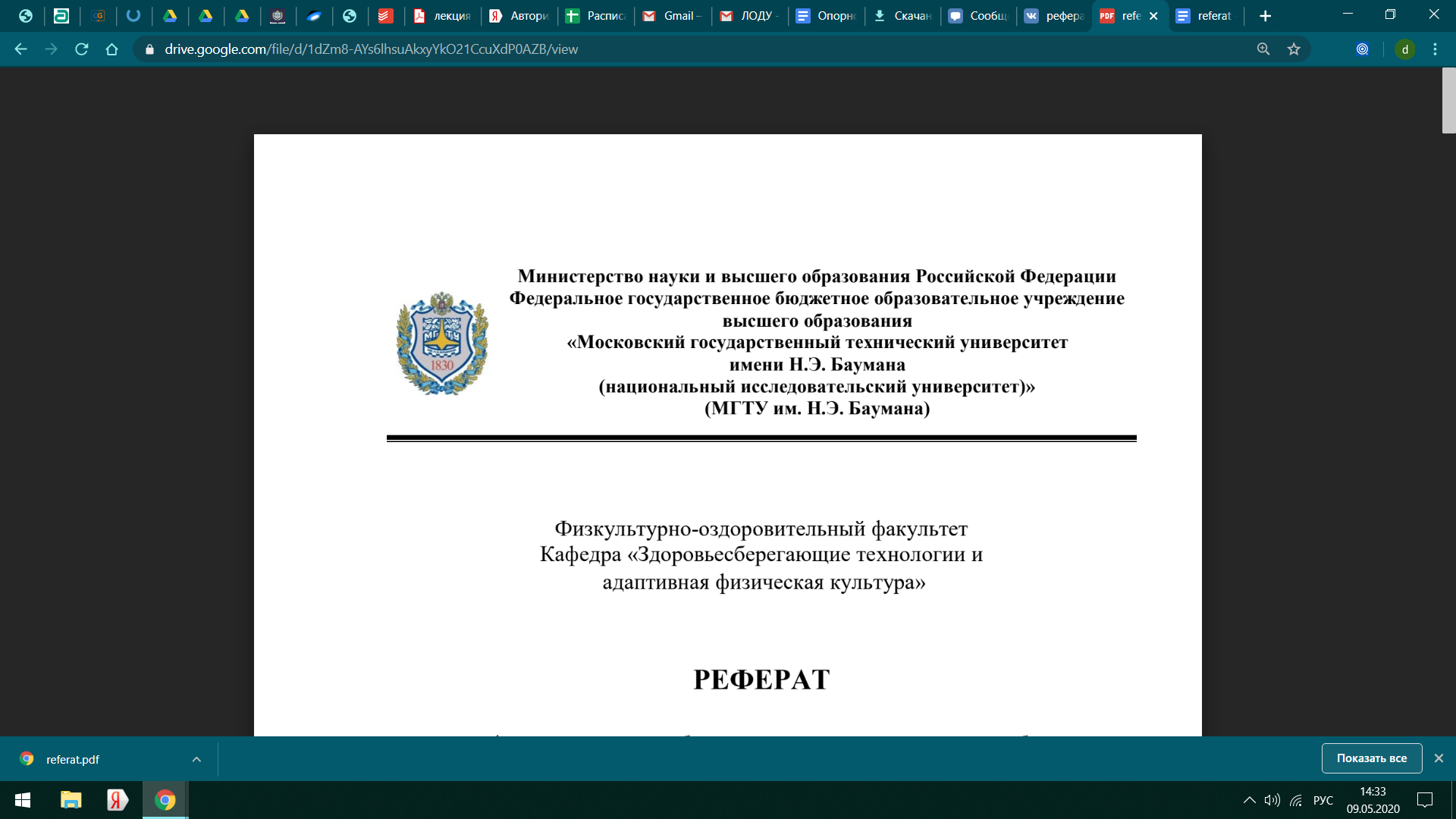
**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)** 

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Системы Автоматического Управления» (ИУ1)

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

Тип практики: учебная

Название предприятия: НУК ИУ МГТУ им. Н.Э. Баумана

Студент:

