Министерство образования и науки Российской Федерации

федеральное государственное бюджетное образовательное

учреждение высшего образования

«Алтайский государственный технический университет

им. И.И. Ползунова»

Факультет \_\_\_информационных технологий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра \_\_\_\_прикладной математики\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность (направление, профиль) \_\_ПИ\_\_\_\_\_\_\_\_\_\_\_\_\_

Курсовой проект

защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.С. Троицкий\_

(подпись руководителя проекта) (инициалы, фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Курсовой проект

Разработка развлекательного программного обеспечения с технологией Ray Casting

(тема курсового проекта)

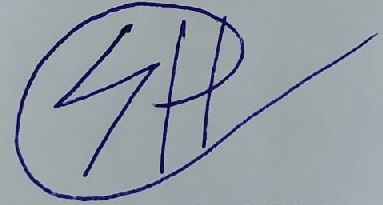
Пояснительная записка

по дисциплине Программирование-3

\_\_\_\_\_\_\_КП 09.03.04.06.000 ПЗ\_\_\_\_\_\_\_

(обозначение документа)

Студент группы ПИ92 Шульпов\_В.М.\_\_\_\_\_\_\_\_\_25.12.2020



(фамилия, имя, отчество) (подпись) (дата)

Руководитель

проекта доцент, к.т.н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.С. Троицкий

(должность, ученое звание) (подпись) (инициалы, фамилия)

БАРНАУЛ 2020Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Алтайский государственный технический университет

имени И.И. Ползунова»

Факультет информационных технологий

Кафедра «Прикладная математика»

З А Д А Н И Е

на курсовой проект по дисциплине «Программирование-3»

студенту группы ПИ-92 Шульпову Виктору Максимовичу

Тема курсового проекта: «Разработка игры с использованием технологии Ray Casting».

Календарный план работы:

|  |  |  |  |
| --- | --- | --- | --- |
| № этапа | Содержание этапа | Дата начала этапа | Количество дней на этап |
| 1 | Получение задания | 01.09.2020 | 20 |
| 2 | Постановка задачи. | 20.09.2020 | 20 |
| 3 | Проектирование программы | 10.10.2020 | 20 |
| 4 | Реализация программы 1 | \*\*.\*\*.2020 | \* |
| 5 | Тестировние программы | \*\*.\*\*.2020 | \* |
| 6 | Оформление пояснительной записки | \*\*.\*\*.2020 | \* |
| 7 | Защита курсового проекта | \*\*.\*\*.2020 | \* |

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Троицкий В.С., доцент

подпись

Дата выдачи задания «1» сентября 2020 г.

число месяц год

Задание принял к исполнению \_\_\_\_\_\_ Шульпов В.М.

подпись

**Содержание**

[Введение 5](#_Toc56120018)

[1 Обзор предметной области и постановка задачи 6](#_Toc56120019)

[1.1 Обзор предметной области 6](#_Toc56120020)

[1.2 Постановка задачи 6](#_Toc56120021)

[1.2.1 Техническое задание 6](#_Toc56120022)

[2 Проектирование 7](#_Toc56120023)

[2.2 Диаграмма классов 7](#_Toc56120024)

[3 Реализация 7](#_Toc56120025)

[3.1 Выбор средств реализации 7](#_Toc56120026)

[3.4 Описание реализации ПО 7](#_Toc56120027)

[8](#_Toc56120028)

[Заключение 10](#_Toc56120029)

[Список использованных источников 10](#_Toc56120030)

[Приложения 11](#_Toc56120031)

[Приложение А Исходный код программы 11](#_Toc56120032)

[Приложение Б Результаты работы программы 12](#_Toc56120033)

# Введение

Cтановлению современных компьютеров, а также повсеместному внедрению и развитию программирования мы обязаны играм. Безусловно, это не единственная причина, по которой человек начал модернизировать вычислительную машину, расширяя её возможности, но это одна из причин. Благодаря развитию игр были изобретены контроллеры и самый первый из них – компьютерная мышь. Игры росли и развивались, рос спрос и интерес общественности, а вместе с тем и требования к конфигурации устройств. Многие современные разработки, которые используются в программных кодах, включая нейронные сети, берут своё начало в примитивных компьютерных играх прошлого.

