

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»

**ХУДОЖЕСТВЕННО-ТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ
КОМПЬЮТЕРНОЙ 3D ИГРЫ В ЖАНРЕ SURVIVAL HORROR**

Выпускная квалификационная работа
по направлению подготовки 09.03.02 Информационные системы и
технологии
профилю подготовки «Информационные технологии в медиаиндустрии»

Идентификационный номер ВКР: 148

Екатеринбург 2018

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»
Институт инженерно-педагогического образования
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ

Заведующая кафедрой ИС

_____ Н. С. Толстова

«__» _____ 2018г.

ХУДОЖЕСТВЕННО-ТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ КОМПЬЮТЕРНОЙ 3D ИГРЫ В ЖАНРЕ SURVIVAL HORROR

Исполнитель:

студентка группы ИТм-402

Т. А. Щепина

Руководитель:

канд. пед. наук, доцент

Т. В. Чернякова

Нормоконтролер:

Н. В. Хохлова

АННОТАЦИЯ

Выпускная квалификационная работа состоит из 3D игры «Brotherly ties» и пояснительной записки на 72 страницах, которая содержит 44 рисунка, 5 таблиц и 30 источников литературы, а также 3 приложения на 4 страницах.

Ключевые слова: UNREAL ENGINE 4, ИГРОВОЙ ДВИЖОК, КОМПЬЮТЕРНАЯ ИГРА, ПРОЕКТИРОВАНИЕ КОМПЬЮТЕРНОЙ ИГРЫ.

Щепина Т. А. Художественно-техническое проектирование компьютерной 3D игры в жанре survival horror: выпускная квалификационная работа / Т. А. Щепина ; Рос. гос. проф.-пед. ун-т, Ин-т инж.-пед. образования, Каф. информ. систем и технологий. — Екатеринбург, 2018. — 75 с.

В работе описана разработка концепта, сценария 3D игры и её реализация на движке Unreal Engine 4.

Цель работы — художественно-технически проектирование компьютерной игры и её реализация с помощью игрового движка Unreal Engine 4. Для реализации этой цели были проанализированы существующие разработки, изучено необходимое программное обеспечение и технологии создания игр. Создана компьютерная 3D игра, состоящая из трёх уровней. Игра запакована в приложение исполняемого формата.

СОДЕРЖАНИЕ

| | |
|---|----|
| Введение..... | 5 |
| 1 Анализ литературы и интернет-источников по теме «разработка видеоигр»7 | |
| 1.1 Анализ и общая характеристика из области разработки 3D компьютерных игр | 7 |
| 1.2 Анализ популярных игр в жанре survival horror | 16 |
| 1.3 Анализ программного обеспечения | 25 |
| 1.4 Общий алгоритм реализации игры Brotherly ties..... | 33 |
| 2 Разработка игры Brotherly ties..... | 36 |
| 2.1 Потенциальная аудитория 3D игры в жанре survival horror | 36 |
| 2.2 Задачи разработки 3D игры..... | 36 |
| 2.2.1 Актуальность разработки 3D игры в жанре survival horror | 36 |
| 2.2.2 Цель и назначение игры Brotherly ties | 37 |
| 2.2.3 Требования к 3D игре | 38 |
| 2.2.4 Входные данные к игре Brotherly ties | 40 |
| 2.2.5 Характеристика оборудования для реализации игры Brotherly ties | 40 |
| 2.3 Этапы разработки игры Brotherly ties | 41 |
| 2.3.1 Разработка концепта игры..... | 41 |
| 2.3.2 Написание сценария..... | 42 |
| 2.3.3 Разработка моделей главного и второстепенного персонажей | 43 |
| 2.3.4 Разработка модели врага | 47 |
| 2.3.5 Моделирование локаций | 49 |
| 2.3.6 Подбор текстур и материалов..... | 51 |
| 2.3.7 Построение уровней | 52 |
| 2.3.8 Подбор головоломок..... | 57 |
| 2.3.9 Программирование основных элементов игры..... | 58 |
| 2.3.10 Создание внутриигровых роликов | 60 |
| 2.3.11 Тестирование | 61 |

| | |
|---|----|
| 2.3.12 Запаковка игры | 62 |
| 2.3.13 Выпуск игры | 63 |
| 2.4 Системные требования игры Brotherly ties..... | 63 |
| 2.5 Калькуляция проекта | 64 |
| Заключение | 65 |
| Список использованных источников | 66 |
| Приложение А | 69 |
| Приложение Б | 71 |
| Приложение В..... | 72 |

ВВЕДЕНИЕ

В последние 30 лет компьютерные игры приобрели невероятную популярность во всём мире. Игры во все времена существования человечества являлись неотъемлемой составляющей развития. Кроме того, игры вносят разнообразие в обыденную повседневность.

Рынок видеоигр является самым большим сегментом мирового рынка цифрового контента. Прогнозы различных аналитических компаний пророчат дальнейшее стремительное развитие рынка в ближайшие два года. Согласно прогнозам J'son and Partners Consulting, объем мирового рынка игр к 2021 году составит почти 130 миллиардов долларов Соединённых Штатов Америки (США), демонстрируя средние ежегодные темпы роста в период 2016–2021 годов на уровне 5,4 %. Российский рынок видеоигр не отстаёт по темпам развития от мирового, а в некоторые годы даже обгоняет по объемам выручки рынок киноиндустрии [18].

Несмотря на смещение сегментов рынка видеоигр в сторону мобильных разработок, компьютерные игры не теряют свою популярность, так как аудитория компьютерных игр является ядром всей игровой аудитории.

Если говорить о жанровых предпочтениях пользователей в одиночных играх, то жанр action является одним из самых популярных. Среди его поджанров в свою очередь самыми популярными являются horror и shooter. Многие люди жаждут испытать опасность и почувствовать прилив адреналина, а именно эти поджанры способны вызвать эмоции тревоги, страха, напряжённости. Несмотря на то, что данные эмоции считаются негативными, медики считают, что сочетание этих эмоций и осознание того, что в реальности человеку ничего не угрожает, ведь всё действие происходит на экране, создаёт чувство удовлетворения и счастья.

Объект исследования — компьютерные игры.

Предмет исследования — методы разработки компьютерной игры.

Цель — художественно-технически проектирование компьютерной игры и её реализация с помощью игрового движка Unreal Engine 4.

Задачи:

- изучить литературу и интернет-источники, связанную с разработкой компьютерных игры, написанию сценариев к играм;
- проанализировать существующие разработки;
- создать концепт игры;
- написать сценарий игры;
- изучить методы работы с движком Unreal Engine 4;
- реализовать продукт в движке Unreal Engine 4.

1 АНАЛИЗ ЛИТЕРАТУРЫ И ИНТЕРНЕТ-ИСТОЧНИКОВ ПО ТЕМЕ «РАЗРАБОТКА ВИДЕОИГР»

1.1 Анализ и общая характеристика из области разработки 3D компьютерных игр

На сегодняшний день компьютерные игры прочно вошли в ежедневный досуг людей. Игры помогают человеку пережить опыт, который он не сможет получить в реальной жизни: сразиться с драконом, путешествовать по мирам и эпохам, ощутить всплеск адреналина, избегая опасные для здоровья и жизни увлечения. Рынок видеоигр — один из самых динамично развивающихся рынков информационных технологий (ИТ-рынков) в мире. По данным аналитического агентства Newzoo (Newzoo's 2017 Global Games Market Report), за последние пять лет доходы от продаж игр в мире увеличились на 56 % по итогам 2017 года и далее доходы будут лишь расти.

Для начала рассмотрим основные термины и понятия из области 3D компьютерных игр.

Игра — это интерактивная структура эндогенного (внутреннего) значения, которая требует от игрока борьбы (преодоления препятствий) для достижения целей [20].

Компьютерная игра (видеоигра) — игра, воспроизводимая на компьютере (игровой консоли), в которой изображения, отображаемые на экране, реагируют на нажатие клавиш на клавиатуре компьютера или взаимодействие с джойстиками на геймпаде [3].

3D игра — компьютерная игра, визуальное пространство которой состоит из трёхмерных объектов, которые вычисляются в трёх направлениях [3].

Жанр — род произведений в пределах какого-либо искусства, отличающийся особыми, только ему свойственными сюжетными и стилистическими признаками [8].

Сеттинг — время и место, в рамках которых происходит действие книги, фильма, игры и т. д.

Разработка компьютерных игр — это процесс создания компьютерных игр.

Разработкой может заниматься как один человек, так и команда разработчиков. Численность штата команды зависит от сложности проекта и его бюджета, чем крупнее проект и его бюджет, тем крупнее команда, работающая над ним.

Команда по разработке компьютерных игр обычно состоит из геймдизайнера, художника, программиста, дизайнера уровней, звукорежиссёра и тестировщика [4].

Геймдизайнер — специалист, который разрабатывает содержание и правила игрового процесса, он лучше других понимает, как будет выглядеть конечный продукт. В крупных проектах может быть несколько геймдизайнеров: один главный и несколько подчинённых, каждый из которых отвечает за отдельную часть игры (игровую механику, персонажей и т. д.).

Художник — специалист, оформляющий визуальный стиль игры. Данный специалист должен обладать навыками работы с 2D и 3D графикой или можно нанимать разных художников для разных целей. В работу с 2D графикой входят создание концепт-артов, разработка пользовательского интерфейса, текстуры, если игра выполнена полностью в 2D графике, то художник так же создает задние фоны, спрайты. В работу 3D художника входит создание моделей персонажей, окружения, анимация моделей (в крупных проектах для этого может быть выделен отдельный специалист).

Программист — специалист, который реализует игровые механики, задуманные геймдизайнером (геймплей), настраивает обработку сигналов, поступающих с различных устройств ввода (клавиатура, мышь, геймпад), занимается разработкой игрового движка — ядра игры, разрабатывает сценарии поведения искусственного интеллекта, и прочее. То есть программист объединяет во едино все составляющие элементы игры и добавляет этим элементам интерактивность.

Дизайнер уровней — специалист, занимающийся разработкой игровых уровней, не только их внешним видом, но и постановкой задач, миссий и препятствий, которые надо преодолеть для продвижения по игре. Присутствует в крупных проектах и является составной частью команды геймдизайнеров. В маленьких проектах его роль выполняет основной геймдизайнер и программист.

Звукорежиссёр — технический специалист, отвечающий за музыкальное и звуковое сопровождение в игре, обычно является наёмным рабочим на данный конкретный проект, а не постоянным членом команды.

Тестировщик — специалист, который проверяет игру на наличие ошибок, игра должна работать так, как это было задумано, все скрипты срабатывать чётко и в нужный момент. Необходимо проверить работоспособность всех функций, совместимость игры с разными компьютерами, операционными системами или платформами, если игра задумывалась, как кроссплатформенная.

Разработка игры обычно включает в себя четыре основных этапа: подготовка к производству, производство, выпуск продукта и его поддержка. Рассмотрим подробнее каждый этап [13, 15].

На первом этапе подготавливается предпроектная документация, разрабатывается концепция игры и дизайн персонажей, создаётся прототип, выбираются средства для реализации проекта, составляется план работы над проектом.

На этапе производства разработчики реализуют составленный ранее план, производят контент для наполнения игры, программируют механики, реализуют «Вертикальный срез» — минимально возможная полноценная версия игры, т. е. в данной версии игры может не хватать каких-либо мелких деталей, быть не до конца проработана графика, но должен быть полностью реализован игровой процесс, это может быть одна полностью проработанная локация, если она включает в себя воплощение всех фишек игры.

Этап выпуска представляет собой полностью завершённую игру, которая поступает в продажу, на этом этапе важно грамотно преподнести игру, подогревая к ней интерес, чтобы получить максимальную прибыль.

Во время поддержки собирается информация от пользователей игры и при возникновении ошибок выпускаются патчи (от английского patch — заплатка) для их исправления, так как не все ошибки можно вычислить в момент тестирования игры, для онлайн игр на данном этапе выпускаются обновления.

Сейчас видеоигры являются частью популярной культуры. Перед разработчиками встаёт вопрос, как адаптировать приёмы классического искусства в уникальном интерактивном продукте? Видеоигры полагаются на те принципы дизайна (перспективу, форму, ценность и т. д.), которые классические художники использовали для создания иллюзии, что холст является окном в воображаемый мир. В этом их эстетическая ценность для визуального повествования.

Крис Соларски — автор книги *Drawing Basics and Video Game Art: Classic to Cutting Edge Art Techniques for Winning Video Game Design* [19] — называет картины художников классической композицией, а видеоигры — динамической композицией. Он пишет, что динамическая композиция раскрывается, если взять основные линии и формы из классической живописи и на их основе создать карту местности. Линии, которые неявно прослеживаются при взгляде на классическую картину, теперь

становятся путями, по которым можно путешествовать по трехмерной среде.

Пути внутри среды — это только одна часть динамической композиции. Чтобы полностью понять динамическую композицию, нужно учитывать пять элементов игры и их отношения друг к другу.

Форма персонажа. Внешний вид персонажа может много рассказать не только о его характере и физических способностях, но и об уровне развития персонажа. В играх, как в кино и литературе, персонаж не должен начинать и заканчивать сюжет в одном и том же состоянии. Например, в RPG персонаж начинает игру в простой тряпичной одежде и с одним видом оружия или вообще без него, но по ходу развития персонажа его обмундирование меняется на более сложное — доспехи и меч — это наглядно показывает игроку, что герой не стоит на месте, а развивается (рисунок 1). Это же относится и к эмоциям, в течение сюжета игрок должен чувствовать изменения в персонаже как в личности, понимать перемены в его психике. Это можно передать через монологи героя, его взаимодействие с другими персонажами, его реакции на окружающую среду и происходящие события.



Рисунок 1 — Герой фракции Инферно на разных ветках развития уровня, игра Might and Magic Heroes VI

Анимации персонажа. Большое значение в последнее время имеет лицевая анимация, она позволяет сильнее прочувствовать эмоции персонажей и чем реалистичнее выглядит персонаж и его мимика, тем сильнее эмоциональная связь между игроком и героем, так как он начинает восприниматься, как живой человек. Анимация помогает игроку отслеживать состояние персонажа, например, в игре Resident Evil главный герой становится физически ослабленным, когда его отравляют или травмируют. Линии движения отображают различные эмоции: тонкие и динамические (кривые линии), медленные и мирные (прямые стойки и горизонтали) и агрессивный (угловой). При разработке движений персонажа важно выбирать линии, которые дополняют эмоции, которые должен испытать игрок.

