**Proiect Prelucrare Grafică**

**- Sat medieval -**

**Marinca Gabriel-Ioan**

**Grupa: 30233**

**15 Ianuarie, 2025**

1. **Cuprins**

1. Cuprins…………………………………………………………………………………………………………..2  
2. Prezentarea temei………………………………………………………………………………………………..3  
3. Scenariul…………………………………………………………………………………………………………3  
    3.1. descrierea scenei și a obiectelor…………………………………………………………………………….3  
    3.2. funcționalități………………………………………………………………………………………………..3  
4. Detalii de implementare……………………………………………………………………………..…………..3  
    4.1. funcții și algortmi…………………………………………………………………………………….……..3  
    4.2. modelul grafic……………………………………………………………………………………………....4   
5. Prezentarea interfeței grafice utilizator / manual de utilizare…………………………………………………...4  
6. Concluzii și dezvoltări ulterioare………………………………………………………………………………..6  
7. Referințe…………………………………………………………………………………………………………6

1. **Prezentarea temei**

Tema proiectului constă într-o simulare a unui sat medieval, incluzând elemente specifice perioadei, precum case construite din lemn și piatră, un castel, înconjurat de ziduri de lemn, terenuri agricole și alte obiecte caracteristice. Scopul proiectului este de a crea o experiență interactivă care permite utilizatorului să exploreze și să interacționeze cu aceste obiecte, ilustrând modul de viață din acea perioadă. **3. Scenariul  
    3.1. descrierea scenei și a obiectelor**

Satul medieval este format din următoarele componente:

* Casele, construite din lemn și piatră, cu acoperișuri din paie. Unele sunt case ale meșteșugarilor, altele locuințe simple.
* Terenurile Agricole, situate la marginea satului, unde este cultivat grâul.
* Copacii, aflați prin sat și la margine
* Țarcuri cu animale, precum vaci și cai
* Un car cu fân, care este animat să se miște
* Moară de măcinat, ale cărei aripi sunt animate să se miște
* Poarta și zidurile fortului, în care este un castel-turn și clădiri importante, precum și o catapultă. **3.2. funcționalități**
* Posibilitatea de a explora satul într-un mod interactiv (mișcare liberă prin scenă).
* Interacțiuni cu obiectele (deschiderea ușilor, miscarea unui car, rotirea aripilor unei mori).
* Generare ceata/ninsoare.  **4. Detalii de implementare  
      4.1. funcții și algortmi**

**Generare fenomen natural:zăpadă**

void initSnow() {

// random positions

for (int i = 0; i < 1000; i++) {

snowdrops.push\_back(glm::vec3(

static\_cast<float>(rand() % 100 - 50),

static\_cast<float>(rand() % 100),

static\_cast<float>(rand() % 100 - 50)

));

}

glGenVertexArrays(1, &snowVAO);

glGenBuffers(1, &snowVBO);

glBindVertexArray(snowVAO);

glBindBuffer(GL\_ARRAY\_BUFFER, snowVBO);

glBufferData(GL\_ARRAY\_BUFFER, snowdrops.size() \* sizeof(glm::vec3), &snowdrops[0], GL\_DYNAMIC\_DRAW);

glVertexAttribPointer(0, 3, GL\_FLOAT, GL\_FALSE, sizeof(glm::vec3), (void\*)0);

glEnableVertexAttribArray(0);

glBindBuffer(GL\_ARRAY\_BUFFER, 0);

glBindVertexArray(0);

}

Funcția initSnow inițializează fenomenul de zăpadă prin generarea a 1000 de poziții aleatoare pentru fulgi în spațiul 3D, utilizând un vector snowdrops de tip **std::vector<glm::vec3>.** Fulgii sunt distribuiți într-un volum delimitat. Vertex Array Object (VAO) este creat pentru a stoca configurația necesară pentru randarea fulgilor, iar buffer-ele sunt configurate astfel încât fiecare fulg să fie reprezentat prin coordonatele sale tridimensionale. La final, se dezactivează VAO și VBO pentru a preveni modificări accidentale. Această configurație permite GPU-ului să gestioneze eficient pozițiile fulgilor și să le afișeze în scenă

**Simulare săritură:**

In Camera.cpp/hpp am adaugat:   
if (direction == MOVE\_UP && !jumping) {

jumping = true;

speedJump = height; // Apply initial upward velocity

}

if (jumping) {

cameraPosition.y += speedJump;

cameraTarget.y += speedJump;

speedJump -= gravity;

if (cameraPosition.y <= 0.0f) {

cameraPosition.y = 0.0f;

cameraTarget.y = 0.0f;

jumping = false;

speedJump = 0.0f;

}

} **4.2. Modelul graphic**

Modelele 3D sunt create utilizând Blender[[1],](#b1) sau descărcate de pe site-uri specifice pentru obiectele de tiă .obj[.[2],](#b2) [[3]](#b3) Ele includ detalii precum texturi realiste. S-au folosit tehnici de optimizare pentru a reduce consumul de resurse hardware. **5. Prezentarea interfeței grafice utilizator / manual de utilizare**

**A screenshot of a video game

Description automatically generated**

**A stone building with towers and a fence

Description automatically generated**

**A boat in a river

Description automatically generated with medium confidence**

**A video game screen of a windmill

Description automatically generated**

**Manual de utilizare:**

|  |  |
| --- | --- |
| **Tasta** | **Actiune** |
| W,A,S,D | Miscare camera inainte,stanga,inapoi si dreapta |
| **←,→,↓,↑** | Schimbare perspectiva camera in stanga, dreapta,jos si in sus |
| Spacebar | Saritura |
| O/C | Deschide/inchide usa |
| P | Miscarea aripilor morii |
| H | Miscare Car inainte - animatie |
| R | Apare/dispare ninsoarea |
| L | Schimbarea unghiului lumini |
| Z,X,V | Vizualizare in modurile wireframe, solid, poligonal |

**6. Concluzii și dezvoltări ulterioare**

Proiectul demonstrează posibilitățile de simulare a unui mediu medieval interactiv. Dezvoltări ulterioare pot include:

* Adăugarea de noi clădiri și obiecte pentru o varietate mai mare.
* Adaugare alte conditii meteo( ploaie, vant)
* Animatii mai complexe **7. Referințe**

[1] Cosmin Nandra – tutoriale Blender <https://www.youtube.com/playlist?list=PLrgcDEgRZ_kndoWmRkAK4Y7ToJdOf-OSM>

Surse Modele:

[2] <https://www.turbosquid.com/>

[3] https://free3d.com/