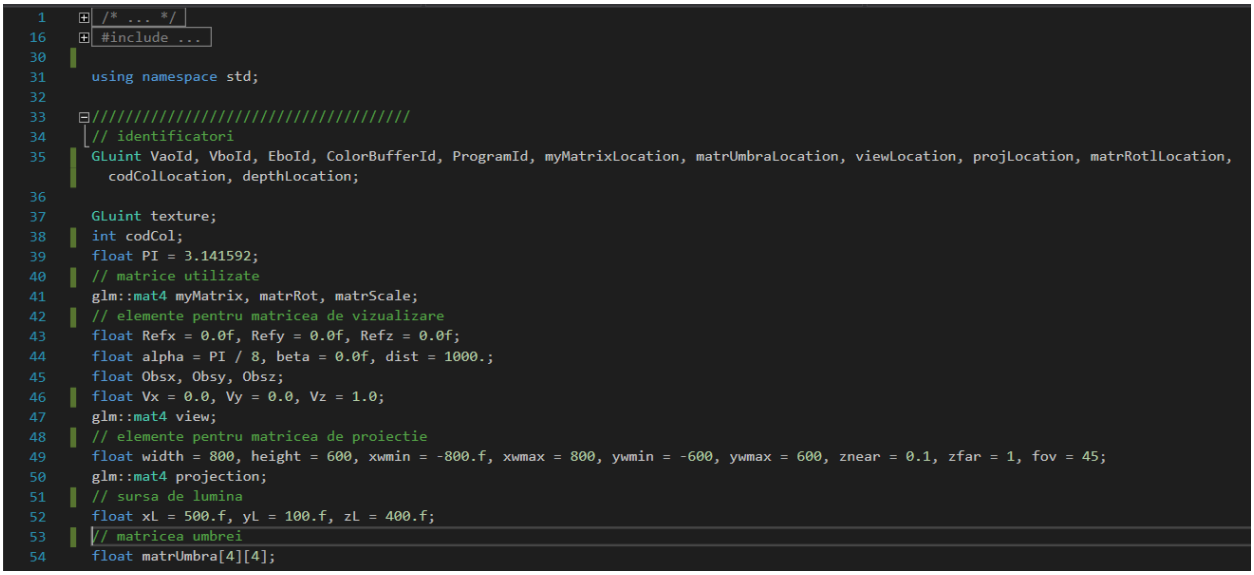
Documentatie tema 5

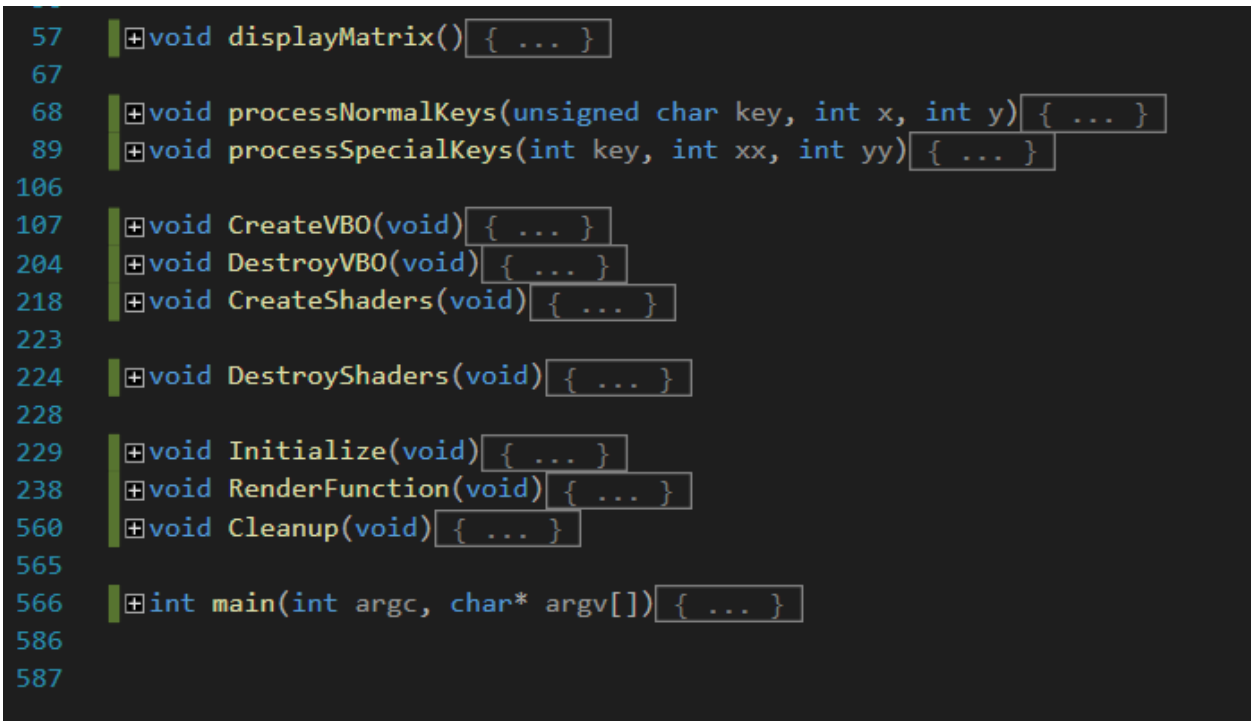
Lungu Andrei, grupa 332

Am avut ca punct de plecare codul suport “12_01_obiecte.cpp” din cadrul laboratorului.

