МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Факультет компьютерных наук

Кафедра программирования и информационных технологий

*Разработка Android-приложения «Расписание занятий»*

Отчёт по проектному семинару «Введение в программирование»

09.03.02 Информационные системы и технологии

Информационные системы и технологии

Допущено к защите в ГЭК

Обучающийся А.Н. Петров, 1 курс, д/о

Руководитель Д.И. Соломатин, ст.преп.

Воронеж 2023

Содержание

[Содержание 2](#_Toc139233757)

[Введение 3](#_Toc139233758)

[1 Постановка задачи 4](#_Toc139233759)

[2 Анализ предметной области 5](#_Toc139233760)

[3 Реализация 8](#_Toc139233761)

[3.1 Этапы разработки 8](#_Toc139233762)

[3.2 Структура движка 9](#_Toc139233763)

[3.3 Итоги разработки 10](#_Toc139233764)

[Заключение 11](#_Toc139233765)

[Список используемых источников 12](#_Toc139233766)

Введение

На сегодняшний день одной из ключевых сфер примененяи программирования является разработка игр. Десятки студий реализуют собственнеые комплекексные системы называемые игровыми движками для упрощения и ускорения производства игр. Каждый из них обладает собственными преимуществами и особенностями, делающими их востребованными для определенных задач.

Данная же работа посвещена извучению классической структуры игрового 2д движка, его принципами и созданию прототипа, на основе которого, в дальнейшем возможно реализовать различный функционал для создания игр.

1. Постановка задачи

Целью данной работы я ставил изучение структуры игровых движков и написание собственного с использованием языка Си++ и дополнительных инструментов, таких как OpenGL и CMake. За основу бралось создание игры выпущенной в 1985 году под названием “Battle City”, как эталон, возможности которой нужно повторить.

Так как поставленная задача являлась крайне объемной, было решено выделлить основные критерии MVP проекта, которые необходимо реализовать в первую очередь, для получения представления о этапах создания игрового движка и игры на нем. Так было решено в первую очередь реализовать основную структуру необходимую для игрового движка, которая необходима для создания игры, а так же пробную игру для демонстрации проделанной работы.

1. Анализ предметной области

В настоящее время на рынке создания игр существует несколько десятков игровых движков, отвечающих различным требованиям и обладающих особенностями необходимыми для реализации тех или иных задач.

Например(устраевшие движки, на которых созданы культовые игры):

* Doom engine;
* Quake engine

Современные движки:

* Unreal Engine 4/5;
* Unity
* CryEngin
* Source 1 / 2
* REDengine

К части из них каждый желающий может получить доступ для разработки собственной игры, часть же из них является собственность фирм создателей и используются исключительно в их стенах.

Однако если изучить историю создания всех этих движков, то можно заметить, что все они написаны на языке C++(более старые написаны на С и асамблере). Это обусловено несколькими факторами, такими как:

Во-первых, C++ обладает высокой производительностью и низким уровнем абстракции, что позволяет разработчикам полностью контролировать ресурсы компьютера и оптимизировать код для максимальной скорости работы. Это особенно важно для игровых движков, которые должны обеспечивать плавную и отзывчивую графику, физику и искусственный интеллект в реальном времени.

Во-вторых, C++ обладает богатым набором инструментов и библиотек, которые могут быть использованы для разработки игровых движков. Например, стандартная библиотека C++ предоставляет множество функций для работы с файлами, сетями, математическими операциями и многими другими аспектами, необходимыми в разработке игр. Кроме того, существуют сторонние библиотеки и фреймворки, такие как DirectX и OpenGL, которые позволяют взаимодействовать с аппаратным обеспечением компьютера для отрисовки графики и работы с аудио.

В-третьих, C++ является мультиплатформенным языком программирования, что означает, что разработанный на нем игровой движок может быть запущен на различных операционных системах, таких как Windows, macOS и Linux, а также на разных типах аппаратного обеспечения. Это дает возможность достичь широкой аудитории игроков и сделать игровой продукт доступным для максимального количества платформ.

Из этого всего следует, что языком для создания собственного игрового движка будет С++ однако возникает вопрос про использования инструментария для отрисовки графики, так сейчас самыми популярными являются следующие:

DirectX: Разработанный компанией Microsoft, DirectX является набором API (интерфейсов программирования приложений), предназначенных для взаимодействия с аппаратным обеспечением компьютера. DirectX обеспечивает доступ к графическому оборудованию, звуковым картам и другим устройствам, что позволяет разработчикам создавать игры с высокой производительностью на платформе Windows.

