1. ТЕХНИЧЕСКОЕ ЗАДАНИЕ
   1. Общие сведения
      1. Наименование

“Вирус”

* + 1. Платформа

Персональные компьютеры под управлением ОС Windows 10.

* + 1. Жанр

2D шутер, рогалик.

* + 1. Целевая аудитория

Из-за сцен насилия игра не рекомендована лицам младше 16 лет. Аудитория, на которую нацелена игра: молодые и средних лет мужчины (от 16 до 45 лет), увлекающиеся играми жанра рогалик.

* + 1. Технические требования

Операционная система: Microsoft Windows 10.

Оперативная память: 1GB.

Процессор: Intel i5 или аналогичные.

Клавиатура.

* + 1. Краткое описание

2D шутер с видом сверху. Герой очнулся посреди подземного убежища, заполненного монстрами. Находясь в безвыходной, безоружной ситуации, только и остается отбиваться камнями. Герою предстоит отбиваться от волн врагов, или остаться в этом подземелье навечно.

* 1. Геймплей
     1. Объектная модель

Герой – игровая сущность, умеющая передвигаться по комнате и кидать камни.

Монстр – игровая сущность, которая атакует героя.

* + 1. Игровой процесс

Вирус – одиночная игра. Игрок управляет передвижением персонажа клавиатурой. Персонаж появляется посреди комнаты, его цель убить как можно больше монстров и набрать максимум игровых очков. Монстр может появится в одной из четырех возможных дверей.

* 1. Интерфейс
     1. Вид

Игра представляет 2D вид сверху на комнату, в которой расположен игрок, вверху показано кол-во набранных очков. При успешной атаке игрока из монстра плещется кровь, при смерти он испаряется.

* + 1. Управление

W, S, A, D – перемещение героя вверх/вниз/влево/вправо.

↑, ↓, ←, → (стрелки) – направление атаки и атака соответственно.

Space – выбор пункта меню или продолжить.

* + 1. Меню

Главное меню, здесь игроку предлагается начать новую игру или выйти.

* 1. Контент

В игре присутствует два вида монстров: червь и зомби. У каждого вида монстра свой вид атаки и поведения. Зомби медленно направляется к герою. Червь ведет себя пассивно и блуждает по комнате, пока герой не окажется на одной горизонтальной или вертикальной прямой относительно червя, после чего червь мгновенно и с увеличенной скоростью атакует в направлении героя по прямой, пока не достигнет стены комнаты. Как только червь достигает стены он снова пассивно блуждает.

1. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ
   1. Выбор темы

Игра “Вирус” является пародией на “The Binding of Isaac”. Это компьютерная инди-игра в жанре Action-RPG, разработанная Эдмундом МакМилленом и Флорианом Химслом. Выпущена 28 сентября 2011 года через цифровой дистрибутив Steam.

В связи с “хардкорностью” и интереса перепрохождения данной игры появилась идея реализации пародии.

Создание игрового приложения способствует приобретению навыков работы с функциями OpenGL и извлечению опыта программирования.

* 1. Выбор языка программирования

C++ — компилируемый, статически типизированный язык программирования общего назначения.

Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование, обеспечивает модульность, раздельную компиляцию, обработку исключений, абстракцию данных, объявление типов (классов) объектов, виртуальные функции.

C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также развлекательных приложений (игр).

* 1. Выбор среды разработки

Microsoft Visual Studio – это разработка компании Microsoft, позволяющая создавать приложения, работающие на платформе .NET. Особенность этой платформы заключается в широком наборе сервисов, которые доступны в различных языках программирования. При этом сервисы реализуются в виде промежуточного кода, который не зависит от базовой архитектуры.

Данная среда включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

* 1. Выбор графического API
     1. OpenGL

OpenGL (Open Graphics Library — открытая графическая библиотека, графический API) — спецификация, определяющая независимый от языка программирования платформонезависимый программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику.

Библиотека OpenGL разработана как обобщенный, независимый интерфейс, который может быть реализован для различного аппаратного обеспечения. По этой причине сама OpenGL не включает функций для создания окон или для захвата пользовательского ввода; для этих операций вы должны использовать средства той операционной системы, в которой вы работаете. По тем же причинам в OpenGL нет высокоуровневых функций для описания моделей трехмерных объектов. Такие команды позволили бы вам описывать относительно сложные фигуры, такие как автомобили, части человеческого тела или молекулы.