Structura generala a proiectului:



```
1   /* ... */
16  #include ...
30
31  using namespace std;
32
33  ////////////////////////////////////
34    // identificatori
35  GLuint VaoId, VboId, EboId, ColorBufferId, ProgramId, myMatrixLocation, matrUmbraLocation, viewLocation, projLocation, matrRotlLocation,
      codColLocation, depthLocation;
36
37  GLuint texture;
38  int codCol;
39  float PI = 3.141592;
40  // matrice utilizate
41  glm::mat4 myMatrix, matrRot, matrScale;
42  // elemente pentru matricea de vizualizare
43  float Refx = 0.0f, Refy = 0.0f, Refz = 0.0f;
44  float alpha = PI / 8, beta = 0.0f, dist = 1000.;
45  float Obsx, Obsy, Obsz;
46  float Vx = 0.0, Vy = 0.0, Vz = 1.0;
47  glm::mat4 view;
48  // elemente pentru matricea de proiectie
49  float width = 800, height = 600, xwmin = -800.f, xwmax = 800, ywmin = -600, ywmax = 600, znear = 0.1, zfar = 1, fov = 45;
50  glm::mat4 projection;
51  // sursa de lumina
52  float xL = 500.f, yL = 100.f, zL = 400.f;
53  // matricea umbrei
54  float matrUmbra[4][4];
```



```
57   void displayMatrix() { ... }
67
68   void processNormalKeys(unsigned char key, int x, int y) { ... }
89   void processSpecialKeys(int key, int xx, int yy) { ... }
106
107  void CreateVBO(void) { ... }
204  void DestroyVBO(void) { ... }
218  void CreateShaders(void) { ... }
223
224  void DestroyShaders(void) { ... }
228
229  void Initialize(void) { ... }
238  void RenderFunction(void) { ... }
560  void Cleanup(void) { ... }
565
566  int main(int argc, char* argv[]) { ... }
586
587
```

Functiile de procesare a tastelor (processNormalKeys, processSpecialKeys) sunt utilizate pentru a permite deplasarea in cadrul scenei – rotatie la stanga sau la dreapta, micsorare sau marire imagine, miscare sus/ jos/ stanga/ dreapta.

In functia CreateVBO initializez varfurile, culorile si normalele primitivelor desenate: terenul, cubul (reprezinta trunchiul bradului si corpul casei), conul (reprezinta acele bradului) si piramida (acoperisul casei), cat si indicii pentru varfuri, bufferele si legarile VAO.

In functia DestroyVBO distrug vao si bufferele create anterior.

In metodele CreateShaders, DestoryShaders creez, respectiv distrug shaderele de varfuri si fragment ale aplicatiei.

Functia de initializare a fost folosita pentru setarea culorii de fond a ecranului si aplicarea metodei de creare shadere.

Cu ajutorul RenderFunction setez pozitia observatorului, reperul de vizualizare, matricea pentru umbra si creez VBO. Transmit variabilele uniforme shaderului de varfuri si shaderului de fragment pentru iluminare si efect ceata. In continuare, desenez primitivele: terenul, 2 randuri de brazi (translatie de a lungul axei ox, respectiv oy), cat si cele 4 case.

```
// desenare rand 1 brazi deplasare pe ox
for (int i = -1300; i < 1500; i = i + 300) {
    codCol = 0;
    glUniform1i(codColLocation, codCol);
    // desenare cub/trunchi
    myMatrix = glm::translate(glm::mat4(1.0f), glm::vec3(0.f + i, 0.f, -50.)) * glm::scale(glm::mat4(1.0f),
      glm::vec3(0.2, 0.2, 1.0));
    myMatrixLocation = glGetUniformLocation(ProgramId, "myMatrix");
    glUniformMatrix4fv(myMatrixLocation, 1, GL_FALSE, &myMatrix[0][0]);
    glDrawElements(GL_TRIANGLES, 36, GL_UNSIGNED_BYTE, (void*)(6));
```

Translatia se face in functie de variabila i din loop (analog pentru conurile bradului).

Pentru axa oy:

```
  // desenare rand 2 brazi deplasare pe oy
  for (int i = -1300; i < 1500; i = i + 300) {
      codCol = 0;
      glUniform1i(codColLocation, codCol);
      // desenare cub/trunchi
      myMatrix = glm::translate(glm::mat4(1.0f), glm::vec3(0.f, 0.f + i, -50.)) * glm::scale(glm::mat4(1.0f),
        glm::vec3(0.2, 0.2, 1.0));
```

La sfarsit eliberez memoria distrugand vbo si shaderele.

Functia main:

```
int main(int argc, char* argv[])
{

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGB | GLUT_DEPTH | GLUT_DOUBLE);
    glutInitWindowPosition(100, 100);
    glutInitWindowSize(1200, 900);
    glutCreateWindow("Tema 5 Lungu Andrei");
    glewInit();
    Initialize();
    glutIdleFunc(RenderFunction);
    glutDisplayFunc(RenderFunction);
    glutKeyboardFunc(processNormalKeys);
    glutSpecialFunc(processSpecialKeys);


    glutCloseFunc(Cleanup);
    glutMainLoop();

}
```