МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение   
высшего образования

«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»

ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра компьютерной инженерии и моделирования

**СОЗДАНИЕ ИГРЫ В ЖАНРЕ “TOWER DEFENSE”**

Курсовая работа

по дисциплине «Программирование»

студента 1 курса группы ПИ-б-о-201

Прокоповича Сергея Валерьевича

направления подготовки 09.03.04 «Программная инженерия»

|  |  |  |
| --- | --- | --- |
| Научный руководитель  старший преподаватель кафедры компьютерной инженерии и моделирования | \_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  \_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | Чабанов В.В. |

Симферополь, 2021

**РЕФЕРАТ**

Тема: создание игры в жанре “tower defense”.

Объём Курсовой 17 листов, на которых размещены 7 рисунков. При написании проекта использовались 3 источника.

Объектом исследования являются язык С++, который изучается посредством создания игры в жанре “tower defense”. В курсовой проект входят: Введение, пять разделов, заключение. В введении ставятся цели и задачи проекта. В первом разделе идет постановка задач, описывается цели проекта, существующие аналоги, основные отличия и ставится техническое задание. Во втором разделе рассматривается программная реализация приложения, анализ инструментальных средств, описание алгоритмов и описание основных модулей проекта. В третьем разделе проводится тестирование программы. В четвертом разделе описываются дальнейшие технические перспективы развития проекта и его возможной монетизации.

В заключении описывается вывод о проекте и полученных знаниях.

**О**ГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 4](#_Toc74394842)

[ГЛАВА 1 ПОСТАНОВКА ЗАДАЧИ 5](#_Toc74394843)

[1.1 Цель проекта 5](#_Toc74394844)

[1.2 Существующие аналоги 5](#_Toc74394845)

[1.3 Основные отличия от аналогов 5](#_Toc74394846)

[1.4 Техническое задание 5](#_Toc74394847)

[ГЛАВА 2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ 6](#_Toc74394848)

[2.1 Анализ инструментальных средств 6](#_Toc74394849)

[2.2 Описание алгоритмов 6](#_Toc74394850)

[2.3 Описание основных модулей 10](#_Toc74394851)

[ГЛАВА 3 ТЕСТИРОВАНИЕ ПРОГРАММЫ 11](#_Toc74394852)

[3.1 Тестирование исходного кода 11](#_Toc74394853)

[3.2 Тестирование интерфейса пользователя и юзабилити 11](#_Toc74394854)

[ГЛАВА 4 ПЕРСПЕКТИВЫ ДАЛЬНЕЙШЕГО РАЗВИТИЯ ПРОЕКТА 12](#_Toc74394855)

[4.1 Перспективы технического развития 12](#_Toc74394856)

[4.2 Перспективы монетизации 12](#_Toc74394857)

[ЗАКЛЮЧЕНИЕ 13](#_Toc74394858)

[ЛИТЕРАТУРА 14](#_Toc74394859)

[**ПРИЛОЖЕНИЕ 1 КОД ОСНОВНЫХ МОДУЛЕЙ ПРОЕКТА** 15](#_Toc74394860)

ВВЕДЕНИЕ

Проект рассчитан на изучения языка С++, посредством создания игры на этом языке. В ходе курсовой работы планируется создать игру в жанре “Tower defense”. Для этого нужно будет определится со средой разработки и библиотеками для начала. В самой игре нужно будет сделать интерфейс, карту, логику для башен и юнитов. После всех этих действий должна получится простоя игра в жанре “Tower defense”.

ГЛАВА 1  
ПОСТАНОВКА ЗАДАЧИ

**1.1 Цель проекта**

Изучение основ программирования в области разработки игр. Итоговая цель – создание игры в жанре “Tower defense”.

**1.2 Существующие аналоги**

“Kingdom rush”, “Prime World: Defenders”, “Empires in Ruins”.

**1.3 Основные отличия от аналогов**

Отличий как таковых от стандарта жанра не планируется.

