МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Отделение информационных технологий и вычислительной техники

# Утверждаю

Руководитель

отделения ИТВТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_Рощенко О.Е\_\_\_\_\_\_

(подпись, фамилия, инициалы)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**ДИПЛОМНЫЙ ПРОЕКТ**

**НА ТЕМУ**

Автор дипломного проекта

(подпись студента, выполнившего дипломный проект)

Карих С.В. Группа И-72   
(фамилия, инициалы студента) (в которой обучался студент)

(факультет)

Специальность 09.02.03 Программирование в компьютерных системах

(код и наименование специальности)

Руководитель дипломного проекта Скорняков С.В.

(подпись, инициалы, фамилия)

Новосибирск, 2021 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Отделение информационных технологий и вычислительной техники

# Утверждаю

Руководитель

отделения ИТВТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_Рощенко О. Е.\_\_\_\_\_\_

(подпись, фамилия, инициалы)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**ЗАДАНИЕ НА ДИПЛОМНЫЙ ПРОЕКТ**

студент*у(ке)* Карих С.В. группы И-72

(фамилия, инициалы) (обучения)

1. Тема Разработка 3D игры в жанре action RPG элементами искусственного интеллекта

Утверждена приказом по НГТУ № \_\_4934/2\_\_ от «\_10\_» \_\_\_декабря\_\_\_\_\_\_ 2020 г.

2. Дата представления проекта к защите «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

3. Цели проекта (исходные данные)

4. Содержание пояснительной записки

4.1

4.2

4.3

4.4

4.5  
  
  
 4.6

5. Перечень графического и (или) иллюстрационного материала

**Руководитель проекта** Скорняков С.В.

(подпись, дата) (фамилия, инициалы)

**Задание принял к исполнению**

(подпись студента, дата) (фамилия, инициалы студента)

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 4](#_Toc72408386)

[ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc72408387)

[1.1 Актуальность темы 5](#_Toc72408388)

[1.2 Средства разработки 8](#_Toc72408389)

[Unity3D 8](#_Toc72408390)

[Fl studio 8](#_Toc72408391)

[Substance designer 8](#_Toc72408392)

[Blender 8](#_Toc72408393)

[Visual studio code 9](#_Toc72408394)

[2.1 Техническое задание 10](#_Toc72408395)

[Список типов объектов 10](#_Toc72408396)

[Характеристики объектов 10](#_Toc72408397)

[Связь/зависимость между объектами 11](#_Toc72408398)

[Список процессов (сценарии работы) 12](#_Toc72408399)

[Список функционала 13](#_Toc72408400)

[Ресурсы?? 16](#_Toc72408401)

[Список элементов на каждом экране 17](#_Toc72408402)

[Геометрия мира, расположение объектов на карте 19](#_Toc72408403)

[Тестирование 19](#_Toc72408404)

[2.2 Описание разработки 20](#_Toc72408405)

[2.3 Инструкция установки программного продукта 22](#_Toc72408406)

[ЗАКЛЮЧЕНИЕ 23](#_Toc72408407)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc72408408)

[Приложения 25](#_Toc72408409)

ВВЕДЕНИЕ

В сегодняшние дни, услышав слово развлечение, первое, что приходит на ум это компьютерные игры. Данная отрасль разработки компьютерных приложений стала очень востребованной. Вследствие такого спроса привело к притоку большого капитала в игровую индустрию. Крупные компании, разрабатывающие игры, получают со своих игр суммы, исчисляемые миллионами долларов.

Однако игровая индустрия интересна не только с точки зрения экономической прибыли, игры могут представлять собой полезные и в плане обучения программы. Например, игра может быть симулятором, помогающим разобраться с управлением сложной техники, такой как самолет, за которую было бы страшно пускать неопытного специалиста. Также игровая индустрия подталкивает специалистов улучшать физику в играх, чтобы она была более похожа на физику реального мира. Такого рода улучшения помогают не только игровой индустрии, но и позволяют эффективно проводить симуляции для решения ряда задач связанных с взаимодействием реальных объектов и физических процессов.

В России, к сожалению, данная сфера плохо развита из-за того, что дошла она до нас довольно поздно. На спрос это никак не повлияло, однако компаний способных конкурировать в данной сфере не существует.

Вследствие выше перечисленного, развитие данной сферы можно считать очень перспективным делом.

ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Актуальность темы

**Актуальность темы исследования обоснована тем, что** в настоящее время нарастает спрос на компьютерные игры, а это в свою очередь способствует развитию игровой индустрии. Данная сфера жизни привлекла к себе людей всех возрастов, так как охватывает множество различных жанров. Такая популярность игр обусловлена не только своим разнообразием, но и возможностями, которые предоставляют игры. В них игроку дается возможность почувствовать себя кем угодно, а также быть в центре событий, в которых они не смогли бы побывать вне игрового мира.

Разработка игр, несмотря на то, что игры это всего лишь развлечение, требует ответственного подхода. В отличие от прикладного ПО игры зачастую имеют в наличие множество сложных программ, реализующих физику или же трассировку лучей и т.д. Чтобы разработчику не приходилось каждый раз создавать уже реализованные механики, существует множество специализированных сред, предоставляющих инструменты для разработки игр, таких как Unity 3D, Unreal engine 4, Corona, Source 2 и многие другие. Ведутся множество споров относительно того, какой из игровых движков лучше. Однако из-за большого количества разработчиков на Unity 3D (вследствие чего существует множество обучающих материалов) было решено выбрать именно его.

В последние годы наметилась тенденция улучшения качества графики, однако из-за этого страдают механики игры, которые приходиться делать либо проще, либо сокращать их количество, а, следовательно, игра получается скучнее. Также разработчики стали злоупотреблять внутриигровыми транзакциями. Это привело к тому, что теперь чуть ли не за каждую текстуру или модель в игре приходиться платить. И более того не всегда это лишь косметические бонусы, зачастую купленные предметы дают преимущество в игре, разрушая тем самым её баланс. Итог этого таков, что игры превращаются в красивую пустышку, где просто нечего делать.

Примером такой игры является Genshin Impact (рис. 1).



Рис. 1. Genshin Impact

Genshin Impact - это компьютерная игра в жанре Action/RPG с открытым миром, разработанная китайской компанией miHoYo Limited. Игра распространяется посредством цифровой дистрибуции по модели free-to-play, но имеет внутри-игровой магазин, использующий реальную валюту. В их игровом магазине они продают “молитвы”, использовав которые, с небольшой вероятностью можно получить персонажа, что дается под контроль игроку. Каждый такой персонаж имеет свои характеристики. И, конечно же, персонажи, полученные из “молитв”, будут куда сильнее, чем те, что приобретаются по ходу игрового процесса бесплатно. И это еще не все. Во внутриигровом магазине можно приобрести за реальную валюту множество вещей, которые будут давать преимущество во время прохождения игры. Без этих вещей игровой процесс превращается в скучнейшую рутину.

Однако в проекте, реализованном мной, все внимание было уделено механике боя. Все силы были сосредоточены на создании интересной схватки с одним единственным противником, подстраивающимся под действия игрока. В данной механике заключается кардинальное отличие от других представителей данного жанра игр, в которых противники зачастую управляются простыми скриптами, целью которых является только атака игрока. Целью же моего скрипта есть максимально эффективный размен здоровья своего на здоровье и выносливость игрока. Некоторые виды атак были созданы лишь для того, чтобы спровоцировать игрока на ошибку, дать ему ложное ощущение превосходства над противником, которое в итоге приведет его к поражению.

Также важной особенностью моего проекта есть возможность игрока получить преимущество над противником, только набравшись опыта в бою с ним и никак иначе. Реальной валютой откупиться не получится, так как продажа внутриигровых бонусов не предусмотрена.

**Цель проекта:** создать 3D action RPG игру с элементами искусственного интеллекта.

Для создания проекта необходимо было решить **следующие задачи:**

* Изучить моделирование и реализовать модели для игровых объектов;
* Изучить процедурное текстурирование и создать необходимые текстуры для всех, требующих того объектах;
* Изучить процесс создания музыкальных композиций и реализовать необходимые для игры;
* Изучить фреймворк, предоставляемый unity, и с его помощью реализовать механики игры.

1.2 Средства разработки

Unity3D

**Unity** — межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие. Также в бесплатном доступе распространяются плагины для unity упрощающие жизнь разработчикам игр.

Данный инструмент был выбран не только из-за его технических характеристик, а также причиной тому послужило наличие большого количества справочной информации и большое сообщество, которое уже решило множество проблем, возникающих при разработке тех или иных механик.

Fl studio

**FL Studio** — цифровая звуковая рабочая станция (DAW) и секвенсор для написания музыки. Музыка создаётся путём записи и сведения аудио- или MIDI-материала.

Substance designer

**Substance designer –** программа создания процедурно генерируемых материалов. Материал составляется из последовательно соединенных узлов, каждый из которых является алгоритмом, обрабатывающим входящее изображение и передающим его в следующий узел.

