

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

Компьютерная графика:
Курсовая работа
«Визуализация 3D-объекта»

Студент гр. 5383

Допира В.Е.

Преподаватель

Герасимова Т.В.

Санкт-Петербург

2018

Задание

Необходимо создать сцену (фотореалистичность желательна). Сцена управляема – можно облететь вокруг, изменить положение источников света.



Теоретические сведения

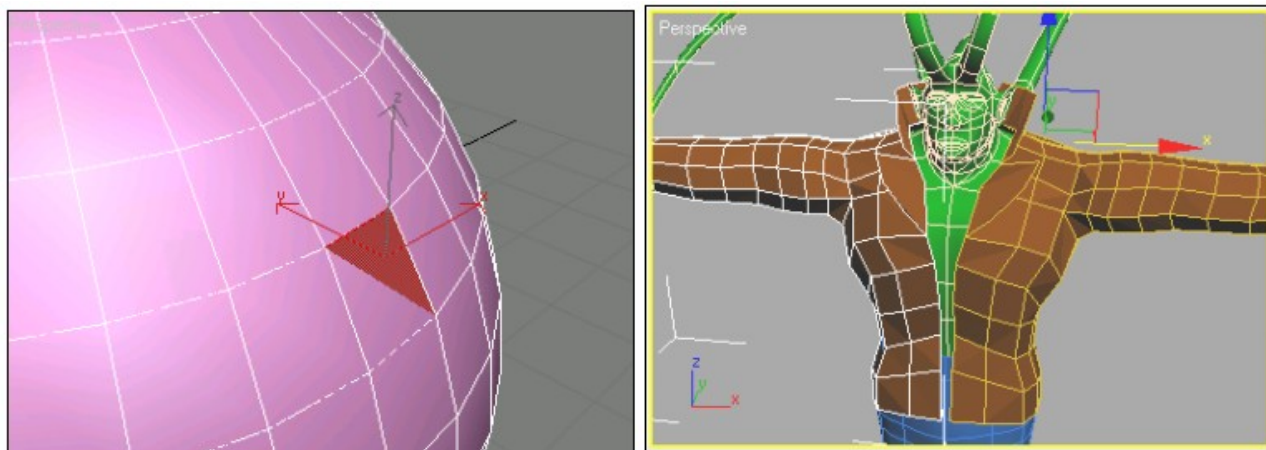
3D графика – это процесс создания объемной модели при помощи специальных компьютерных программ.

Необходимость в компьютерном моделировании сцен реального и виртуального миров возникает во многих областях современной человеческой деятельности. При работе с трехмерной графикой существуют пять последовательных этапов, которые необходимы для получения готового продукта:

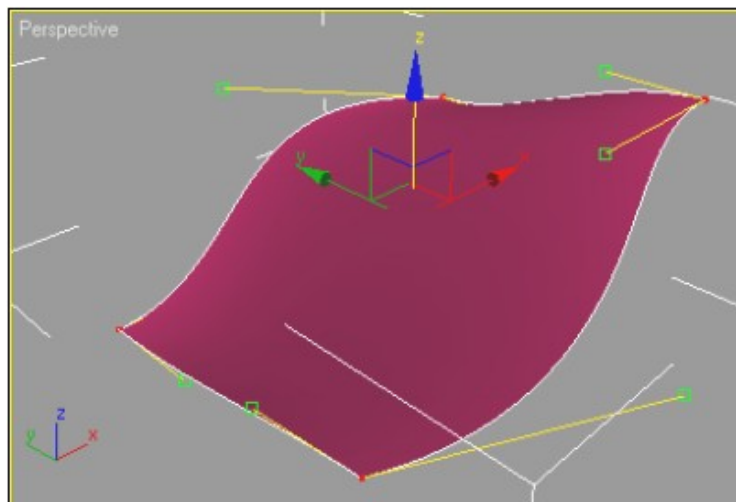
1. Моделирование — создание объектов, которые будут на сцене.
2. Текстурирование (использование материалов) — определение свойств поверхностей объектов для имитации различных свойств реальных предметов (цвет, фактура, прозрачность, яркость и т. д.).
3. Освещение — добавление и размещение источников света подобно тому, как это делается в театральной студии или на съемочной площадке.
4. Анимация — создание движения по ключевым кадрам.
5. Визуализация — создание конечного изображения или анимации.