## 1.4 Техническое задание

Необходимо создать уровни, в которые будут входить фон, дороги для монстров, башни и места для башен. Игроку дается в начале деньги на одну башню, а затем он получает деньги с убийства монстров и ставит больше башен. В противовес монстров становится больше и увеличивается их “HP” то есть жизненная сила. Главная задача игрока, не дать дойти врагам до своей “красной зоны” иначе если игрок пропустит 5 монстров, то он проиграет и начнет заново. Также надо разработать интерфейс для начала игры, настроек и для отображения текущего количества “HP” и денег.

ГЛАВА 2  
ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ

## 2.1 Анализ инструментальных средств

В ходе создания проекта использовалась программная среда Visual Studio 2019. Графическая часть проекта была создана с помощью библиотеки SFML.

Выбор пал на Visual Studio 2019, как самую популярную IDE для языка C++. Библиотека SFML была выбрана для удобного создания графики для игры.

## 2.2 Описание алгоритмов

Для создания окна воспользуемся алгоритмом, представленным на рисунке 2.1.

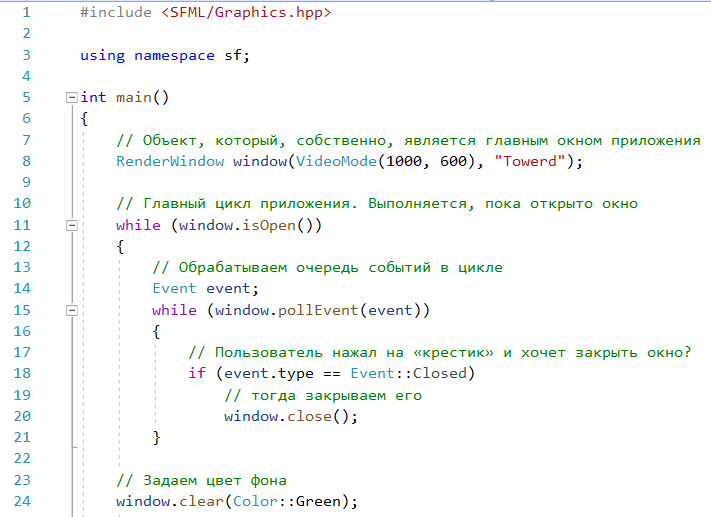


Рисунок 2.1 – алгоритм отображения окна программы.

Так же необходимо загрузить текстуры для элементов графики. Алгоритм показан на рисунке 2.2.

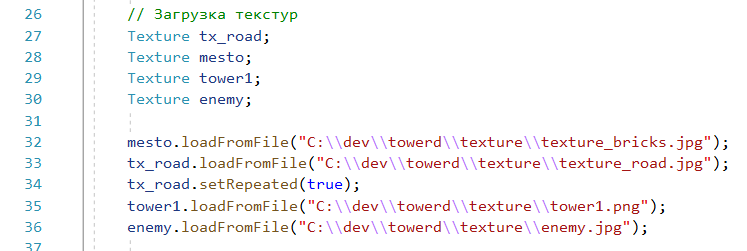


Рисунок 2.2 – алгоритм загрузки текстур.

Для создания различных элементов воспользуемся следующим алгоритмом, показанным на рисунке 2.3.

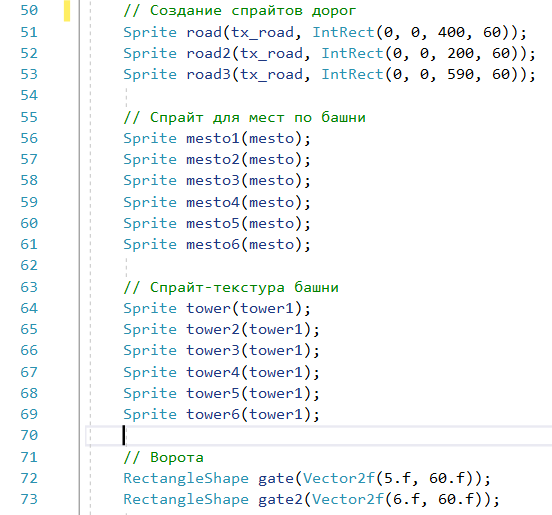


Рисунок 2.3 – создание спрайтов.

