**Proiect la Prelucrare Grafica**

**Student**: Juhasz Norbert

**Grupa**: 30234

**Profesor**: Constantin Nandra „Cosmin”

**1.Cuprins**

1. Cuprins  
2. Prezentarea temei  
3. Scenariul  
    3.1. descrierea scenei și a obiectelor  
    3.2. funcționalități  
4. Detalii de implementare  
    4.1. funcții și algoritmi  
           4.1.1. soluții posibile  
           4.1.2. motivarea abordării alese  
    4.2. modelul grafic  
    4.3. structuri de date  
    4.4. ierarhia de clase  
5. Prezentarea interfeței grafice utilizator / manual de utilizare  
6. Concluzii și dezvoltări ulterioare  
7. Referințe

**2. Prezentarea temei**

Scopul acestui prooect este de a implementa o aplicatie in OpenGL prin intermediul limbajulul C++ si GLSL. Pentru miscare am implementat navigarea prin tastele w,a,s,d, precum si cu ajutorul sagetilor.

Ca si scena am importat o insula japoneza care se afla intr-un glob plutitor unde am adaugat niste obiecte, Digimoni, avand si anumite efecte depsre care voi vorbi in alte capitole.

**3. Scenariul  
    3.1. Descrierea scenei și a obiectelor**

Scena principala este alcatuita dintr-un templu langa o mare care se afla intr-un glob imens plin de plante. Pe langa scena principala am importat si niste Digimoni. Pe pod putem vedea batalia dintre bine si rau, binele incercand sa salveze gradinita de Digimoni care se afla sus in templu. Deasemenea invatatorul de la gradinita se afla langa un felinar langa pod pentru a sprijini lupta binelui cu raul. Pe partea de jos a mapei inainte de a ne urca pe scari spre templu putem observa un Metal Garurumon care se plimba din stanga spre dreapta sub forma sinusoidala. In templu avem acea gradinita care trb aparata de bine. Putem observa si niste oua de digimoni, un digimon care se balanseaza pe un calut, precum si un digimon care se joaca in nisip.

**3.2. Funcționalități**

In primul rand pentru a ne putea deplasa prin scena vom folosi tastele W, A, S, D cu ajutorul carora putem naviga in fata si in spate prin mapa. Cu ajutorul tastelor de up si down am implementat un efect de lift, astfel incat putem sa vedem de sus sau de mai jos toata mapa. Pe tastele de stanga si dreapta am implementat rotatia spre stanga si spre dreapta a camerei. Prin apasarea tastei T am implementat efectul de fulger. Prin apasarea tastei B am implementat modul de noapte, in care putem observa lumina punctiforma, aceasta fiind o lanterna care se misca in aceasi directie cu camera, astfel simuland o lanterna antica japoneza, care se contopeste cu atmosfera creata de scena. Pentru modul de lumina vom apasa tasta N. Deasemenea am implementat efectul de ploaie care o sa apara prin apasarea tastei R. Pe langa aceste efecte am implementat si modul wireframe si cel punctiform prin apasarea tastelor Z, X si C. Pentru a putea misca lumina directionala vom apasa pe tastele J si L.

**4. Detalii de implementare**

**4.1. Funcții și algoritmi**

Aici voi vorbi despre anumite functionalitati pe care le-am implementat in proiectul meu:

1. Efectul de ploaie: pentru a implementa efectul de ploaie deasupra podului am luat initial un punct al podului de unde sa inceapa ploaia. Dupa care am parcurs dimensiunea podului si o inaltime a acestuia cu 3 for-uri imbricate pentru a putea simula un paralelipiped dreptunghic. In aceasta zona am desenat random o picatura de ploaie mereu cand acea valoare random era 5.
2. Am implementat bounding sphere-ul pentru a putea pune efectul de coliziune asupra obiectelor. Prima data am creat o structura care sa retina aceste date: centrul sferei fiind un vec3 si raza sferei, cat de mare sa fie sfera. Dupa care am incadrat 2 obiecte: un felinar si personajul negativ de pe pod in 2 sfere diferite si am verificat de fiecare data pozitia camerei, adica in cazul in care eu ma misc cu camera si noua pozitie a camerei ar intra in acea sfera atunci sa nu execute operatia de move a camerei.
3. Am implementat deasemenea si efectul sinusoidal al lui Metal Garurumom care se misca din stanga spre dreapta a mapei. Pentru a implementa acest efect am crescut axele x, y, z cu anumite valori constante. In cazul in care y ajungea la o anumite valoare maxima at scadeama cea valoare pana la o valoare minima. Deasemenea am numarat si nr de oscilatii pe care sa le faca, in cazul in care ajungea la vbaloarea maxima de 3 ori in acel moment garurumon trb sa se roteasca 180 de grade si sa mearga in directia opusa. Pentru a implementa rotatia, prima oara am rotit obiectul in origine, dupa care l-am rotit cu 180 de grade, mai apoi l-am translatat inapoi in pozitia initiala.
4. Efectul de noapte l-am implementat in shadere, astfel incat am trimis o valoare int, iar in cazul in care acea valoare era 1 at inmulteam cele 3 valori: ambientala, speculara si difuza cu o valoare constanta astfel incat culoarea obiectelor sa para mai intunecata, idem pentru efectul de lumina.
5. Efectul de fulger l-am implementat tot in shadere, astfel ca la apasarea unei taste cresc intensitatea luminii foarte mult, adica inmultesc speculara, ambientala si difuza cu o valoare foarte mare. Deasemenea am implementat in functia render un timer pe care il folosesc pentru a calcula distanta dintre 3 fulgere care o sa apara in momentul apasarii tastei.
6. Efectul de ceata a fost implementata tot in shadere, existand metoda de computeFog() care foloseste o exponentiala pentru a reda intr-un mod cat mai realistic aceasta ceata.
7. Umbrele le-am implementat tot in shadere folosind functia computeShadow(), calculand acei pixeli a caror culoare vor fi neagri ,acesti pixeli fiind cei corespunzatori umbrei fiecarui obiect. Aceste umbre uneori o sa fie pixelate din cauza luminii directionale, acesta s-ar putea rezolva prin micsorarea dimensiunii acelui paralelipiped dreptunghic din functia ortho.
8. Deasemenea am implementat si o prezentare a scenei care se va executa in momentul in care apasam pe tasta P. in timpul efectului nu vom putea misca camera. Pentru implementarea acestui effect am folosit pur si simplu operatiile de move ale camerei: move up, down, left, right, etc.

**4.2. Modelul grafic**

Obiectele din scena au fost sau facuta de mine: acea cutie de nisip, sau luate de pe internet. Scena mare in sine cu obiectele japoneze am luat-o de pe <https://www.deviantart.com/>, iar obiectele de Digimon le-am luat de pe <https://www.models-resource.com/>. Pentru a crea scena am folosit blender-ul pentru a pune texturile obiectelor care sunt sub forma de png. Pentru a mapa textura asupra obiectelor am folosit un fisier mtl in care avem functia map\_Kd care pune textura pe o zona a obiectului. Pentru picatura de ploaie am folosit o functie din blender pentru a reduce numarul de varfuri ca sa putem desena cat mai multe picaturi de ploaie fara a scadea fps-urile cu care va rula aplicatia. Pentru obiectele transparente din mapa am folosit o mixare din blender pe care dupa am exportat-o si am folosit shaderele de frunza pe care le-am implementat la laborator.

