МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра информационных систем и цифровых технологий

Контрольная работа

по дисциплине: «Компьютерная графика»

Выполнил: Волков В.В.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 91ПГ

Проверил: Чижов А.В.

Отметка о зачете:

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2023 г.

Орел, 2022

**КРАТКОЕ ОПИСАНИЕ ПРОЕКТА**

**Проект представляет собой модель тюрьмы. Сцена представляет из себя комнату с предметами соответствующими тюрьме и освещением. Представленная модель затекстурирована пятью видами текстур, при этом текстуры можно отключить по нажатию кнопки. Кроме того модель освещена двумя источниками света: направленным (солнце) и точечным (лампа), которые могут быть отключены по отдельности по нажатию соответствующей кнопки. Реализована возможность перемещения наблюдателя по сцене при помощи кнопок WS и стрелочек. Все кнопки управление не регистрозависимые, но обязательно язык должен стоять на английском раскладке. Ниже и в консоли представлено управление в проекте.**

**Управление:**

**Стрелочки - перемещение камеры**

**WS/ws – Перемещение наблюдателя вперед и назад**

**T/t - включение/выключение текстур**

**R/r - включение/выключение источника направленного света (солцна)**

**E/e - включение/выключение источника точечного света (лампы)**

**КРАТКОЕ ОПИСАНИЕ МОДЕЛИ**

**Модель представляет из себя модель тюрьмы, которая представляет из себя комнату со стенами, полом, кроватью, тумбочкой, лампы, солнцем. Все объекты затекстурированы и освещены. Внешний вид модели представлен на рисунке 1.**

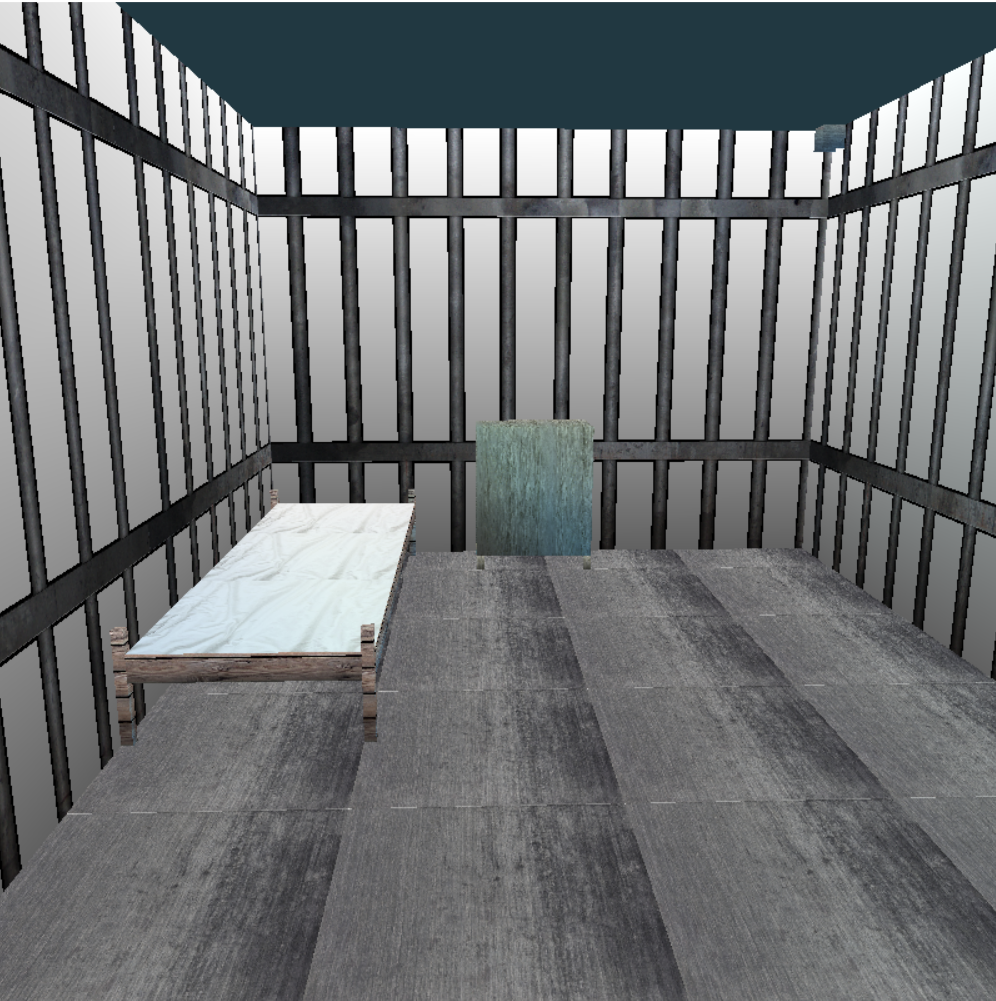


Рисунок 1 – Модель с позиции наблюдателя

**СОЗДАНИЕ МОДЕЛИ**

**Для отображения модели на экране необходимо проинициализировать соответствующие значения.**

**Инициализация модели происходит при вызове функции** Initialize()**. Перед началом инициализации подключаются расчет освещения, два источника освещения, свойства материала и автоматическую нормализацию нормалей.**