Окружающая среда. Окружение персонажа является ключевой частью динамической композиции, потому что окружающая среда обычно занимает большую часть визуального пространства в кадре. Под окружающей средой так же подразумеваются враги и второстепенные персонажи. Через формы в окружающей среде можно понять находится ли персонаж в безопасности или ему что-то угрожает. Когда окружающая среда и персонаж гармонируют друг с другом, то игрок ощущает состояние покоя, чувствует домашний уют (рисунок 2), если персонаж и окружающая среда находятся в состоянии диссонанса, то сразу чувствуется угроза, нависшая над персонажем.



Рисунок 2 — Игра Journey, персонаж и окружающая среда сочетаются по форме и цветовой гамме

Пути внутри игры. Передвижение персонажа по локации придерживается определённых форм для поддержания настроения игры: округлые формы — спокойствие, острые формы — динамика и агрессия. Конечно, с распространением игр с открытым миром у игроков появилось больше просто в перемещении и соблюдение данных форм в путях становится менее очевидным, тем не менее дизайнеры стараются сохранять единство форм окружающей среды и путей, по которым предстоит двигаться игроку.

Жесты игрока. Этот аспект особенно важен для игр, которые используют контроллеры движений в своём игровом процессе. Взаимосвязь анимации персонажа на экране и движений игрока — это взаимодействие, уникальное для видеоигр (рисунок 3).



Рисунок 3 — Момент в игре Until Dawn, когда нельзя шевелить геймпадом для спасения персонажа

Изучая компьютерные игры, нельзя акцентировать внимание только на их эстетической составляющей, стоит также рассмотреть такой термин, как техническое проектирование компьютерных игр. Техническое проектирование подразумевает разработку архитектуры, написание технического задания и формальное планирование производственного цикла [23].

Архитектура компьютерной игры определяет из каких элементов состоит игра и как эти элементы взаимодействуют. Компьютерные игры

имеют сложную архитектуру и к каждой нужен индивидуальный подход. Наиболее распространённые это объектно-ориентированный (ООП) и компонентно-ориентированный (КОП) подходы. В ООП объекты представляют собой классы, от которого могут наследоваться другие классы или который сам наследует кого-то. В КОП объекты — это наборы компонентов, где каждый компонент отвечает за свою функцию, например, объект «персонаж» состоит из компонента управления персонажем, обработчика столкновений или взаимодействия с другими объектами, компонента анимации и так далее.

Для взаимодействия различных объектов в игре используются машины состояний (в русскоязычной литературе также используется термин конечный автомат) и деревья поведений. Логика объектов в машине состояний разбивается на текущее состояние, действие, которое выполняется в этом состоянии, переход между 2 состояниями и событие, которое вызывает смену состояний, то есть активирует переход от одного состояния в другое. Дерево поведений — это иерархическая древовидная структура, где в качестве узлов выступают блоки игровой логики (рисунок 4). Дерево поведений обычно состоит из следующих узлов: действие — функция, который должна выполняться при посещении данного узла; условие для выполнения, следующего за ним узла; последовательность — выполняет вложенные в неё узлы по порядку, пока какой-либо из них не завершится неудачей или пока все не завершатся успехом; селектор действует как последовательность, но завершается только при успешном прохождении всех вложений; итератор используется для прохождения цикла заданное количество раз (рисунок 5) [11].

Какой именно подход для реализации взаимодействия объектов выбрать зависит от конкретной игры, над которой ведётся работа. Можно также объединять машину состояний и дерево поведений, так как реализация дерева поведений считается более сложной для объектов, которые

получают информацию из вне (управляются игроком или искусственным интеллектом).



Рисунок 4 — Машина состояний

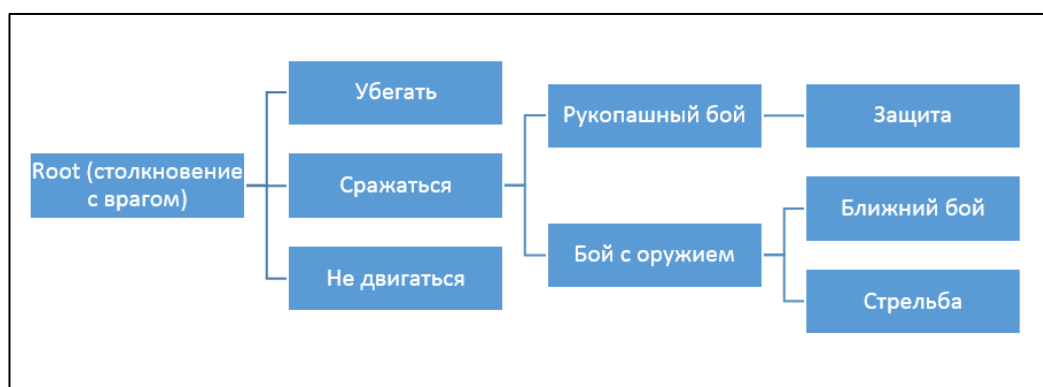


Рисунок 5 — Дерево поведений

Важный этап технического проектирования игры — это подготовка документации. Особое внимание стоит уделять проектной документации проектировщика, чем подробнее описаны все модули, которые необходимо воплотить в игре, тем меньше изменений придётся делать по ходу разработки, что экономит и время, и деньги. Этот шаг особенно важен для крупных команд, где над проектом работает несколько программистов и дизайнеров, в таком случае проектной документацией ни в коем случае нельзя пренебрегать.

1.2 Анализ популярных игр в жанре survival horror

Жанр survival horror является довольно популярным в среде видеоигр, особенно среди инди-разработчиков. Чтобы не затеряться среди конкурентов и выпустить интересный продукт необходимо провести сравнительный анализ наиболее популярных игр в данном жанре. Стоит отметить, что не во всех играх данного жанра присутствует оружие и иногда от врагов можно только убегать и прятаться, такие игры рассматриваться не будут, хотя некоторые и являются очень известными, потому что в своём проекте планируется использование огнестрельного оружия для взаимодействия с противниками.

Для анализа были выбраны игры, выпущенные в течение последних 10 лет: Siren: Blood Curse, Silent Hill: Homecoming, Dying Light, Resident Evil 7, The Evil Within 2 [5, 6, 22].

Siren: Blood Curse (Siren: New Translation) — эксклюзивная игра для PlayStation 3, разработанная SCE Japan Studio на движке Havok и выпущенная Sony Computer Entertainment в 2008 году. Игра является переосмыслением первой части серии Forbidden Siren. Графику нельзя назвать выдающейся даже для 2008 года (рисунок 6). Текстуры не отличаются высокой детализацией, однако это не бросается в глаза из-за темных локаций и слабого освещения от фонаря, что играет только на руку атмосфере игры. Модели людей и зомби довольно качественные и детализированные, благодаря чему люди выглядят живыми, а монстры жуткими.

Управление простое, но иногда не очень удобное. Данный небольшой недостаток некоторые игроки выделяют в достоинство, как бы странно это ни звучало. Управление давало ощущение, что главные герои простые люди, а не бессмертные супергерои, что только добавляла чувство страха и тревоги. Игра представляет вид от третьего лица, камера находится за спиной главного персонажа.



Рисунок 6 — Скриншот игры Siren: Blood Curse

В игре интересная подача сюжета — играть предстоит за нескольких персонажей, всего семь человек, от лица каждого будет показан какой-то эпизод истории, иногда дороги героев пересекаются, что позволяет посмотреть на одну ситуацию с разных сторон. Геймплей и задания будут зависеть от того, за кого предстоит играть в данный момент, например, один из персонажей является десятилетняя девочка и единственный способ выживания для неё — это убегать от врагов и не попадаться им на глаза.

Одной из наиболее интересных особенностей игры является то, что можно смотреть на происходящее глазами врага (зомби) и одновременно продолжать игру, в такие моменты экран делится на две части (рисунок 7). Так же в игре есть возможно переключаться на вид от первого лица в моменты сражений.

К недостаткам игры можно отнести то, что локации коридорные из-за чего геймплей очень линейен, многие вещи сразу отмечены на карте, что губит исследовательскую часть, после нахождения огнестрельного оружия враги перестают столь сильно пугать, так как уничтожаются буквально с двух выстрелов.



Рисунок 7 — Слева вид от лица главного персонажа, справа — от лица врага

В целом игра интересная, имеет необычные элементы в геймплее и довольно интересный сюжет.

Silent Hill: Homecoming — кроссплатформенная игра, разработанная студией Double Helix Games и выпущенная компанией Kanami в 2008 году, является шестой частью серии игр Silent Hill. Игра объединяет в себя два жанра: survival horror и action-adventure. Так же, как и предыдущая игра Silent Hill: Homecoming разработана на движке Havok.

Игровой процесс состоит из исследования местности, решения головоломок, сражений с монстрами. В данной части впервые в серии добавили возможность выбирать ответы в диалогах с персонажами. Тот или иной ответ мог повлиять на развитие сюжета.

Игра имеет вид от третьего лица, стандартное положение камеры — за спиной героя, в момент применения огнестрельного оружия камера плавно перемещается ближе — за плечо героя. Управление очень податливое, главный герой подвижный, хорошо дерётся и умело управляется с оружием, это объясняется тем, что герой бывший военный. Огнестрельное оружие в игре имеет отдачу. Врагов как правило можно уничтожить в ближнем бою, поэтому малое количество патронов для огнестрельного оружия доставляет проблемы только в схватках с боссами. Игра имеет но-

вовведения в плане геймплея только, если сравнивать её с предыдущими частями серии, никаких новых фишек в рамках жанра нет.

Графика средняя, как и в предыдущем примере, качество текстур скрывают слабым освещением (рисунок 8). Из плюсов графики — хорошо проработанные лицевые анимации. Каждый из героев игры активно использует мимику для выражения эмоций.

Выделить можно наличие нескольких концовок, что является фишкой серии Silent Hill. Вид концовки зависит от того, какие выборы игрок делал в ключевых моментах, таких моментов три. В итоге имеется хорошая концовка с бонусной сценой, которую показывают если собрать все фотографии в игре или пройти её на высокой сложности, три концовки, которые можно считать плохими и одна шуточная, которая копирует хорошую с небольшим изменением — вмешательством инопланетян (одна из фишек серии Silent Hill — концовка с инопланетянами).



Рисунок 8 — Скриншот из игры Silent Hill: Homecoming

Dying Light — кроссплатформенная игра в жанре survival horror и action RPG от первого лица с открытым миром, разработанная студией Techland на движке Chrome Engine 6 и изданная холдингом Warner Bros в 2015 году.

В игровом процессе стоит выделить активное использование паркура при перемещении в застроенном городе, из-за чего на первый план скорее выходит жанр action. Сражаться нужно не только с зомби, но и с людьми за ресурсы. В игре реализована смена дня и ночи, которая непосредственно влияет на игровой процесс, ночью враги более агрессивны и встречаются чаще и в большем количестве. Некоторые люди в игре просят помощи главного героя, можно помогать им или игнорировать, что тоже скажется на дальнейшем прохождении.

Так же в игре присутствует кооперативный режим и в этом режиме можно играть не за человека, а за зомби и охотиться на других игроков (рисунок 9).



Рисунок 9 — Игра Dying Light в режиме Be the Zombie

Игра имеет хорошую графику, локации хорошо проработаны и детализированы, что является большим плюсом особенно для открытого мира.

Текстуры не замыленные и хорошо выглядят, как на моделях персонажей, так и на окружающем пространстве. Модели людей и зомби так же на высоком уровне. Люди не похожи один на другого, монстры интересные и пугающие (рисунок 10). Хорошая работа со светом в дневное время, тени динамичные, солнечный свет реалистичен как на улице, так и внутри помещений.

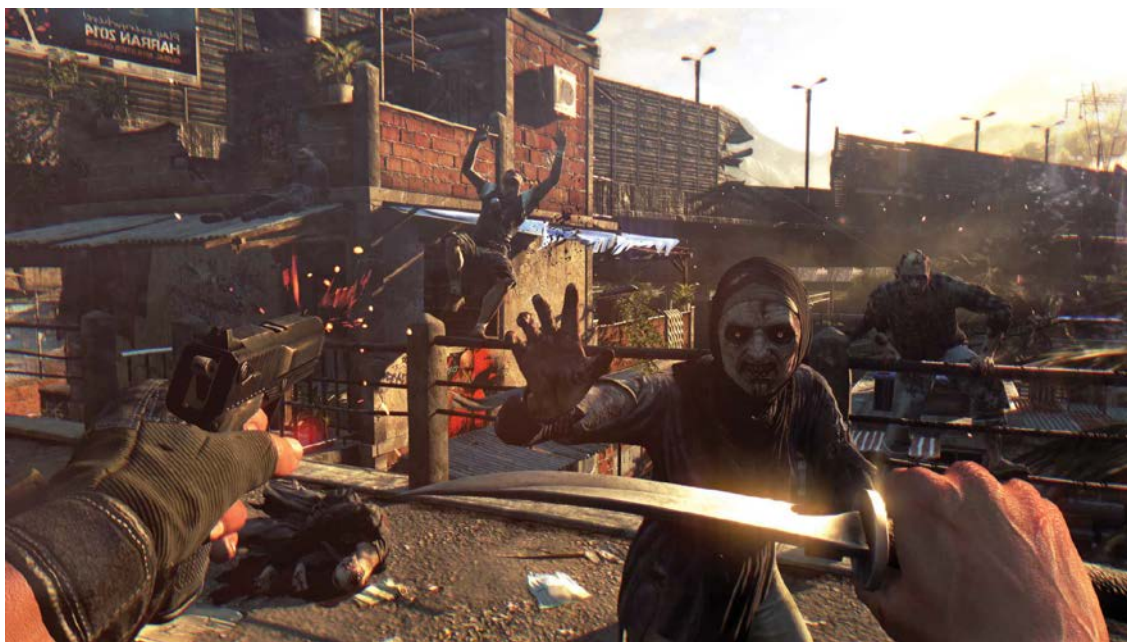


Рисунок 10 — Скриншот из игры Dying Light

Resident Evil 7: Biohazard — кроссплатформенная игра, разработанная на движке RE Engine и выпущенная компанией Capcom в 2017 году, является седьмой частью в серии Resident Evil. Версия для PlayStation 4 поддерживает режим виртуальной реальности через шлем PlayStation VR.

