Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Чувашский государственный педагогический университет

им. И.Я. Яковлева»

Физико-математический факультет

Информатики и информационно-коммуникационных технологий

КУРСОВАЯ РАБОТА

Создание ПО для просмотра 3D объектов на языке программирования Python

с использованием framework Numpy и Pygame.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_ Григорьев Ю.В.

подпись, дата

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ Пигальцев Т.Е.

номер группы, зачетной книжки подпись, дата инициалы,

Чебоксары2021

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc70372924)

[1.ИЗУЧЕНИЕ ОСНОВ ГРАФИЧЕСКИХ СИСТЕМ 3D-ДВИЖКОВ 5](#_Toc70372925)

[1.1. Точки, векторы и базовый принцип трехмерного пространства 5](#_Toc70372926)

[1.2. Система координат 6](#_Toc70372927)

[1.3. Взаимная перпендикулярность 6](#_Toc70372928)

[1.4. Точки и векторы 7](#_Toc70372929)

[1.5. Основы линейных преобразований 8](#_Toc70372930)

[1.6. Повороты 8](#_Toc70372931)

[1.7. Масштабирование 9](#_Toc70372932)

[1.8. Отсечение 10](#_Toc70372933)

# ВВЕДЕНИЕ

Представление данных на мониторе компьютера в графическом виде впервые было реализовано в середине 50-х годов для больших ЭВМ, применявшихся в научных и военных исследованиях. С тех пор графический способ отображения данных стал неотъемлемой частью подавляющего числа компьютерных систем, в особенности персональных.

Компьютерная графика — это специальная область информатики, изучающая методы и средства создания и обработки изображений с помощью программно-аппаратных вычислительных комплексов. Она охватывает все виды и формы представления изображений, доступных для восприятия человеком либо на экране монитора, либо в виде копии на внешнем носителе (бумага, киноплёнка, ткань и прочее).

Развитие технологий привело к быстрому росту в области компьютерной техники и программного обеспечения. Обычным явление благодаря массовому распространению программного обеспечения по созданию компьютерной графики и трёхмерного моделирования. Программы трёхмерной графики - самый интересный по своим возможностям, но сложны в освоениях приложения.

Трёхмерная графика уже прочно вошла в нашу жизнь и не замечаем её. Разглядывая рекламный щит, играя в игру, в которой взрывается машина это результат мастера трёхмерной графики. Область применения трёхмерной графики необычайно широк начиная от: рекламы, киноиндустрий и дизайна интерьера и производства компьютерных игр.

Использование компьютерных технологий при проектировании и разработке дизайна интерьера помогает увидеть конечный вариант задолго до того, как обстановка будет воссоздана. Трехмерная графика позволяет создавать трехмерные макеты различных объектов (кресел, диванов, стульев и т.д.), повторяя их геометрическую форму и имитируя материал, из которого они созданы. Чтобы получить полное представление об определенном объекте, необходимо осмотреть его со всех сторон, с разных точек, при различном освещении. Трехмерная графика позволяет создать демонстрационный ролик, в котором будет запечатлена виртуальная прогулка по этажам будущего коттеджа, который только начинает строиться.

Цель курсовой работы: Создание ПО для просмотра 3D объектов

Для достижения этой цели, необходимо решить следующие задачи:

Изучить основы графических систем 3D-движков;

Изучить основы языка программирования Python и framework’s Django и Pygame;

# 1.ИЗУЧЕНИЕ ОСНОВ ГРАФИЧЕСКИХ СИСТЕМ 3D-ДВИЖКОВ

## 1.1. Точки, векторы и базовый принцип трехмерного пространства

Для трёхмерной графики требуется концепция трёхмерного пространства. Наиболее часто из всех видов пространств используется декартово пространство, которое позволяет нам применять декартовы координаты (стандартная запись (x, y) и двухмерные графики).

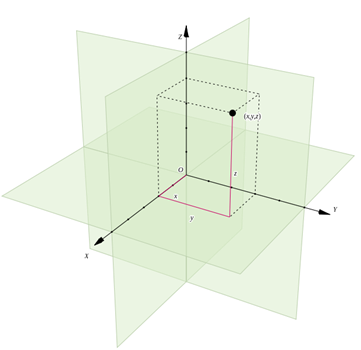
Трёхмерное декартово пространство даёт нам оси x, y и z (описывающие положение по горизонтали, вертикали и в глубину). Координаты любой точки в этом пространстве обозначаются как несколько чисел (в нашем случае три числа, потому что у нас три оси). На двухмерной плоскости запись обозначается как (x, y), а в трёхмерном пространстве — как (x, y, z). Эта запись (кортеж) показывает положение точки относительно исходной точки пространства (которая обычно обозначается как (0, 0, 0). Внешний вид трёхмерного декоратора смотрите на рис. 1.1.