Для корректного отображения спрайтов воспользуемся следующим алгоритмом, показанным на рисунке 2.4 и рисунке 2.5

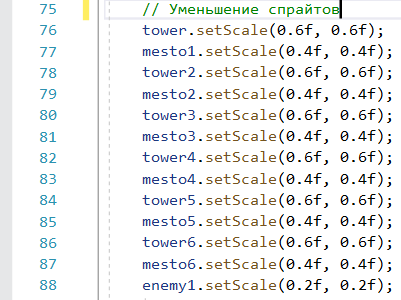


Рисунок 2.4 – уменьшение спрайтов.

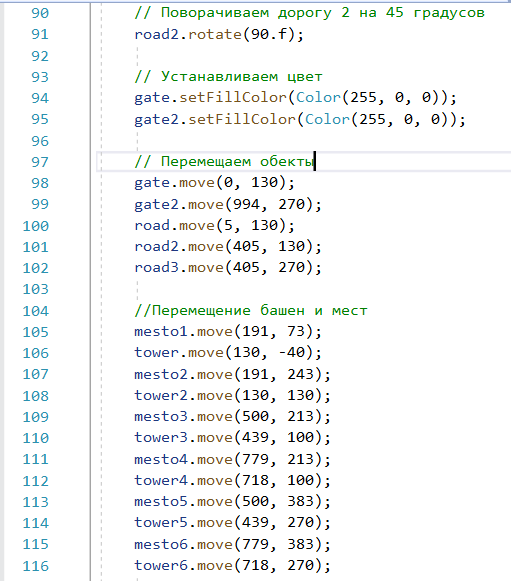


Рисунок 2.5 – операции над спрайтами.

Для отображения конечного результата воспользуемся алгоритмом, показанным на рисунке 2.6.

Эти алгоритмы описывают базовое отображение статичной графики.

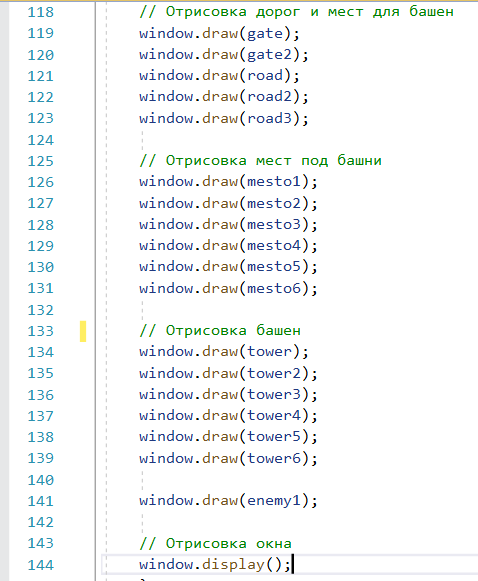


Рисунок 2.6 – отображение окна с графикой.

## 2.3 Описание основных модулей

Создан только один модуль – main.cpp. В него вошел весь код, имеющийся программы. Результат модуля показан на рисунке 2.7.

