**

Колледж космического машиностроения и технологии

**КУРСОВОЙ ПРОЕКТ**

**По дисциплине «Прикладное программирование»**

**Тема: «Реализация компьютерной игры типа «Марио» на C++»**

Выполнил студент

Растопчин Андрей Романович

Группа П2-17

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_ (Дата сдачи работы)

Королев, 2020

Оглавление

[**Введение** 3](#_Toc43390983)

[**Глава 1. Теоретическая часть** 3](#_Toc43390984)

[**1.1.** **Описание предметной области** 3](#_Toc43390985)

[**1.2. Описание существующих разработок** 7](#_Toc43390986)

[**Глава 2. Проектная часть** 8](#_Toc43390987)

[**2.1. Диаграмма прецедентов** 8](#_Toc43390988)

[**2.2. Выбор инструментов** 9](#_Toc43390989)

[**2.3. Проектирование сценария** 10](#_Toc43390990)

[**2.4. Диаграмма классов** 11](#_Toc43390991)

[**2.5. Описание главного модуля** 12](#_Toc43390992)

[**2.6. Описание спецификаций к модулям** 12](#_Toc43390993)

[**2.7. Описание модулей** 12](#_Toc43390994)

[**2.8. Описание тестовых наборов модулей** 12](#_Toc43390995)

[**2.9. Описание применения средств отладки** 12](#_Toc43390996)

[**2.10. Анализ оптимальности использования памяти и быстродействия** 12](#_Toc43390997)

[**Глава 3. Эксплуатационная часть** 12](#_Toc43390998)

[**3.1.** **Руководство оператора** 12](#_Toc43390999)

[**Заключение** 12](#_Toc43391000)

[**Список литературы и интернет-источников** 13](#_Toc43391001)

[**Приложения** 13](#_Toc43391002)

# **Введение**

Данный курсовой проект посвящен созданию 2D игры на языке C++.

Выбору такого жанра компьютерной игры, как платформер, послужили: возможность понимания основных механик создания игры и получение полезного опыта при работе с визуальной составляющей.

Цель курсового проекта заключается в понимании и использовании основных методов создания 2D-игр, изучения свободной, кроссплатформернной и мультимедийной библиотеки SFML, а также полезный опыт работы со средой разработки Microsoft Visual Studio.

Для осуществления поставленной задачи служат следующие пункты:

1. Изучение литературы, документации и программных решений по SFML и C++;
2. Подбор мультимедии и подходящих шаблонов для создания будущего персонажа;
3. Написание кода;
4. Описание руководства оператора.

# **Глава 1. Теоретическая часть**

## **Описание предметной области**

**Платформер** - жанр компьютерных игр, в которых основной чертой игрового процесса является прыгание по платформам, лазанье по лестницам, собирание предметов, обычно необходимых для завершения уровня.

Противники (называемые «врагами»), всегда многочисленные и разнородные, обладают примитивным искусственным интеллектом, стремясь максимально приблизиться к игроку, либо не обладают им вовсе, перемещаясь по круговой дистанции или совершая повторяющиеся действия. Соприкосновение с противником обычно отнимает очки жизни у героя или вовсе отправляет в начало уровня. Чтобы справиться с противником обычно достаточно обойти его (если это необязательный персонаж) или устранить его прыгнув ему на голову или из оружия, если оно есть. Устранение противника обычно изображается символически (существо исчезает или проваливается под карту).

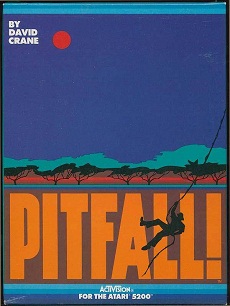
Уровни как правило наполнены большим количеством потайных (невидимых) комнат или бонусами, что упрощает прохождение и подогревает к нему интерес.

Игры такого жанра как правило характеризуются мультяшностью окружения и персонажей.

