

Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Архитектура вычислительных систем

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
на тему

«3D движок»

БГУИР КП 1-40 04 01 009 ПЗ

Студент гр. 753504
А.Д. Матюшонак
Руководитель
Ассистент кафедры информатики
А. В. Леченко

Минск 2019

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»
Факультет компьютерных систем и сетей
Кафедра информатики

УТВЕРЖДАЮ
Заведующий кафедрой ИИТП
_____ Волорова Н. А.
« ____ » _____ 2019 г.

ЗАДАНИЕ
по курсовому проекту

Группа 753504

Студенту Маиюшонку Александру Денисовичу

- 1.Тема проекта:** 3D движок
- 2.Сроки сдачи студентом законченного проекта:** 21.12.2019 г.
- 3.Исходные данные к проекту:** Для написания курсового проекта была выбрана среда разработки Microsoft Visual Studio 2019 а также язык программирования C#.
- 4.Содержание расчетно-пояснительной записки** (перечень подлежащих разработке вопросов):
Описание работы 3D движка и его алгоритмов
Список использованных источников.
- 5.Перечень графического материала** (с указанием обязательных чертежей и графиков):

- 6.Консультанты по проекту:** Леченко А. В.
- 7.Дата выдачи задания:** 20.09.2019 г
- 8.Календарный график работы над проектом на весь период проектирования** (с указанием сроков выполнения и трудоемкости отдельных этапов):

№ п/п	Наименование этапов курсового проекта	Срок выполнения этапов проекта	Примечание
1.	1-я опрoцентoвка (введение, раздел 1)	16.10.2019	30%
2.	2-я опрoцентoвка (раздел 2, раздел 3, раздел 4.)	28.10.2019	60%
3.	3-я опрoцентoвка (демонстрация, заключение)	12.11.2019	100%
4.	Защита курсового проекта	21.12.2019	Согласно графику

Руководитель _____ (Леченко А. В.)
Задание принял к исполнению 15.09.2019 _____ (_____)

Содержание

1. Описание работы 3D движка и его алгоритмов.....	4
1.1. Представление объекта для 3D проецирования	4
1.2. Получение цвета примитива.....	4
1.3. Поворот объекта	5
1.4. Перемещение объекта	5
1.5. проецирование 3D объекта на плоскость экрана	6
1.6. Загрузка объектов	6
1.7. Отрисовка объектов.....	6
Список использованной литературы.....	7

1. Описание работы 3D движка и его алгоритмов

1.1. Представление объекта для 3D проецирования

Для отображения объекта нам важны следующие вещи: возможность достижения высокой детализации, и не большое количество суммарных данных которые его хранят, что кроме меньшего количества данных для хранения так же означает меньшее количество данных для обработки и ускоряет потенциальную работу рендеринга.

И так, отображать данные мы собираемся на мониторе, т.е. в 2D плоскости, тогда в качестве “составных кирпичиков” наших объектов разумно будет выбрать не 3D фигуры, а 2D, которые будут составлять лишь оболочку нашей внутри пустой фигуры. И так, так-же нам нужно как можно меньше точек в нашей фигуре, чтобы обеспечить наибольшую плавность и детализацию фигур, линия нам не подойдет, поскольку она плоская, а нам нужна ширина, будем использовать треугольник. Так же на первый взгляд неплохо может подойти круг, действительно для него даже потребуется хранить 2 параметра: точку центра и радиус в отличие от того же треугольника с его тремя вершинами. Однако с кругами будет много проблем в плане производительности при их отрисовке, определении границ и других, которые мы рассмотрим далее.

Для всего объекта будем использовать mesh: список, который хранит все треугольники, которые составляют объект.

1.2. Получение цвета примитива

Для получения нового цвета примитива будем пользоваться: источником света, изначальным цветом плоскости и углом поворота примитива к источнику света.