Соловьев Михаил Антонович, группа ИУ1-22 (1 курс) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

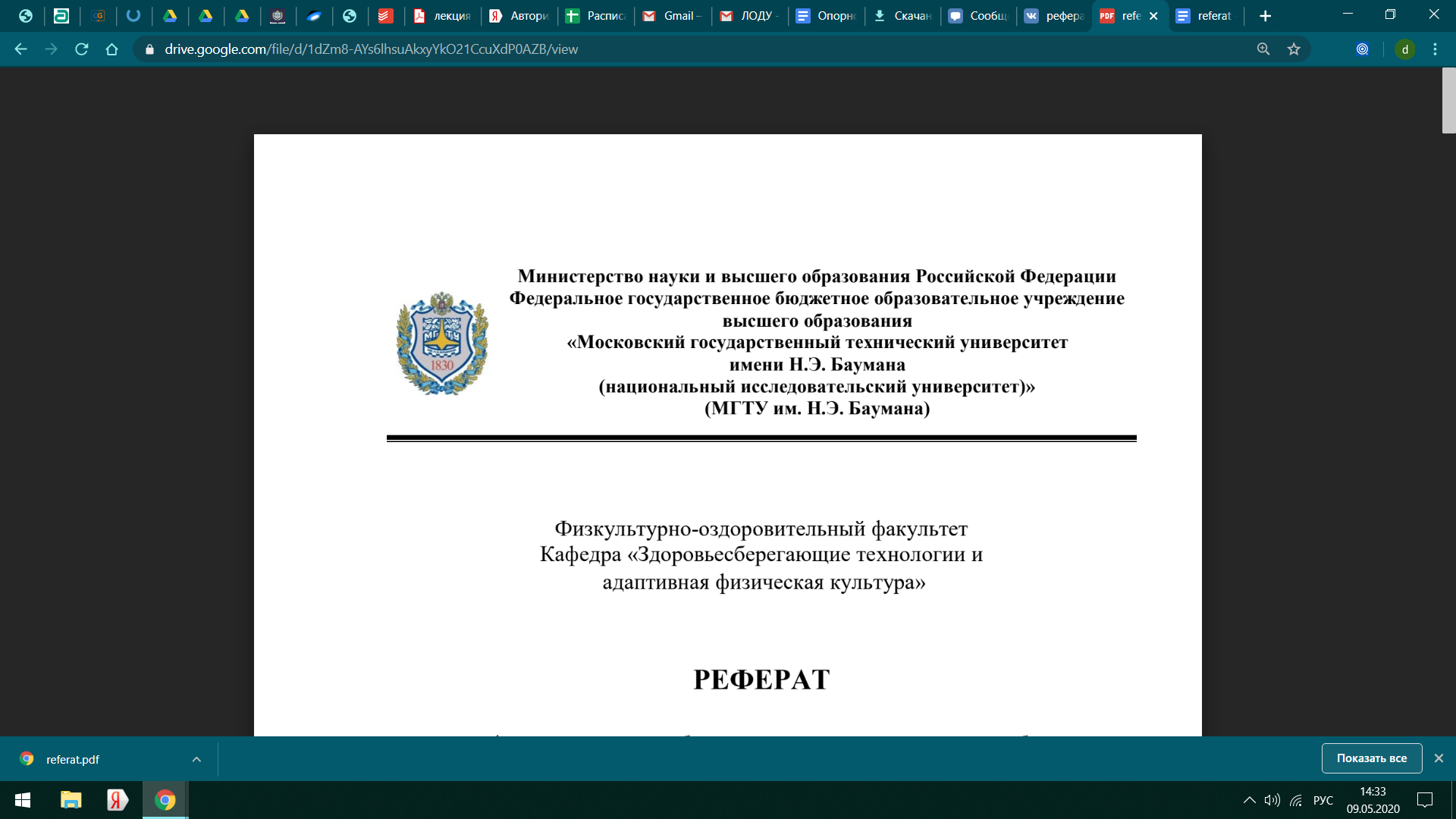
(подпись, дата)