Рис. 1.1

Каждый из элементов кортежа является скалярным числом, определяющим положение вдоль базисного вектора. Каждый базисный вектор должен иметь единичную длину (его длина равна 1), то есть такие кортежи, как (1, 1, 1) и (2, 2, 2) не могут быть базисными векторами, потому что они слишком длинные.

Мы определим три базисных вектора в нашем пространстве:

(1.1)

## 1.2. Система координат

Точку начала координат системы координат можно обозначить точкой O, которая описывается кортежем из трёх элементов (0,0,0). Это значит, что математическое представление системы координат можно изобразить так:

(1.2)

Этой записью мы можем сказать, что (x, y, z) представляют собой положение точки относительно начала координат. Такое определение означает, что любую точку P, (a, b, c) можно представить как:

(1.3)

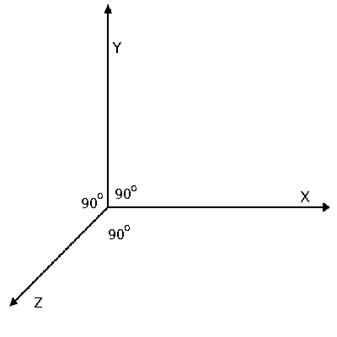
где a, b и c – это скаляры;

X, Y, Z – это базисные векторы;

(1.4)

## 1.3. Взаимная перпендикулярность

Система координат, которую используем, обладает очень ценным свойством: взаимной перпендикулярностью. Это значит, что в пересечении каждой из осей на своей соответствующей плоскости угол между ними равен 90 градусам. Внешний вид системы координат взаимной перпендикулярностью смотрите на рис. 1.2.

Рис. 1.2

Векторное произведение можно определить следующим уравнением (при условии, что у нас есть два кортежа из трёх элементов):

(1.5)

## 1.4. Точки и векторы

Аксиома — это логическое утверждение, считаемое достаточно очевидным для принятия без доказательств.

При работе с точками и векторами мы будем использовать две основные аксиомы.

Аксиома 1: разность между двумя точками — это вектор, то есть

(1.6)

Аксиома 2: сумма точки и вектора — это точка, то есть

(1.7)

## 1.5. Основы линейных преобразований

Линейные преобразования имеют следующую форму:

(1.8)

Есть функция преобразования F (), в качестве входных данных используется вектор A, а на выходе мы получим вектор B

Каждую из этих частей — два вектора и функцию — можно представить в виде матрицы: вектор B — как матрицу 1x3, вектор A — как ещё одну матрицу 1x3, а линейное преобразование F () — как матрицу 3x3 (матрицу преобразований).

(1.9)

(1.10)

## 1.6. Повороты

Поворот, по самому определению — это круговое движение объекта вокруг точки поворота. Точка поворота в нашем пространстве может принадлежать плоскости XY, плоскости XZ или плоскости YZ где каждая плоскость составлена из двух базисных векторов.

Три точки поворота означают, что у нас есть три отдельных матрицы вращения:

Матрица поворота по XY:

(1.11)

Матрица поворота по XZ:

(1.12)

Матрица поворота по YZ:

(1.13)

Для поворота точки AA вокруг плоскости XY на 90 градусов (π/2 радиан — в большинстве математических библиотек есть функция преобразования градусов в радианы.

(1.14)

То есть если начальная точка A имела координаты (3,4,5), то выходная точка B будет иметь координаты (−4,3,5).

## 1.7. Масштабирование

Масштабирование — это преобразование, увеличивающее или уменьшающее объект в соответствии с заданным масштабом.

Преобразование масштабирования требует двух типов входных данных: входного вектора и кортежа масштабирования из трёх элементов, который определяет масштаб входного вектора по каждой из осей пространства.

Матрица масштабного преобразования имеет следующий вид (где , и — это элементы кортежа масштабирования):

(1.15)

Входной вектор A (a0, a1, a2) вдвое больше по оси X (используя кортеж S = (2, 1, 1), вычисления должны иметь следующий вид:

(1.16)

При входном векторе A = (3, 4, 0) выходной вектор BB будет равен (6, 4, 0).

## 1.8. Отсечение

Отсечение — это выборка объектов из большей группы объектов. В трехмерном движке меньшей группой будут точки, которые нужно отрисовать на экране. Большей группой объектов будет множество всех существующих точек.

Благодаря отсечению движок значительно снизит потребление памяти системы. Он будет отрисовывать только то, что камера может увидеть, а не весь мир точек.

Пространство обзора будет задаваться по всем трём традиционным осям: x, y и z. Границы по x состоят из всего между левой и правой границами окна, границы по y — из всего между верхней и нижней границами окна, а границы по z находятся в пределах от 0 куда установлена камера до расстояния видимости камеры мы будем использовать произвольно выбранное значение 100.

Перед отрисовкой точки класс камеры будет проверять, находится ли точка в пространстве обзора. Если находится, то точка отрисовывается, в противном случае — не отрисовывается.

2.