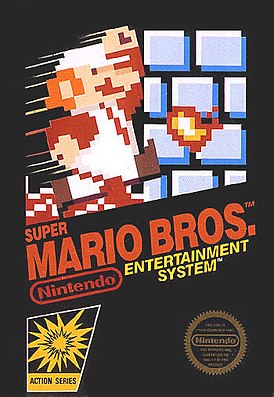
Путь платформеров начался в 1980-х годах с 2D графики, когда игровые консоли не были достаточно мощными, чтобы отображать трехмерную графику или видео. Они были ограничены статическими игровыми мирами, которые помещались на один экран, а игровой герой был виден в профиль. Персонаж лазал вверх и вниз по лестницам или прыгал с платформы на платформу, часто сражаясь с противниками и собирая предметы, улучшающие характеристики. Первыми играми этого типа были *Space Panic* и его последователь *Donkey Kong*, эти игры относились к аркадному жанру.



Вскоре процесс прохождения уровня перестал быть в основном вертикальным и стал горизонтальным с появлением длинных многоэкранных прокручивающихся игровых миров. Считается, что начало этому положила выпущенная фирмой Activision в 1982 году игра *Pitfall!* для консолей Atari 2600. Стоит отметить, что особенностью игры являась резкая смена кадров при перемещении по карте, а не всем так примычное следование камеры за игровым персонажем.



В 1985 году фирма «Nintendo» выпустила для приставки Nintendo Entertainment System революционный платформер Super Mario Bros.



Игра была наполнена большими и сложными уровнями, и стала примером для последующих создателей игр, и даже сегодня многие люди считают её одной из самых лучших видеоигр.

**Трёхмерный платформер** - геймплей, включающий все три измерения, или использование трёхмерных полигонов в реальном времени для отрисовки уровней и героев, или и то и другое. Появление трёхмерных платформеров принесло изменение конечных целей некоторых платформеров. В большинстве двумерных платформеров игроку нужно было достичь на уровне только одной цели, однако во многих трёхмерных платформерах, каждый уровень необходимо прочесывать, собирая кусочки головоломок (*Banjo-Kazooie*) или звезды (*Super Mario 64*).



**Изометрические платформеры** - поджанр и дву-, и трёхмерных платформеров. Они отображают трёхмерную сцену с помощью двумерной графики, которая отображает мир с жёстко ориентированной камеры без учёта перспективы. И хотя изометрические платформеры не были первыми изометрическими играми, ранними примерами изометрических платформеров в мире аркадных автоматов является игра 1983 года *Congo Bongo*.



***Super Mario*** - серия компьютерных игр в жанре платформер, издаваемых компанией Nintendo. Часть медиафраншизы Mario. Первая игра в серии - *Super Mario Bros.* - вышла в 1985 году, последняя - *Super Mario Maker 2* - в 2019 году. Серия *Super Mario* является основной линейкой игр в своей франшизе.

