**

Колледж космического машиностроения и технологии

**КУРСОВОЙ ПРОЕКТ**

**По дисциплине «Прикладное программирование»**

**Тема: «Разработка игры "Танчики"»**

Выполнил студент

Стрельников Сергей Дмитриевич

Группа П2-17

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

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

Королев, 2020

Оглавление

[Задание на курсовой проект 2](#_Toc43403660)

[Введение 3](#_Toc43403661)

[Глава 1. Теоретическая часть 3](#_Toc43403662)

[1.1. Описание предметной области 3](#_Toc43403663)

[1.1.1. Жанр Аркад 3](#_Toc43403664)

[1.1.2. Описание библиотеки SFML 5](#_Toc43403665)

[1.1.3. Описание языка программирования С++ 5](#_Toc43403666)

[1.1.4. Составляющие персонажа и противников («Танчики») 6](#_Toc43403667)

[1.2. Описание существующих разработок 10](#_Toc43403668)

[Глава 2. Проектная часть 11](#_Toc43403669)

[2.1. Диаграмма прецедентов 11](#_Toc43403670)

[2.2. Выбор инструментов 12](#_Toc43403671)

[1.2.1. Библиотека SFML 12](#_Toc43403672)

[1.2.2. Текстовый редактор Sublime Text 3 12](#_Toc43403673)

[1.2.3. Компилятор G++ 12](#_Toc43403674)

[2.3. Проектирование сценария 13](#_Toc43403675)

[2.4. Диаграмма классов 13](#_Toc43403676)

[2.5. Описание главного модуля 13](#_Toc43403677)

[2.6. Описания спецификаций к модулям 13](#_Toc43403678)

[2.7. Описание модулей 13](#_Toc43403679)

[2.8. Описание текстовых наборов модулей 13](#_Toc43403680)

[2.9. Описание применения средств отладки 13](#_Toc43403681)

[2.10. Анализ оптимальности использования памяти и быстродействия 13](#_Toc43403682)

[Глава 3. Эксплуатационная часть 13](#_Toc43403683)

[3.1. Руководство оператора 13](#_Toc43403684)

[Заключение 13](#_Toc43403685)

[Список литературы и интернет-источников 13](#_Toc43403686)

[Приложення 13](#_Toc43403687)

Задание на курсовой проект

Разработать игру «Танчики» с использованием библиотеки SFML и языка C++

Введение

В современном мире создание видеоигр является одним из наиболее крупных сегментов индустрии развлечений. С каждым годом игры показывают, что могут являться способом творческого самовыражения, не

уступающим литературе или живописи, а финансовый успех таких игровых франшиз, как Call of Duty или Assassin’s Creed – живое доказательство тому, что индустрия игр развивается очень быстро.

Все больше людей придерживаются мнения, что игры – это не просто искусство, но и вообще его абсолют, объединяющий в себе остальные проявления творчества.

Выбранная мною темя является желанием понимать основные принципы создания игр и возможность самому написать интересный продукт с оглядкой на существующие решения.

Цель курсового проекта заключалась в создании 2D аркадной видеоигры

Для осуществления обозначенной цели служат следующие задачи:

1. Изучение документации по SFML;
2. Создание текстур для игровых объектов и карты;
3. Создание интерфейса;
4. Создание компонентов игры;
5. Написание кода
6. Описание руководства оператора

Глава 1. Теоретическая часть

* 1. Описание предметной области
     1. Жанр Аркад

Аркада (англ.*arcade game*)— жанр компьютерных игр, характеризующийся коротким по времени, но интенсивным игровым процессом.У жанра “аркад” есть множество поджанров, включающих в себя платформеры, простейшие симуляторы, файтинги, и шутеры

Классические аркады характеризуются следующими свойствами:

* ***Игра на одном экране*.** Все действия происходят на одном экране чтобы игроки в любой момент времени могли видеть весь игровой мир и принимать решения. У многих игр было больше одного экрана, каждый экран был отдельным уровнем**.** Примеры таких игр: *Joust*, *Pac-Man*, *Mario Bros.*
* ***Бесконечная игра.*** Заключается в том, что игроки могут играть в аркаду бесконечное время, и соответственно, не могут выиграть. Игроков удерживало то, что они делали вызов сами себе – на сколько долго можно продержаться. Относительно геймдизайна в аркадах игрок никогда не выигрывал, и каждая игра заканчивалась поражением. Со временем играть становилось все сложнее из-за повышающего уровня слолжности
* ***Множество жизней***. Обычно, классическая аркада предлагает игроку несколько попыток. Это делается для того, чтобы игрок анализировал свои ошибки, учился на них и не проходил игру или уровень заново.
* ***Игровой счёт.*** Практически все классические аркады включают в себя игровой счёт. Когда игрок получает очки за выполнение различных целей
* ***Быстрое обучение****, простой игровой процесс*. Для многих аркад характерен не сложный игровой процесс, в который легко понять даже новичку. От профессионала его будет отличать лишь умение грамотно пользоваться небольшим количеством функций
* ***Нет сюжета/истории***. Во многих, но не во всех аркадах, нет сюжета. Он попросту не нужен, потому что игрок сам в состоянии понять что происходит на экране, и сам игровой процесс не нуждается в какой-либо истории.
* ***Искусственный интеллект*** (ИИ) аркадных игр как правило, представляет собой простые действия и не реагирует на поведение игрока. И, зачастую, сложность увеличивается просто количеством противников
  + 1. Описание библиотеки SFML

SFML это аббревиатура, расшифровывающаяся как Simple and fast multimedia library, что в переводе простая и быстрая мультимедийная библиотека. Данная библиотека нацелена на создание двухмерных игр, которые можно реализовать различными и основными приёмами, такие как использование спрайтовой анимации, использование графических примитивов, С SFML ваше приложение может компилироваться и запускаться из коробки в самых распространенных операционных системах: Windows, Linux, macOS и, в скором времени, на Android и iOS.

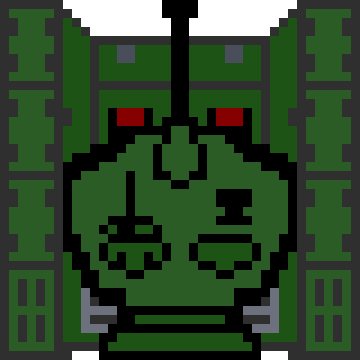
SFML многоязычный, имеет официальные привязки для языков C и .Net. А благодаря активному сообществу он также доступен на многих других языках, таких как Java, Ruby, Python, Go и других.

* + 1. Описание языка программирования С++

Язык программирования С++ это высокоуровневый компилируемый язык программирования со статической типизацией, который подходит для создания самых различных приложений. На сегодняшний день С++ является одним из самых популярных и распространенных языков.

Своими корнями он уходит в язык Си, который был разработан в 1969—1973 годах в компании Bell Labs программистом Деннисом Ритчи (Dennis Ritchie). В начале 1980-х годов датский программист Бьерн Страуструп (Bjarne Stroustrup), разработал С++ как расширение к языку Си. Фактически вначале C++ просто дополнял язык Си некоторыми возможностями ООП.

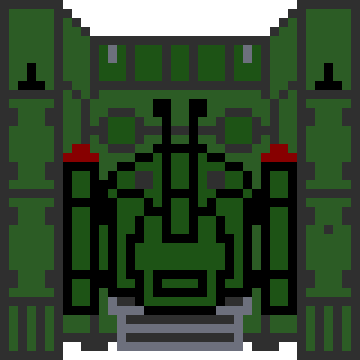
* + 1. Составляющие персонажа и противников («Танчики»)

Все представленные в курсовом проекте текстуры «Танчиков» срисованы с реальных образцов военной техники.

**ОБТ Т-90 «Владимир»**

**«Танчик Т-90»**

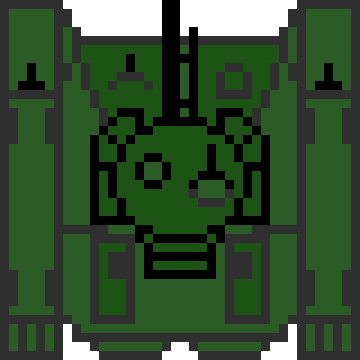
«Танчик Т-90» срисован с российского основного боевого танка Т-90 «Владимир»



**«Танчик БМПТ «Терминатор»»**

**«**Танчик БМПТ «Терминатор»» срисован с российской боевой машины поддержки танков БМПТ-2 «Терминатор»

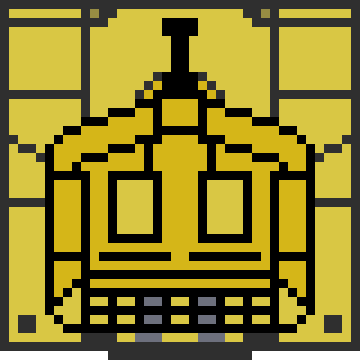
**БМПТ-2 «Терминатор»**



**«Танчик БМП-3»**

«Танчик БМП-3» срисован с российской боевой машины пехоты БМП-3

**БМП-3**

«Танчик М1 «Абрамс»» срисован с основного боевого танка США M1A2 «Abrams»

M1A2 «Abrams»

**«Танчик М1 «Абрамс»»**

* 1. Описание существующих разработок

В основе курсового проекта лежат механики из уже существующих

проектов.