Игра имеет вид от первого лица и фотореалистичную графику (рисунок 11). При создании моделей людей и окружающих предметов студия использовала фотограмметрию, то есть загримированный человек фотографировался с разных сторон, и специальная программа создавала 3D модель на основе этих фотографий. Данный приём, увы, нельзя применять для создания моделей монстров, так как грим в данном случае займёт больше времени, чем моделирование монстра в 3D редакторе. Анимации как лицевые, так и анимации тел живые и хорошо проработанные, тексту-

ры чёткие и реалистичные, грамотное динамическое освещение только подчёркивает все плюсы графики.



Рисунок 11 — Скриншот из игры Resident Evil 7

В игре скудное количество боеприпасов для огнестрельного оружия, а врагов довольно тяжело убить даже из него, не то что в ближнем бою. Локации как правило представляют собой небольшие комнаты или узкие коридоры. Совокупность этих двух факторов держит игрока постоянно в напряжении. Так же в игре есть враги, которых невозможно убить до определенного момента в игре, поэтому придётся прятаться или убегать. Имеются интересные головоломки для открытия проходов в новые помещения. Управление сделано так, что буквально ощущаешь, что главный герой простой человек, а не солдат готовый сражаться — герой не очень ловкий, медленно бежит, устаёт при беге, прицел тугой и долго наводится на врага, при этом плавно покачивается из стороны сторону.

Сюжет подаётся от лица трёх персонажей, за одного предстоит играть в настоящем времени, эпизоды с двумя другими — это моменты из прошлого, записанные на видеокассету. Переключение между персонажами происходит так — главный герой находит в доме кассету, затем ищет телевизор с проигрывателем и как только кассета вставлена игра пере-

ключается на героя от чьего лица эта кассета записана. В игре две концовки (хорошая и плохая), которые определяются выбором в одном ключевом моменте, но эти концовки мало чем отличаются друг от друга, поэтому их введение выглядит глупым и неоправданным.

The Evil Within 2 (Psychobreak) — кроссплатформенная игра, разработанная компанией Tango Gameworks на движке STEM Engine (на основе движка id Tech 5) и выпущенная компанией Bethesda Softworks в 2017 году. Является прямым продолжением игры The Evil Within.

Игра имеет открытый мир, по которому можно свободно перемещаться, но в некоторые места нельзя пройти, пока сюжет того не позволит, локации представляет из себя небольшой городок, состоящий из трёх маленьких районов. Помимо прямого сюжета есть побочные квесты, которые плотно связаны с основным сюжетом и подробнее раскрывают его.

Прицел тут потряхивает, подобно рассмотренной ранее Resident Evil 7, что усложняет борьбу с врагами. Враги здесь различные зомби и мутировавшие люди, поэтому быстрый и надежный способ их убить — попасть пулей чётко в голову. Боеприпасов мало, поэтому приходится экономить и скрываться от врагов. Имеется несколько интересных загадок для открытия проходов.

Игра имеет вид от третьего лица с уже упоминаемой ранее камерой, привязанной к спине главного героя и приближающейся к плечу при прицеливании. Графика качественная, текстуры хорошо проработаны, модели персонажей и врагов отлично детализированы. Освещение часто скудное, так как в игре постоянно ночь, но свет расположен очень грамотно. Постановку кат-сцен с участием фотографа можно назвать шедеврами, за ними визуально интересно наблюдать, так как они сделаны очень кинематографично (рисунок 12).



Рисунок 12 — Скриншот игры The Evil Within 2

Из проведённого анализа можно сделать следующие выводы:

- в играх жанра survival horror большое внимание нужно уделить атмосфере, которая будет поддерживать в игроке чувство тревоги и страха;
- не стоит злоупотреблять так называемыми скримерами — приём, используемый для вызова испуга по средствам громких звуков, резких движений, ярких цветов и так далее — лучше вообще отказаться от данного приёма и сконцентрироваться на создании и поддержании чувства тревоги, так как скримеры быстро надоедают, сбрасывают напряжение и через некоторое время перестают пугать игрока;
- необходимо уделить внимание сюжету, так как игр в данном жанре на данный момент много, из общей массы его может выделить именно интересный сюжет, так же грамотное обоснование мотивов главного героя, его характера и истории позволит вызвать в игроке больше переживаний за судьбу главного героя;
- присутствие оружия или других средств и способов самозащиты является чертой жанра survival horror, которая позволяет отличить данный жанр от очень похожего на него quest с элементом horror, где как правило

игрок может лишь убегать от врага или врага нет вообще, и тревога создаётся лишь с помощью атмосферы и чувства неизвестности;

- главной отличительной чертой жанра является малое количество боеприпасов и аптечек, еды и воды (если такое предусмотрено геймплеем), что является частью выживания.

Вышеперечисленные выводы стали основными критериями, на которые шёл упор в разработке собственной игры.

1.3 Анализ программного обеспечения

На данный момент любой желающий может самостоятельно начать разрабатывать игры при помощи разнообразных игровых движков, которыми пользуются не только начинающие или инди-разработчики, но и крупные всемирно известные компании.

Игровой движок — это программная среда, предназначенная для создания и разработки видеоигр или других интерактивных приложений с графикой, обрабатываемой в реальном времени. Движок игры включает в себя визуализатор, физический движок, звук, систему скриптов, анимацию, искусственный интеллект, сетевой код, управление памятью и многопоточность [6].

Изначально все игры создавались с нуля, но в середине 1990-х годов, после появления знаменитой игры Doom совершился переворот в создании игр. Doom имел четкую архитектуру, которая легко делилась на центральные компоненты (компонент для передвижения, компонент для стрельбы, компонент для обработчика столкновений и т. д.) и графические ресурсы, образующие игровой мир. На основе такой архитектуры стали создаваться игровые движки, теперь компании разработчиков могли использовать один набор инструментария для разных игр, что сокращало время и средства в процессе разработки игры. В следствии чего в последнее время игры создаются либо на общедоступных движках, либо студии

создают движки на основе своих ранее выпущенных игр, дорабатывая их в последствии под тенденции технического прогресса.

Сравнительная характеристика наиболее популярных движков представлена в таблице 1.

Таблица 1 — Сравнительная характеристика игровых движков

| Название | Платформы | Язык программирования | Исходный код | Лицензия | Общие сведения |
|-----------------|--------------------|---|--------------|---|--|
| 1 | 2 | 3 | 4 | 5 | 6 |
| Unreal Engine 4 | Кроссплатформенный | C++, встроенная визуальная скриптовая система — Blueprint | Открытый | Если доходы от игры превышают 3000 \$ в квартал, то 5 % продаж уходит Epic Games | Высокие графические возможности. Имеет мощный инструмент для дизайна игровых уровней прямо в движке. |
| Unity3D | Кроссплатформенный | C#, JavaScript, Visual Basic | Закрытый | Indie-версия: free Pro-версия: 1500 \$ | Хорошее качество графики, не требующая при создании игр больших познаний в языках программирования. |
| CryEngine 3 | Кроссплатформенный | C++ и скриптовый язык Lua | Закрытый | Если доход игры превышает 5000 \$ за первый год, то Cretek получает 5 % от продаж | Продвинутые технологии визуализации и музыкального сопровождения. |

Окончание таблицы 1

| 1 | 2 | 3 | 4 | 5 | 6 |
|-----------------|--------------------------------------|-----|----------|---------------------------|--|
| Navok Physics 8 | Кроссплатформенный физический движок | C++ | Закрытый | Коммерческий | Симулирует физическое взаимодействие в реальном времени, что делает мир игры более живым и реалистичным. |
| Id Tech 6 | Кроссплатформенный | C++ | Закрытый | Коммерческая/Свободное ПО | Есть возможность создавать открытые миры. |

Для разработки собственной однопользовательской игры в жанре survival horror был выбран движок Unreal Engine 4, так как он имеет интуитивно-понятный интерфейс, удобную скриптовую систему Blueprint (рисунок 13), является наиболее инновационным, имеет мощное отзывчивое сообщество пользователей. В проектной документации описан инструментарий движка, имеется справочник по работе с Blueprint и ответы на часто задаваемые вопросы, например, пошаговая инструкция, как реализовать виртуальную клавиатуру. Движок является полностью бесплатным, но при коммерческом использовании 5 % от прибыли игры переходят компании Epic Games при условии, что доходы игры составляют более \$ 3000 за квартал.

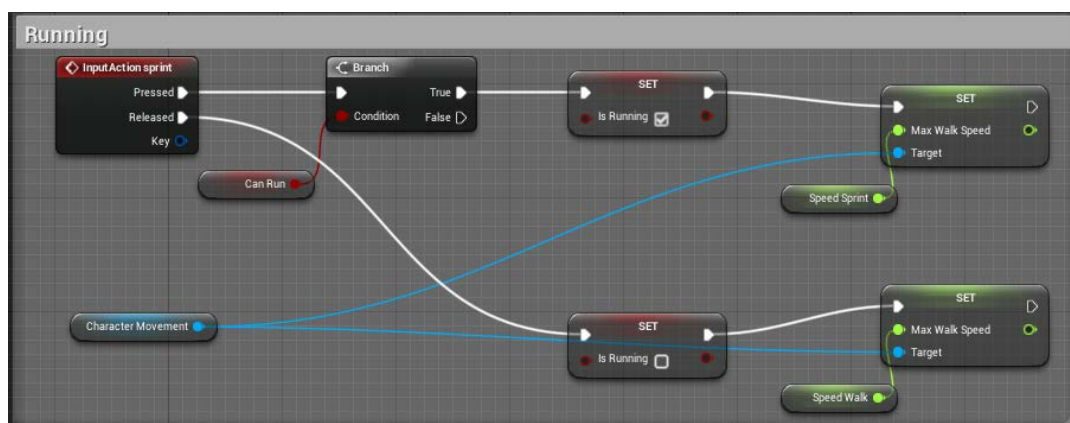


Рисунок 13 — Blueprint, реализующий бег игрового персонажа

При создании игрового персонажа необходимо уделить большое внимание топологии сетки модели и построению скелета для правильной анимации. Для начала рассмотрим эти два термина.

Тополо́гия (от др.-греч. τόπος — место и λόγος — слово, учение) — раздел математики, изучающий явление непрерывности, свойства пространств, которые остаются неизменными при непрерывных деформациях. Например, связность, ориентируемость.

Есть два основных типа информации, необходимой для определения полигональной сетки — геометрия и соединения, другими словами, это набор точек в пространстве (вершины) и набор соответствующих связей между точками (границы). Без информации о связях, полигональная сетка является неструктурированной, а потому и неопределённой. Введение набора граней — это шаг, который, в конечном счёте, актуализирует сетку и задаёт её характер с точки зрения непрерывности, сходимости и связности. Эта сетевая структура и называется топологией пространства. Другими словами, топология — это способность сетки корректно реагировать на деформации, например, при разработке мимики лица, одежды и движений персонажа. Для корректного поведения сетки следует грамотно продумать и построить полигональную сетку персонажа (рисунок 14).

После топологии стоит уделить внимание скелету модели (рисунок 15). Он представляет собой древообразную структуру костей, в которой каждая последующая кость соединена с предыдущей, то есть повторяет за ней движения и повороты с учётом иерархии в скелете. Таким образом, при движении отдельной кости двигаются и все вершины, привязанные к ней.

Ещё один вид скелетной анимации — анимация с развесовками, которая представляет собой более продвинутый вариант скелетной анимации, в ней каждая вершина модели может быть связана не с одной, а с несколькими костями. При этом для каждой кости определяется свой вес, то есть величина влияния этой кости на перемещение вершины. Чем больше вес какой-то кости, тем сильнее вершина смещается под её влиянием.

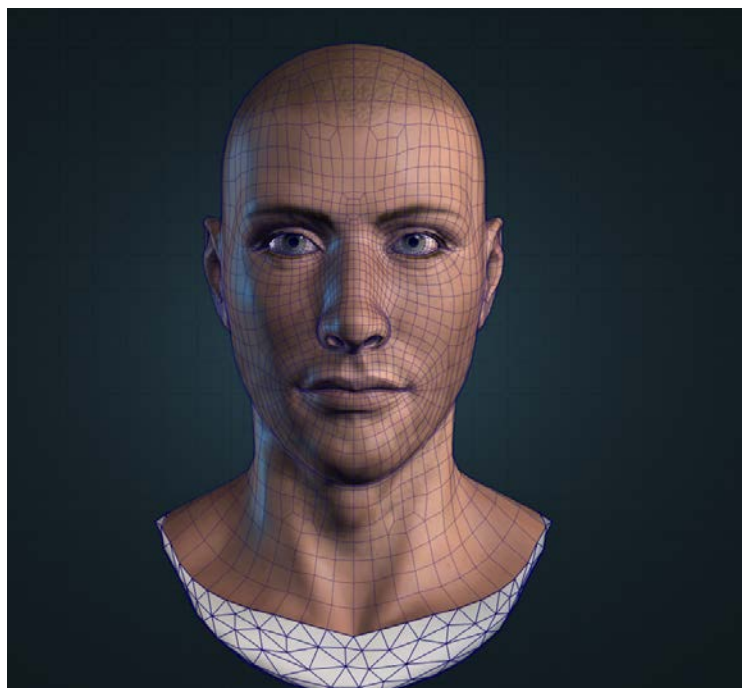


Рисунок 14 — Правильная топологическая сетка

Благодаря развесовкам можно анимировать плавные изгибы поверхностей, например, движения ткани.