* [*Super Mario Bros.*](https://ru.wikipedia.org/wiki/Super_Mario_Bros.) ([NES](https://ru.wikipedia.org/wiki/NES))
* [*Super Mario Bros.: The Lost Levels*](https://ru.wikipedia.org/wiki/Super_Mario_Bros.:_The_Lost_Levels) (NES)
* [*Super Mario Bros. 2*](https://ru.wikipedia.org/wiki/Super_Mario_Bros._2) (NES)
* [*Super Mario Land*](https://ru.wikipedia.org/wiki/Super_Mario_Land) ([Gameboy](https://ru.wikipedia.org/wiki/Gameboy" \o "Gameboy))
* [*Super Mario Bros. 3*](https://ru.wikipedia.org/wiki/Super_Mario_Bros._3) (NES)
* [*Super Mario World*](https://ru.wikipedia.org/wiki/Super_Mario_World) ([SNES](https://ru.wikipedia.org/wiki/SNES))
* [*Super Mario Land 2: 6 Golden Coins*](https://ru.wikipedia.org/wiki/Super_Mario_Land_2:_6_Golden_Coins) (Gameboy)
* [*Wario Land: Super Mario Land 3*](https://ru.wikipedia.org/w/index.php?title=Wario_Land:_Super_Mario_Land_3&action=edit&redlink=1) (Gameboy)
* [*Super Mario All Stars*](https://ru.wikipedia.org/w/index.php?title=Super_Mario_All_Stars&action=edit&redlink=1) (SNES)
* [Super Mario World 2: Yoshi's Island](https://ru.wikipedia.org/wiki/Super_Mario_World_2:_Yoshi%27s_Island) (SNES)
* [*Super Mario 64*](https://ru.wikipedia.org/wiki/Super_Mario_64) ([Nintendo 64](https://ru.wikipedia.org/wiki/Nintendo_64" \o "Nintendo 64))
* [*Super Mario Sunshine*](https://ru.wikipedia.org/wiki/Super_Mario_Sunshine) ([Nintendo GameCube](https://ru.wikipedia.org/wiki/Nintendo_GameCube))
* [*New Super Mario Bros.*](https://ru.wikipedia.org/wiki/New_Super_Mario_Bros.) ([Nintendo DS](https://ru.wikipedia.org/wiki/Nintendo_DS))
* [*Super Mario Galaxy*](https://ru.wikipedia.org/wiki/Super_Mario_Galaxy) ([Wii](https://ru.wikipedia.org/wiki/Wii" \o "Wii))
* [*New Super Mario Bros. Wii*](https://ru.wikipedia.org/wiki/New_Super_Mario_Bros._Wii) (Wii)
* [*Super Mario Galaxy 2*](https://ru.wikipedia.org/wiki/Super_Mario_Galaxy_2) (Wii)
* [*Super Mario 3D Land*](https://ru.wikipedia.org/wiki/Super_Mario_3D_Land) ([3DS](https://ru.wikipedia.org/wiki/3DS))
* [*New Super Mario Bros. 2*](https://ru.wikipedia.org/wiki/New_Super_Mario_Bros._2) (3DS)
* [*New Super Mario Bros. U*](https://ru.wikipedia.org/wiki/New_Super_Mario_Bros._U) ([Wii U](https://ru.wikipedia.org/wiki/Wii_U))
* [*New Super Luigi U*](https://ru.wikipedia.org/w/index.php?title=New_Super_Luigi_U&action=edit&redlink=1) (Wii U)
* [*Super Mario 3D World*](https://ru.wikipedia.org/wiki/Super_Mario_3D_World) (Wii U)
* [*Super Mario Maker*](https://ru.wikipedia.org/wiki/Super_Mario_Maker) ([Wii U](https://ru.wikipedia.org/wiki/Wii_U) / [3DS](https://ru.wikipedia.org/wiki/3DS))
* [*Super Mario Run*](https://ru.wikipedia.org/wiki/Super_Mario_Run) ([Android](https://ru.wikipedia.org/wiki/Android) / [iOS](https://ru.wikipedia.org/wiki/IOS))
* *Super Mario Odyssey* (Nintendo Switch)
* *Super Mario Maker 2* (Switch)

Особо можно отметить [*Super Mario 64*](https://ru.wikipedia.org/wiki/Super_Mario_64), эта часть стала первой игрой с трехмерной графикой во франшизе.

## **1.2. Описание существующих разработок**

**Jet Fire** - игра жанра платформер основной целью которой является увеличение пройденного расстояния.

Игра вызывает тягу к получению все новых и новых рекордов, а возможность улучшения персонажа посредством сбора монет во время игры еще больше подкрепляет интерес к игре.

Из улучшений программной составляющей хотелось бы предложить увеличение скорости персонажа, более резкие падение и прыжок, а также возможность перемещения персонажа вперед, но и назад.



Ссылка на источник: <http://flashdozor.ru/play-53367.html>

**Toto Double Trouble** - игра жанра платформер и представляет собой игру сделанную исключительно для 2 игроков, целью которых является вместе дойти до конца уровня.

Персонажи полностью зависят друг от друга и ни один, ни второй не смогут дойти до конца без взаимопомощи.

Плюсом игры является необходимость обдумывания своих действий и работа в команде, а также сложность, которая растет с каждым новым уровнем.

А из минусов стоит отметить необходимость нахождения второго игрока непосредственно за одним компьютером с первым, так как персонажи управляются с одного компьютера.