Руководитель от кафедры:

старший преподаватель кафедры ИУ7 Рогозин \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Николай Олегович (подпись, дата)

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_

**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Системы Автоматического Управления» (ИУ1)

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ**

Тип практики: учебная

Название предприятия: НУК ИУ МГТУ им. Н.Э. Баумана

Сроки практики: с Место для ввода даты. по Место для ввода даты

Специальность / направление: Системы управления летательными аппаратами

Специальность / профиль: 24.05.06 «Системы управления летательными аппаратами»

За время прохождения практики студенту надлежит согласно программе практики:

* **Изучить …**
* **Собрать** материал ….
* **Получить практические навыки …..**

Студент:

Соловьев Михаил Антонович, группа ИУ1-22 (1 курс)

(подпись, дата)

Руководитель от кафедры:

старший преподаватель кафедры ИУ7 Рогозин \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Николай Олегович (подпись, дата)

**ОГЛАВЛЕНИЕ**

4 – Введение

4-6 – Основная часть. Начало

7 – Основная часть. Разработка приложения – подготовка

7-15 – Основная часть. Разработка приложения – программирование

15 – Заключение

15 – Список литературы

16 – Приложения

**ВВЕДЕНИЕ**

Основными целями и задачами практики были: ознакомление с принципом

разработки игрового приложения, изучение возможностей графической

библиотеки SFML, реализация концепций объектно-ориентированного

программирования на примере языка C++, отладка и тестирование программы,

написание отчета.

**ОСНОВНАЯ ЧАСТЬ**

**Начало**

В чем заключается проект? Приложение, о котором дальше пойдет речь в отчете

представляет собой игру Space Battle, состоящую из трех окон, одно из которых

отведено под консоль. Идея игры проста: игрок при запуске приложения

выбирает область, отведенную под кнопку начала игры или выхода (Рис. 1). В

случае, если игрок кликает левой кнопкой мыши по круговой области с

надписью “START”,

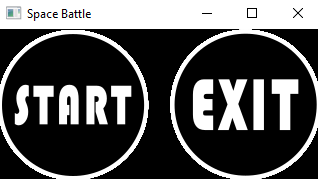


Рис. 1 (Меню)

то открывается игровое окно размером 1024х1024 (Рис. 2), где в дальнейшем

будут сосредоточены действия пользователя. В случае, если пользователь

нажмет левой кнопкой мыши по круговой области с надписью “EXIT” или

кликнет на крестик, то окно закроется – программа завершит свою работу.