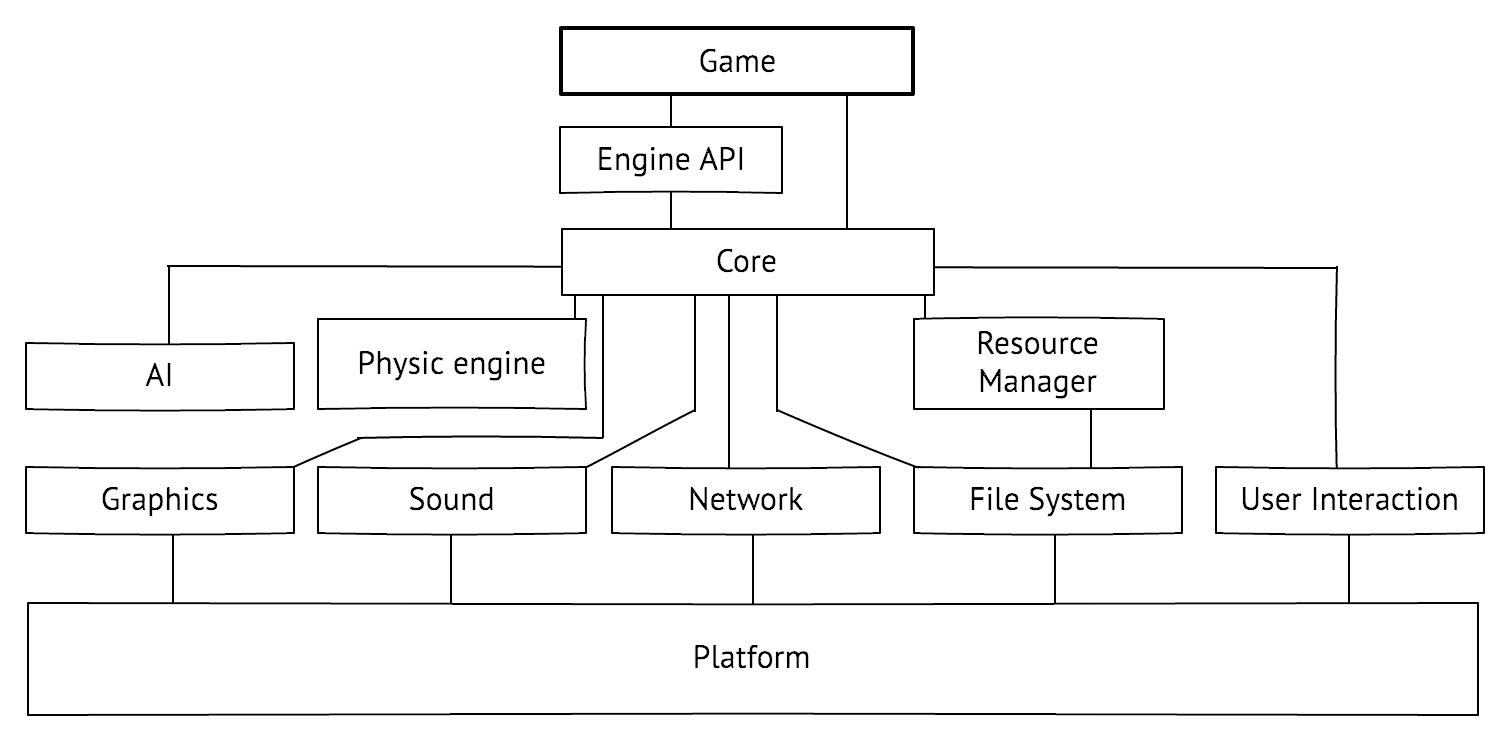
OpenGL: Это открытый стандарт API для отрисовки 2D и 3D графики. OpenGL позволяет разработчикам создавать кросс-платформенные игры, которые могут работать на различных операционных системах, включая Windows, macOS и Linux. Он также предоставляет широкий набор функций для работы с трехмерной графикой, текстурами, освещением и другими графическими эффектами.

Vulkan: Это новое поколение API для графики, разработанное консорциумом Khronos Group. Vulkan обладает высокой производительностью и низким уровнем абстракции, что позволяет разработчикам полностью контролировать графический процессор и создавать игры с высокой скоростью отрисовки. Он также предоставляет возможность распараллеливания задач и оптимизации использования ресурсов, что особенно полезно для разработки игр на мобильных устройствах и консолях.

Для своего же проекта я решил использовать OpenGL.

Так же я решил использолвать Cmake, как средство для автоматизации сборки программного обеспечения из исходного кода.

В ходе анализа предметной области были расмотерны несколько примеров игровых движков, изучены их способы создания и был подобран набор инструментов, с помощью которых будет реализовываться моя игра. А так же исследованы основные элементы, которые должны присутствовать в движке.