Источник игры: <http://flashdozor.ru/play-53360.html>

# **Глава 2. Проектная часть**

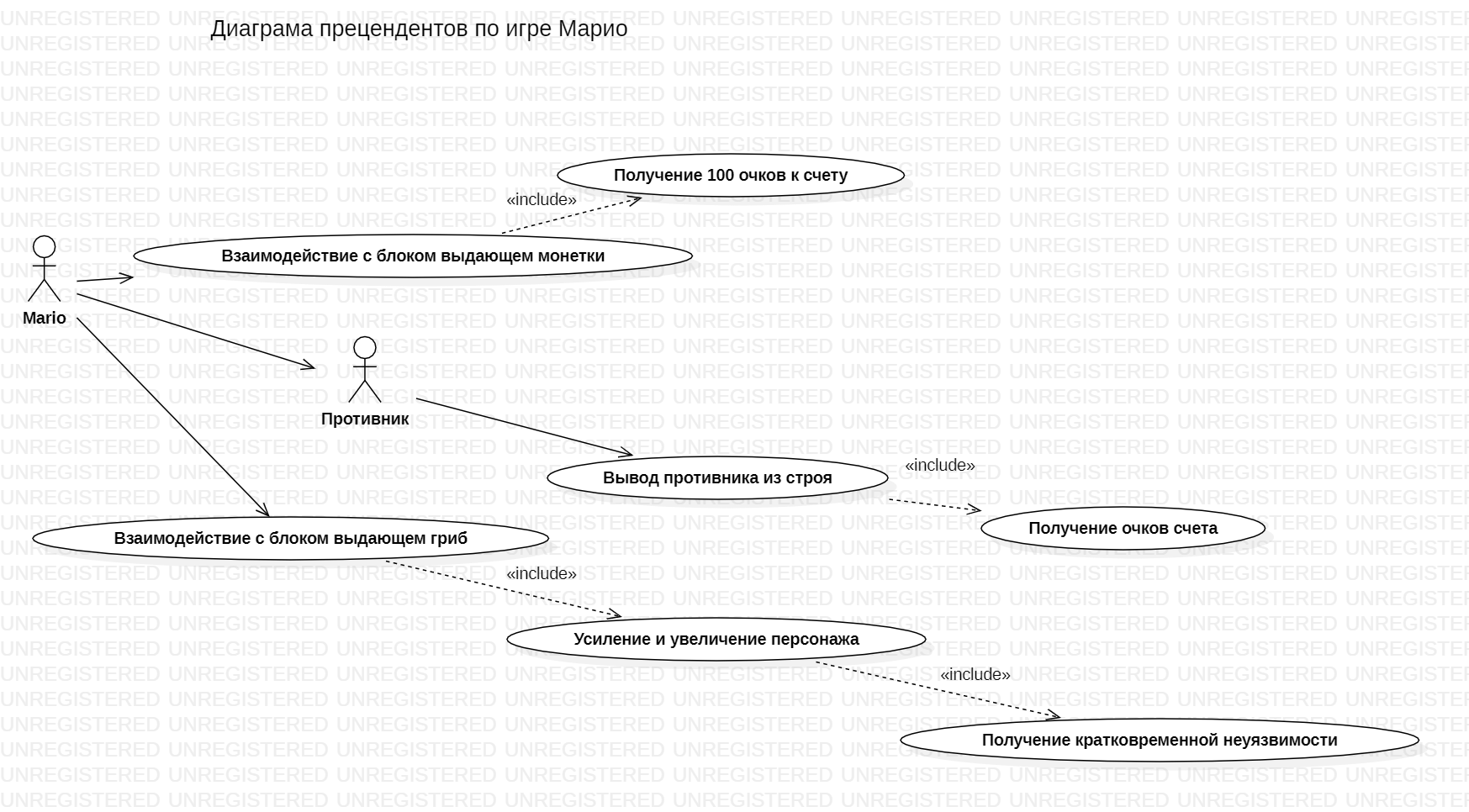
## 

## **2.1. Диаграмма прецедентов**

Происходит взаимодействие игрока с персонажем (Mario).

В свою очередь персонаж взаимодействует с объектами и другими персонажами:

1. Взаимодействие с блоком выдающем монеты приводит к получению очков, которые прибавляются к счету игрока.
2. Взаимодействие с “Противником” приводит к его устранению и как следствию получения очков.
3. Взаимодействие с блоком выдающем грибы приводит к подбору гриба персонажем игрока и временному усилению персонажа (неуязвимости).



## 

## **2.2. Выбор инструментов**

Выбор инструментов является основополагающей задачей, стоящей перед разработчиком.