Для успешного моделирования важно предварительно продумать, каким образом наблюдаемые объекты можно превратить в компьютерные модели. Моделирование поверхностей основано на создании произвольных поверхностей. Для этого используются различные математические модели. Выделяют несколько видов моделирования:

1. Многоугольные каркасы — это сложные модели, созданные из множества многоугольных поверхностей, которые сглажены в процессе визуализации. Такое моделирование основано на манипулировании непосредственно вершинами, ребрами и гранями. Плоскости, образующие многогранник, называются гранями (Polygon). Линии пересечения граней называются ребрами (Edges). Точки пересечения ребер называются вершинами (Vertex). Три вершины в пространстве образуют треугольную грань (Face).



2. Лоскутки (Patches) строятся на основе сплайнов и могут изменяться с помощью контрольных точек. Образующие сплайны располагаются по краям создаваемой поверхности.



3. Неоднородные рациональные В-сплайны (NURBS) — технология, предназначенная для создания плавных форм и моделей. Она основана на специальном математическом аппарате.

Создаваемые трехмерные модели представляют собой набор координат в пространстве. Для того чтобы их можно было увидеть, должны быть закрашены их грани — многоугольные плоскости, образуемые ребрами. Только после этого модель становится видимой.

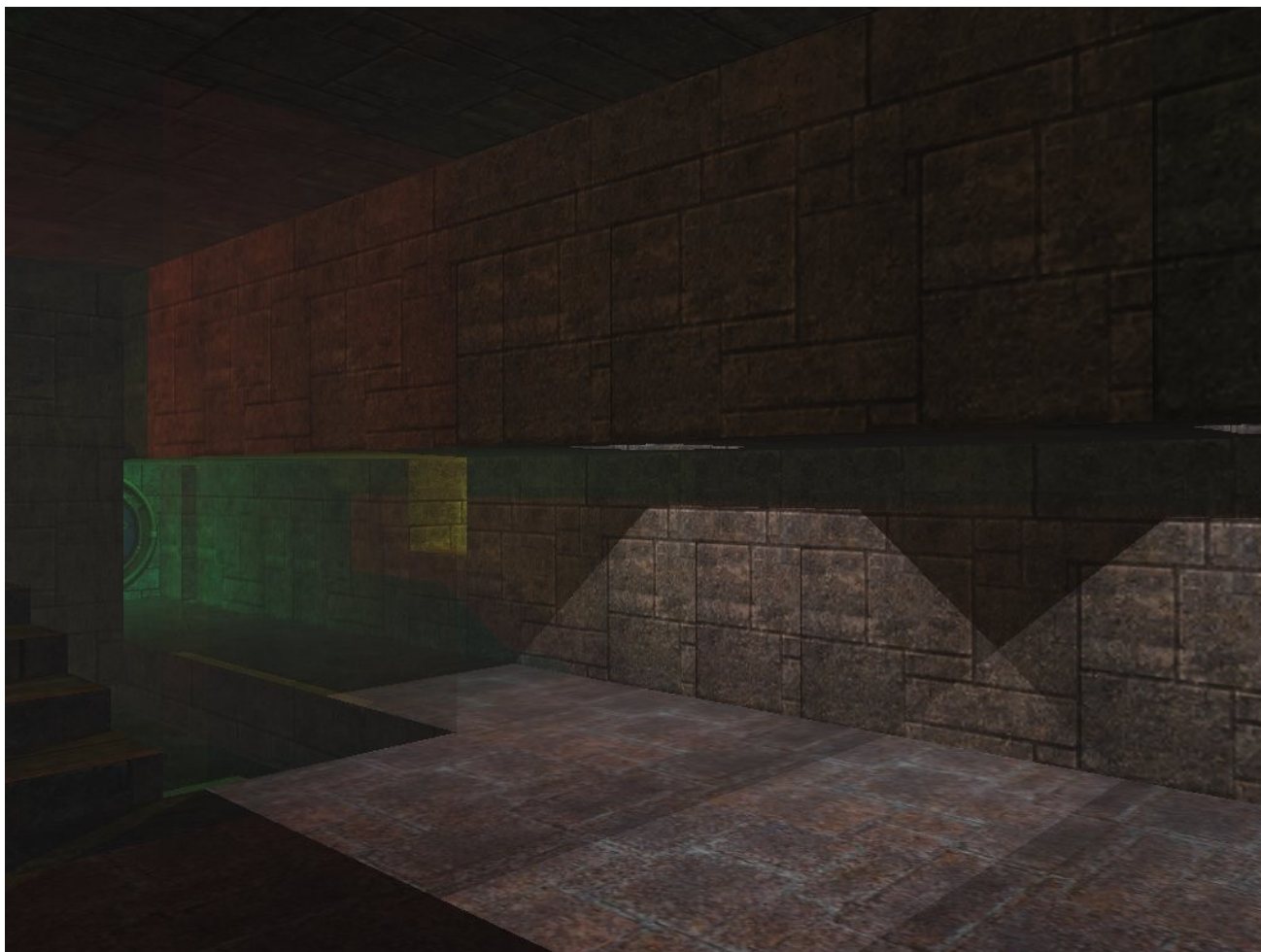
На этапе текстурирования поверхностям моделей придают вид реальных материалов: дерева, металла, пластика и так далее.

