# АННОТАЦИЯ

В пояснительной записке описывается модель игры для операционной системы Windows, спроектированной и разработанной на базе объектного подхода.

Документ содержит описание назначения, технических требований, проектного решения и программной реализации. Так же в документе предоставлено детальное пояснение структуры моделируемой системы, взаимодействия программных модулей, описание реализации основных классов и их методов. В пояснительную записку включено подробное руководство по эксплуатации системы и критерии качества моделируемой системы.

# СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

В современное время, объектно-ориентированный подход является наиболее актуальным и распространенным, так как позволяет эффективно проектировать и создавать сложные программные системы. Со временем размеры систем возрастают, и это не должно сказываться на их надежности и эффективности. Объектно-ориентированный подход предоставляет возможности для расширения масштаб систем, и при необходимости легко расширяя их функционал.

В данной курсовой работе поставлена задача на изучение современного подхода к программированию на основе объектно-ориентированной технологии и приобретение навыков программирования на примере написания программы на языке с поддержкой ООП. Разработка программы ведется на основании технического задания, выданного 09 февраля 2018 года кафедрой информационных систем Севастопольского государственного университета.

Целями курсового проектирования является изучение современного подхода к программированию на основе объектно-ориентированной технологии, приобретение навыков написания программных систем со сложным характером взаимодействия значительного количества элементов.

В данном курсовом проекте требуется организовать детальное проектирование и программную реализацию системы, моделирующую процессы создания и взаимодействия группы объектов.

В данной пояснительной записке содержится полная информация о разрабатываемом проекте, разбитая по пунктам.

Пункт «Список исполнителей» содержит информацию о ролях разработчиков и их вкладе в курсовой проект.

Пункт «Постановка задачи» описывает цель разработки, условия выполнения программы, а также накладываемые ограничения.

В пункте «Проектное решение» содержится информация о структуре системы, ее иерархии классов, их свойствах, а также жизненных циклов объектов и программы в целом.

Пункт «Программная реализация» предоставляет обоснования при выборе языка написания программы, а так же описывает особенности программной реализации основных классов и их методов. Так же в этом пункте дается оценка качества и полноты разработанной системы.

В пункте «Заключение» подводятся итоги работы, а так же пункт содержит выводы, сделанные при разработке курсового проекта.

# СПИСОК ИСПОЛНИТЕЛЕЙ

Данный курсовой проект разрабатывался студентами Волковой А. А. и Генераловым Н. Н.

Выбор языка программирования, подбор программных средств, жанра игры, целей, механики игры, а так же разработка иерархии классов проводился совместно.

Волкова А. А. реализовала часть иерархии классов отвечающих за статические объекты, характеристики персонажей и интерфейс пользователя.

Генералов Н. Н. реализовал классы для взаимодействия игрока, компьютера и всех внутриигровых объектов.

# ПОСТАНОВКА ЗАДАЧИ

В разрабатываемой системе моделируется противоборство главного героя, которым управляет игрок и множества юнитов, управляемых компьютером.

В рамках курсового проектирования ведется разработка игры в жанре Action Adventure.

В системе выделен ряд объектов, с которыми может взаимодействовать главный герой:

* объекты местности (деревья, строения, водные объекты);
* вражеские юниты;
* снаряды, выпущенные главным героем и другими юнитами;

Объектами местности имеют характеристики, содержащие информацию о разрушаемости объектов, изменении их свойств, к примеру, размера и количеству ударов, требуемых для их уничтожения.

Все юниты, в зависимости от их типа, имеют ряд характеристик: количество жизней, очков действий, скорость движения, скорость чтения заклинаний, урон, разброс урона, защита, а так же ряд модификаторов данных атрибутов.

В зависимости от типа, снаряды, выпускаемые юнитами, при столкновении с объектам либо исчезают, либо продолжают свое движение.

Игровое поле состоит из набора плиток размера 32х32, текстуры для которых выбираются на основе ID, хранящихся в виде целочисленных значений, загружаемых из бинарного файла.

Цель игры состоит в том, чтобы выполнить основные задания, уничтожив несколько основных врагов - боссов.