**Для генерации сцены создадим функцию DrawRect(), которая имеет 3 параметра длина, ширина, высота, которая отрисовывает прямоугольник по заданным параметрам. Также внутри этой функции пропишем нормали для корректного освещения.**

**Все объекты нашей сцены будут рисоваться при помощи прямоугольников построенных нашей функцией. Для отрисовки каждого объекта создана соответствующая функция.**

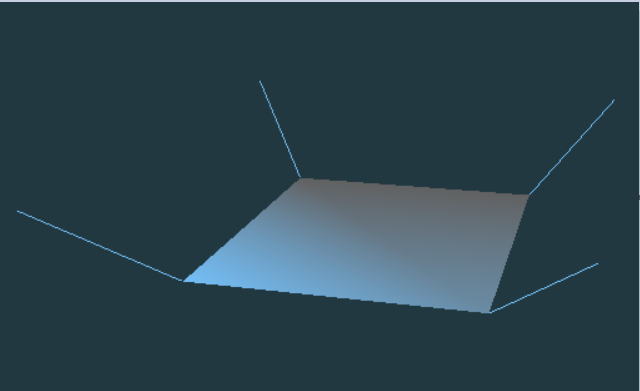
**Таблица для всякий нужд**

|  |  |  |  |
| --- | --- | --- | --- |
| **Тест1** | **Тест1** | **Тест1** | **Тест1** |
| **Тест2** | **Тест2** | **Тест2** | **Тест2** |
| **Тест3** | **Тест3** | **Тест3** | **Тест3** |
| **Тест4** | **Тест4** | **Тест4** | **Тест4** |
| **Тест5** | **Тест5** | **Тест5** | **Тест5** |

**ОСВЕЩЕНИЕ СЦЕНЫ**

**Для настройки освещения вызывается функция** InitLight()**.**

**Для создания корректного освещения сцены нормали были установлены в функции создания прямоугольника. Расположение векторов нормали представлено на рисунке 2.**



**Рисунок 2 – Расположение векторов нормалей**

**Первый источник света будет направленным, он же наше солнце, которое находится над нашей комнатой. Оно находится в бесконечности и свет из него будет распространяться в заданном направлении. Второй источник света лампочка расположенная в углу комнаты и имеет тип "Прожектор".**

**ТЕКСТУРИРОВАНИЕ ОБЪЕКТОВ МОДЕЛИ**

**Для загрузки текстур воспользуемся сторонней библиотекой stb\_image.h, которую необходимо добавить в проект. Все текстуры для нашей модели будут загружаться из файлов с расширением ".png". Для загрузки текстуры используется две функции LoadTextute() и InitTexture(). InitTexture() – запускает цикл по всем элементам контейнера map хранящем id – текстуры загруженной в видеокарту и путь к картинке и вызывает LoadTextute(), в которую передаем все параметры из нашего контейнер. При выходе за границы текстура будет повторяться. При уменьшении или увеличении текстуры цвета будут браться от ближайшей точки.**

**Все текстуры, используемые в модели располагаются в папке с кодом.**

**Все текстуры были взяты из открытых Интернет-источников**

1. **Пол – https://www.pngegg.com/ru/png-ndflj/download**
2. **Стены – https://phonoteka.org/27452-fon-reshetka-tjurmy.html**
3. **Простыня – https://designboom.ru/catalog/product/prostynya-220kh240-sm-belaya/**
4. **Все виды дерева – https://opengameart.org/content/wall-grass-rock-stone-wood-and-dirt-480**

**Перед применением текстуры разрешается использование текстур при помощи команды glEnable(GL\_TEXTURE\_2D). Перед непосредственным применением текстура устанавливается как активная.**

**Текстуры для всех объектов нашей модели накладываются на прямоугольники. Для этого используем команду** glBindTexture(GL\_TEXTURE\_2D, id\_текстуры), которая в target GL\_TEXTURE\_2D помещает наш объект текстуры, передав туда его id перед каждым прямоугольником который хотим затекстурировать.

**ПЕРЕМЕЩЕНИЕ НАБЛЮДАТЕЛЯ ПО СЦЕНЕ**

**Для имитации движения камеры используются функции** glutKeyboardFunc(keyboardFunc), glutSpecialFunc(specialkeys). Первая функция нужна для обработки событий клавиш клавиатуры (WS,E,R,T). Для движения вперед и назад мы синус от угла поворота по каждой оси умножаем на коэффициент перемещения.

**Для поворота камеры используются функция** glutSpecialFunc, которая обрабатывает **стрелочки на клавиатуре, которые позволяют осматриваться по осям y и z на 360 градусов.**