Цвет — один из простейших свойств материала. Однако даже использование цвета имеет множество аспектов. Цвет может быть основным, определяющим покрытие всего объекта, обтекающим определяющий влияние фонового освещения, зеркальным определяющим цветом наиболее ярких участков блестящей поверхности объекта и т. д.

В процессе создания материалов широко используются карты текстур, в простейшем виде растровые изображения реальных объектов. Кроме этого используются процедурные карты — изображения, которые генерируются программным путем. В процессе создания материала можно использовать несколько карт текстур. Точное размещение материала на поверхности объекта достигается, благодаря так называемым координатам проецирования (UVW Map), когда растровое изображение интерактивно размещается на поверхности объекта. Умелое использование материалов позволяет сэкономить много времени и добиться прекрасных результатов.



Освещение не должно быть навязчивым. Оно подчеркивает свойства сцены, выполненной в результате моделирования и использования материалов. На рисунке изображен пример с несколькими источниками света:



В области компьютерной графики одной из наиболее важных задач является получение реалистичного и правдоподобного конечного изображения. Основным критерием фотореалистичности и правдоподобности трехмерного изображения является точное отображение освещения, теней, отражающих и поглощающих свойств материалов объектов. Визуализация является заключительным этапом работы над моделируемой сценой. На этом этапе компьютер превращает математическую модель сцены в форму доступную для визуального восприятия. Этот процесс называется рендерингом (render). На этом этапе существенную роль играет быстродействие компьютера и параметры видеокарты.

Анимация — иллюзия движения, созданного через просмотр быстро сменяющихся кадров.