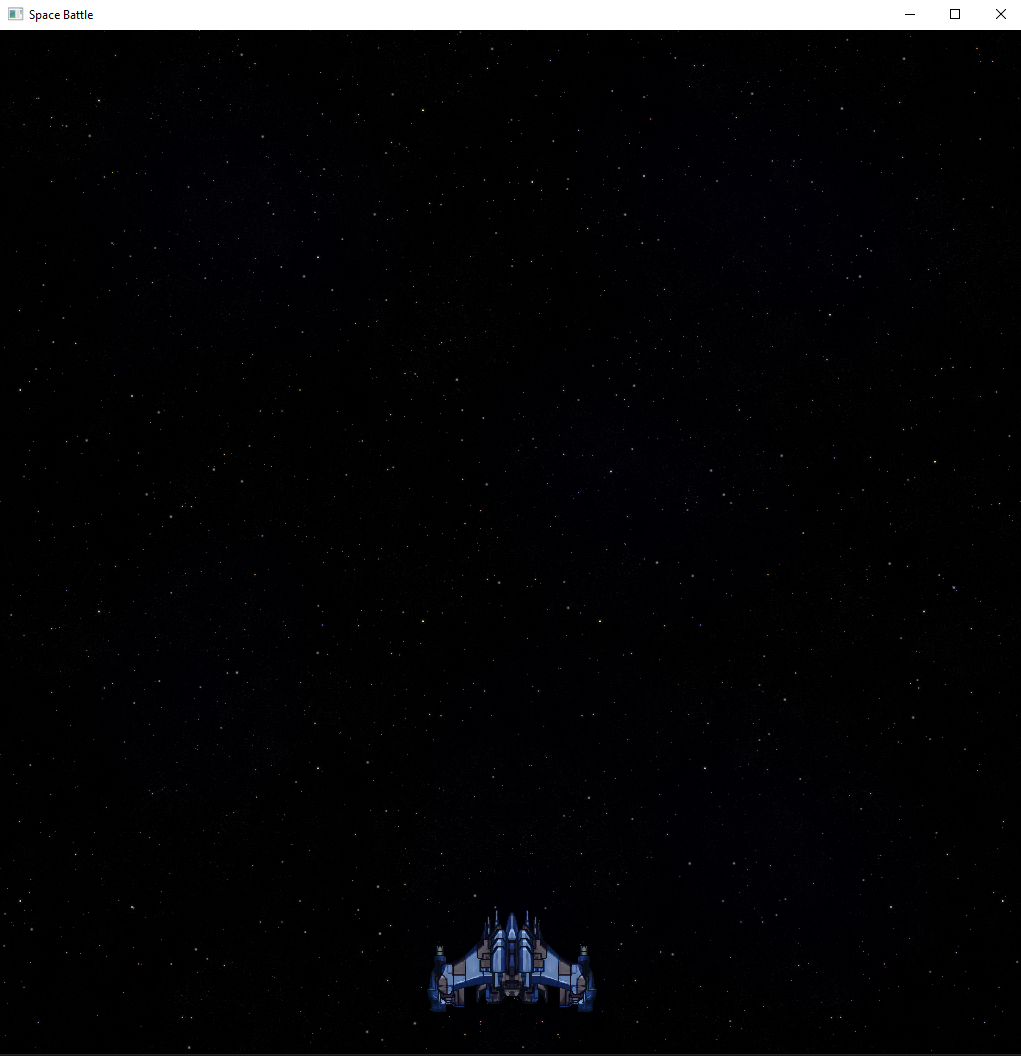


Рис. 2 (Игровое окно)

Суть игры заключается в уничтожении противников (Рис. 5), которые

генерируются при нажатии игроком кнопки “Q” в количестве одной штуки.

Действия, которые может совершать пользователь:

* Движение вверх, вниз, влево, вправо – по нажатию клавиш “W”, “S”, “A”,

“D”

* Выстрел большим желтым лучом слева и справа – по нажатию правой

кнопки мыши (Рис. 3)

* Выстрел маленьким красным лучом слева и справа – по нажатию левой кнопки мыши (Рис. 4)

Рис. 3 Рис. 4

* Уничтожение врагов – посредством лучей, причем в зависимости от того,

какое количество урона было получено. Урон, получаемый врагом,

считается следующим образом: если луч – красный и оба луча (левый и

правый) попадают по противнику, то из начальных 500 единиц жизней

противника будет снято 100, если попадает только один луч (левый или

правый), то будет снято 50 единиц жизней. При попадании левым и

правым желтыми лучами, будет снято 500 единиц, но такое невозможно,

в связи с тем, что расстояние между центрами левого и правого желтых

лучей превосходит ширину спрайта врага, поэтому, в случае поражения

врага желтым лучом, его здоровье уменьшится на 250 единиц.

Уничтожение врага сопровождается 11-кадровой анимацией взрыва:

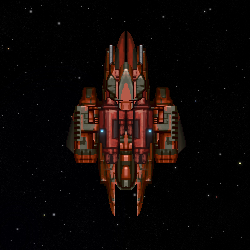
 

Рис. 5 (Противник до взрыва) Рис. 6 (Кадр взрыва)

**Разработка приложения - подготовка**

Средой разработки была выбрана Visual Studio 2019, для обеспечения

взаимодействия с графической составляющей Space Battle был выбран NuGet

пакет sfml\_graphics версии 2.5.1, так же, были немного изменены свойства

проекта для работы с библиотекой SFML (Рис. 7).