Свет будет тем ярче чем меньше угол между нормалью поверхности с направлением света, для определения этого параметра возьмем скалярного произведения нормализованной нормали примитива и нормализованным направлением света: $\text{double dp} = \text{normal.DotProduct}(\text{lightDirection});$

В дальнейшем этот параметр будет передан в функцию getColor, которая получит параметр степени яркости $a = 255 * \text{dp}$, если яркость максимальная то цвет отобразится в своем изначальном заданном виде, иначе станет темнее $1/\text{dp}$ раз. Если $a < 25$, то оставим $a = 25$, чтобы примитив был минимально виден при любом даже нулевом освещении.

1.3. Поворот объекта

Для поворота объекта в пространстве будем отдельно использовать матрицы поворота объекта по осям: X, Y, Z, для каждого из которых объявлена матрица, в которой:

- Вращение вокруг оси x :

$$M_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}.$$

- Вращение вокруг оси y :

$$M_y(\alpha) = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix}.$$

- Вращение вокруг оси z :

$$M_z(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Мы предварительно перемножим все наши матрицы поворота чтобы получить общую матрицу поворота, после чего умножим на неё каждый из векторов (точек) нашего треугольника и в конечном счете повернем каждый из треугольников нашего объекта.

1.4. Перемещение объекта

Для перемещения нашего объекта в пространстве будем использовать матрицу, в которой все диагональные элементы равны 1, все элементы 3-ей строки равны соответственно x , y , z координатам в которые мы перемещаем наш объект. После чего умножив эту матрицу на каждую из точек каждого из треугольников нашего объекта, мы получим что каждая из точек переместилась в координаты x , y , z .

1.5. проецирование 3D объекта на плоскость экрана

Т.К. экран может быть любого разрешения, то будем использовать коэффициент соотношения сторон $a = h/w$, где h – высота экрана, w – ширина экрана.

Объект будет занимать в нашей области пространство зависящее от нашего угла обзора, чем она больше тем больше пространства для обзора нам доступна и следовательно тем меньше на нем будет занимать места объект, область видимости нашей камеры в свою очередь регулируется углом O . Размеры объекта по ширине и высоте изменятся на величину $F = 1/\tan(O/2)$.

Ширина и длинна объектов на проекции уменьшаются пропорционально тому как увеличивается их дальность, z , тогда $x = x/z$, $y = y/z$

Итоговое преобразование будет выглядеть как: $[x * (a * F) / z, y * F / z, z]$

1.6. Загрузка объектов

Для хранения и последующей загрузки будем использовать 1 из вариантов файлов, что использует blender. В начале он хранит список из всех вершин с номерами и координатами, помечая это префиксом v а после ссылается на них, в префиксе f который означает треугольную плоскость.

1.7. Отрисовка объектов

Будем отрисовывать треугольники для которых мы уже определили яркость в зависимости от источника света, заполняя их содержимое заданным цветом, так-же в методе определена отрисовка точек треугольников и выделение их границ, позволяя лучше разглядеть из каких составных частей состоит фигура.

Чтобы отрисовать весь объект определим единичную матрицу над которой выполним все предварительные преобразования вроде поворота, перемещения, и тп, чтобы впоследствии один раз выполнить все преобразования над нашими точками.

Если угол между нормалью к треугольнику над которым мы проверили все преобразования по перемещению в пространстве и направлением нашей камеры < 90 , то мы его не видим, следовательно не нужно его отрисовывать.

Умножим каждый из векторов нашего треугольника на матрицу проекции на 2D пространство, нормализуем полученный результат. Чтобы отрисовывать треугольники в правильном порядке (сначала самые отдаленным, а поверх них те что ближе) находим среднее арифметическое между всеми точками треугольников и сортируем их по этому параметру от большего к меньшему.

Список использованной литературы

1. 3D engine [Электронный ресурс]. - Электронные данные. - Режим доступа: (<https://youtu.be/ih20l3pJoeU>)
2. Краткий курс компьютерной графики: пишем упрощённый OpenGL своими руками [Электронный ресурс]. - Электронные данные. - Режим доступа: (<https://habr.com/ru/post/248153/>)
3. Создаём собственный программный 3D движок [Электронный ресурс]. - Электронные данные. - Режим доступа: (<https://habr.com/ru/post/334580/>)