Реализация

Для построения всех объектов на сцене был написан класс SimpleObject3d, который содержит в себе вершинный и индексный буфер для построения объекта, а также информацию о повороте, перемещении и масштабе для конкретного объекта.

```
class SimpleObject3D : public Transformational
{
public:
    SimpleObject3D();
    SimpleObject3D(const QVector<VertexData> &vertexData, const QVector<GLuint> &indexes,
const QImage &texture);
    ~SimpleObject3D();
    void init(const QVector<VertexData> &vertexData, const QVector<GLuint> &indexes, const
QImage &texture);
    void rotate(const QQuaternion &r) override;
    void translate(const QVector3D &t) override;
    void scale(const float &s) override;
    void setGlobalTransform(const QMatrix4x4 &g) override;
    void draw(QOpenGLShaderProgram *program, QOpenGLFunctions *functions) override;
    static QVector2D rotateMatrix(QVector2D point, double angle);
private:
    QOpenGLBuffer m_vertexBuffer;
    QOpenGLBuffer m_indexBuffer;
    QOpenGLTexture *m_texture;
    // составляющие модельно-видовой матрицы
    QQuaternion m_rotate;
    QVector3D m_translate;
    float m_scale;
    QMatrix4x4 m_globalTransform;
};
```

Этот класс - наследник класса Transformational. Он является интерфейсом (абстрактный класс без полей), который необходимо реализовывать всем классам, которые будут отображаться на сцене.

```

class Transformational
{
public:
    virtual ~Transformational() {}
    virtual void rotate(const QQuaternion &r) = 0;
    virtual void translate(const QVector3D &t) = 0;
    virtual void scale(const float &s) = 0;
    virtual void setGlobalTransform(const QMatrix4x4 &g) = 0;
    virtual void draw(QOpenGLShaderProgram *program, QOpenGLFunctions *functions) = 0;
};

```

В сцене для визуализации объекты моделируются с помощью следующих фигур:

- 1) Диск (Disk);
- 2) Пояс (Belt)- две окружности в плоскости XY, сшитые между собой с помощью треугольников;
- 3) Цилиндр (Cylinder).

Для генерации этих объектов был написан отдельный класс FigureBuilder.

```

class FigureBuilder
{
public:
    FigureBuilder();
    static SimpleObject3D *initDiskSector(const QImage &texture, QVector3D point =
QVector3D(0.0f, 0.0f, 0.0f), double r = 1.0, double angle = M_PI / 2, double step = 0.1, bool invert =
false);
    static SimpleObject3D *initBelt(const QImage &texture, QVector3D center1, QVector3D center2,
double r1, double r2, double step = 0.1);
    static SimpleObject3D *initCylinder(QVector3D center1, QVector3D center2, double r1, double
r2, double step);
};

