МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Лабораторная работа №7

По предмету «Интерактивные графические системы»

Лабиринт

Выполнил студент: Пшеничный Д.О.

Факультет, группа: ФКФН, ПО(б)-81

Руководитель работы: Богачев И.В.

Хабаровск – 2021г.

**Цель:** продемонстрировать знание основных возможностей графической библиотеки OpenGL, в том числе работу с системами координат, освещением, текстурированием, спецэффектами (прозрачность, туман).

**Задание:** написать проект, реализующий сложную трехмерную сцену с элементами анимации и интерактивности.

**Матрица карты лабиринта:**

//1 - стена

//2 - прозрачная дверь

int labirintMap[10][10] = {

{1, 1, 1, 1, 1, 1, 1, 1, 1, 1},

{1, 0, 0, 0, 1, 0, 0, 0, 0, 1},

{1, 0, 0, 0, 1, 0, 1, 0, 0, 1},

{1, 1, 2, 1, 1, 0, 1, 0, 0, 1},

{1, 0, 0, 0, 1, 0, 1, 0, 0, 1},

{1, 0, 0, 0, 2, 0, 1, 0, 0, 1},

{1, 1, 1, 1, 1, 0, 1, 1, 1, 1},

{1, 0, 0, 0, 0, 0, 0, 0, 0, 1},

{1, 0, 0, 0, 0, 0, 0, 0, 0, 1},

{1, 1, 1, 1, 1, 1, 1, 1, 1, 1},

};

**Расстановка объектов:**

for (unsigned int i = 0; i < 10; i++)

{

for (unsigned int j = 0; j < 10; j++)

{

if (labirintMap[i][j] == 1)

{

// настройка шейдера блока стены

glm::mat4 model = glm::mat4(1.0f);

model = glm::translate(model, glm::vec3(1.0f \* i, 0.0f, 1.0f \* j));

lightingShader.setMat4("model", model);

glDrawArrays(GL\_TRIANGLES, 0, 36);

}

else if (labirintMap[i][j] == 2)

{

float angle = 0.0f;

//поворот двери

if (labirintMap[i - 1][j] == 1 && labirintMap[i + 1][j] == 1)

angle = 0.0f;

else if (labirintMap[i][j - 1] == 1 && labirintMap[i][j + 1] == 1)

angle = 90.0f;

windowShader.use();

glBindVertexArray(transparentVAO);

glBindTexture(GL\_TEXTURE\_2D, transparentTexture);

glm::mat4 model = glm::mat4(1.0f);

if(angle == 90.0f)

model = glm::translate(model, glm::vec3(1.0f \* i, 0.0f, 1.0f \* j + 0.5f));

else if(angle == 0.0f)

model = glm::translate(model, glm::vec3(1.0f \* i - 0.5f, 0.0f, 1.0f \* j));

model = glm::rotate(model, glm::radians(angle), glm::vec3(0.0f, 1.0f, 0.0f));

windowShader.setInt("texture1", 1);

windowShader.setMat4("model", model);

windowShader.setMat4("projection", projection);

windowShader.setMat4("view", view);

glDrawArrays(GL\_TRIANGLES, 0, 6);

}

}

}

**Проверка столкновения со стеной:**

GLfloat velocity = this->MovementSpeed \* deltaTime;

if (direction == FORWARD)

if(labirintMap[(int)round((this->Position + this->Front \* velocity).x)][(int)round((this->Position + this->Front \* velocity).z)] != 1)

this->Position += this->Front \* velocity;

if (direction == BACKWARD)

if (labirintMap[(int)round((this->Position - this->Front \* velocity).x)][(int)round((this->Position - this->Front \* velocity).z)] != 1)

this->Position -= this->Front \* velocity;

if (direction == LEFT)

if (labirintMap[(int)round((this->Position - this->Right \* velocity).x)][(int)round((this->Position - this->Right \* velocity).z)] != 1)

this->Position -= this->Right \* velocity;

if (direction == RIGHT)

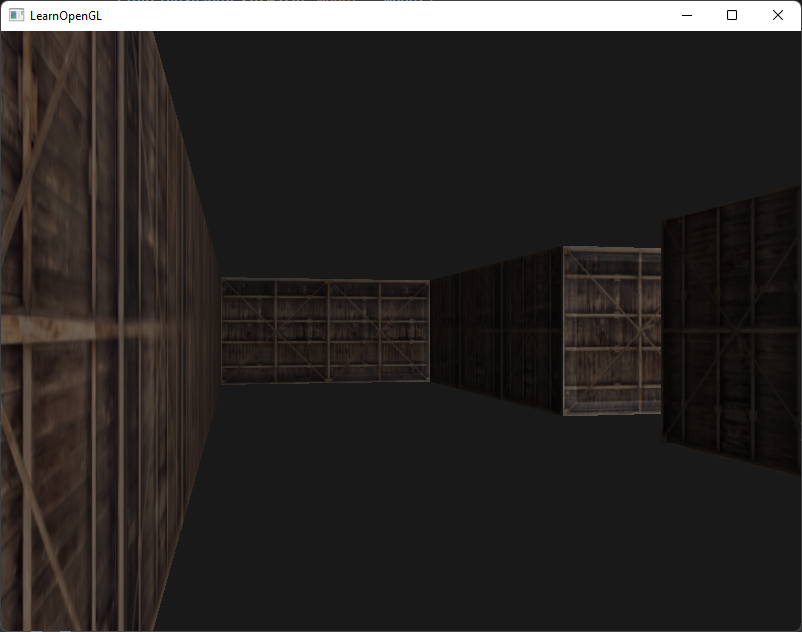
if (labirintMap[(int)round((this->Position + this->Right \* velocity).x)][(int)round((this->Position + this->Right \* velocity).z)] != 1)

this->Position += this->Right \* velocity;

**Результат работы:**

Изображение выглядит как текст

Автоматически созданное описание



Изображение выглядит как текст

Автоматически созданное описание