В их числе аркада 1980 года Battle City

Battle City – это аркадная игра в которой основной задачей является защита базы от танков противника. В процессе игры га поле будут появляться различные бонусы: гранаты, лопаты, дополнительные жизни и другое.

Достоинства:

- Затягивающий игровой процесс;

- Запоминающееся звуковое сопровождение;

- Возможность редактирования карт

Недостатки:

 - Отсутствую. Игра является классической и все недостатки пришли только из-за технического устаревания игры

Battle City

Из игры Battle city присутствуют следующие механики:

* Полигон действий виден сверху
* Управлением игроком «Танчиком»
* Появление врагов в разных местах карты

Аркадный танковый экшн Armored Warfare 2015 года.

Представляет собой попытку создать конкурента игре World of Tanks только в современном сеттинге. Разрабатывалась obsidian entertainment, после была передана mail.ru group. Игровой процесс это – онлайн бой 15 на 15 игроков на разных видах современной бронетехнике

Преимущества:

- Динамичный игровой процесс

- Разнообразие видов техники

- Красивая графика

Недостатки

- Требовательна к железу компьютера

- Навязывает финансовые вливания для комфортной игры

 - Маленькое комьюнити

Armored Warfare

Из игры Armored Warfare были взяты следующие решения:

* Механика ПТУРов (Управляемых снарядов)
* Механика Авто-Пушек
* Механика неуправляемых снарядов

Глава 2. Проектная часть

* 1. Диаграмма прецедентов

Для определения вариантов использования к проекту была построены диаграммы прецедентов.

 Диаграмма прецедентов «Пользователь» показывает возможные действия пользователя после запуска приложения. Пользователь может запустить саму игру, просмотреть окно помощи или выйти из приложения

Диаграмма прецедентов Пользователь

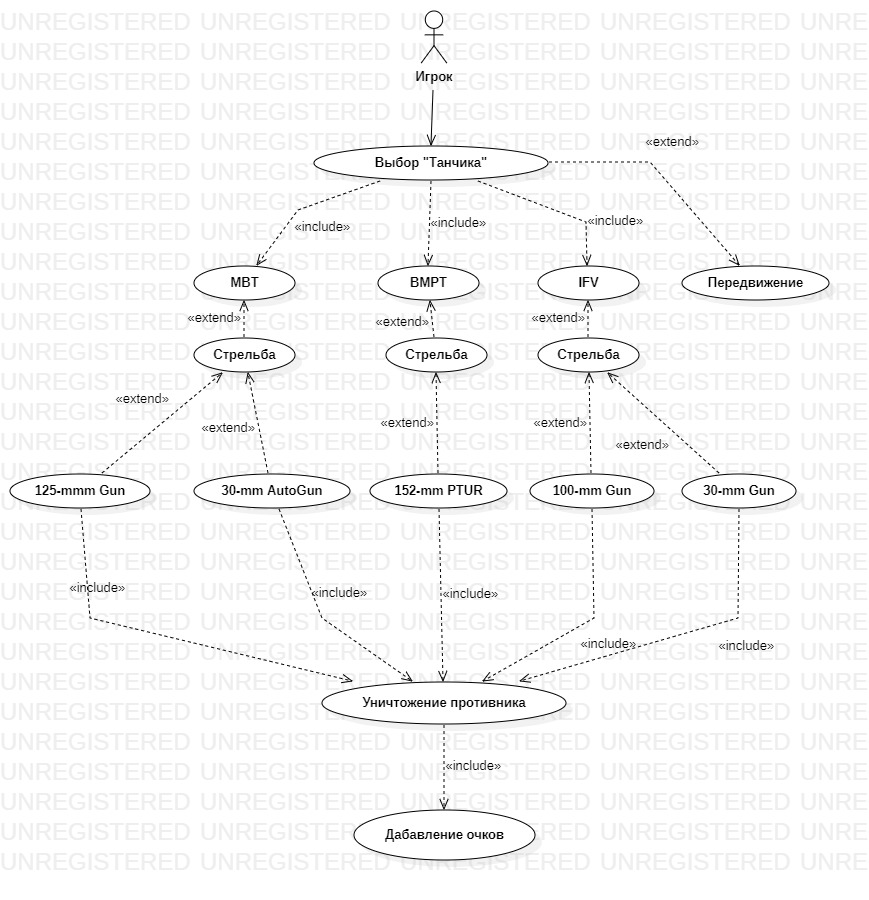
Диаграмма прецедентов «Игрок» показывает возможные действия игрока во время игры. Игрок может выбрать «Танчик» и в последствии управлять им, уничтожать противников из доступных вариантов вооружения и получать за это очки.