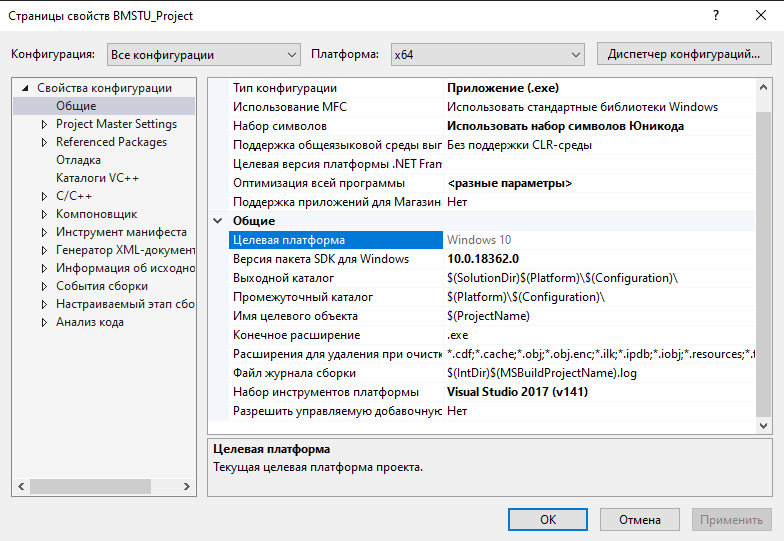


Рис. 7

**Разработка приложения – программирование**

Для того, чтобы использовать функционал SFML библиотеки, необходимо

подключить заголовочный файл SFML/Graphics.hpp в предварительно

созданном файле Source.cpp пустого приложения C++. Для удобства

сразу выбирается пространство имен sf.

Создадим класс point (Рис. 7), объекты данного класса будут часто

использоваться в разных местах программы.

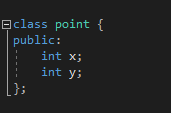


Рис. 7

Перейдем к созданию функции, отвечающей за окно с меню (Рис. 8)

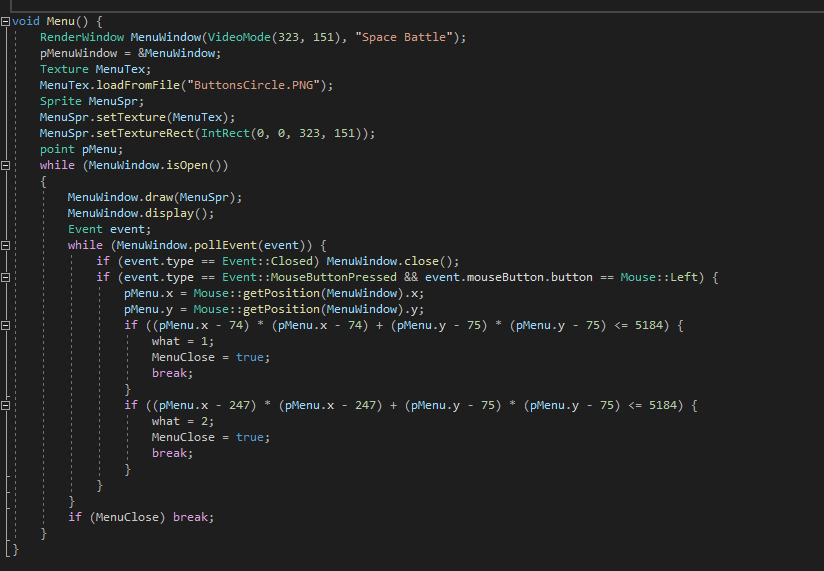


Рис. 8

Создадим объект класса sf::RenderWindow и назовем его MenuWindow –

это будет окно расширением 323х151, с названием “Space Battle”, присвоим

значение указателю pMenuWindow на данное окно (потребуется в дальнейшем).

Создадим текстуру, которая будет загружена из файла (Рис. 9).

