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 №4

По предмету «Интерактивные графические системы»

Свет

Выполнил студент: Пшеничный Д.О.

Факультет, группа: ФКФН, ПО(б)-81

Руководитель работы: Богачев И.В.

Хабаровск – 2021г.

**Цель:** научиться создавать в графическом приложении источники света с различными характеристиками.

**Задание:** расположить в произвольных точках пространства несколько источников света. Задать характеристики источников света: один источник должен быть точечный, другой — направленный (типа «прожектор»). Цвет источников подбирается из эстетических соображений.

**Отрисовка источников света:**

// Настройка источников света

// Направленный свет

lightingShader.setVec3("dirLight.direction", -0.2f, -1.0f, -0.3f);

lightingShader.setVec3("dirLight.ambient", 0.05f, 0.05f, 0.05f);

lightingShader.setVec3("dirLight.diffuse", 0.4f, 0.4f, 0.4f);

lightingShader.setVec3("dirLight.specular", 0.5f, 0.5f, 0.5f);

// Настройка одного из точечных источников света

lightingShader.setVec3("pointLights[0].position", pointLightPositions[0]);

lightingShader.setVec3("pointLights[0].ambient", 0.05f, 0.05f, 0.05f);

lightingShader.setVec3("pointLights[0].diffuse", 0.8f, 0.8f, 0.8f);

lightingShader.setVec3("pointLights[0].specular", 1.0f, 1.0f, 1.0f);

lightingShader.setFloat("pointLights[0].constant", 1.0f);

lightingShader.setFloat("pointLights[0].linear", 0.09f);

lightingShader.setFloat("pointLights[0].quadratic", 0.032f);

// Включение шейдера лампочек

lightCubeShader.use();

lightCubeShader.setMat4("projection", projection);

lightCubeShader.setMat4("view", view);

// Отрисовка источников(лампочек)

glBindVertexArray(lightCubeVAO);

for (unsigned int i = 0; i < 4; i++)

{

model = glm::mat4(1.0f);

model = glm::translate(model, pointLightPositions[i]);

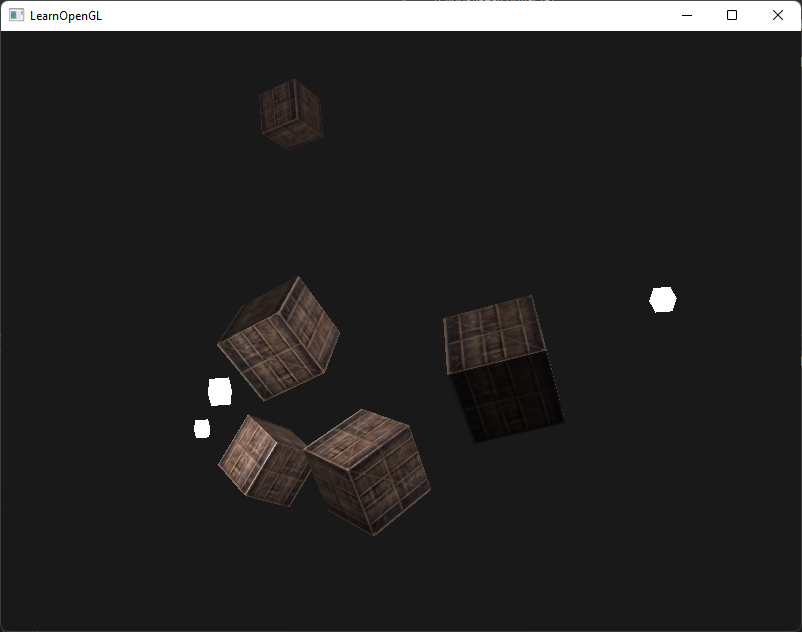
model = glm::scale(model, glm::vec3(0.2f)); // Используется VAO куба, поэтому уменьшаем размер

lightCubeShader.setMat4("model", model);

glDrawArrays(GL\_TRIANGLES, 0, 36);

}

**Результат работы:**

****

Изображение выглядит как текст

Автоматически созданное описание