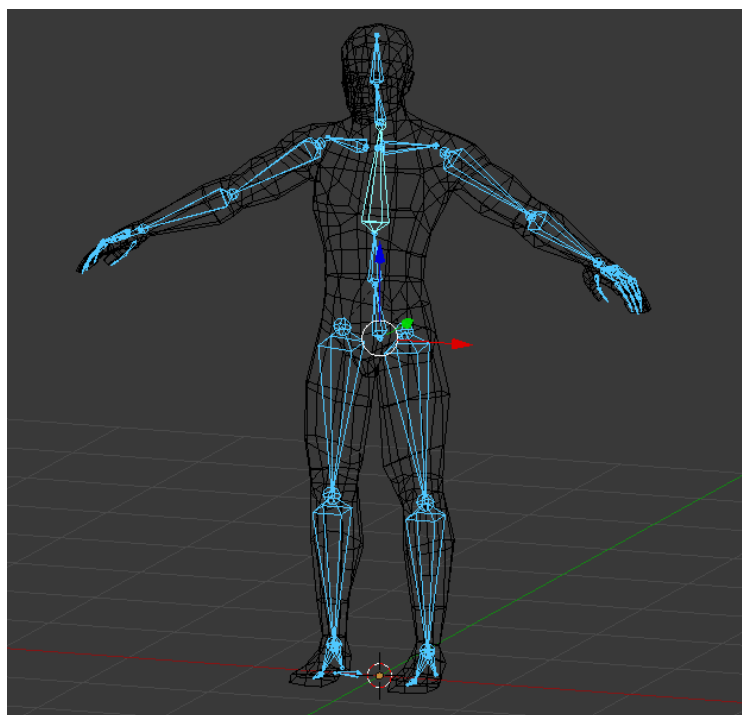


Рисунок 15 — Скелет 3D модели

С учётом вышеперечисленных факторов можно выделить три бесплатные программы, которые предоставляют возможности для создания правильной топологии сетки и скелета модели — Mixamo Fuse, Daz 3D, MakeHuman.

Сравнительная характеристика программ для создания 3D персонажей приводится в таблице 2.

Таблица 2 — Сравнительная характеристика программ для создания персонажа

| Название | Лицензия | Дополнительные материалы | Скелет | Экспорт в .fbx | Возможность анимирования |
|-------------|--|--|--|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| Mixamo Fuse | Free, если устанавливать версию Steam. | Нет, но в программе есть встроенный пакет с одеждой, причёсками, разными типами тел. | При выгрузке модели на сайт mixamo.com для неё автоматически создается скелет. | Косвенный . Программа может экспортировать в формат *.obj, после добавления модели на сайт mixamo.com и её анимирования её можно сказать в формате *.fbx. | На сайте представлено множество бесплатных анимаций, которые автоматически накладываются на модель. |
| Daz 3D | Free | Обширный каталог очень качественных дополнительных материалов (модели, одежда, причёски, обувь | При создании персонажа ему автоматически создается скелет. | Есть | Есть возможности для скелетной анимации модели. |

Окончание таблицы 2

| 1 | 2 | 3 | 4 | 5 | 6 |
|-----------|------|---|-------------------------------|------|-----|
| | | и т. д.), в основном платные. | | | |
| MakeHuman | Free | Небольшой каталог бесплатных материалов, в основном плохого качества. | Есть четыре скелета на выбор. | Есть | Нет |

Стоит отметить, что модель, созданную в MakeHuman, так же можно загрузить на сайт Mixamo.com и проанимировать, если модель имеет формат *.obj или *.fbx. Скелет моделей Daz 3D слишком детализирован и алгоритмы mixamo.com не могут сопоставить его со своим стандартным скелетом, поэтому при загрузке на сайт данных моделей даже в нужных форматах будет выдаваться сообщение об ошибке, но можно удалить из модели скелет, например, в программе 3Ds Max и после этого загрузить модель на сайт, скелет будет создан на сайте автоматически.

Для реализации собственного персонажа была выбрана программа Mixamo Fuse, так как в ней достаточное количество встроенных материалов для моделирования человека и есть удобный редактор для изменения формы всех частей тела. Программа сразу создает правильную топологическую сетку (рисунок 16).

После этого необходимо экспортировать модель в формат *.obj (единственный формат для экспорта). Теперь есть два пути. Первый, открыть модель в 3Ds Max, построить в программе скелет и привязать его к модели, после чего модель можно анимировать в самом 3Ds Max или других программах для анимации, предварительно сохранив модель в формате *.fbx, например, Maya или MotionBuilder. Второй путь, загрузить модель в формате *.obj или *.fbx на сайт mixamo.com, для работы с сайтом необходимо иметь Adobe ID для авторизации на сайте. Здесь автоматически модели присваивается

скелет, после чего её можно просто скачать в формате *.fbx или добавить модели анимацию и скачать её в этом же формате уже с анимацией (рисунок 17).



Рисунок 16 — Персонаж, созданный в Mixamo Fuse

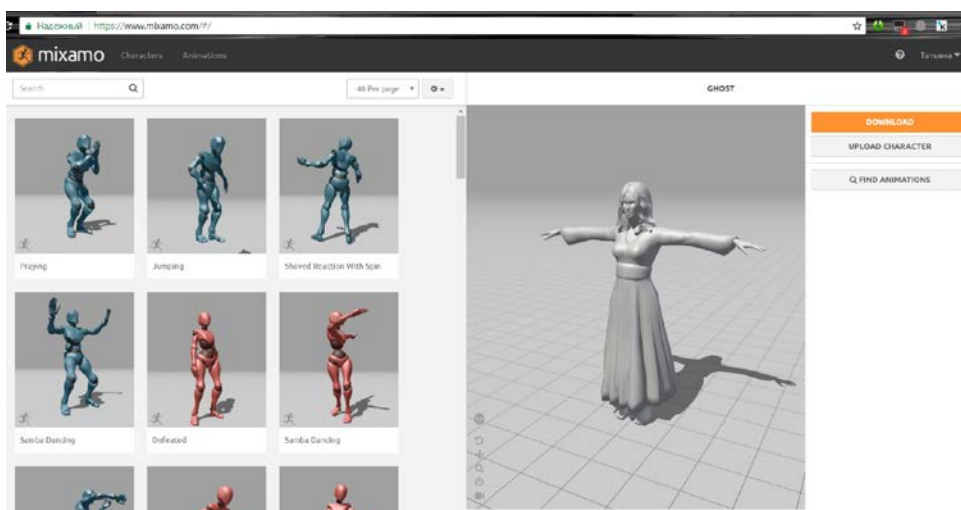


Рисунок 17 — Интерфейс ресурса www.mixamo.com

Для того, чтобы сократить работу над графической составляющей игры и уделить больше внимания технической, было решено пойти по второму пути: модели, созданные в программе Fuse, загружались на сайт,

где происходил подбор анимаций, необходимых для реализации игры — бег, ходьба, состояние покоя и прочее.

1.4 Общий алгоритм реализации игры *Brotherly ties*

Реализация проекта состояла из тринадцати этапов.

Первый этап — разработка концепта игры: о чём игра, какие цели стоят перед игроками, кто является целевой аудиторией. Выбор жанра игры и её вида (2D/3D, от первого лица/от третьего лица). На этом же этапе, после выбора вида и жанра происходил анализ игр в схожем жанре, чтение статей и книг по разработке видеоигр.

Второй этап — написание сценария. Так как жанром игры был выбран survival horror, для вдохновения и поиска идей были просмотрены фильмы в жанре триллер и ужас. Изучались статьи по написанию сценариев и приёмов вызова чувства страха в фильмах и играх. После того, как был придуман сеттинг, основное направление сюжета и главный герой, сценарий был записан в документ Microsoft Word, после чего несколько раз дополнялся и изменялся.

Третий этап — разработка модели главного персонажа и брата (используется в кат-сценах). Для экономии времени персонаж создавался в программе, предназначенной для быстрого моделирования 3D моделей людей и зомби — Mixamo Fuse. После чего готовая модель персонажа была экспортирована на сайт Mixamo.com, где ей добавили скелет и присвоили основные анимации необходимые в игре.

Четвёртый этап — разработка модели врага. Первой версией врага был призрак женщины. Модель создавалась в программе Daz 3D, так как в данной программе можно сделать болезненно худого персонажа. В программе 3Ds Max у модели был удалён скелет, так как данный скелет не подходит для алгоритмов mixamo.com, после чего модель анимировалась так же как модель главного героя. Но после смены сюжета и небольших

геймплейных изменений противника с призрака было решено сменить на монстров. Так как программы, которые рассматривались для создания персонажа человека не предоставляли возможности смоделировать желаемого монстра, модель врага было решено скачать с сайта mixamo.com из бесплатной библиотеки персонажей, для анимации были использованы анимации зомби и мутанта.

Пятый этап — моделирование локаций. Квартира и здание детского приюта были смоделированы в программе Archicad 21, из библиотеки объектов этой же программы были взяты некоторые предметы для интерьера квартиры и приюта. Недостающие элементы смоделированы в программе 3Ds Max или скачаны с сайтов 3ddd.ru, turbosquid.com и free3d.com по бесплатному тарифу.

Шестой этап — подбор текстур и настройка материалов. Текстуры были скачаны с сайтов 3ddd.ru, render911.ru, junior3d.ru, textures.com. На основе текстур в движке Unreal Engine 4 были созданы материалы, которые присваивали моделям.

Седьмой этап — построение уровней. После присвоения материалов модели, если с ними нет взаимодействия по ходу игры, перемещались на сцену сразу, если же планируется взаимодействие, например, дверь нужно открывать и закрывать, то для модели создавался Blueprint, где программировались условия для взаимодействия и анимация. В таком случае на сцену помещается не сама модель, а Blueprint, созданный на её основе. После размещения объектов настраивается реалистичный свет, накладываются звуковые эффект, скаченные с сайтов wav-library.net и freesound.org.

Восьмой этап — подбор головоломок для игры. Для того, чтобы разнообразить игровой процесс в игру было решено добавить головоломки. Внешний вид головоломок создавался в программе Adobe Photoshop, а логика (программирование) производилось в движке Unreal Engine 4.

Девятый этап — программирование. Движок Unreal Engine 4 даёт возможно программировать двумя способами: на языке программирования C++ или с помощью визуального языка Blueprint. Выбран был второй способ, так как он интуитивно более понятен. На данном этапе просматривались видео уроки по работе с Blueprint, а также была прочитана проектная документация на официальном сайте docs.unrealengine.com. После программирования основных возможностей главного персонажа на стандартной модели-манекене, в движок были загружены модели главного героя и монстра, для них настраивалась машина состояний для смены анимаций, после чего манекены были заменены на подгруженные модели.

Десятый этап — создание внутриигровых роликов. Внутриигровые ролики или кат-сцены помогают раскрыть сюжет, в проекте были использованы два вида таких роликов. Первый вид — ролик в реальном времени, то есть происходит смена вида с камеры и игрок наблюдает часть игры без возможности взаимодействия. Второй вид — пререндеренные ролики, ролик создаётся в движке, рендерится, а затем с помощью пользовательского интерфейса воспроизводится на экране.

Одиннадцатый этап — тестирование. Проверка работоспособности происходила после добавления новых возможностей искусственному интеллекту и главному персонажу, построения локаций, программирования главного меню, меню паузы и меню настроек. Данные тесты проводятся до запаковки игры.

Двенадцатый этап — запаковка игры в режиме разработчика и тестирование готового продукта. Некоторые ошибки могут выявиться только после запаковки игры или запуске игры на другом компьютере.

Тринадцатый этап — выпуск игры. После проведения всех тестов и исправления возникших ошибок игра запаковывается для распространения и продажи.

2 РАЗРАБОТКА ИГРЫ BROTHERLY TIES

2.1 Потенциальная аудитория 3D игры в жанре survival horror

Несмотря на приток женской аудитории в среду видеоигр, целевым потребителем по-прежнему являются мужчины (59 % игроков), возрастом от 18 до 35 лет [2]. Любители жанра horror обычно в играх обращают внимание на атмосферу, наличие саспенса и сюжет, поэтому можно сделать вывод, что по модели сегментации игроков по психологическим типам придуманной Ричардом Аланом Бартлом профессором Университета Эссекса, данный игрок является исследователями (explorers). Им интересно изучать игровой мир и раскрывать его тайны, такие игроки любят квесты, ценят сюжет и увлечённо изучают механики игры [12].

Проанализировав целевую аудиторию игры, было решено, что продукт должен обязательно иметь сюжет, предлагать игроку разные механики (поиск предметов, стрельба, решение головоломок), локации должны быть устроены так, чтобы по ним можно было побродить, а не идти из одного пункта в другой по строгому маршруту, для поддержания атмосферы нужно использовать звуковое сопровождение и правильно выстроенный свет.

2.2 Задачи разработки 3D игры

2.2.1 Актуальность разработки 3D игры в жанре survival horror

В качестве проекта выпускной квалификационной работы выступает 3D игра в жанре survival horror с видом от первого лица.

Всемирный рынок игр с каждым годом растёт, а информационные агентства пророчат лишь дальнейший рост отрасли. Аналитическая компания Newzoo опубликовала прогноз по рынку игр на 2018 год. Эксперты ком-

пании считают, что доходы с игр по итогу 2018 года составят \$ 137,9 млрд. Это на 13,3 % больше, чем индустрия заработала в прошлом году, что свидетельствует о большом спросе на такой продукт как видео-игры, несмотря на то, что большую часть прибыли приносят мобильные игры, производство компьютерных и консольных игр не теряет свою популярность, так как игры для телефонов и компьютеров, как правило, имеют разную целевую аудиторию и разные цели. Игры для телефонов обычно создаются с целью «убийства» времени, в них играют пока едут в метро, ждут своей очереди в больнице и так далее, в то время, как компьютерные игры часто стремятся рассказать какую-либо историю [21].

С большим отрывом в рейтинге компьютерных игр лидирует жанр action, среди которых в свою очередь самыми популярными являются игры с элементами ужаса, horror, а также shooter. Данный жанр является одним из самых популярных, потому что состояние высокой активности, в котором человек находится в момент страха, может иметь позитивную окраску. Некоторые из основных химических процессов, протекающих в мозге, когда человек напуган, задействованы также и в других эмоциональных реакциях — например, в счастье, волнении или азарте [10].

Данный проект является актуальным из-за выше представленных факторов — росту рынка игровой индустрии и желанию человека испытывать страх, избегая при этом реальной угрозы жизни, что способствует выбросу гормонов счастья.