Рис. 9

Затем из данной структуры создадим спрайт, который как бы отсечет самую

правую круговую область (она нам не нужна). Теперь необходимо создать

экземпляр класса point для того, чтобы хранить в нем координаты

мышки относительно созданного окна. До тех пор, пока открыто окно,

мы рисуем в нем созданный спрайт при помощи метода draw() и отображаем

в при помощи display(). Теперь создадим объект класса sf::Event для получения

событий, полученных в ходе взаимодействия с окном. До тех пор, пока есть

события, которые можно обработать, мы определяем его тип:

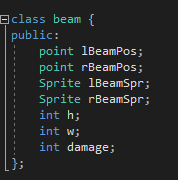
* Нажат крестик – закрываем окно;
* Нажата левая кнопка мыши – если попали в область “START”,

то присваиваем глобальной переменной what единицу и закрываем окно;

если попали в область “EXIT”, присваиваем what 2 и закрываем окно.

Для определения границ областей было использовано уравнение окружности.

Создадим класс beam, поля которого дадут представление о луче (Рис. 9).

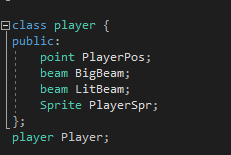
 Рис. 9

Полями данного класса являются позиции левого и правого лучей, их спрайты,

высота и ширина спрайтов и урон, наносимый лучом.

Создадим класс player с информацией об игроке и сразу же объявим глобально

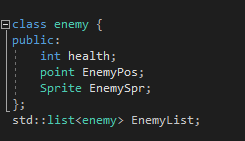
объект этого класса (Рис. 10).

 Рис. 10

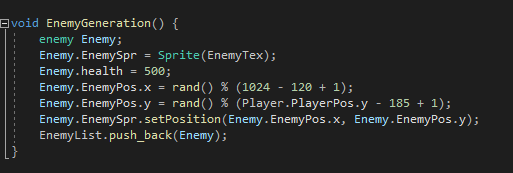
Создадим класс enemy с информацией о враге и сразу же создадим

глобальный список объектов данного класса, не забыв использовать

пространство имен std и назвав список EnemyList (Рис. 11).

 Рис. 11

Реализуем функцию добавления врага (Рис. 12).

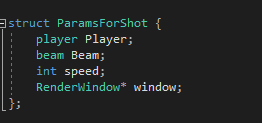
 Рис. 12

После загрузки спрайта и установки его на случайные позиции (y зависит от

текущей позиции игрока для того, чтобы никогда не быть ниже, чем игрок),

добавим в список EnemyList объект класса Enemy с определенными свойствами.

Создадим структуру ParamsForShot, она потребуется дальше (Рис. 13).

 Рис. 13

Полями структуры являются: объект класса player, объект класса beam, значение

скорости стрельбы, указатель на окно, в котором будет реализована стрельба.

Создадим функцию стрельбы ShotingByPlayer с аргументом в виде

экземпляра структуры ParamsForShot (Рис. 14).



Рис. 14

Создадим переменные, ответственные за время, - они понадобятся для

реализации плавной стрельбы. Пока не произошло столкновение луча

с вражеским кораблем, луч плавно летит. Указателю на окно, в котором

происходят боевые действия, сообщаем через метод draw(), что нужно

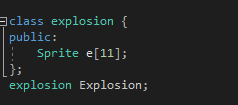
отобразить в окне после очистки (необходимость в очистке обусловлена

плавностью движения луча). Если столкновение все-таки произошло, то

фиксируем урон и выходим из цикла.

Создадим класс explosion с полем в виде массива спрайтов на 11 элементов

(Рис. 15).

 Рис. 15

Перейдем к созданию функции Game, в которой будет присутствовать

практически все, что было написано до этого (Рис. 16, Рис. 17).