На ЯП С# для создания графических интерфейсов с помощью платформы .NET применяются разные технологии - Window Forms, WPF, приложения для магазина Windows Store (для ОС Windows 8/8.1/10). Однако наиболее простой и удобной платформой до сих пор остается Window Forms или формы.

На ЯП Python для создания графических интерфейсов (в частности игр) применяется Pygame — набор [модулей](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8C_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) (библиотек) языка программирования [Python](https://ru.wikipedia.org/wiki/Python" \o "Python), предназначенный для написания [компьютерных игр](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0) и [мультимедиа](https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BC%D0%B5%D0%B4%D0%B8%D0%B0)-приложений. Pygame базируется на мультимедийной библиотеке [SDL](https://ru.wikipedia.org/wiki/Simple_DirectMedia_Layer).

Этот курсовой проект является примером простой игры из прошлого. ПО будет похоже на игру “Doom”, но в ней пока что не планируется создавать стены с текстурами (с помощью спрайтов), оружие, выстрелы из него и др. Игра будет представлять собой рендеринг карты по принципу raycasting. В начале пользователь будет появляться в опеределенной точке старта и искать квадрат, являющийся точкой выхода.

**Цель данной работы:**

Изучение технологии Ray Casting, разработка программного обеспечения с использованием методалогии ООП на языках С# и Python. Изучении работы

**Главная задача:**

Написать ПО, позволяющее пользователю играть, а именно: находить выход из лабиринта. Пользователь будет появляеться на точке старта, а затем искать выход из лабиринта. Сделать 3 формы (главную, в которой будет общая информация и кнопка «играть», форму с уровнями, в которой будут создаваться кнопки для каждого уровня и та, в которой будет происходить игра (панели 3д карты, панель 2д карты).

**Дополнительная задача (для будующей разработки**):

Добавить монетки в лабиринт, чтобы можно было копить деньги, противников, которые будут препятсвтовать прохождению лабиринта, возможность стрелять в противников, текстуры.

# 1 Обзор предметной области и постановка задачи

## 1.1 Обзор предметной области

Объе́ктно-ориенти́рованное программи́рование (ООП) — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

В ООП класс - это основной элемент, в рамках которого осуществляется конструирование программ. Класс содержит в себе данные и код, который управляет этими данными. Класс зачастую описывает объект реального мира.

Form (форма) представляет окно или диалоговое окно, которое составляет пользовательский интерфейс приложения.

Button (кнопка) представляет элемент управления

События сигнализируют системе о том, что произошло определенное действие. И если нам надо отследить эти действия, то как раз мы можем применять события.

## 1.2 Постановка задачи

### 1.2.1 Техническое задание

Разработать трёхмерную компьютерную игру – лабиринт. В каждом уровне есть своя карта с точками появления пользователя и выхода из лабиринта. Создать несколько уровней.

В управлении должны быть реализованы повороты игрока (камеры) и движение вперёд. Управление нужно реализовать с помощью кнопок на форме или с помощью клавиш на клавиатуре.

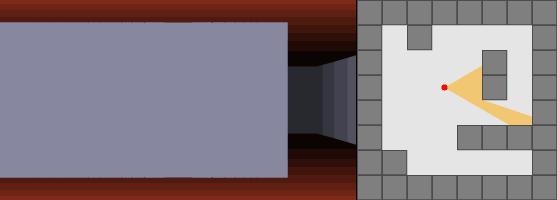
Стены должны быть одного цвета с тенями. Небо и потолок – другими цветами. Небо – верхний прямоугольник, пол – нижний.

Реализовать 3 формы (главную, со списком уровней и ту, в которой будет запускать игра), используя среду разработки Visual Studio и PyCharm.

На главной форме должна быть кнопка играть на красивом фоне, верхние windows-кнопки со справочной информацией. В форме с уровнями – отдельная кнопка для каждого уровня, по которой можно перейти на форму с игрой с соотвествующей картой, кнопка - назад.

В главной форме должно показывать 3d пространство и 2d карта.