2.2.2 Цель и назначение игры *Brotherly ties*

Продукт создается с целью развлечения потенциальных игроков. Игра предназначена для заполнения свободного времени, но в отличие от просмотра фильма или прочтения книги, даёт возможно человеку почувствовать себя причастным к событиям происходящим в сюжете, а также испытать различные эмоции и отвлечься от ежедневной рутины.

Игра должна вызвать эмоциональный отклик и запомниться игроку, чтобы привлечь его внимание к разработчику и его следующим работам.

2.2.3 Требования к 3D игре

Конечный продукт должен представлять собой приложение формата *.exe (исполняемый файл в операционных системах Windows) для 64-разрядной архитектуры, так как большинство, около 90 %, пользователей Steam (самый популярный интернет-магазин, распространяющий игры) имеют 64-битную операционную систему Windows, то для оптимально работы игры лучше выбрать 64-битную версию.

Игра должна состоять из трёх уровней, которые связаны друг с другом: первый переносит на второй, второй переносит на третий. В игре должны быть реализованы автоматические точки сохранения, после смерти главного героя игра начинается с последней точки сохранения.

Для каждого уровня должна быть создана своя локация. Первый уровень — лабиринт, второй уровень — квартира главного героя, третий уровень — заброшенный приют. На первом и последнем уровне должны присутствовать враги и ощущаться атмосфера жанра horror, которая создаётся с помощью освещения и звукового сопровождения, второй уровень должен быть безопасным и служить для передышки игрока и раскрытия сюжета.

В продукте должны быть реализованы следующие пользовательские интерфейсы:

- главное меню, включает в себя кнопки продолжить (начать с последней точки сохранения), новая игра (начать игру с самого начала), настройки (перенаправляет на интерфейс настроек), выход (закрывает игру);
- меню паузы, включает в себя кнопки продолжить (возобновление игрового процесса), настройки (перенаправляет на интерфейс настроек),

выход в главное меню (перенаправляет на интерфейс главного меню), выход из игры (закрывает игру);

- настройки игры — основные настройки графики и настройка чувствительности мыши;
- титры — информация о проекте, его авторе, кнопки с выходом в главное меню и выходом из игры.

В игре для главного персонажа должны быть реализованы функция передвижения (ходьба, бег), стрельба, взаимодействие с предметами, головоломки, решение которых открывает запертые двери, система изменения жизни (уменьшается при ударах врага, восстанавливается после подбора аптечки), система восполнения боеприпасов.

Для искусственного интеллекта реализовать функции передвижения, атаки и реакции на игрового персонажа.

Игра должна иметь графику, приближенную к реалистичной для лучшего погружения игрока в процесс. С этой же целью выбран вид от первого лица — положение камеры (рисунок 18).



Рисунок 18 — Игра с видом от первого лица

Хронометраж игры должен составлять около 30 минут при первом прохождении, когда не известно, что где лежит и куда конкретно нужно идти.

В игре должны присутствовать кат-сцены (внутриигровые ролики), как минимум по одной на каждом уровне, которые будут раскрывать сюжет.

2.2.4 Входные данные к игре Brotherly ties

В проекте были задействованы модели, скаченные с сайтов 3ddd.ru, turbosquid.com и free3d.com, а также модель монстра, использовалась в качестве врага, скаченная с сайта mixamo.com. Все модели скачаны по бесплатным тарифам. При необходимости модели корректировались в программе 3Ds Max.

Для построения лабиринта, квартиры и здания приюта, были проанализированы планы однокомнатных, двухкомнатных квартир, квартир-студий, зданий больниц, школ и детских приютов, а также рисунки различных лабиринтов.

Текстуры были скачаны с сайтов 3ddd.ru, render911.ru, junior3d.ru, textures.com. по бесплатным тарифам. С помощью текстур внутри движка Unreal Engine 4 создавались материалы, которые потом присваивались 3D моделям (приложение В).

Аудиофайлы необходимые для звукового сопровождения игры скачены с сайтов wav-library.net и freesound.org. Так как Unreal Engine 4 работает с аудиофайлами формата *.wav, аудиофайлы, имеющие другой формат, конвертировались на сайте online-convert.com.

2.2.5 Характеристика оборудования для реализации игры Brotherly ties

Для реализации проекта был использован ноутбук фирмы MSI модель GP70 2QE-645RU и графический планшет Wacom One S (Small) A6.

Характеристики ноутбука представлены в таблице 3, характеристики графического планшета указаны в таблице 4.

Таблица 3 — Характеристика ноутбука MSI GP70 2QE-645RU

| Характеристика | Описание |
|------------------------------|--------------------------------------|
| Процессор | Intel Core i7-4720HQ CPU 2.6 ГГц x64 |
| Количество ядер процессора | 4 |
| Операционная система | Windows 8.1 |
| Тип системы | 64-разрядная операционная система |
| Размер оперативной памяти | 8,00 Гб |
| Частота оперативной памяти | 1600 МГц |
| Модель дискретной видеокарты | NVIDIA GeForce 940M |
| Модель встроенной видеокарты | Intel HD 4600 |
| Объём видеопамати | 2,00 Гб |

Таблица 4 — Характеристика графического планшета Wacom One S (Small) A6

| Характеристика | Описание |
|----------------------------|---|
| Поддерживаемые ОС | Windows XP, Windows Vista, Windows 7, Windows 8, Mac OS X 10.6 или выше |
| Способ ввода | перьевой |
| Размер рабочей области | 152x95 мм |
| Разрешение | 2540 lpi |
| Чувствительность к нажатию | 1024 уровня |
| Точность пера | 0.25 мм |

2.3 Этапы разработки игры Brotherly ties

2.3.1 Разработка концепта игры

Первое, что стоит обратить внимание при разработке игры — это её концепции (идея). О чём игра, какие цели стоят перед игроками, какой у игры жанр, сеттинг и так далее [7].

За время работы над проектом концепт игры менялся два раза. Изначально игра планировалась, как небольшой horror, главной целью которого является раскрытие истории, поиск записок, ключей. Пугать игроков должна была атмосфера, скримеры и появление призрака в конце, убегая

от которого нужно было покинуть дом. Игра состояла бы из одной локации старого дома.

После изучения планов различных домов и коттеджей, удовлетворяющих сюжет, был сделан вывод, что данная локация будет слишком маленькой и игра получится не интересной. В следствии чего, был немного изменён первоначальный сценарий и замысел. Игра по-прежнему состояла из одной локации, но дом заменили на особняк. Тем не менее, геймплей оставался бы довольно скудным и скучным. Поэтому для разнообразия механик в игру было решено добавить огнестрельное оружие. Стало очевидно, что одного уровня для раскрытия сюжета и механик мало.

Игра полное изменение сценария, в следствии чего изменились враг и локации. Теперь игра приобрела черты жанра survival horror: наличие оружия, скудное количество припасов.

Несмотря на неоднократную переработку идеи игры, сеттинг оставался неизменным, так как игроки как правило чувствуют себя уютно и комфортно в уже знакомом мире, события игры происходят в 21 веке в Соединённых Штатах Америки.

Жанр horror, а позже и survival horror, был выбран, потому что это один из популярных жанров, а так как на сегодняшний день на рынке horror игр представлено множество игр клонов таких проектов как Slender и Five Nights at Freddy's, то игры отличные от них по механике привлекают интерес.

2.3.2 Написание сценария

На этапе разработки концепта игры был написан черновой сценарий, где был определён главный герой, его цели и место действия. Теперь нужно было развить этот концепт в полноценный сценарий с репликами и действиями персонажей.

На этом этапе для поиска фишек и идей просматривались фильмы в жанрах horror, анализировались игры жанра survival horror. Так же изучались статьи на тему написания сценария [9, 16] и статьи по приёмам, которые используются в играх и фильмах для вызова чувства страха и тревоги у зрителя или игрока [1, 17].

Сценарий был оформлен в виде таблицы в программе Microsoft Excel, где были указаны основные события и реплики персонажа (приложение Б).

2.3.3 Разработка моделей главного и второстепенного персонажей

Персонажи создавались в программе, предназначенной для быстрого моделирования 3D моделей людей и зомби — Mixamo Fuse. Данную программу можно приобрести бесплатно в интернет-магазине Steam или купить новую версию Fuse CC в Adobe Creative Cloud.

Программа не локализована на русский язык, но имеет интуитивно понятный интерфейс, поэтому проблем с работой не возникает (рисунок 19). Создание персонажа похоже на создание персонажей в играх, выбирается форма лица, тела, рук, ног, все части можно подкорректировать. Затем на модель надевается одежда, обувь и причёска, у этих объектов можно корректировать только текстуру (цвет).

Главным героем игры является мужчина 25 лет европейской внешности и стройного телосложения, который должен быть одет в современную одежду. Поэтому для построения модели были выбраны части тела Male Fit A. Цвет волос и глаз светлый. Одежда в повседневном стиле — футболка, джинсы и кеды. Программа сразу назначает объектам материалы и стоит правильную топологическую сетку (рисунок 20).

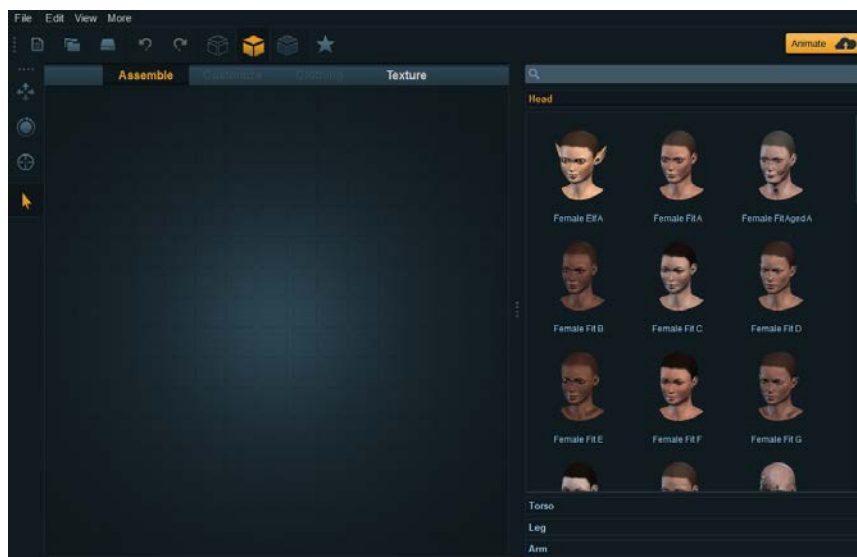


Рисунок 19 — Интерфейс программы Mixamo Fuse



Рисунок 20 — Модель главного героя игры с отображением топологической сетки

Модель второстепенного персонажа, который по сюжету является братом главного героя, создавалась по такому же принципу, как и модель главного героя. Для построения данной модели использовались части Male Fit Teen A, то есть аналог предыдущей модели, только младшего возраста (рисунок 21).

Обе модели экспортировались в формат *.obj и загружались на сайт mixamo.com, где им присваивался скелет и подбирались анимации. У главного героя есть два состояния — в квартире он без оружия, в лабиринте и заброшенном приюте он с оружием, поэтому необходимо было

скачать анимации, которые предусматривали бы ношение оружия и анимации для безоружных персонажей.

Список анимации главного героя:

- состояние покоя с оружием и без;
- анимация шага с оружием и без;
- анимация бега с оружием и без;
- анимация состояния покоя в положении сидя с оружием и без;
- перемещение в положении сидя без оружия;
- реакция на урон;
- анимация смерти.



Рисунок 21 — Модель второстепенного персонажа с отображением топологической сетки

Модель брата использовалась только в кат-сценах, поэтому для него требовалось меньше анимации:

- состояние покоя;
- анимация бега;
- покой в положение лёжа;
- анимация ползания;
- анимация объятий;

- анимация приветствия.

Каждая анимация скачивается отдельно в формате *.fbx, все они импортируются в движок, при импорте нужно указать, что это Skeletal Mesh, чтобы помимо самой модели импортировался и скелет, где для них программируется анимационный Blueprint с машиной состояний (рисунки 22, 23).

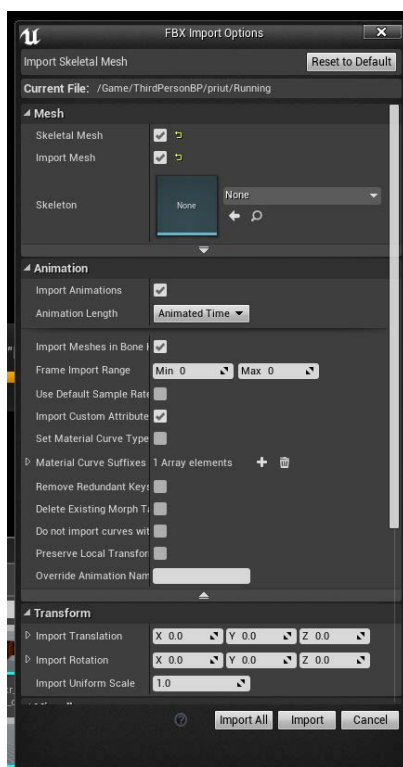


Рисунок 22 — Импорт модели skeletal mesh

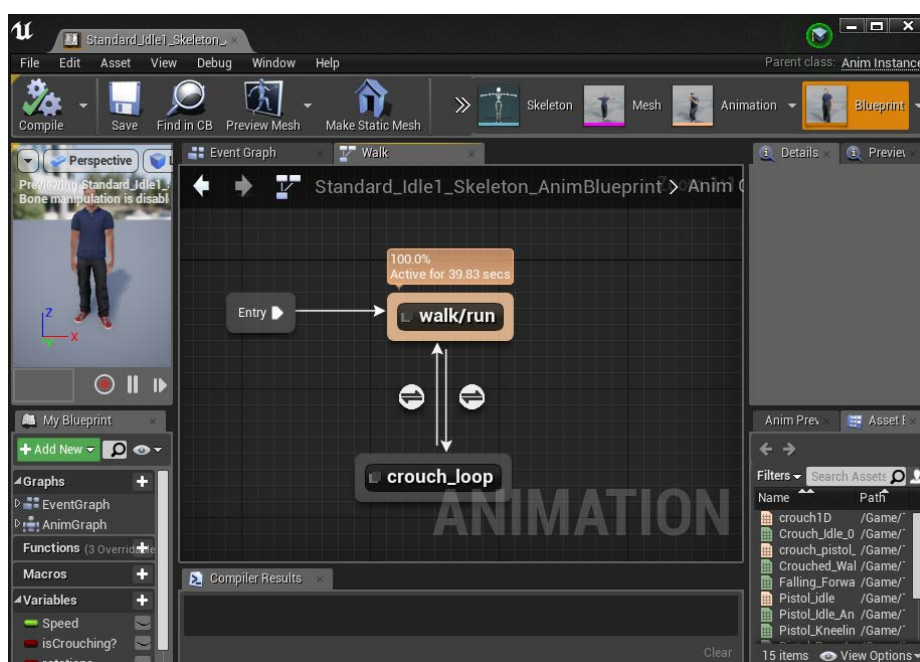


Рисунок 23 — Анимационный Blueprint машина состояний

В таком Blueprint настраиваются условия, по которым меняются анимации, например, если скорость передвижения игрового персонажа изменилась с 0 до 200 единиц, значит надо изменить анимацию с состояния покоя на анимацию шага.