1. Общая схема структуры игрового движка
2. Реализация
   1. Этапы разработки

Для воплощения собственной реализации игры “Battle City”, как и при создании любой игры, необходимо было разделить весь процесс разработки на определнные этапы:

1. Отрисовка текстур
2. Создание текстурных атласов и анимаций
3. Создание карты
4. Создание игровых объектов
5. Создание физического движка
6. Создание интерфейса меню
7. Попытка создания искусственного интеллекта противников
   1. Структура движка

В процеесе разработки были написаны следущие модули:

* Main – осноыной класс активации игры;
* Game – являющийся классом в котором происходит основные действия по созданию окна и запускаемых сценариев;
* Level – класс в котором происходит сборка уровня;
* PhysicsEngine – класс в котором описывается физическое взаимодействие объектов и их коллизия;
* Renderer – отвечает за рендериг графики с импользованием OpenGl;
* ResourceManager – отвечает за управление ресурсами игры, такими как шейдеры, текстуры и спрайты.
* ShaderProgram – отвечает за управление ресурсами игры, такими как шейдеры, текстуры и спрайты.
* Sprite – отвечает за параметры спрайта
* SpriteAnimation – параметры анимации спрайта
* StartScreen – отрисовка стартового экрана

И некоторые другие калссы описывающие игровые объекты и их свойства.

* 1. Итоги разработки

По итогу было реализовано:

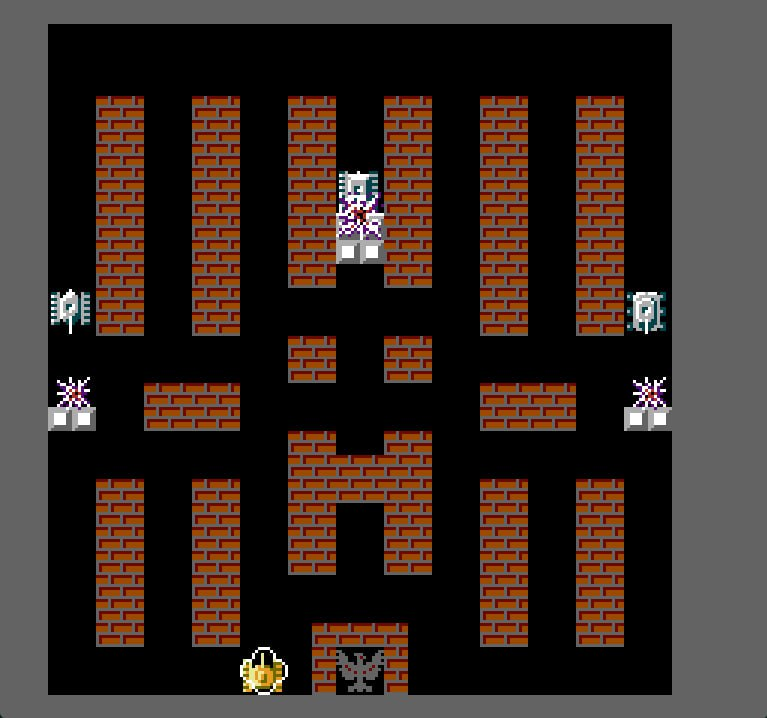
* Стартовое меню с выбором режима игры
* 2 режима игры, для одного игрока и для двоих, с управлением с помощью клавиш «WASD + Space» и «Стрелочек + L Shift»
* Карта с разрушаемыми стенами от выстрелов
* Начата работа над искусственным интеллектом противника (не завершено)

Требует доработки:

* Уничтожение противника
* Завершение уровня после уничтожения всех противников

Демо версия игры состоит из:

* Экрана меню
* Игрового поля

1. Интерфейс игры

Заключение

В данном отчете была описана разработка игрового движка с демо игрой «Battle City» с возможностью последующей модификации игры. Цель изучения структуры и инструментария для создания собственного движка изучена и применены на практике, пока движок реализован в формате нескольких классов изменяя которые можно изменять игру по собственному усмотрению, что открывает возможность для дальнейшего усовершенствования проекта и доведения его до состояния, когда с меньшим взаимодействием с кодом можно будет создавать уровни, изменять сложность противников и другие параметры. Возможно добавление интерфейса и других возможностей, присущих знакомым большенству игровым движкам.

Список используемых источников

1. Структура игрового движка [электрοнный ресурс] URL: https://dtf.ru/flood/21907-struktura-igrovogo-dvizhka
2. Как и зачем создавать собственный игровой движок [электрοнный ресурс] URL: https://habr.com/ru/companies/playgendary/articles/650241/
3. Статьи по работе с OpenGL [электрοнный ресурс] URL: https://learnopengl.com/
4. Игровой движок на с++ и OpenGL с нуля [электрοнный ресурс] URL: <https://www.youtube.com/watch?v=wf37ukItjE8&list=PL6x9Hnsyqn2XU7vc8-oFLojbibK91fVd-&index=1>
5. Ностальгия: роемся у «Танчиков» под капотом [электрοнный ресурс] URL: https://habr.com/ru/articles/142126/