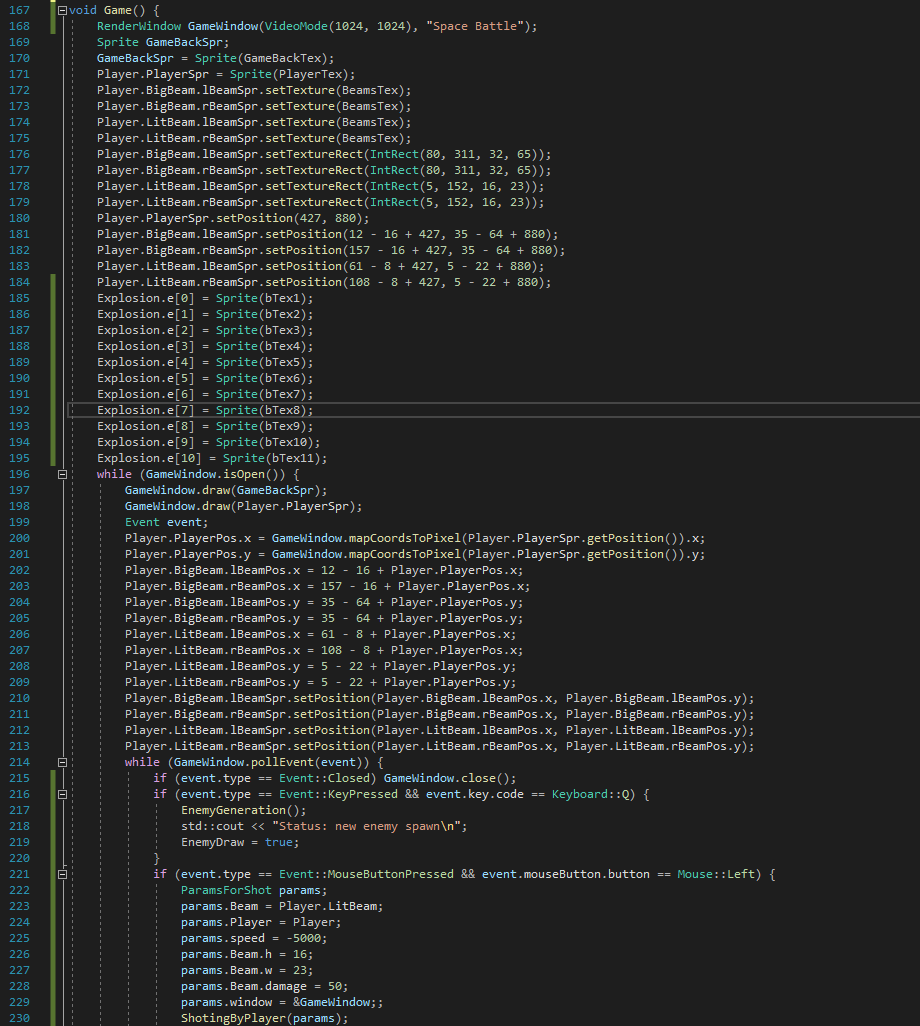


Рис. 16

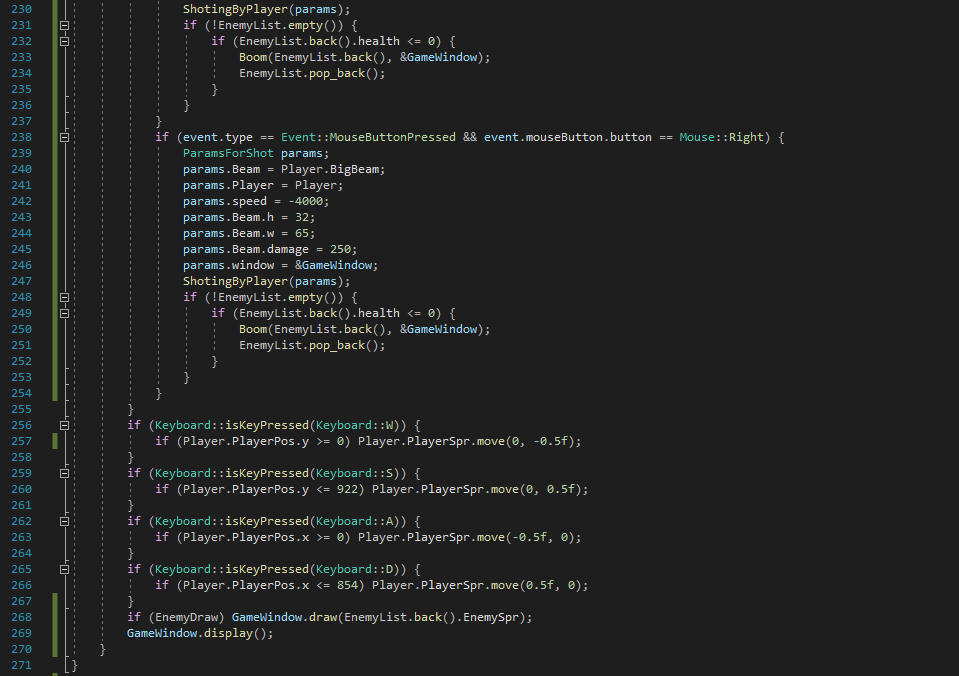


Рис. 17

Создадим окно GameWindow размером 1024х1024, определим свойства

объектов Player и Explosion, затем, пока окно открыто, создаем объект

класса Event, отображаем в окне фон и спрайт космического корабля игрока,

определяем позицию игрока, положение всех лучей относительно

GameWindow. Пока есть события, считываем их:

* Клик по крестику – закрываем окно,
* Нажата левая кнопка мыши – создаем экземпляр структуры

ParamsForShot params для красного луча, заполняем его поля и передаем

полученный экземпляр в функцию стрельбы (ShotingByPlayer); если у

врага кончилось здоровье, то вызываем функцию для анимации взрыва

(Рис. 18) и удаляем последний элемент списка (врага),

* Нажата правая кнопка мыши – аналогично левой кнопки мыши, но

для желтого луча;

если игрок жмет на клавиши “W”, “A”, “S”, “D”, то таким образом он

обеспечивает перемещение спрайта своего космического корабля и неотображаемых спрайтов лучей.