2.3.4 Разработка модели врага

Изначально в качестве врага выступал призрак женщины. Первые попытки его создания проходили так же в программе Fuse, но в данной программе предусмотрены лишь модели живых людей и зомби, поэтому модель создавалась в программе Daz 3D, в ней можно сделать человека болезненно худым (рисунок 24).

После в программе Adobe Photoshop CC текстура кожи видоизменялась, чтобы сделать её бледнее и добавить кровавые брызги, раны и синяки. Модель создавалась без волос и одежды, они привязывались к модели после в программе 3Ds Max, там же у модели удалялся родной скелет, так как он вступает в разногласия с алгоритмами mixamo.com (рисунок 25).

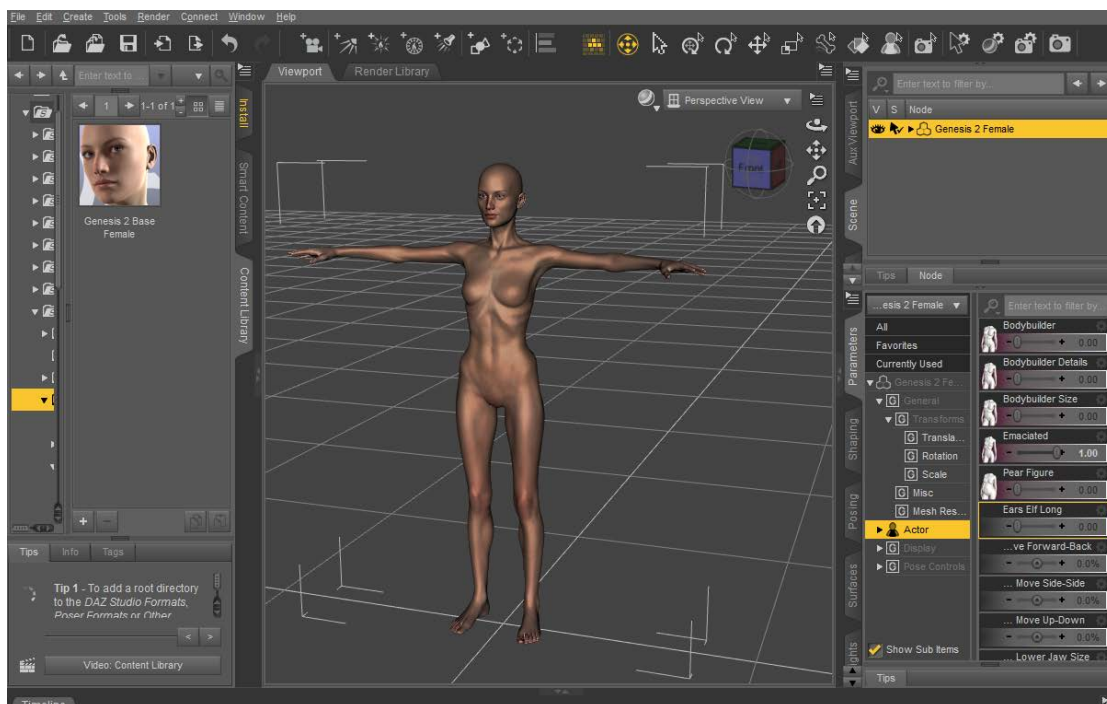


Рисунок 24 — Заготовка для модели врага, созданная в программе Daz 3D



Рисунок 25 — Финальный вид призрака

После смены сценария появилась необходимость сменить модель врага с призрака на монстра, так как данный призрак смотрелся не гармонично. Так как ни одна из доступных программ не даёт возможности создать желаемого монстра, а время на создание игры ограничено, было решено скачать монстра из бесплатной библиотеки сайта mixamo.com, там же модель анимировалась (рисунок 26).

Для монстра понадобились анимации:

- состояния покоя;
- перемещения — шаг и бег;
- атаки;
- реакции на урон;
- смерти;
- анимация поедания для кат-сцены.

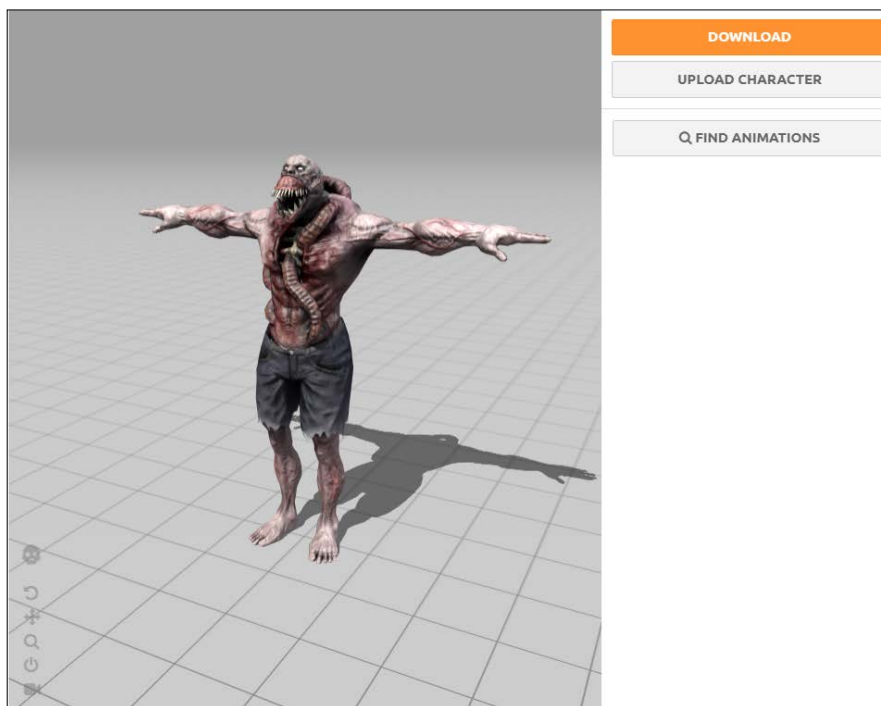


Рисунок 26 — Модель монстра

Анимации скачивались по одной и импортировались в Unreal Engine 4, где для них создавался анимационный Blueprint с машиной состояний.

2.3.5 Моделирование локаций

Квартира и здание детского приюта были смоделированы в программе Archicad 21. Данная программа позволяет быстро смоделировать здания и добавить основные предметы интерьера и экстерьера.

Квартира была полностью смоделирована и обставлена мебелью в программе Archicad 21. Некоторые мелкие предметы интерьера скачивались из Интернета (сайты 3ddd.ru, turbosquid.com, free3d.com), чтобы придать квартире более живой вид.

Здание приюта было построено в программе Archicad 21, после дорабатывалось в программе 3Ds Max, предметы интерьера так же брались из библиотеки Archicad 21 и скачивались из Интернета.

Обе модели импортировались в 3Ds Max для подготовки их переноса в Unreal Engine 4. Необходимо было создать предметам коллизии, с помощью примитивов типа Box, настроить корректное отображение текстур, проверить индексы полигонов, доработать некоторые модели (рисунок 27).

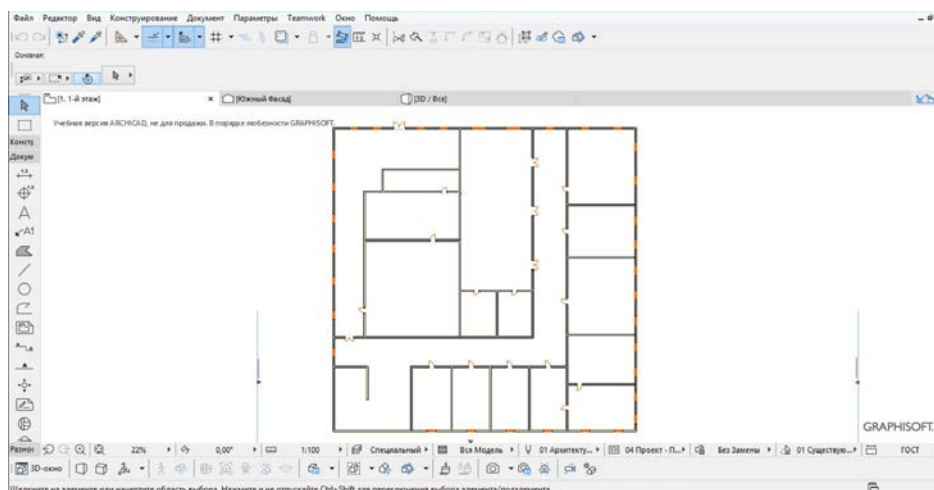


Рисунок 27 — План первого этажа приюта в программа Archicad 21

Модель лабиринта создавалась в программе 3Ds Max из примитивов типа Box.

Все модели импортируются в Unreal Engine 4 (рисунок 28).



Рисунок 28 — Импорт модели Static Mesh в движок

Движок может автоматически создавать коллизии, но они могут построиться не корректно, поэтому лучше создавать их вручную в 3Ds Max, что уже было сделано. Так же стоит убрать флажок с пункта Skeletal Mesh этот тип используется для моделей персонажей, так как у них есть скелет.

2.3.6 Подбор текстур и материалов

Одни из важнейших этапов в создании реалистичного окружения в игре — подбор текстур и создание материалов на их основе. Текстуры должны иметь хорошее качество и быть бесшовными. Сайты 3ddd.ru, render911.ru, junior3d.ru, textures.com. предоставляют текстуры в большом разрешении, не все и них бесшовные, но таковые всё же имеются.

Одной хорошей текстуры для реалистичного материала мало, необходимы ещё различные карты, одна из самых важных — карта нормалей (имитирует неровности поверхности).

Для создания таких карт есть специализированные программы, например, CrazyBump, программа позволяет создавать различные карты на основе изображения с текстурой.

Для создания карт нормалей в данном проекте была использована программа Adobe Photoshop CC, фильтры данной программы позволяют создавать карты нормалей и рельефа (рисунок 29).

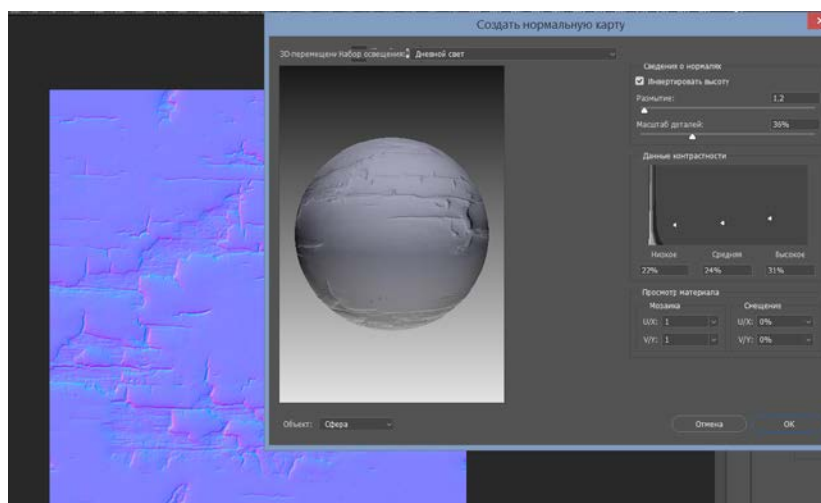


Рисунок 29 — Создание карты нормалей в программе Photoshop CC

На основе текстур и карт нормалей в движке Unreal Engine 4 создаются материалы, которые потом применяются на модели. При комбинации нескольких текстур и использование параметров блеска и прозрачности, можно создать интересные новые материалы. На рисунке 30 представлен материал грязной, потёртой стали, который создан с помощью текстуры старой стены.

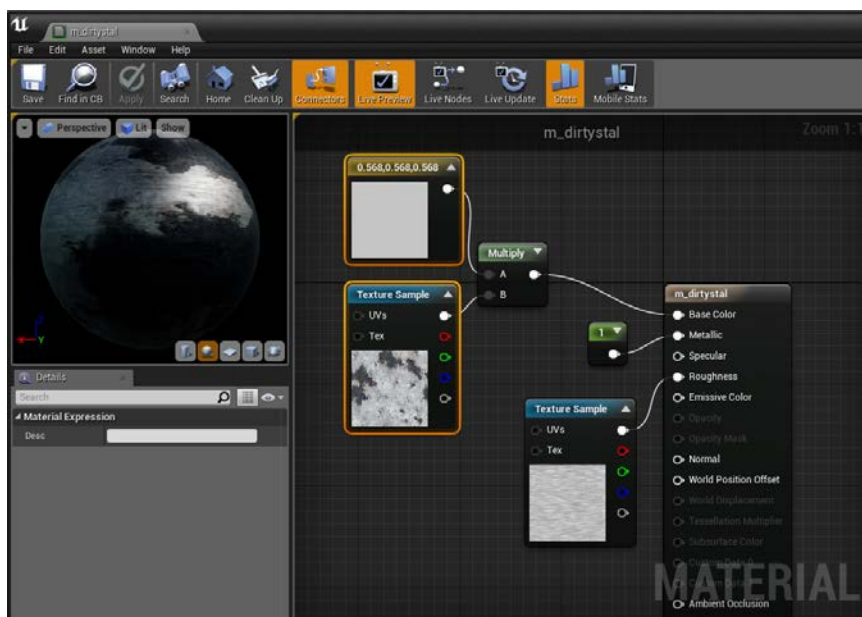


Рисунок 30 — Blueprint материала старой стали

Если полигонам модели были присвоены индексы, то каждому индексу будет соответствовать свой слот под материал, таким образом на одну модель можно накладывать несколько материалов.