**4.3. Structuri de date**

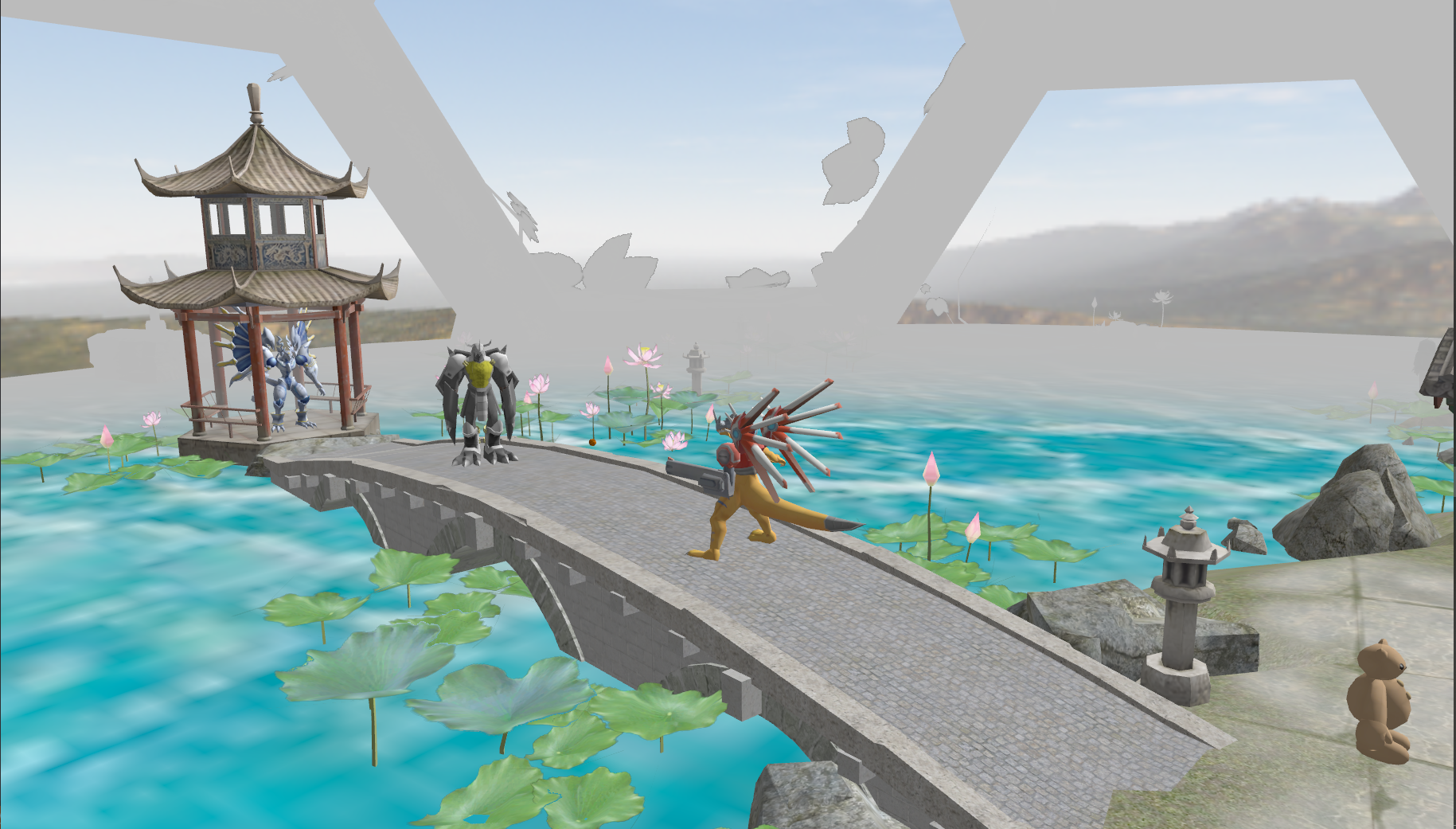
Ca si structuri de date am folosit cel mai mult mat4, vec4 si vec precum si glunit-uri. Deasemenea am creat so o structura de date pentru a putea retine componentele bounding sphere-ului.