Диаграмма прецедентов Игрок

* 1. Выбор инструментов

Для написания проекта были выбраны: библиотека SFML, текстовый редактор sublime text 3 и компилятор языка C++ GNU GCC на основе следующих таблиц сравнения

Важность критерия определяется по шкале от 0 до 10 исходя из своих предпочтений и знаний

|  |  |  |  |
| --- | --- | --- | --- |
|  | SFML | SDL | GTK+ |
| **Простота синтаксиса** | 7 | 3 | 7 |
| **Документация на Русском языке** | 10 | 6 | 8 |
| **Кроссплатформенность** | 10 | 10 | 10 |
| **Итог:** | 27 | 19 | 15 |
|  |  |  |  |
|  | Sublime Text 3 | Atom | Vim |
| **Интерфейс** | 9 | 9 | 6 |
| **Простота использования** | 6 | 3 | 3 |
| **Популярность** | 6 | 4 | 10 |
| **Итог:** | 21 | 16 | 19 |
|  |  |  |  |
|  | Pythom | C++ |  |
| **Простота синтаксиса** | 9 | 5 |  |
| **Документация** | 4 | 8 |  |
| **Скорость** | 5 | 9 |  |
| **Использование памяти** | 4 | 10 |  |
| **Итог:** | 22 | 32 |  |

* + 1. Библиотека SFML

SFML это графическая библиотека для создания простой 2-d графики и спрайтовой анимации. Для написания проекта использовались модули:

* sfml-graphics
* sfml-system
* sfml-window
  + 1. Текстовый редактор Sublime Text 3

Sublime text 3 – это проприетарный текстовый редактор.

Функциональность придают такие возможности как быстрая навигация, подсветка синтаксиса, высокая степень настраиваемости, богатые визуальные схемы.

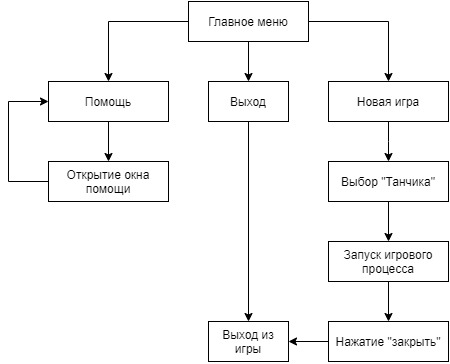
* + 1. Компилятор GNU GCC

Свободно распространяемый компилятор языка с++. Позволяет получать полную и точную информацию о работе о программе при отладке.

* 1. Проектирование сценария

Данная программа ориентирована на следующие 2 последовательных сценария использования:

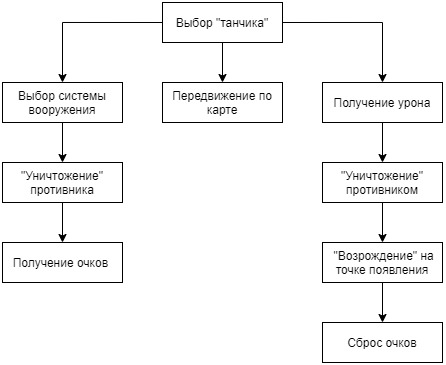
1. Сценарий пользователя при запуске приложения:



Сценарий использования

После запуска программы пользователь имеет следующие варианты действия

* Запуск новой игры
* Просмотр окна помощи
* Выход из программы

1. Сценарий пользователя при запуске игрового процесса

После запуска игрового процесса и выбора «танчика» игрок имеет

Следующие варианты действия:

* Передвижение по игровой карте
* Выбор системы вооружения
* «Уничтожение» противников
* Получение урона от противников
* «Уничтожение» противником

После «Уничтожения» противника игрок получает фиксированное количество очков к своему счёту.