2.3.7 Построение уровней

После того, как все модели импортированы и им присвоены материалы можно начинать создавать локации для уровней. В игре создано пять уровней или карт — на трёх уровнях происходят действия игры, ещё два являются вспомогательными для главного меню и титров.

При создании нового уровня Unreal Engine 4 предлагает выбор из нескольких шаблонов: уровень по умолчанию, VR уровень и пустой уровень (рисунок 31).

Для создания лабиринта использовался, а для квартиры и приюта уровень по умолчанию. Уровень с лабиринтом не имеет окон и проходит в замкнутом тёмном пространстве, поэтому небо и солнечный свет здесь не нужны. Уровни с квартирой и приютом имеют окна и протекают в дневное время суток, поэтому для них создавался шаблон, имеющий небо и солнце.

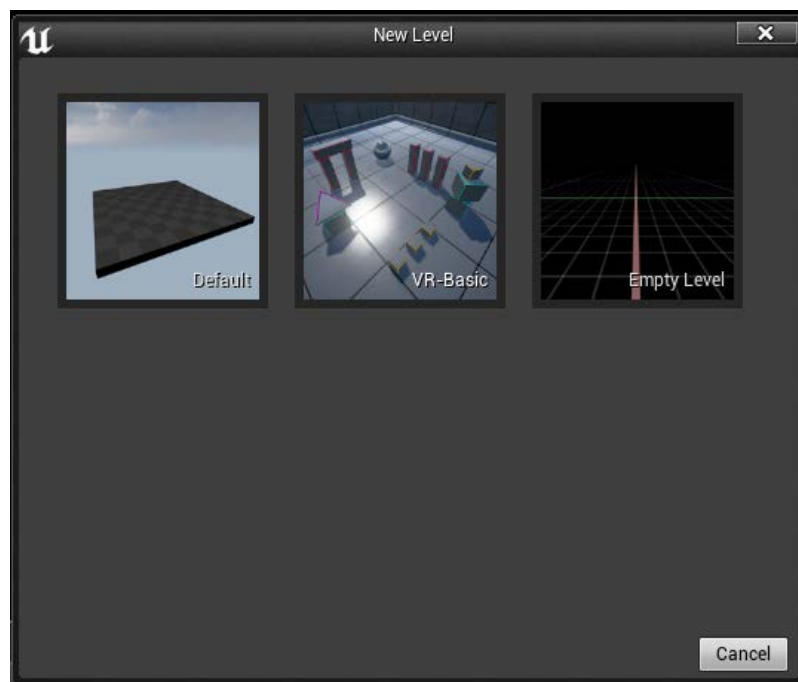


Рисунок 31 — Шаблоны уровней

Если с моделями нет взаимодействия по ходу игры, то они перемещались на сцену сразу, если же планируется взаимодействие, то для модели создавался blueprint, где программировались условия для взаимодействия и анимация (рисунок 32), тогда на сцену помещается не модель, а Blueprint, созданный на её основе.

После размещения объектов на сцене настраивался реалистичный свет.

На первом уровне это был самосветящийся материал для имитации свечения лампы (рисунок 33) и два источника света типа Point Light (рисунок 34).

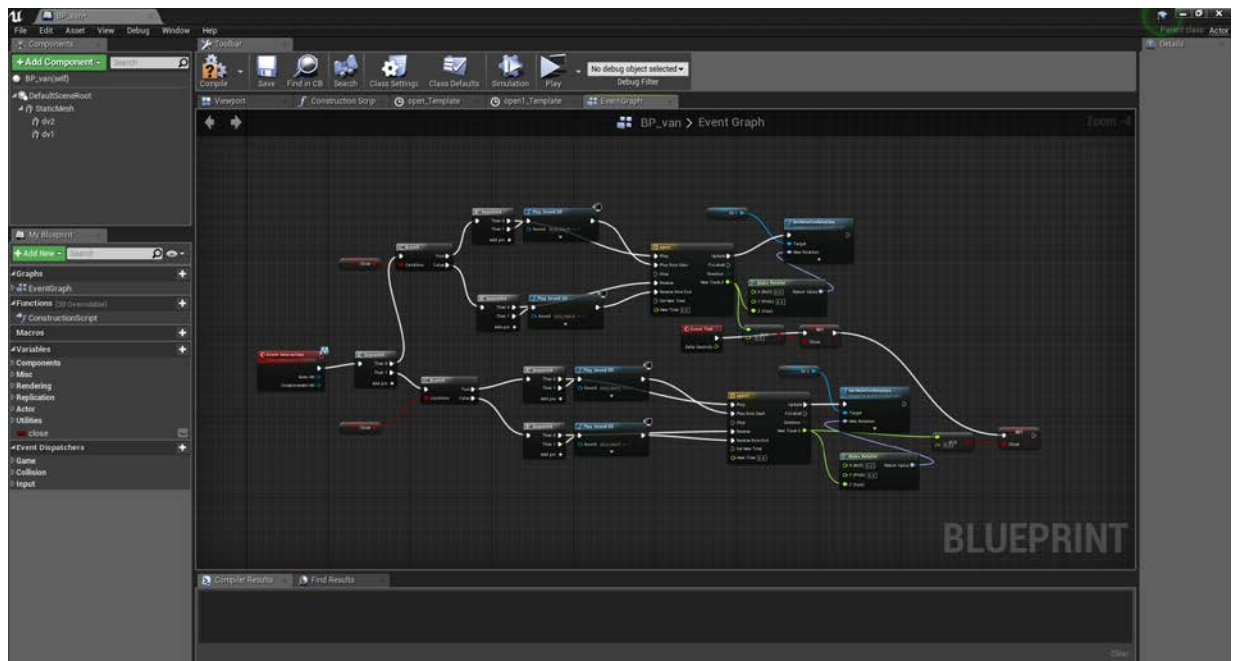


Рисунок 32 — Blueprint для открытия/закрытия дверей ванного шкафчика

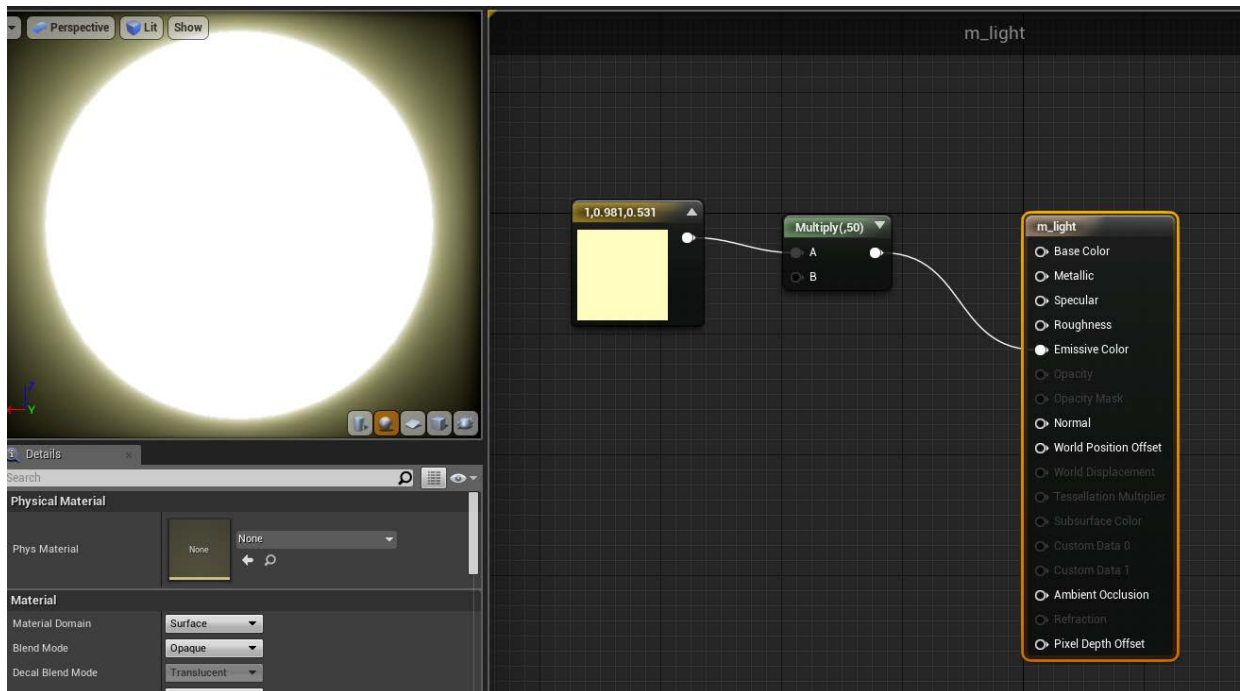


Рисунок 33 — Blueprint самосветящегося материала

Для реалистичного солнечного света на уровне с квартирой были использованы источники света Light Source — имитирует солнечный свет, и SkyLight — имитирует небесное освещение путем захватывания панорамного изображения (рисунок 35) [14].



Рисунок 34 — Источник света Point Light на сцене



Рисунок 35 –Освещение второго уровня игры

Третий уровень освещался по принципу второго, здесь использовались те же источники света, только уменьшалась интенсивность источника SkyLight, чтобы помещение оставалось мрачным (рисунки 36, 37).

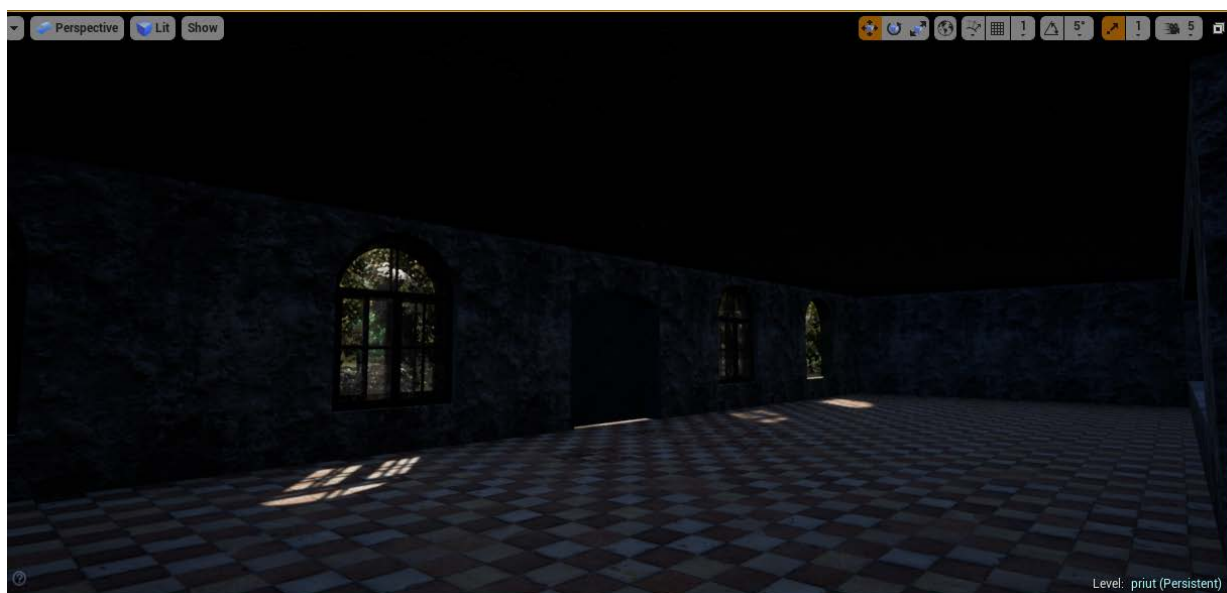


Рисунок 36 — Освещение третьего уровня внутри здания



Рисунок 37 — Освещение третьего уровня снаружи здания

После построения локаций на них добавлялись звуковые эффекты. Скрипы у дверей воспроизводились при открывании и закрывании, они настраивались через Blueprint двери. Звук шагов был привязан к анимации хождения и бега. Фоновые звуки перемещались на сцену, где настраивались их радиус действия и громкость. Все звуки скачивались из Интернета с сайтов, распространяющих звуковые эффекты для видео и игр на бесплатной основе. Это сайты wav-library.net и freesound.org.

2.3.8 Подбор головоломок

Для того чтобы разнообразить игровой процесс в игру было решено добавить головоломки. После небольшого опроса было решено делать головоломки в виде пользовательских интерфейсов.

Всего в игре пять головоломок:

- кодовый замок сейфа, нужно найти код спрятанный в квартире;
- кодовый замок на двери приюта, подсказка в одной из комнат приюта;
- головоломка, состоящая из квадратов размером 3x3 нужно сменить цвет всех квадратов, при этом нажатие на один из них меняет цвет сразу у нескольких;
- головоломка на нахождение правильного пути, состоит из квадратов размером 4x4, подсказка в одной из комнат приюта;
- головоломка состоящие из квадратов, необходима зажечь правильную комбинацию.

Логика головоломок программировалась в движке Unreal Engine 4 (рисунок 38).