Blender

**Blender** — профессиональное cвободное и открытое программное обеспечение для создания трёхмерной компьютерной графики, включающее в себя средства моделирования, скульптинга, анимации, симуляции, рендеринга, постобработки и монтажа видео со звуком, компоновки с помощью «узлов», а также создания 2D-анимаций. В настоящее время пользуется большой популярностью среди бесплатных 3D-редакторов в связи с его быстрым стабильным развитием.

Visual studio code

**Visual Studio Code** — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации.

ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Техническое задание

Список типов объектов

* Менеджер компонентов
* Физика движений
* Обработчик анимаций
* Контроллер орудий игрока
* Компонент характеристик
* Фаза противника
* Объект состояния
* Атака
* Машина состояний
* Вид характеристик
* Обработчик столкновений с атакующими объектами

Характеристики объектов

* Менеджер компонентов хранит в себе публичные ссылки на все компоненты объекта, к которому он принадлежит.
* Физика движений – компонент, хранящий процедуры, рассчитывающие и меняющие положение игрового объекта, к которому данный компонент прикреплен.
* Обработчик анимаций – компонент, отвечающий за запуск анимаций по их названию, а также за смену состояний флага взаимодействия.
* Контроллер орудий игрока – компонент, который хранит координаты для размещения орудий игрока, также содержит методы, отвечающие за включение и выключение коллайдера урона.
* Компонент характеристик – компонент, хранящий важные характеристики игрока, такие как: здоровье, выносливость.
* Фаза противника – интерфейс для создания объектов фазы противника.
* Объект состояния – интерфейс для создания объектов состояния противника. Имеет метод тика, в который передается менеджер компонентов, исполняющий логику данного состояния.
* Атака – компонент, хранящий в себе основную информацию об атаке. Данная информация содержит анимацию атаки, эффект необходимый для данной атаки, также имеет в себе количество наносимого урона.
* Машина состояний – объект, обрабатывающий тики состояний, а также отвечает за переходы между ними.
* Вид характеристик – объект, занимающийся графическим отображением компонента характеристик при помощи методов и объектов фреймворка unity.
* Обработчик столкновений с атакующими объектами – объект, который обрабатывает столкновения строго на установленном слое. После столкновения данный объект получает компонент атаки и исполняет процедуру получения урона.

Связь/зависимость между объектами

*Машина состояний/Объект состояния* – связь данных заключается в том, что машина состояний исполняет логику состояния, последовательно вызывая метод тика состояния, пока не будет возвращено другое состояние.

*Обработчик столкновений с атакующими объектами/Атака* – связь заключается в получении объекта атаки обработчиком столкновений для дальнейшего использования характеристик объекта атаки для нанесения урона родительскому объекту.

Список процессов (сценарии работы)

1. Появление в игровом мире

По появлении в игровом мире, игроку сначала кратко объясняется сюжет в текстовом виде. Затем игроку передается управление над персонажем и камерой.

1. Загрузочный экран

Пока сцена с игровым миром загружается, игроку будет отображаться изображение с инструкциями по управлению.

1. Первая фаза боя

Как только игрок попадает на арену, где будет происходить битва, он оказывается в зоне видимости противника и начинается битва.

1. Вторая фаза боя

Наступает после того как количество жизней противника будет меньше половины. После смены фазы модель противника меняется на модель мечника.

1. Финал

Игра заканчивается анимацией добивания противника. После чего игрока отправляют в главное меню.

1. Смерть

После того как шкала жизни игрока достигает 0, все игровые объекты ставятся на паузу и на экран выводиться надпись, сообщающая игроку о его смерти, а также предоставляется две возможности отправиться в главное меню или начать игру снова.

Список функционала

1. Игрок
   1. Перемещение

Перемещение по игровому миру в осях x и z. Вертикальное перемещение может быть обусловлено только в случае падения с возвышенности или же вертикальным смещением ландшафта игрового мира.

Таблица 1

Виды передвижений

|  |  |  |
| --- | --- | --- |
| Название | Описание | Управление |
| Ходьба | Проигрывается анимация ходьбы. | WASD |
| Бег | Проигрывается анимация Бега. | WASD + SPACE (удерживать) |
| Кувырок | При исполнении игрок неуязвим. | WASD + SPACE |

* 1. Атака

Таблица 2

Виды атак

|  |  |  |
| --- | --- | --- |
| Название | Описание | Управление |
| Легкая атака | Проигрывается анимация легкой атаки. | ЛКМ |
| Тяжелая атака | Проигрывается анимация тяжелой атаки. | ПКМ |
| Легкая атака (комбинированная) | Проигрывается анимация легкой атаки (комбинированная версия). | ЛКМ + ЛКМ |

* 1. Вращение камеры

При помощи мыши происходит вращение камеры вокруг стержня установленного в центре игрока. Когда камера сталкивается с другими объектами, которые мешают видеть игрока, она фиксирует свою позицию относительно вектора направленного от стержня в точку столкновения с мешающим объектом.