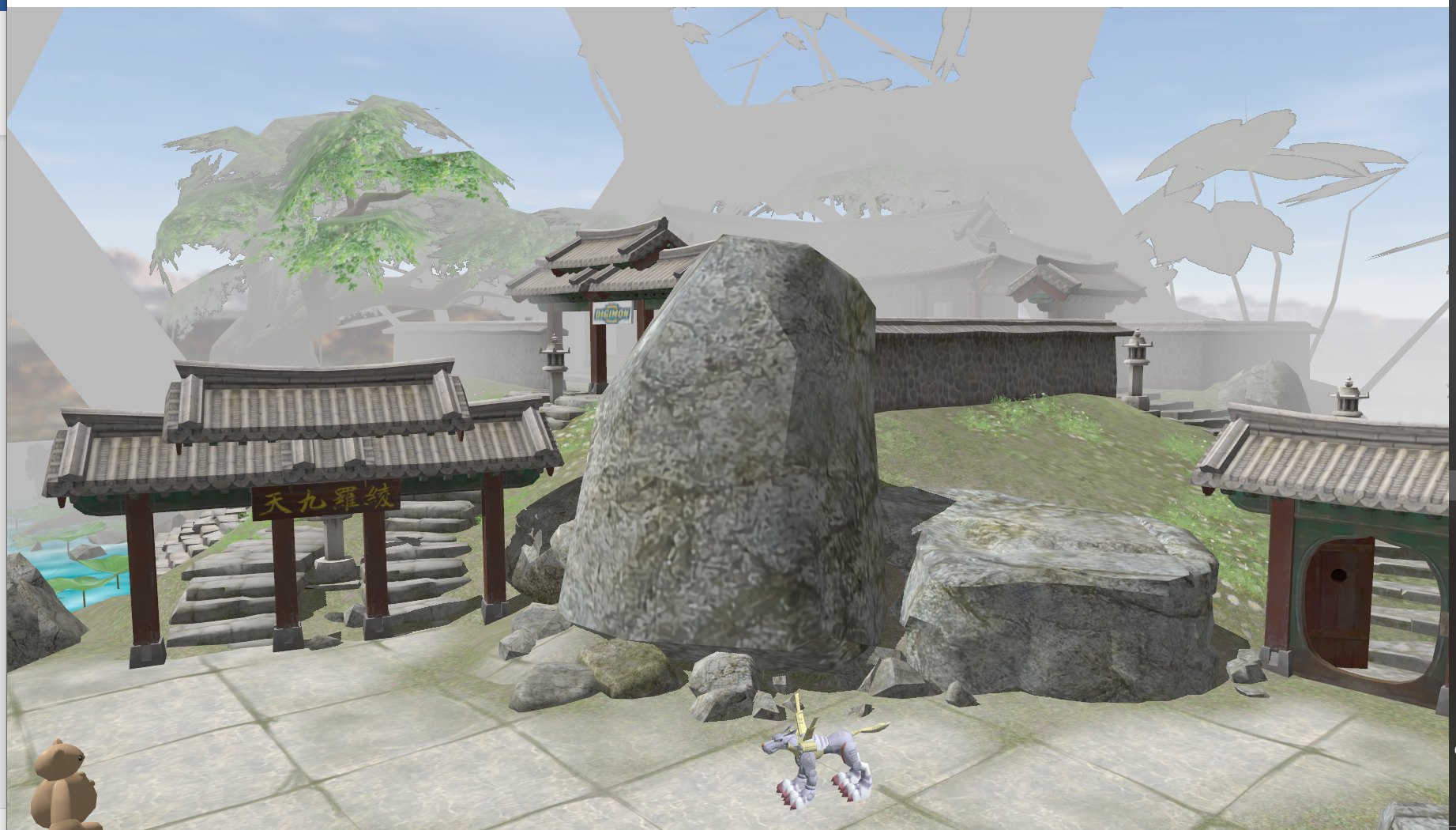
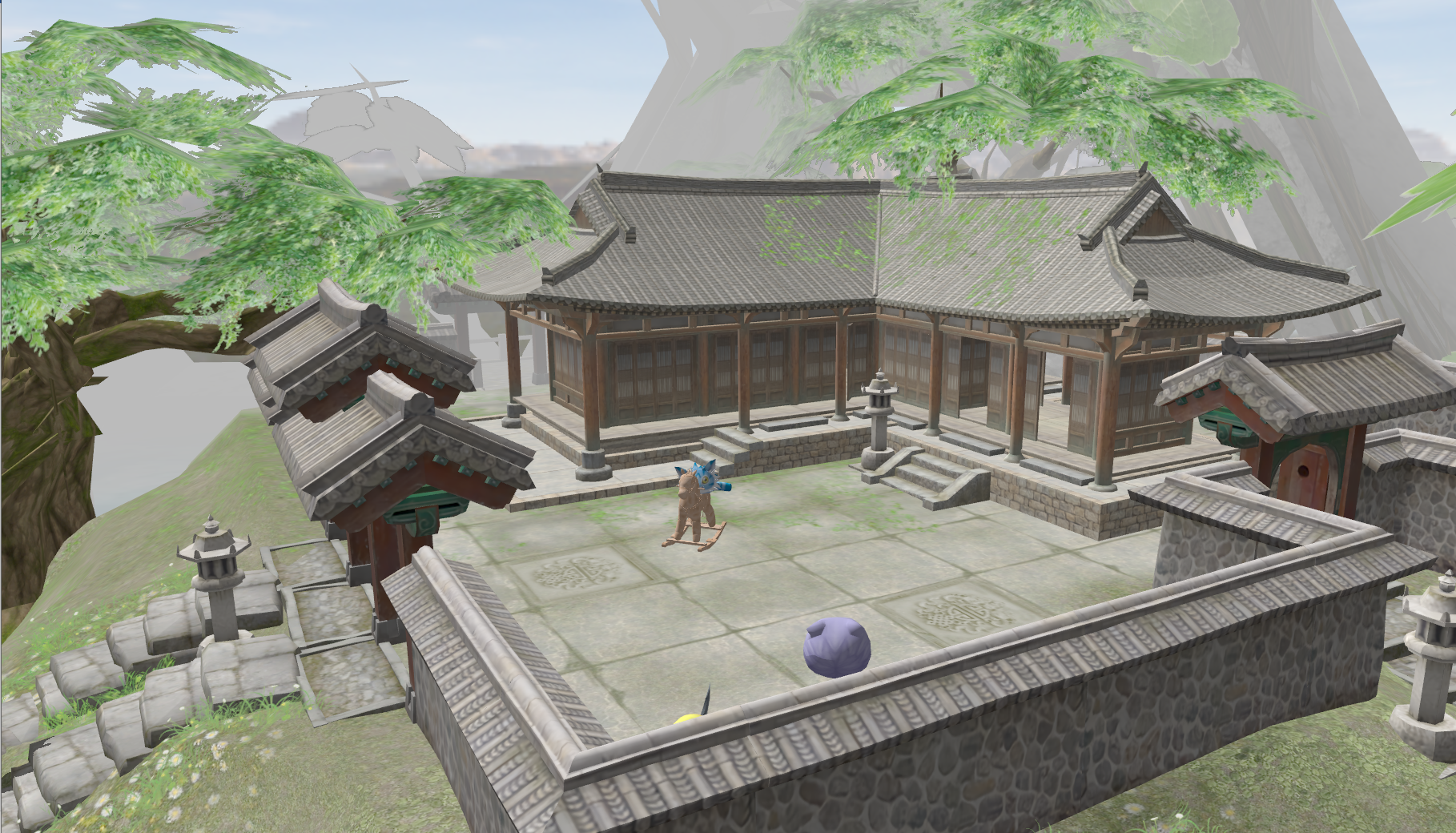
**4.4. Ierarhia de clase**