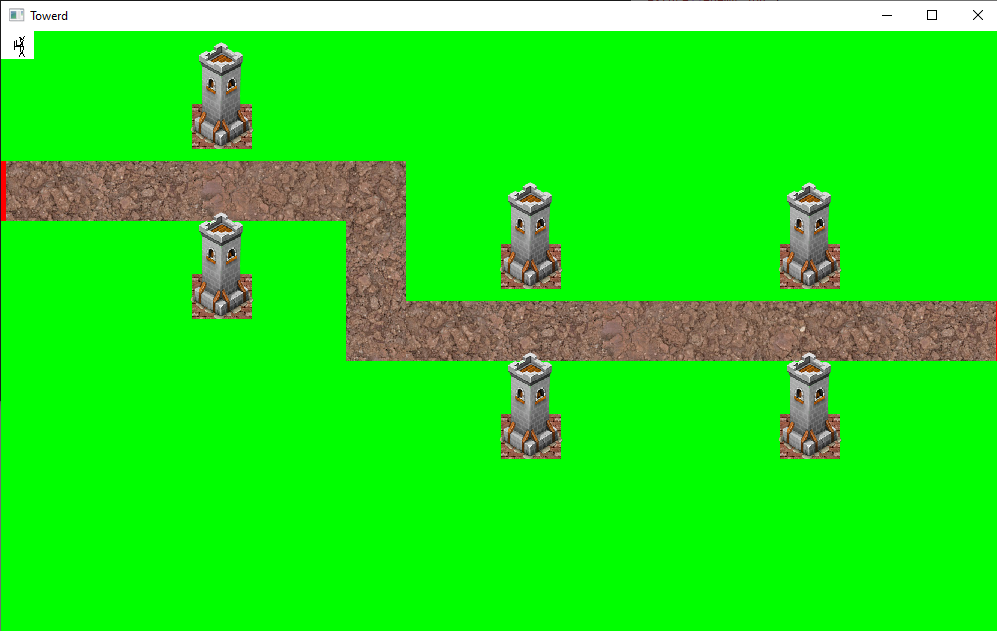


Рисунок 2.7 – результат работы программы.

ГЛАВА 3  
ТЕСТИРОВАНИЕ ПРОГРАММЫ

**3.1 Тестирование исходного кода**

Тестирование не проводилось по причине незаконченности программы

**3.2 Тестирование интерфейса пользователя и юзабилити**

После запуска программы отображается карта с башнями.

ГЛАВА 4  
ПЕРСПЕКТИВЫ ДАЛЬНЕЙШЕГО РАЗВИТИЯ ПРОЕКТА

**4.1 Перспективы технического развития**

Проект не был доведен до конца, поэтому необходимо доделать логику для игры. Создать двигающихся врагов, сделать так чтобы можно было ставить башни и что бы от них шел врагам урон, нанесения урона должно отображаться графически в виде стрел или другого спецэффекта. После полной реализации проекта можно сделать компанию с сюжетом, где игроку будут ставится различные задачи и добавляться на выбор различные башни. Так же игру планируется портировать на Android платформу.

**4.2 Перспективы монетизации**

Такую игру можно продать в “Play market” по относительно невысокой цене. Или сделать бесплатной, но встроить рекламу.

ЗАКЛЮЧЕНИЕ

В результате работ проект не был завершен. Сделана только одна карта для уровня и полностью отсутствует логика для игры. В ходе работы над проектом я познакомился с работой библиотеки SFML и немного изучил язык С++.

ЛИТЕРАТУРА

1. ГОСТ 19.002-80 Схемы алгоритмов и программ. Правила выполнения [Текст] – Введ. с 01.07. 1981 г. М.: Изд-во стандартов, 1981. – 9 с.
2. ГОСТ 19.003-80 Схемы алгоритмов и программ. Обозначение условные графические [Текст] – Введ. с 01.07. 1981 г. М.: Изд-во стандартов, 1981. – 9 с.
3. Оформление выпускной квалификационной работы на соискание квалификационного уровня «Магистр» («Бакалавр»): методические рекомендации. / сост. Бержанский В.Н., Дзедолик И.В., Полулях С.Н. – Симферополь: КФУ им. В.И.Вернадского, 2017. – 31 с.