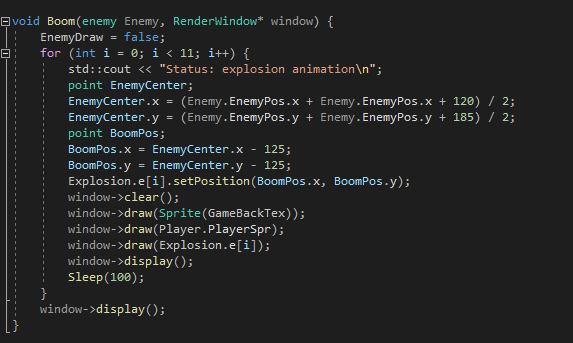


Рис. 18 (Анимация взрыва)

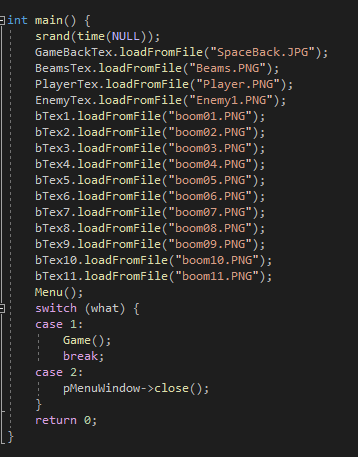
В главной функции (Рис. 19) происходит загрузка текстур в глобальные

переменные, текстуры загружаются из файлов проекта. В дальнейшем,

происходит вызов функции Menu и, если не была выбрана область “EXIT”,

вызывается функция Game, в противном случае – через указатель на окно с меню

закрываем его.

 Рис. 19

**ЗАКЛЮЧЕНИЕ**

Все основные задачи и цели практики были выполнены в достаточной

мере.

**СПИСОК ЛИТЕРАТУРЫ**

<https://www.sfml-dev.org>

<https://habr.com/ru/post/279957/>

<https://habr.com/ru/post/280153/>

<https://devtribe.ru/p/programming/sfml-events-processing>

<https://www.cyberforum.ru>

<https://ru.cppreference.com/w/>

<https://progressor-blog.ru/sfml/otkrytie-i-upravlenie-oknom-sfml/>

<https://ravesli.com/graficheskaya-biblioteka-sfml-sprajty-i-tekstury/>

<https://kychka-pc.ru>

<https://ru.stackoverflow.com>

**ПРИЛОЖЕНИЯ**

Все текстуры были взяты с сайта: <https://opengameart.org>