Передвижение игрового персонажа предполагается осуществлять с помощью набора клавиш W, A, S, D.

Прицеливание и атаку осуществлять при помощи манипулятора мыши.

Для использования умения персонажем необходимо сначала набрать определенную комбинацию клавиш (1, 2, 3) и нажать пробел, для генерации умения. Умения могут быть атакующие, усиливающие и исцеляющие персонажа.

Для запуска игры требуется предустановленная 64-х разрядная операционная система Windows 7 или более поздней версии, пакет Microsoft Visual C++ Redistributable 2010 или выше, а так же требуется наличие графических файлов расположенных в корневом каталоге игры.

По современным меркам, требования к техническим средствам предъявляются не высокие. Для работы игры потребуется:

* Процессор с частотой не менее 2.5 Ггц
* 200 мб оперативной памяти
* 300мб памяти на жестком диске
* видео адаптер
* Клавиатура
* Мышь

# ПРОЕКТНОЕ РЕШЕНИЕ

1. Абстрагирование и выделение объектов.

При абстрагировании были выделены игровые объекты:

уровень, миссия, карта, объекты местности, земля, главный герой (игрок), враг, снаряд.

А так же ряд объектов интерфейса, необходимых для взаимодействия с пользователем:

внутриигровое окно, кнопка, индикатор здоровья, индикатор способностей.

1. Информационная модель

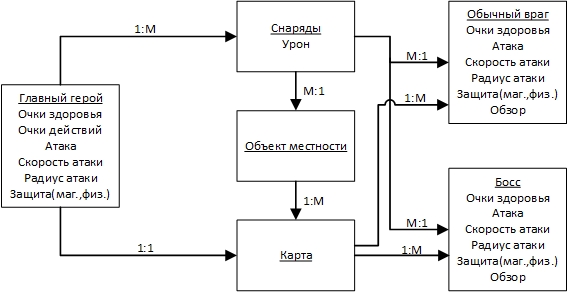


Рисунок 2.1 – Информационная модель

Главный герой – снаряды. Главный герой создает объект «снаряд». Он может создавать несколько снарядов разных типов.

Главный герой – карта. На одной карте может находиться только один главный герой.

Cнаряды – Объекты местности. Снаряды сталкиваются с объектами местности.

Объекты местности – Карта. Объекты местности препятствуют передвижению персонажей.

Снаряды – Обычный враг. Снаряды сталкиваются с врагом уменьшая его очки здоровья.

Снаряды – Босс. Снаряды сталкиваются с боссом уменьшая его очки здоровья.

Карта – Обычный враг, Карта – Босс. На карте располагается множество врагов и боссов.

1. Иерархия класса «Физические объекты»

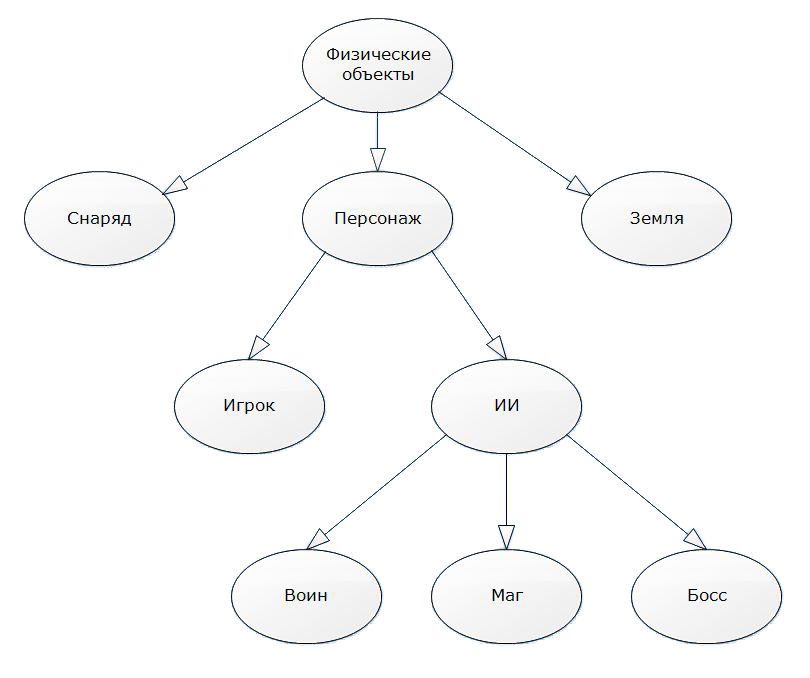


Рисунок 2.2 – Иерархия класса «Физические объекты»