Clasele principale folosite in aceasta aplicatie sunt camera.cpp pentru functionalitatea camerei,model3d.cpp pentru incarcarea modelelor dorite clasele pentru shader pentru incarcarea shadere-lor precum si main-ul in care s-au implementat toate functionalitatile aplicatiei.Mai sunt puse la dispozitie fisierele de shader aplicate pe obiecte ,folosite tot in main.cpp.

Clasele principale folosite pt aceasta aplicatie sunt: camera.cpp, model3d.cpp si main.cpp. In camera.cpp am implementat in plus anumite metode de getter si setter pentru a putea verifica coloziunea inainte de a apela metoda de move, iar in main am implementat restul functionalitatilor precum: operatiile pe taste si efectele asociate tastelor, precum si desenarea obiectelor si animatiilor acestora.

**5. Prezentarea interfeței grafice utilizator / manual de utilizare**



Pentru folosirea functionalitatilor se vor folosi urmatoarele taste:

w-miscare camera fata

s-miscare camera spate

a-miscare camera stanga

d-miscare camera dreapta

mouse-rotatia camerei

up/down- efectul e lift

left/right- rotatia camerei

z-mod polygonal

x-mod wireframe

c-mod punctiform

r-start/sfarsit ploaie

b-mod noapte

n-mod zi

j si l-mutare cub lumina pentru testare umbre

t- fulgerul

p- mod de prezentare

**6. Concluzii și dezvoltări ulterioare**

Moduri de dezvoltare ulterioare ale proiectului:

* Sa adaug anumite obiecte random ca de exemplu fantome in timpul modului de noapte, sau sa apara niste easter egg-uri amuzante care sa apara doar daca camera se afla intr-un anumit punct al mapei

Proiectul a fost unul foarte interesant si am invatat foarte multe despre openGL precum si despre GLSL.

**7. Referințe**

**https://www.models-resource.com/**

**https://www.deviantart.com/**