**Слайд 1**

Добрый день уважаемая комиссия, сегодня я представлю вам свой дипломный проект.

**Слайд 2**

Основные задачи, поставленные в работе, состояли в изучении разнообразных методов создания моделей и анимаций, преимущественно в программе Blender, для дальнейшего их использования в проектах на движке Unreal Engine.

В рамках практической части разрабатывался прототип боевой системы на Unreal Engine, включающий в себя все базовые действия перемещения, атак и защиты, дополнительные подсистемы и различные визуальные элементы.

**Слайд 3**

GameDev (разработка видеоигр) привлекает в последнее время все больше людей. Особенно это связано с тем, что государство и частные компании в России стали выступать с инициативами по поддержанию отечественных разработчиков видеоигр. Это и известные ИРИ (институт развития интернета), и компания VK, готовые вкладываться в конкурентноспособные продукты.

Также появляются и независимые от крупных компаний разработчики (так называемые инди-студии), создающие уникальные продукты и привлекающие к себе начинающих разработчиков и энтузиастов для работы над их проектами.

Яркие примеры созданных проектов последних лет представлены на экране.

**Слайд 4**

В рамках основной практической части создавался проект на движке Unreal Engine. Движок позволяет писать код как на C++, так и на собственном блочном языке Blueprint.

В процессе работы были написаны несколько классов на C++: два наследника класса character Unreal Engine’а для персонажей и управляющий класс. Анимации и другие, более мелкие элементы были написаны на Bluepint’ах.

Все основные этапы представлены на слайде.

В процессе работы были решены различные проблемы, связанные с взаимодействием персонажей, правильным отображением анимаций и связки C++ кода и Blutprint’ов. В том числе были решены проблемы с импортом моделей из Blender в UE: инвертация нормалей поверхностей и неприменение материалов.

**Слайд 5**

Среди двух разработанных персонажей, одним полностью управляется человеком через клавиатуру и мышь. Он может свободно перемещается по карте, вращать камерой во всех направлениях и выполнять другие действия.

Другой персонаж, противник, находится под управлением компьютера. Все его действия заранее прописаны, и зависят от действий игрока или же, в некоторых случаях, выбираются случайно.

На рисунке представлены диаграммы состояний для обоих персонажей.

**Слайд 5**

Взаимодействие в бою происходит через считывание позиции противника в момент атаки. Но противник получает урон не в каждый момент отображения анимации, а лишь в определенном диапазоне времени. При этом учитывается, куда повернут в этот момент герой (не промахнулся ли он) и на каком он расстоянии.

Если все условия соблюдены, то особая булева переменная становится истиной, что провоцирует в классе противника запуск функции получения урона с отображением соответствующей анимации.

**Слайд 7**

Как и во многих подобных проектах, в данном была добавлена система захвата внимания, это особый режим, в который игрок может войти через нажатие определенной клавиши (СКМ). В данном режиме “взгляда” героя (синяя стрелка) всегда направлена по направлению вектору между персонажами – красная стрелка.

Для сохранения этого положения выполняется изменение контроллера персонажа, управляющего поворотом камеры, а также выполняется поворот модели самого персонажа.

**Слайд 8**

Никакой проект в подобном жанре не обходится без системы здоровья. Каждому персонажу был добавлен счетчик здоровья в виде виджета: у противника висящий прямо над его головой; у героя, закреплённый в левом верхнем углу экрана.

Также при смерти противника происходит удаление его старого объекта с тегом is\_dead и создание нового в случайной позиции.

**Слайд 9**

Для проекта были использованы десятки различных анимаций, для качественного отображения всех действий. Переход между анимациями осуществляется в Animation Blueprint, однако все переменные в условиях передаются напрямую из C++ кода.

Обычные анимации применяются при перемещении персонажа, когда его поворот по оси Z совпадает с направлением его движения, которое указывает человеком.

Но в особом режиме захвата внимания, когда персонаж всегда повернут в сторону его противника, анимации будут зависеть от того, куда он будет двигаться. Такие анимации сделаны с помощью Blend Space Unreal Engine’а, это особый ассет, содержащий в себе набор обычных анимаций и смешивающий их между собой, в зависимости от параметров, подаваемых на вход.

**Слайд 10**

Для завершения проекта была проделана большая работа по созданию его визуальной составляющей.

Были добавлены визуализаторы здоровья для обоих персонажей, а также эффекты получения урона.

Для обоих видов персонажей были добавлены специальные модели воинов и была создана арена из различных ассетов.

**Слайд 11**

Результаты работы представлены на экране.

В итоге, созданный мною прототип игровой системы уже сейчас хорошо выглядит и в него можно поиграть. Но при дальнейшем развитии проекта, его легко можно развить до полноценной игры на смартфоне или PC, добавив другие механики и разнообразив визуальную часть новыми локациями и героями.

В конце хочу сказать, что за время работы был получен огромные опыт в разработке на UE, который я бы хотел и далее применять.