Нужно соотнести все плюсы и минусы имеющихся вариантов и выбрать тот инструмент, которым вам комфортнее пользоваться.

|  |  |  |
| --- | --- | --- |
| **Критерии** | **CodeBlocks** | **Microsoft Visual Studio** |
| участие в корпоративном проекте | 8 | 10 |
| простота сопровождения | 10 | 8 |
| наличие библиотек | 7 | 10 |
| наличие документации на русском языке | 6 | 10 |
| скорость разработки и т.д. | 8 | 9 |
| Итого: | 39 | 47 |

Выбор пал на Microsoft Visual Studio.

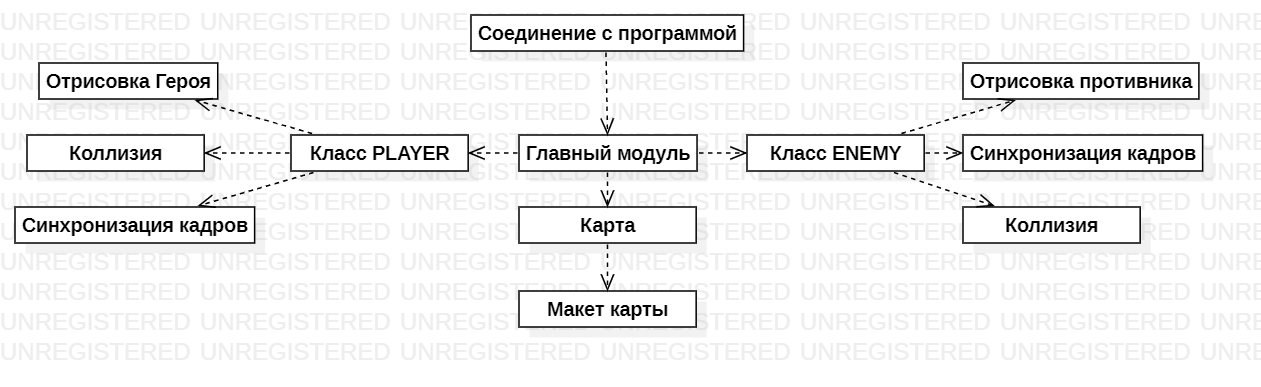
Отдельно стоит отметить, что работа в Microsoft Visual Studio значительно упрощает подключение дополнительных библиотек, таких как SFML и значительно упрощает отладку программы, поиск и исправление ошибок в коде.

## 

## **2.3. Проектирование сценария**

**Сценарий** – план по которому будут развиваться события в программе.

Сценарий является важным аспектом любой проектной деятельности. Именно благодаря сценарию любой другой человек или сам разработчик может понять и представить как будет выглядеть работа в конечном плане.



## **2.4. Диаграмма классов**

**Диаграмма классов** - структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

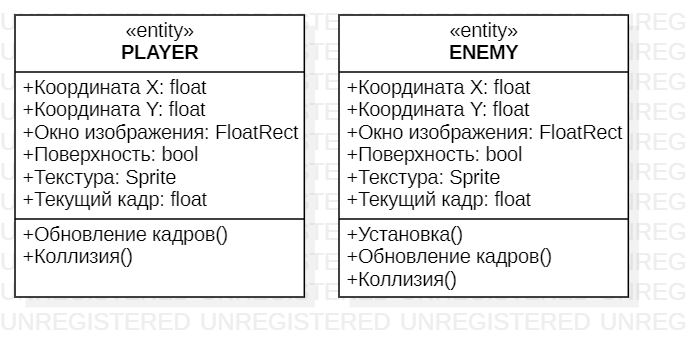
В диаграмме представлены два класса PLAYER (Игрок) и ENEMY(противник).

В классе PLAYER используется спецификатор доступа public, что говорит нам о том, что доступ открыт всем, кто видит определение данного класса.

Блок стоящий после PLAYER показывает нам объявленные переменные, а знак “+” указывает на спецификацию доступа, в данному случае только public.

После блока с атрибутами, расположен блок с операциями (функциями класса).

У класса ENEMY структура полостью совпадает со структурой класса PLAYER, за исключением еще одной операции (Установка()).