Существует класс – физический объект. Объект «снаряд» имеет ряд характеристик: скорость перемещения, дальность полета, время жизни, радиус поражения. Объект «персонаж» имеет характеристики: очки здоровья, очки действий, сила атаки, радиус атаки, радиус обзора, скорость передвижения, скорость чтения заклинаний, физическая и магическая защита. Объекты подклассов «ИИ» имеют разные характеристики и поведение.

1. Жизненный цикл программы.



Рисунок 2.3 – Жизненный цикл программы

S1 – Запуск игры

S2 – Создание меню

S3 – Обработка событий манипулятора мыши

S4 – Запуск игрового процесса, на основе критериев выбранных пользователем

S5 –Генерация объектов игрового процесса: карты, объектов местности, главного героя и врагов

S6 – Генерация интерфейса пользователя

S7 – Считывание действий пользователя

S8 ­– Обработка действий пользователя

S9– Проверка состояния игры

S10 – Удаление объектов игрового процесса

S11 – Завершение игры

1. Жизненный цикл объектов:



Рисунок 2.5 – Жизненный цикл объекта «Враг»

1. Диаграмма переходов состояний:



Рисунок 2.6 – Диаграмма переходов состояний объекта класса «Враг»

1. Диаграмма потоков данных и действий:



Рисунок 2.7 – Диаграмма поток данных и действий основного персонажа

S1 – Создание главное героя

S2 – Считывание событий клавиатуры и манипуляторы мыши

S3 – Создание набора элементов для генерации умений

S4 – Генерация умения

S5 – Перемещение

S6 – Атака

S7 – Столкновение

S8 – Получение урона

S9 – уничтожение

# ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

1. Обоснование выбора языка и средств программирования.

Игровая система была написана на языке C++, с использованием мультимедийной библиотеки SFML версии 2.0. Кодирование производилось в интегрированной среде разработки Visual Studio 2017.

Язык C++ был выбран, потому что он оказал огромное влияние на другие языки программирования, и в настоящее время сам динамично развивается и вбирает в себя лучшее из других языков. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также развлекательных приложений (игр). Данный язык поддерживает такие парадигмы программирования, как: процедурное программирование, объектно-ориентированное программирование, обобщённое программирование.

Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков.

Язык С++ обладает высокой производительностью и большими возможностями низкоуровневого управления, предоставляя разработчику контроль над памятью и ее адресами, что отличает его от большинства современных языков.

Выбор в пользу среды Microsoft Visual Studio 2017, был сделан, так как данная среда содержит множество средств для упрощения написания кода, тестирования и отладки программных систем. Имеет встроенные инструменты осуществляющие авто коррекцию кода, предотвращающую синтаксические ошибки при написании программ, а так же инструменты для работы с системами контроля версий в частности Git.

Не маловажную роль при выборе среды разработки и мультимедийных средств, сыграло отсутствие коммерческой составляющей.

1. Общее описание взаимодействия программных модулей.

Данная программная система разбита на модули, каждый из которых отвечает за свой собственный класс, и в редких случаях за классы которые тесно связанны между собой.

Модули physOb\_t, ground\_t содержат информацию о физических объектах, их расположении, состоянии и свойствах

Модули character\_t, player\_t, npc\_t, BossNpc\_t содержат информацию о свойствах, характеристиках, поведении и состоянии как главного героя, так и врагов.

Модуль bullet\_t содержит информацию о динамических объектах, которые генерируются игровыми персонажами и взаимодействуют с другими объектами игрового мира.

Модули, которые имеют приставки – “Interface” в названии, отвечают за графическое представление, а так же обработку событий при взаимодействии с графическим интерфейсом.

Модуль “additional” содержит константы и функции которые используются