Отрисовка карты должна происходить с помщью метода raycasting — бросание лучей) — одного из методов рендеринга в компьютерной графике, при котором сцена строится на основе замеров пересечения лучей с визуализируемой поверхностью.



Принцип метода Ray Casting

# 2 Проектирование

## 2.2 Диаграмма классов

## 2.3 Структура форм

В данном продукте реализованы:

Формы:

MainForm – главная форма проекта. Содержит кнопки (объект Button) «играть» для начала игры и «справочная информация».

LevelsForm – форма с уровнями. Содержит количетсво кнопок, соотвествующее количеству уровней. Имеется кнопка «назад», возвращающая пользователя на главную форму.

GameForm – форма, в которой происходит самое интересное – игра. Состоит из двух панелей (объект Panel), в которых отрисовываются 3D карта и 2D карта.

## 2.4 Структура классов

Programm.cs – класс, запускающий программу, создает объект MainForm

MainForm.cs – класс главной формы

LevelsForm.cs – класс формы с уровнями

Player.cs – класс с данными игрока (позиция, угол, расстояние до стены)

Drawing.cs – класс, отвечающий за отрисовку

Map.cs – класс, хранящий карту в текстовом виде, в нём карта преобразуется в массив точек для карты.

MathHelper.cs – класс с некоторыми математическими методами

Settings.cs – класс, в котором хранятся все настройки в виде static полей.

# 3 Реализация

## 3.1 Выбор средств реализации

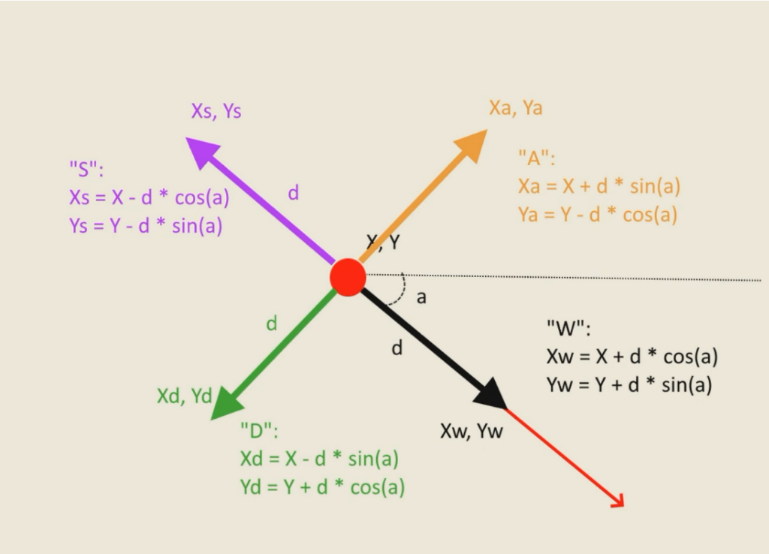
Данное ПО реализовано на языке С# с использованием среды разработки программного Microsoft Visual Studio 2019.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет строгую статическую типизацию, поддерживает полиморфизм, перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML.

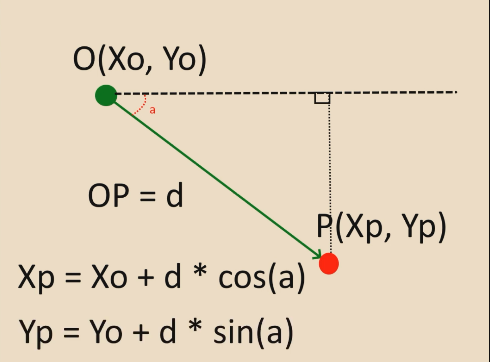
Вторая реализация ПО выполнена на языке Python. Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой набор полезных функций. Основная библиотека – pygame.

Pygame — набор модулей языка программирования Python, предназначенный для написания компьютерных игр и мультимедиа-приложений. Pygame базируется на мультимедийной библиотеке SDL.

## 3.2 Описание реализации ПО

Сначала создается пустое окно с точкой и обработчик нажатий, который позволяет перемещать эту точку. Затем по формулам, приведенным на рисунке ниже, расчитывается положение точки, указывающей направление игрока.