При использовании библиотеки OpenGL вы должны строить необходимые модели при помощи небольшого набора геометрических примитивов – точек, линий и многоугольников (полигонов). Тем не менее, библиотека, предоставляющая описанные возможности может быть построена поверх OpenGL.

В следующем списке коротко описаны основные графические операции, которые выполняет OpenGL для вывода изображения на экран.

1. Конструирует фигуры из геометрических примитивов, создавая математическое описание объектов (примитивами в OpenGL считаются точки, линии, полигоны, битовые карты и изображения).
2. Позиционирует объекты в трехмерном пространстве и выбирает точку наблюдения для осмотра полученной композиции.
3. Вычисляет цвета для всех объектов. Цвета могут быть определены приложением, получены из расчета условий освещенности, вычислены при помощи текстур, наложенных на объекты или из любой комбинации этих факторов.
4. Преобразует математическое описание объектов и ассоциированной с ними цветовой информации в пиксели на экране. Этот процесс называется растеризацией (или растровой разверткой).
   * 1. GLUT

OpenGL Utility Toolkit - библиотека утилит для приложений под OpenGL, которая в основном отвечает за системный уровень операций ввода-вывода при работе с операционной системой. Из функций можно привести следующие: создание окна, управление окном, мониторинг за вводом с клавиатуры и событий мыши. Она также включает функции для рисования ряда геометрических примитивов. GLUT даже включает возможность создания несложных всплывающих меню.

* + 1. GLEW

OpenGL Extension Wrangler Library - кроссплатформенная библиотека на C/C++, которая упрощает запрос и загрузку расширений OpenGL. GLEW обеспечивает эффективные run-time механизмы для определения того, какие OpenGL расширения поддерживаются на целевой платформе. Все расширения OpenGL размещаются в одном заголовочном файле, который автоматически генерируется из официального списка расширений. GLEW доступна на множестве операционных систем, включая Windows, Linux, Mac OS X, FreeBSD, IRIX и Solaris.

* 1. Выводы

Языком программирования был выбран язык C++. Данный выбор обусловлен возможностями языка осуществлять как процедурную модель программирования, так и объектно-ориентированную, а также тем, что этот язык предоставляет возможность программирования с использованием библиотек OpenGL.

В качестве среды разработки была выбрана Microsoft Visual Studio из-за простоты в использовании, удобного интерфейса и удобной отладки программы.

Таким образом, была поставлена задача в создании двумерной игры “Вирус” на языке программирования C++ с использованием графических библиотек GLEW, GLUT и среды разработки Microsoft Visual Studio.

1. РУКОВОДСТВО ПРОГРАММИСТА
   1. Структура программы
      1. Краткое описание классов

Структура программы представляет собой совокупность классов, которые взаимодействуют между собой.

В программе используются следующие классы:

* Game – содержит методы и правила взаимодействия объектов в кадре.
* Drawable – представляет объект который возможно отрисовать.
* UI – наследуется от Drawable, представляет пользовательский интерфейс в игре.
* Room – наследуется от Drawable, представляет само игровое поле.
* Body – наследуется от Drawable, представляет игровой объект, который может существовать на поле.
* Shot – наследуется от Body, представляет игровой объект камень и методы его поведения.
* Worm – наследуется от Body, представляет игровой объект червь и методы его поведения.
* Zombie – наследуется от Body, представляет игровой объект зомби и методы его поведения.
* Character – наследуется от Body, представляет игровой объект герой и методы его поведения.
* Drawer – представляет очередь из Drawable и содержит метод для отображения этих объектов в кадре.
* GameState – хранит состояния и события приложения.
* TexHelper – хранит текстуры и методы их отображения.
* Keyboard – определяет правило изменения игровых событий при использовании клавиатуры.
* Poof – содержит методы для отображения покадровой “poof” анимации.

Структуры:

* Coord – используется для хранения позиции на текстуре.
* Size – используется для хранения размеров.
* Orientation – используется для хранения ориентации в пространстве.
* Position – используется для хранения позиции объекта.
* HandTimer – используется в качестве покадрового таймера.
* TexImage – используется для хранения информации о текстуре.
* TexProperty – используется для хранения характеристик текстуры.
* DrawPropertie – используется для хранения характеристик отрисовываемого обьекта.
* Level – используется для хранения монстров на уровне.
* State – используется для хранения состояния игры.
  + 1. ATD формат основных классов
       1. ADT Game

Даные:

Операции:

Вход:

Значение:

Процесс:

Вход: нет.

Предусловия: нет.

Процесс: устанавливает, что в клетку был совершен выстрел.

Выход: нет.

Постусловия: нет.