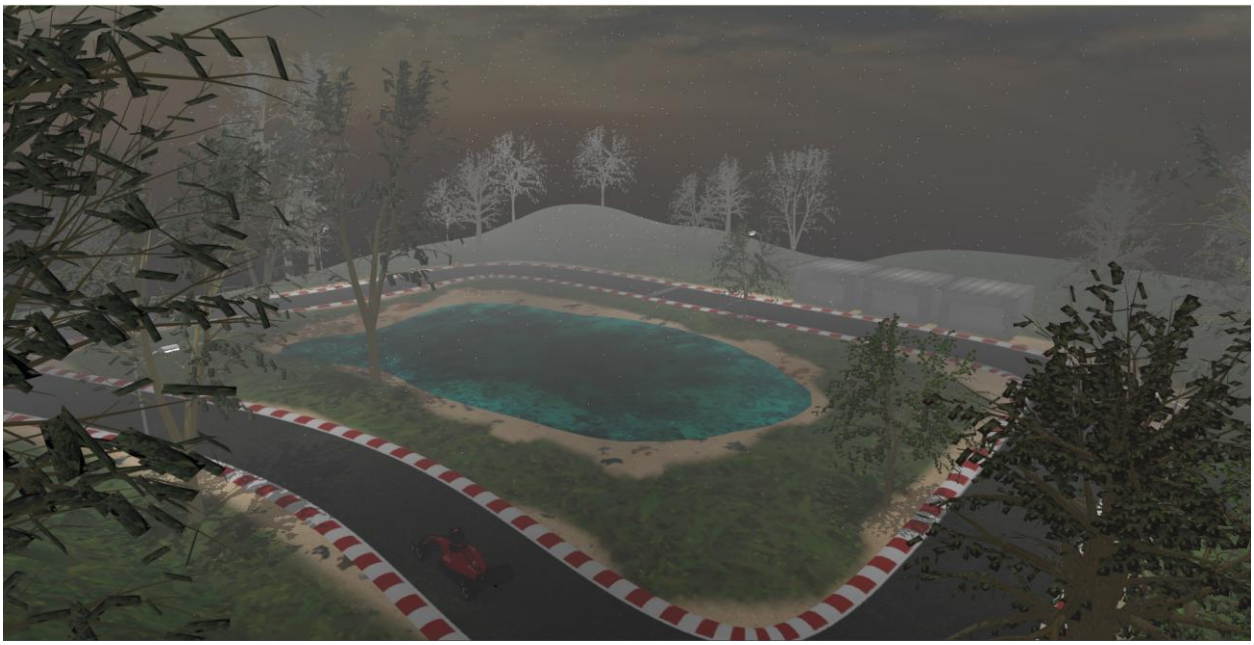
3.4. IERARHIA DE CLASE

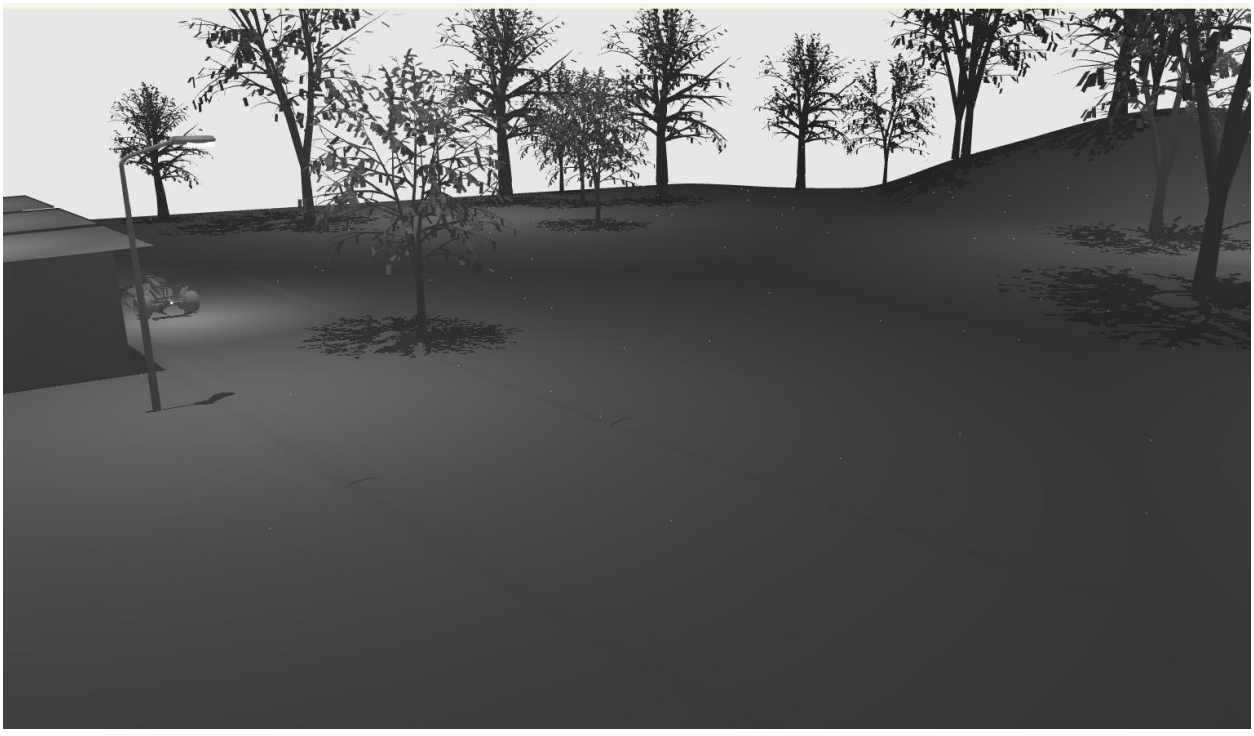


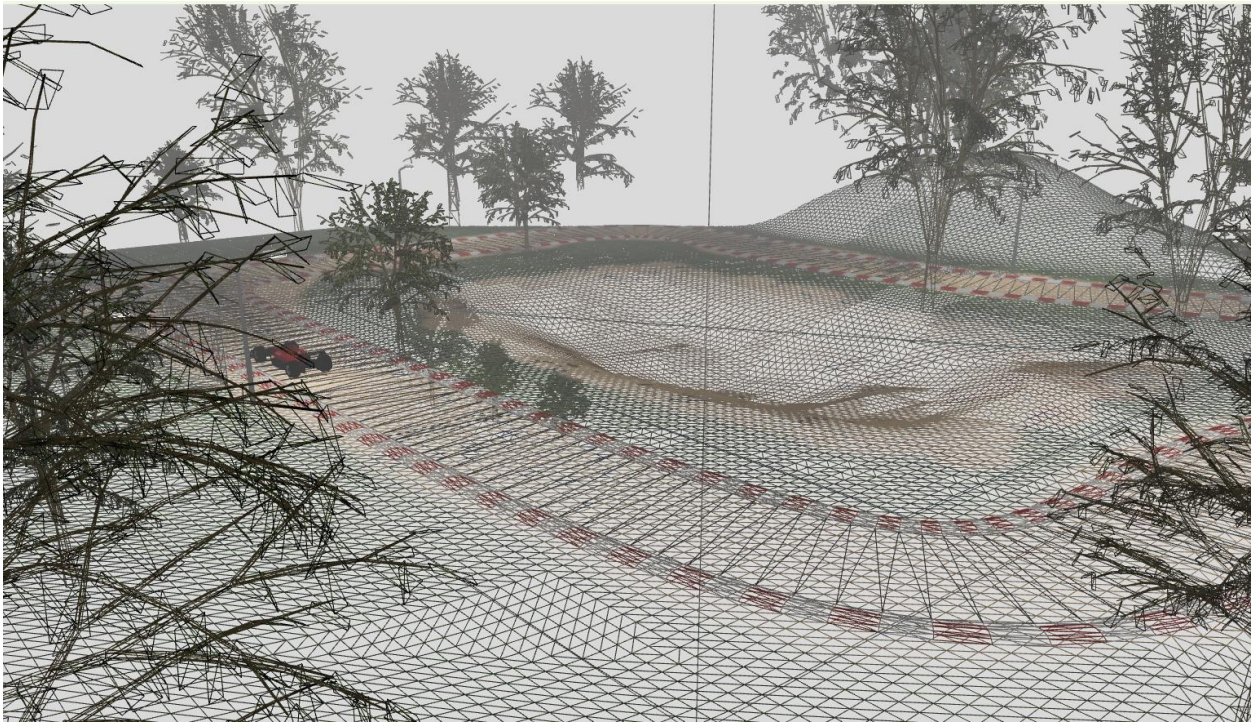
4. PREZENTAREA INTERFEȚEI GRAFICE UTILIZATOR / MANUAL DE UTILIZARE

PREZENTAREA INTERFEȚEI GRAFICE:

Odată cu rularea aplicației utilizatorul este introdus direct în scenă de unde poate începe vizualizarea. Acestea sunt câteva poze din aplicație:







MANUAL DE UTILIZARE:

- Deplasarea prin scenă se face folosind tastele: **W**, **A**, **S**, **D** și mouse-ul.
- Rotirea scenei se face folosind tastele: **Q** și **E**
- Vizualizarea în modul wireframe: **T**
- Vizualizarea în modul solid: **Y**
- Activare/dezactivare animație mașină: **F**
- Poziție următoare mașină: **N**
- Activare/dezactivare animație ploaie: **Z**
- Activare/dezactivare animație cameră: **V**
- Mutare mașină în scenă: **I**, **K**, **J**, **L**, **G**, **H**
- Rotire mașină pe axa Y: **U**, **O**
- Ajustarea luminii ambientale: **1**, **2**
- Ajustarea efectului de ceață: **3**, **4**

5. CONCLUZII ȘI DEZVOLTĂRI ULTERIOARE

În concluzie aplicația oferă o scenă complexă cu multiple surse de lumină, umbre, animație pentru obiecte, animație de ploaie și efect de ceață. Ca dezvoltări ulterioare se poate realiza controlarea mașinii într-un mod realist de către utilizator, implementarea unui sistem de coliziuni, și multe alte lucruri pentru creșterea nivelului de realism.

6. REFERINȚE

1. Lucrările de laborator
2. https://www.cgtrader.com/free-3d-models/car/racing-car/f1-2022-concept-car
3. https://learnopengl.com/