Реализация управления в игре

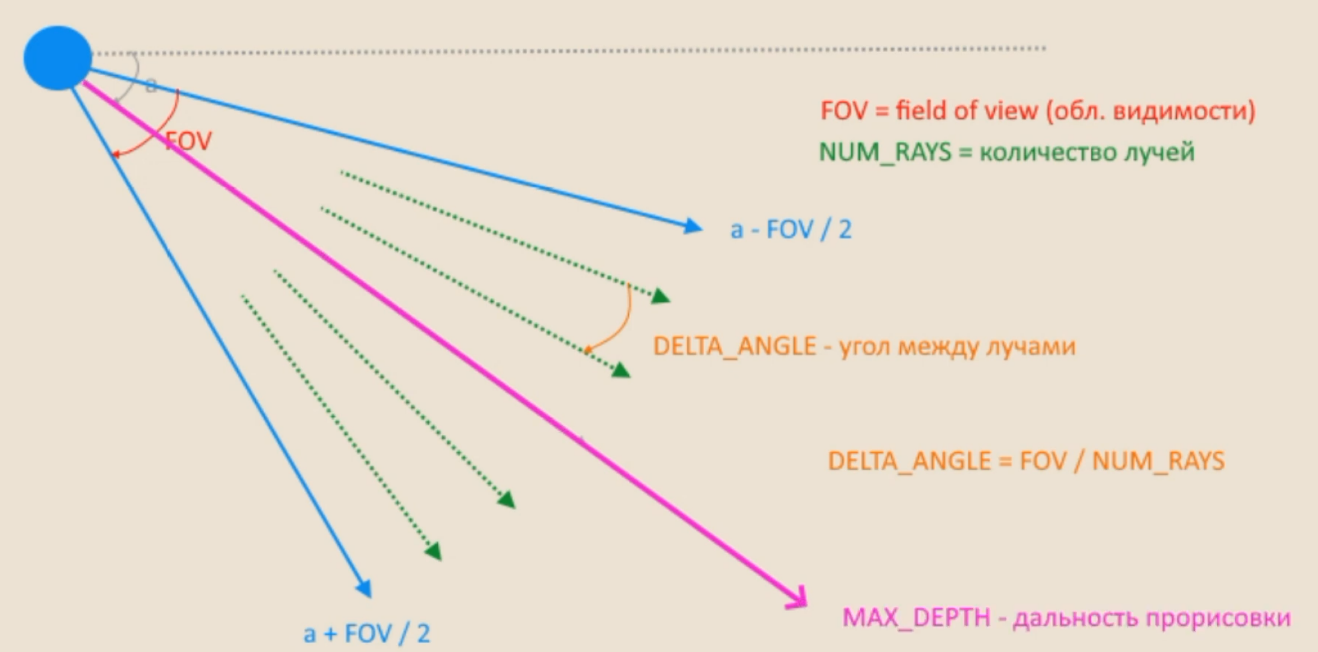


Реализация нахождения точки на луче, в сторону которого направлен игрок:

x0, y0 – текущее позиция игрока

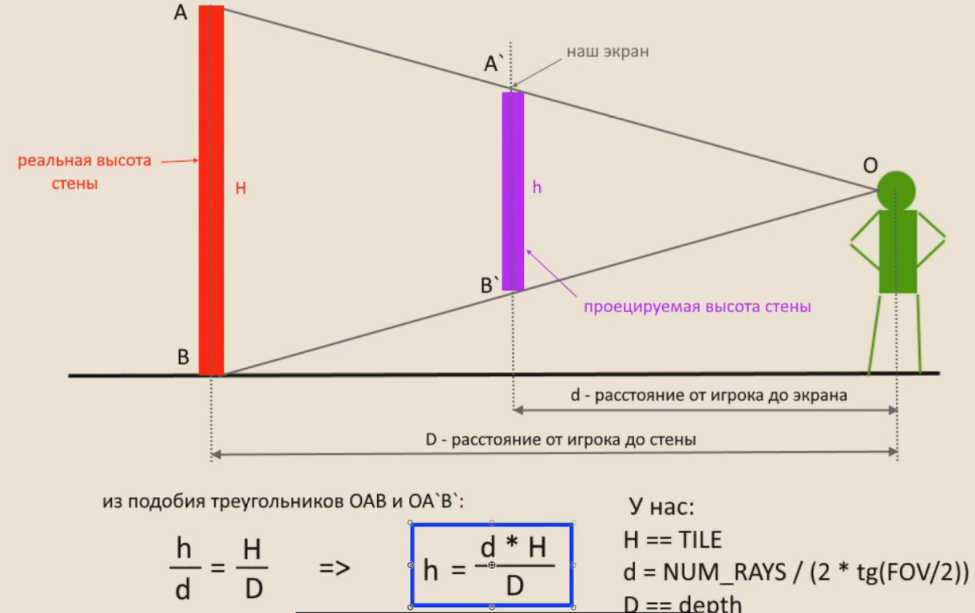
xp, yp – позиция точки, находящейся на расстоянии d от текущей позиции игрока в направлении угла a.

Отрисовка

Реализация отрисовки лучей на 2d карте (для рендеринга 3d карты):

После этого можно рисовать стены, результат будет похоже на 3D картину, которая состоит из столбиков разной высоты, которая будет зависеть от расстояния до этой стены. Каждому лучу, «брошенному» от игрока будет соотвествовать свой столбик.

Цвет столбика так же будет зависеть от расстояния: чем дальше стена, тем она темнее.

Реализация отрисовки стен в 3d пространстве:

Depth – глубина (расстояние до стены в текущем направлении)

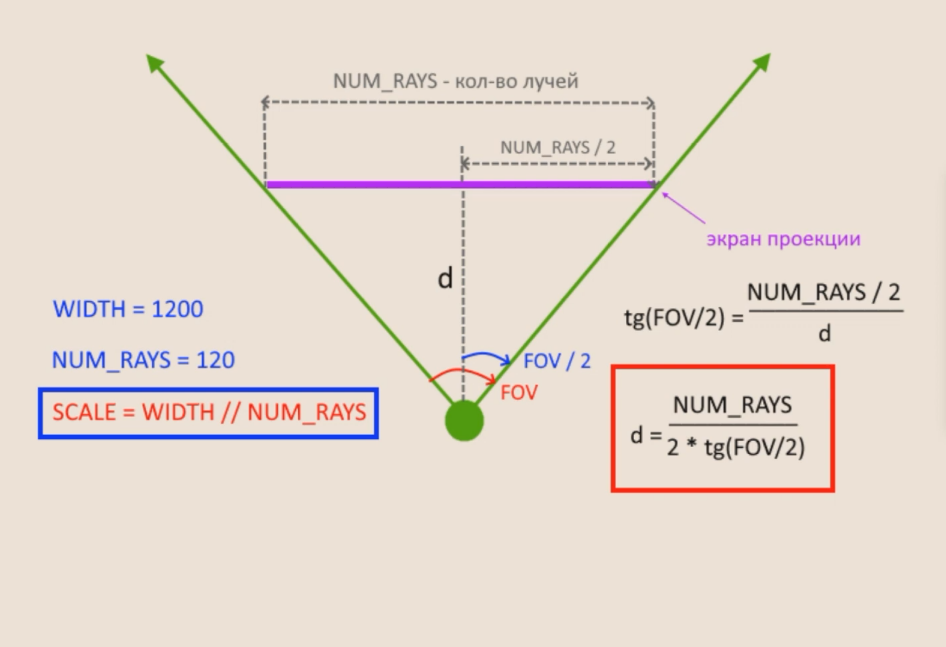
TILE – размер квадратика на карте (его ширина или высота)

Лучи будут «бросаться» от угла current\_angle - FOV/2 до current\_angle + FOV/2, где FOV – угол обзора, current\_angle – текущий угол (угол, в сторону которого направлен игрок).

NUM\_RAYS – количество лучей

WIDTH – ширина карты

SCALE – ширина столбика



# Заключение

# Список использованных источников

# Приложения

## Приложение А Исходный код программы

## Приложение Б Результаты работы программы