```

Созданные объекты объединяются в группу, к которой можно применять различные преобразования (поворот, перемещение, масштаб), применяющиеся ко всем объектам в группе. Именно эта группа и является объектом на сцене (бочка).


```

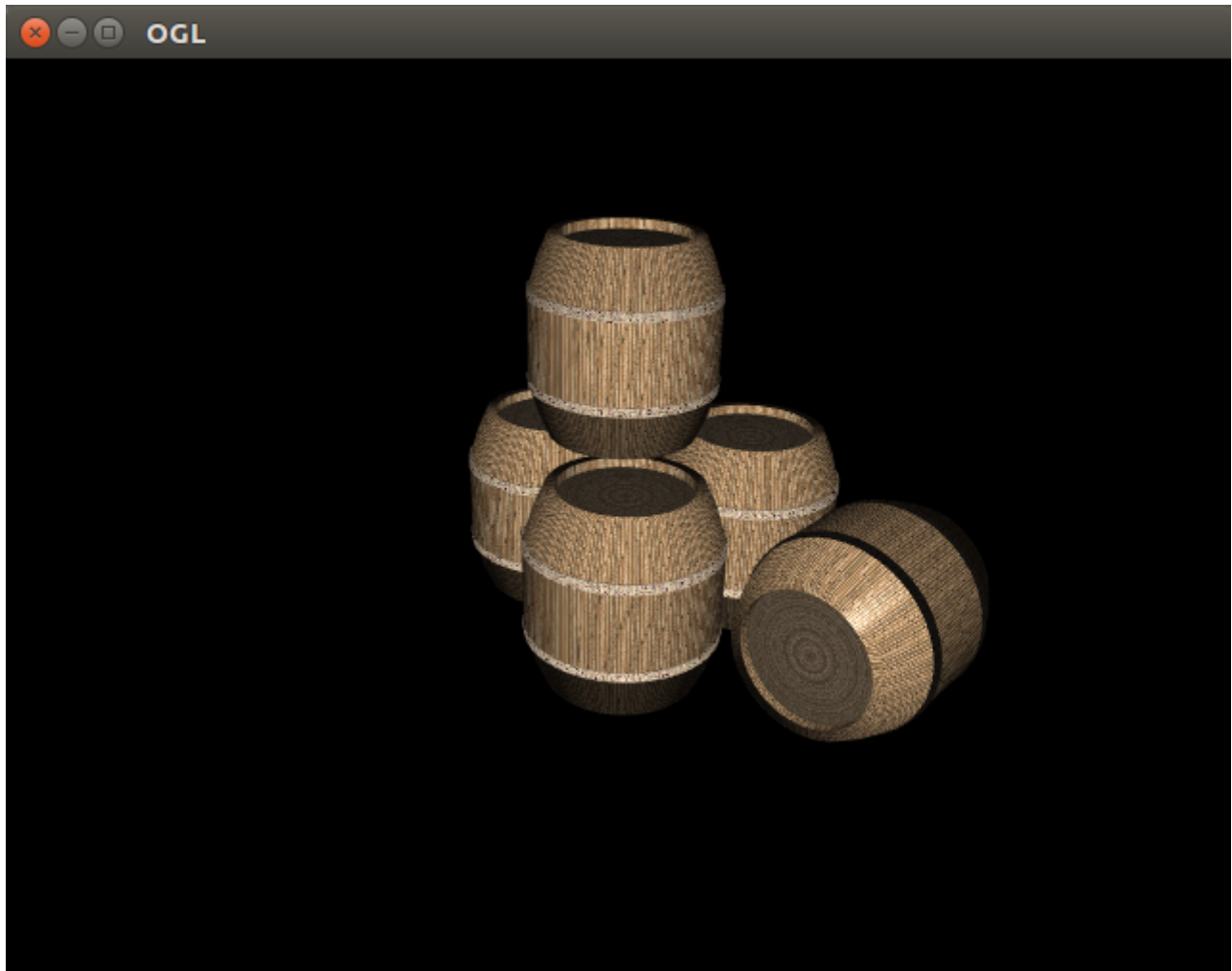
class Group3D : public Transformational
{
public:
    Group3D();
    ~Group3D();
    void rotate(const QQuaternion &r) override;
    void translate(const QVector3D &t) override;
    void scale(const float &s) override;
    void setGlobalTransform(const QMatrix4x4 &g) override;
    void draw(QOpenGLShaderProgram *program, QOpenGLFunctions *functions) override;
    void addObject(Transformational* obj);
    void removeObject(Transformational* obj);
    void removeObject(int index);
private:
    // составляющие модельно-видовой матрицы
    QQuaternion m_rotate;
    QVector3D m_translate;
    float m_scale;
    QMatrix4x4 m_globalTransform;
    QVector<Transformational*> m_objects;
    void setViewMatrix();
};

```

Группа рисуется в виджете с наложением текстур и шейдерным освещением.

Тестирование

Результаты тестирования представлены на снимках экрана.



Вывод

В курсовой работе на практике была визуализирована интерактивная 3D сцена.

Список использованной литературы:

1. Меженин А.В. Технологии 3d моделирования для создания образовательных ресурсов. Учебное пособие. СПб., 2008.
2. Дж. Ли, Б. Уэр. Трёхмерная графика и анимация. Вильямс, 2002.
3. Д. Херн, М. П. Бейкер. Компьютерная графика и стандарт OpenGL. 2005.