После «Уничтожения» противником игрок перемещается на место старта и происходит сброс набранных очков

* 1. Диаграмма классов

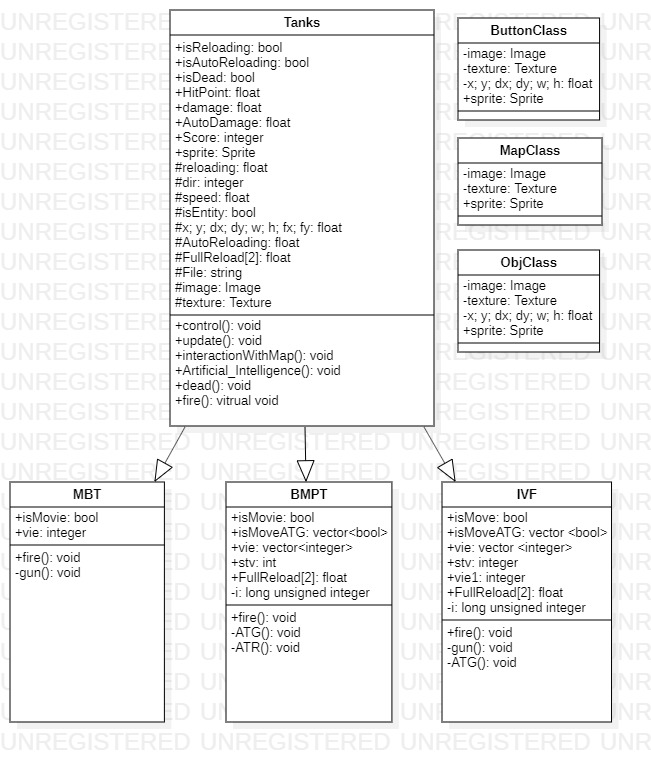
После определения функциональных требований к проекту, была описана предметная область с целью построения диаграммы классов

Диаграмма коассов

* 1. Описание главного модуля

Главный модуль представляет с собой файл Game.cpp.

В главный модуль разработки входит вызов функции главного меню, которая отвечает за дальнейшее использование приложения.

Данный модуль запускается единожды и является самым первым в запуске приложения. Состоит из кода, который приведён в листинге 1.

**Листинг 1:**

#include <iostream>

#include <vector>

#include <cstring>

#include <sstream>

#include <SFML/Graphics.hpp>

//Подключение файла содержащий функции главного меню

#include "resourse/MainMenu.hpp"

using namespace std;

using namespace sf;

int main()

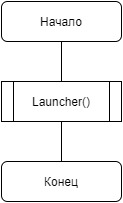
{

//Вызов функции главного меню

Launcher();

return 0;

}

**Блок-схема к главному модулю:**

* 1. Описания спецификаций к модулям

Описание всех публичных членов модулей курсового проекта

Главный модуль Game.cpp

В главном модуле только вызывается функция главного меню.

Модуль главного меню MainMenu.cpp

Press() – отвечает за подсветку кнопок

FrameMenu() – отвечает за смену кадров анимации фона

Launcher() – отвечает за работу главного меню

Модуль игрового процесса mainGame.cpp

Press() – отвечает за подсветку при выборе «Танчика»

MapFunct() – отвечает за отрисовку карты

Choice() – отвечает за выбор «Танчика»

isFire() – отвечает за вывод статуса перезарядки

Score() – отвечает за подчёт очков и их вывода

Game() – отвечает за запуск игрового процесса

Модуль родительского класса Tanks.hpp

control() – отвечает за управление «Танчиком»

update() – отвечает за передвижение «Танчиков»

interactionWithMap() - отвечает за проверку столкновения с препятствиями

Artificial\_Intelligence() – отвечает за передвижение противников

dead() – отвечает за обработку попаданий по «Танчикам». Является перегруженной

fire() – отвечает за функции ведения огня «Танчиков». Является виртуальной

Модуль дочернего класса ClassMBT.hpp

Fire() – отвечает за функцию ведения огня «Танчика» класса MBT

Модуль дочернего класса ClassBMPT.hpp

Fire() – отвечает за функцию ведения огня «Танчика» класса BMPT

Модуль дочернего класса ClassIFV.hpp

Fire() – отвечает за функцию ведения огня «Танчика» класса IFV

* 1. Описание модулей

В данном пункте будут описаны все оставшиеся модули курсового проекта.

* Модуль главного меню

Данный модуль представляет собой

* Модуль игрового процесса
* Модуль родительского класса
* Модуль дочернего класса MBT
* Модуль дочернего класса BMPT
* Модуль дочернего класса IFV
* Модуль карты
* Модуль класса для кнопок

Представляет собой набор переменных для создания спрайтов кнопок из указанного файла, задание местоположения.

* Модуль класса для объектов

Представляет собой набор переменных для создания спрайтов объектов из указанного файла, установки их границ и задание местоположения.

* 1. Описание тестовых наборов модулей
  2. Описание применения средств отладки
  3. Анализ оптимальности использования памяти и быстродействия

Глава 3. Эксплуатационная часть

3.1. Руководство оператора

Заключение

Список литературы и интернет-источников

Приложення