**ПРИЛОЖЕНИЕ 1  
КОД ОСНОВНЫХ МОДУЛЕЙ ПРОЕКТА**

main.cpp

#include <SFML/Graphics.hpp>

using namespace sf;

int main()

{

// Объект, который, собственно, является главным окном приложения

RenderWindow window(VideoMode(1000, 600), "Towerd");

// Главный цикл приложения. Выполняется, пока открыто окно

while (window.isOpen())

{

// Обрабатываем очередь событий в цикле

Event event;

while (window.pollEvent(event))

{

// Пользователь нажал на «крестик» и хочет закрыть окно?

if (event.type == Event::Closed)

// тогда закрываем его

window.close();

}

// Задаем цвет фона

window.clear(Color::Green);

// Загрузка текстур

Texture tx\_road;

Texture mesto;

Texture tower1;

Texture enemy;

mesto.loadFromFile("C:\\dev\\towerd\\texture\\texture\_bricks.jpg");

tx\_road.loadFromFile("C:\\dev\\towerd\\texture\\texture\_road.jpg");

tx\_road.setRepeated(true);

tower1.loadFromFile("C:\\dev\\towerd\\texture\\tower1.png");

enemy.loadFromFile("C:\\dev\\towerd\\texture\\enemy.jpg");

Sprite enemy1(enemy);

// Горизонтальное перемещение

//int dx = 997;

//for (int i = 0; i < 4; i++) a[i].x += dx;

//int n = 3;

// Создание спрайтов дорог

Sprite road(tx\_road, IntRect(0, 0, 400, 60));

Sprite road2(tx\_road, IntRect(0, 0, 200, 60));

Sprite road3(tx\_road, IntRect(0, 0, 590, 60));

// Спрайт для мест по башни

Sprite mesto1(mesto);

Sprite mesto2(mesto);

Sprite mesto3(mesto);

Sprite mesto4(mesto);

Sprite mesto5(mesto);

Sprite mesto6(mesto);

// Спрайт-текстура башни

Sprite tower(tower1);

Sprite tower2(tower1);

Sprite tower3(tower1);

Sprite tower4(tower1);

Sprite tower5(tower1);

Sprite tower6(tower1);

// Ворота

RectangleShape gate(Vector2f(5.f, 60.f));

RectangleShape gate2(Vector2f(6.f, 60.f));

// Уменьшение текстур

tower.setScale(0.6f, 0.6f);

mesto1.setScale(0.4f, 0.4f);

tower2.setScale(0.6f, 0.6f);

mesto2.setScale(0.4f, 0.4f);

tower3.setScale(0.6f, 0.6f);

mesto3.setScale(0.4f, 0.4f);

tower4.setScale(0.6f, 0.6f);

mesto4.setScale(0.4f, 0.4f);

tower5.setScale(0.6f, 0.6f);

mesto5.setScale(0.4f, 0.4f);

tower6.setScale(0.6f, 0.6f);

mesto6.setScale(0.4f, 0.4f);

enemy1.setScale(0.2f, 0.2f);

// Поворачиваем дорогу 2 на 45 градусов

road2.rotate(90.f);

// Устанавливаем цвет

gate.setFillColor(Color(255, 0, 0));

gate2.setFillColor(Color(255, 0, 0));

// Перемещаем обекты

gate.move(0, 130);

gate2.move(994, 270);

road.move(5, 130);

road2.move(405, 130);

road3.move(405, 270);

//Перемещение башен и мест

mesto1.move(191, 73);

tower.move(130, -40);

mesto2.move(191, 243);

tower2.move(130, 130);

mesto3.move(500, 213);

tower3.move(439, 100);

mesto4.move(779, 213);

tower4.move(718, 100);

mesto5.move(500, 383);

tower5.move(439, 270);

mesto6.move(779, 383);

tower6.move(718, 270);

// Отрисовка дорог и мест для башен

window.draw(gate);

window.draw(gate2);

window.draw(road);

window.draw(road2);

window.draw(road3);

// Отрисовка мест под башни

window.draw(mesto1);

window.draw(mesto2);

window.draw(mesto3);

window.draw(mesto4);

window.draw(mesto5);

window.draw(mesto6);

// Отрисовка башен

//

window.draw(tower);

window.draw(tower2);

window.draw(tower3);

window.draw(tower4);

window.draw(tower5);

window.draw(tower6);

window.draw(enemy1);

// Отрисовка окна

window.display();

}

return 0;

}