## **2.5. Описание главного модуля**

**Главный модуль программы** – часть программы которая обеспечивает общее функционирование системы, а также всех прочих модулей программы.

В проекте “Марио” главный модуль программы находится в файле main.cpp и отвечает за функционирование классов ENEMY и PLAYER и заголовочного файла с картой, реализует подключение картинок, звуков и музыки, осуществляет работу кнопок управления персонажем, реализует создание карты из картинок по макету из map.h, реализует отображение текста в окне, а также создает меню игры и окно демонстрации изображения.

## **2.6. Описание спецификаций к модулям**

## **2.7. Описание модулей**

## **2.8. Описание тестовых наборов модулей**

## **2.9. Описание применения средств отладки**

## **2.10. Анализ оптимальности использования памяти и быстродействия**

# **Глава 3. Эксплуатационная часть**

## **Руководство оператора**

**Назначение программы**

**Функциональное назначение программы**

Основной функцией проекта “Mario” является развлекательная составляющая. Игра помогает отвлечься от реального мира и погрузиться в увлекательное прохождение уровней и сражения с противниками.

**Эксплуатационное назначение программы**

Основное назначение проекта “Mario” – показать концепцию всех игр жанра платформер, визуализировать методы работы с текстурами, звуками и текстом, а также реализовать связь между созданными моделями и действиями игрока.

**Состав функций**

**Эмоциогенная функция**

Игра меняет эмоциональное состояние человека, как правило в лучшую сторону, повышает настроение и провоцирует рост воодушевления. Игра пробуждает целый спектр чувств у пользователя, это и удовольствие, и радость и даже страх.

**Терапефтическая функция**

Согласно с выводами Бёрна, в игры чаще всего играют люди с нарушенным душевным равновесием, игры помогают в снятии стресса и отвлечения от реальных проблем, чтобы было время расслабиться и вернуться к ним с новыми силами.

**Условия выполнения программы**

**Минимальный состав аппаратных средств**

Минимальный состав используемых технических (аппаратных) средств:

* Windows 10 версии 1703 или выше;
* Процессор с тактовой частотой не ниже 1,8 ГГц.
* 2 ГБ ОЗУ; рекомендуется 8 ГБ ОЗУ
* Место на жестком диске: до 200 ГБ (20 – 50 ГБ для Microsoft Visual Studio 2017, 100 МБ проект);
* Видеоадаптер с минимальным разрешением 720p (1280 на 720 пикселей);

**Минимальный состав программных средств**

Системные программные средства, используемые проектом “Mario”, должны быть представлены локализованной версией операционной системы Windows 7, Windows 8.1 или Windows 10.

Также для функционирования программы Mem.exe на ПК необходимо предустановленное программное обеспечение стороннего разработчика, программа Microsoft Visual Studio 2017 и пакет библиотеки SFML, для 32-х и 64-х разрядной системы.

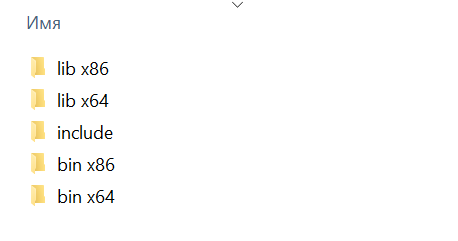
**Требование к персоналу (Пользователю)**

Конечный пользователь программы (оператор) должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

**Выполнение программы**

**Загрузка и запуск программы**

Загрузка и запуск проекта “Mario” осуществляется по следующим пунктам:

1. Необходимо вставить диск с программой в дисковод и скачать пакет с проектом.
2. Распаковать проект в папку, находящуюся на локальном диске.
3. Скачать библиотеку SFML с официального сайта вида: Visual C++ 15 (2017) (32-х и 64-х битную версии), в папку на локальном диске.
4. Создать общую папку из 32-х и 64-х битных версий библиотеки, поместив туда папки: от 32-х битной версии (include, lib, bin), изменяя названия папок lib и bin на lib32 и bin32; От 64-х битной версии только lib и bin меняя названия на lib64 и bin 64. Вот что должно получиться: 
5. Открыть проект и запустить файл Mario.sln через Microsoft Visual Studio 2017.

# **Заключение**

# **Список литературы и интернет-источников**

# **Приложения**