1. Противник

Таблица 3

Атаки на первой фазе

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Характеристики | Условия | Описание |
| Болото |  |  | При попадании в него, игрок замедляется. |
| Огненный шар |  |  | Снаряд, летящий в игрока. На данный шар действует гравитация, поэтому он может упасть и не долететь до игрока. |
| Кольцо огня |  |  | Окружает противника, и не дает подойти к нему. |

Таблица 4

Атаки на второй фазе

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Характеристики | Условия | Описание |
| Вертикальный срез |  |  | Луч, движущийся на игрока, а также оставляющий за собой кратеры. |
| Круговой срез |  |  | Атака во вращении. Оставляет след вокруг противника, который наносит урон. |
|  |  |  |  |

1. Эффекты

Таблица 5

Список эффектов

|  |  |
| --- | --- |
| Название | Описание |
| Болото |  |
| Кратер |  |
| Вертикальный разрез |  |
| Луч |  |
| Хвост меча |  |

1. Анимации

Таблица 6

Список анимаций

|  |  |  |
| --- | --- | --- |
| Родительский объект | Название | Описание |
| Игрок | Ходьба |  |
| Бег |  |
| Кувырок |  |
| Легкая атака 1 |  |
| Легкая атака 2 |  |
| Тяжелая атака |  |
| Противник | Ходьба |  |
| Заклинание-выстрел |  |
| Заклинание по площади |  |
| Вертикальная атака |  |
| Горизонтальная атака |  |
|  |  |

Ресурсы??

Таблица 7

Список моделей

|  |  |
| --- | --- |
| Название | Описание |
| Игрок |  |
| Противник (Маг) |  |
| Противник (Мечник) |  |

Таблица 8

Музыка

|  |  |
| --- | --- |
| Название | Описание |
| Меню |  |
| Первая фаза |  |
| Вторая фаза |  |

Список элементов на каждом экране

* Главное меню
  + Название-логотип (Logo)
  + Кнопка начать (Start)

Запускает скрипт загрузки игрового мира и переносит на загрузочный

* + Кнопка выход (Exit)

Завершает процесс игры.

**

Рис. 2 Макет главного меню игры

* Меню смерти
  + Кнопка начать сначала

Перезапускает игровую сцену.

* + Кнопка выхода в главное меню

Загружает сцену главного меню.

* HUD
  + Шкала здоровья

Отражает значение переменной здоровья игрока.

* + Шкала выносливости

Отражает значение переменной выносливости игрока.

* + Шкала здоровья противника

Отражает значение переменной здоровья противника.

**

Рис. 3 Макет экрана смерти

Геометрия мира, расположение объектов на карте

*Схематичный макет карты с обозначениями*

Тестирование

Технические проблемы были выявлены в процессе разработки статическим анализатором Unity 3d, после чего устранены.

Логические проблемы были выявлены и устранены при помощи тестирования использованием, т.е. все механики игры были проверены вручную.

2.2 Описание разработки

Создание проекта в unity

Геймплей

Персонаж

Камера

Передвижение

Характеристики

Оружие

Противник

Характеристики

Машина состояний

Нейронная сеть

След оружия

Пользовательский интерфейс

Меню

Игровая сцена

Экран смерти

Шейдеры

Вода

эффекты

Модели

Персонаж

Противник

Дерево

Балка

Меч

Топор

Текстуры

Трава

Песок

Заросшая дорога

Скала

стена

Музыка

2.3 Инструкция установки программного продукта

Разархировать в текущую директорию.

ЗАКЛЮЧЕНИЕ

пав

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Adobe Substance Designer. Процедурное текстурирование. <https://habr.com/ru/company/softline/blog/510252/>
2. Создание персонажей в Blender и Unity. <https://habr.com/ru/post/422255/>
3. Основы работы с Shader Graph в Unity. <https://habr.com/ru/post/479302/>
4. Unity: системы частиц. <https://habr.com/ru/post/351458/>
5. Getting Started with Blender’s Particle System. <https://hub.packtpub.com/getting-started-blenders-particle-system/>
6. Unity user manual. <https://docs.unity3d.com/ru/2019.4/Manual/index.html>
7. Простой конечный автомат на Unity. <https://habr.com/ru/post/206120/>

Приложения