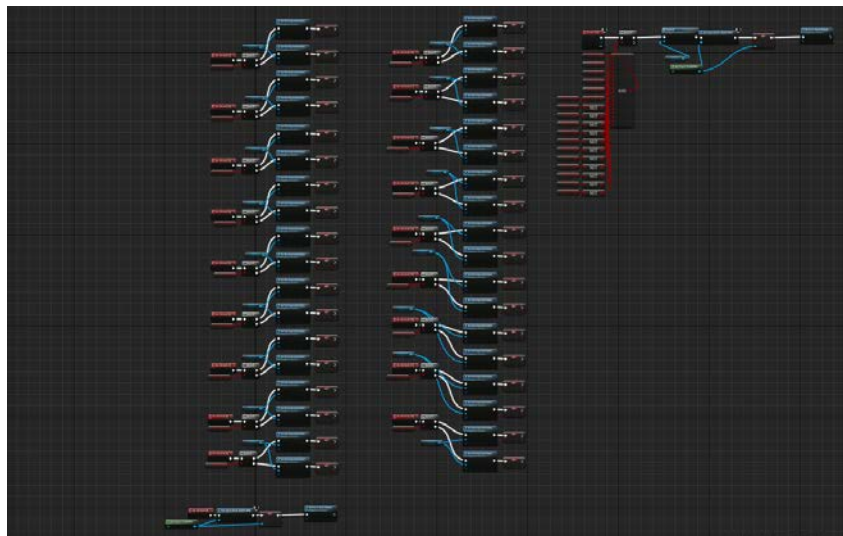


Рисунок 38 — Blueprint последней головоломки

После успешного решения головоломки открывалась запертая дверь, рядом с которой находилась панель запускающая интерфейс головоломок.

2.3.9 Программирование основных элементов игры

Движок Unreal Engine 4 даёт возможно программировать двумя способами: на языке программирования C++ или с помощью визуального скриптового языка Blueprint [14, 24].

Для разработки проекта был выбран визуальный язык Blueprint, так как по нему больше полезной информации на сайте движка Unreal Engine 4, он более понятен и быстро осваивается, язык C++ имеет больше нюансов и раньше сталкиваться с ним не приходилось.

С помощью Blueprint, как уже упоминалось ранее, программировались головоломки, настраивались материалы, создавалась логика взаимодействия с некоторыми объектами и настраивалась машина состояний для анимаций.

Но самый важный аспект — это программирование главного игрового персонажа и искусственного интеллекта врага [25, 26, 27]. Враг в игре исполняет мало функций, поэтому его программирование не заняло много времени (рисунок 39).

Функционал врага:

- перемещение по локации в пределах заданного радиуса;
- если враг видит персонажа, то он бежит за ним;
- если персонаж попал в зону атаки, враг атакует.

Реализованный функционал игрового персонажа (рисунок 40):

- перемещение;
- бег;
- включение/выключение фонарика;
- стрельба из пистолета;
- прицеливание;
- взаимодействие с предметами;
- система жизни (уменьшение и восполнение);
- уставание при беге;

- система передвижения в полуприсяде.

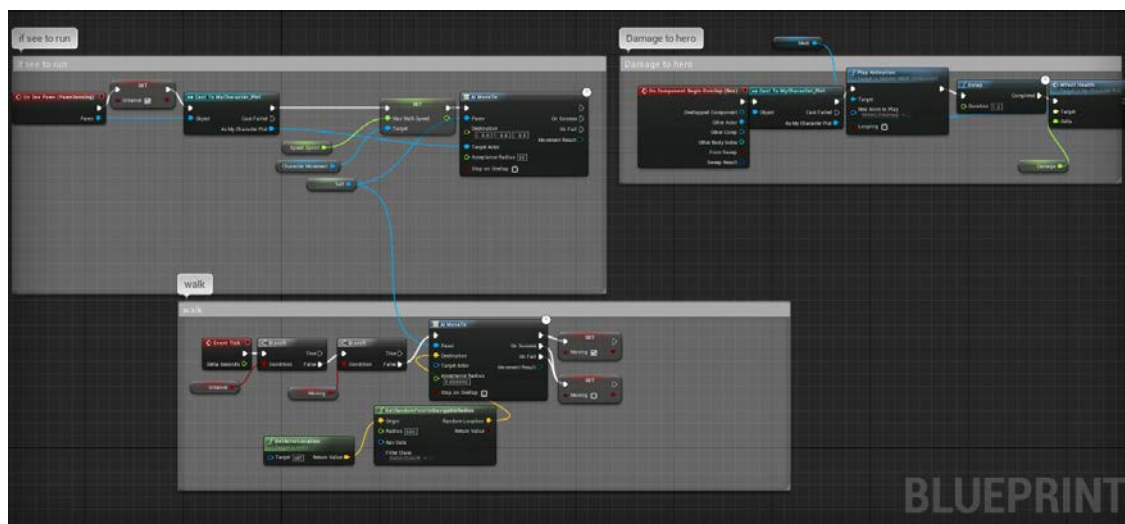


Рисунок 39 — Blueprint искусственного интеллекта

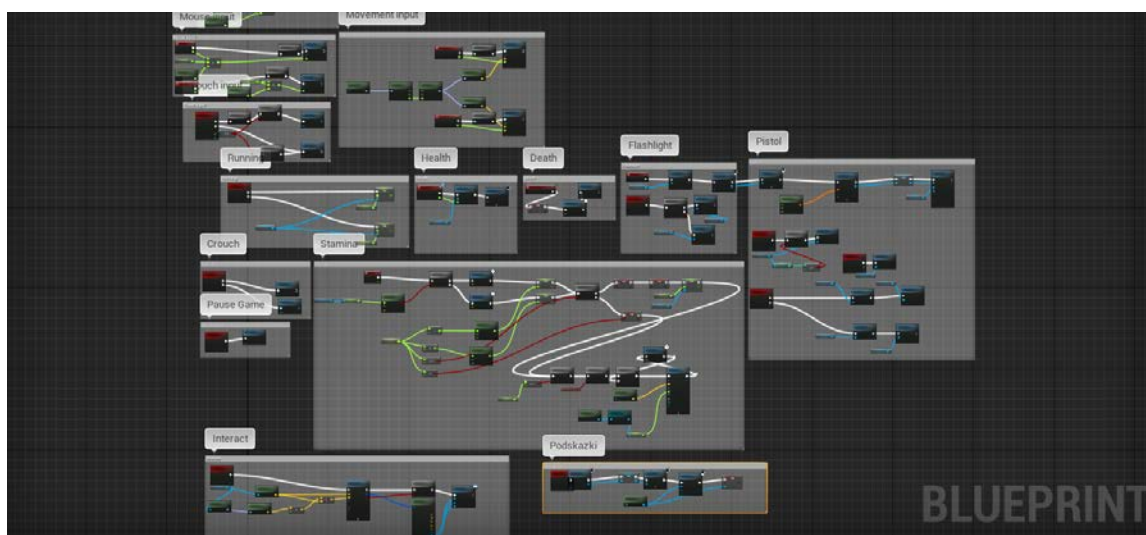


Рисунок 40 — Blueprint игрового персонажа

С помощью Blueprint программировались пользовательские интерфейсы главного меню, титров, подсказок, меню настроек и меню паузы [29]. В этих интерфейсах у кнопок был обработчик событий, который при нажатии на кнопку выполнял какое-либо действие, например, если в интерфейсе главного меню нажать на кнопку «настройки», то открывался новый интерфейс, содержащий меню настроек (рисунок 41).

Помимо этого, программировался пистолет, в его Blueprint настраивалась стрельба, к этому Blueprint обращается Blueprint главного героя при нажатие левой кнопки мыши для осуществления выстрела, при попадании пули в модель врага у врага отнималось значения жизни [28, 29].

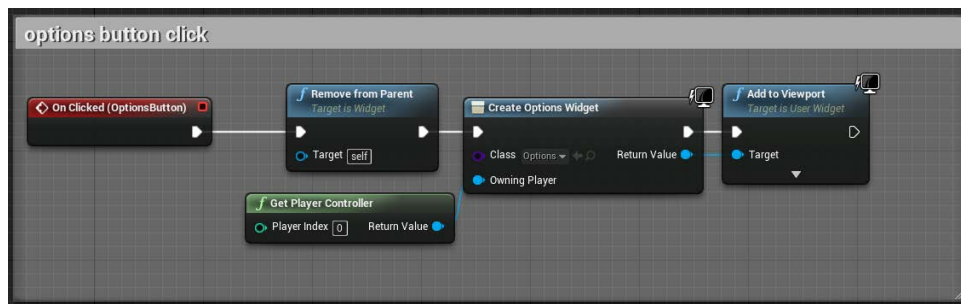


Рисунок 41 — Обработчик события пункта меню «настройки»

Из важных деталей можно отметить так же Blueprint аптечек и магазинов с патронами. Аптечки обращались к Blueprint главного героя и пополняли значение жизни, а патроны обращались к Blueprint пистолета и пополняли значение патронов.

Так же настраивались события, запускаемые по триггеру. Персонаж заходил в зону триггера, после чего происходило определённое действие. С помощью триггеров в игре осуществляются сохранения и воспроизводятся кат-сцены.

2.3.10 Создание внутриигровых роликов

Внутриигровые ролики, так же называемые кат-сценами, используются для раскрытия сюжета. Всего в игре пять кат-сцен: две на первом уровне, одна на втором, две на третьем уровне.

Ролик на втором уровне протекает в реальном времени, то есть на сцене присутствует камера, которую не видно при игре и во время кат-сцены вид с камеры персонажа переключается на камеру, задействованную в кат-сценах, после её воспроизведения, вид опять переходит на камеру игрового персонажа.

Остальные ролики создавались с помощью инструмента Level Sequence, на сцену добавлялась камера, вспомогательные персонажи, настраивалась анимация персонажей и камеры по ключевым кадрам (рисунок 42) [29].

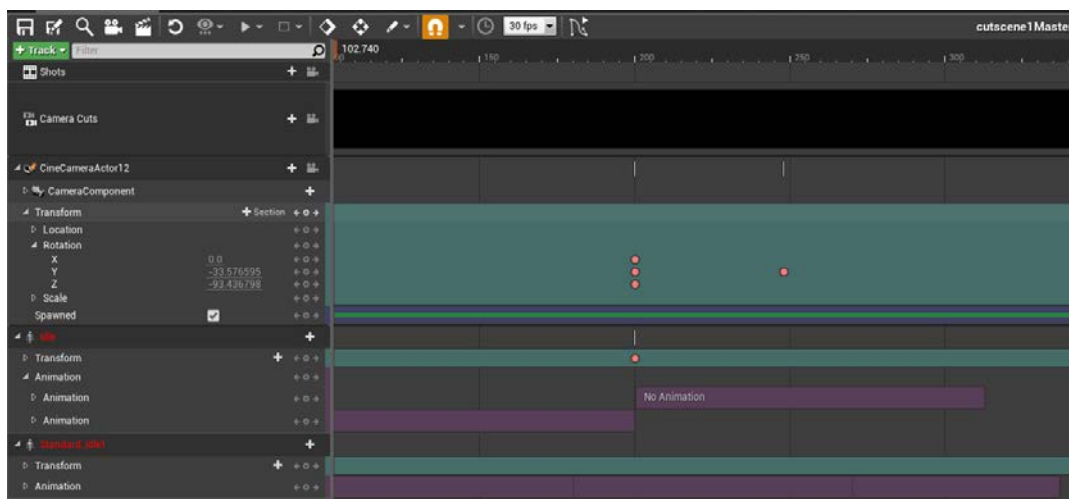


Рисунок 42 — Создание кат-сцены с помощью Level Sequence

Эти ролики рендерились, монтировались, если это необходимо и воспроизводились с помощью интерфейса. Персонаж входил в зону триггера, после чего запускался пользовательский интерфейс и воспроизводилось видео, по его окончании интерфейс закрывался, а игра продолжалась. Такой способ оказался удобнее, так как не пришлось создавать лишние Blueprint для второстепенного персонажа и монстров.

2.3.11 Тестирование

Данный этап протекал на протяжении всего процесса разработки, так как каждая новая функция тестировалась после её добавления и при возникновении ошибок Blueprint исправлялись.

Данные тесты помогли определиться с методом патрулирования у врага. Изначально враги перемещались от одной точки к другой, но данный метод работал некорректно и иногда враги не доходили до второй точки из-за сложных маршрутов или застревали на какой-то из точек. Смена метода на случайное перемещение в пределах радиуса показала себя как более жизнеспособная, поэтому после повторных тестов было решено оставить её.

Так же данные тесты помогли определиться с методом стрельбы, показали, что важные объекты стоит подсветить, так как ключи сложно разглядеть в темноте.

2.3.12 Запаковка игры

Финальный этап в разработке — это запаковка продукта. В Unreal Engine 4 продукт можно запаковать для нескольких платформ: Android, HTML5, Windows x32 и x64, Linux, iOS и tvOS. Так как игра предназначена для компьютеров, а по статистике большинство пользователей интернет-магазина Steam (потенциальные покупатели) имеют устройства с операционной системой Windows x64, то для запаковки игры выбран именно этот формат. Перед тем как начать запаковку, необходимо в настройках указать стартовый уровень, по умолчанию в настройках стоит пустой уровень, удалить из контента лишние папки, если таковые имеются, чтобы не увеличивать размер игры. Запаковка будет проходить в фоновом режиме, по окончании раздастся звуковой сигнал [30].

После этого игра будет представлять из себя приложение формата *.exe и несколько вспомогательных папок (рисунок 43).

| Имя | Дата изменения | Тип |
|----------------------------|------------------|--------------------|
| brotherly ties | 03.06.2018 15:00 | Папка с файлами |
| Engine | 03.06.2018 15:02 | Папка с файлами |
| brotherly ties | 03.06.2018 13:46 | Приложение |
| Manifest_NonUFSFiles_Win64 | 03.06.2018 13:48 | Текстовый докум... |

Рисунок 43 — Запакованная игра

После запаковки необходимо провести ещё несколько тестов, так как некоторые проблемы не обнаружить в движке.

После первого теста был обнаружен дефект с материалами травы и деревьев (рисунок 44). Материалы были утеряны. Проблема решилась обновлением материалов у данных моделей.



Рисунок 44 — Неисправность материалов

Так же после запаковки игру можно переносить на другие компьютеры с операционной системой Windows x64, чтобы проверить её работоспособность. На сторонних компьютерах игра запускается корректно, а значит можно её выпускать.

2.3.13 Выпуск игры

На данном этапе игру повторно запаковывают в режиме Shipping — режим для продажи. Добавить игре описание, добавить шифрование для сохранения контента от копирования, сжать некоторые файлы или использовать средства оптимизации, если тесты показали, что в этом есть необходимость. После финальной запаковки игру можно начать распространять.

2.4 Системные требования игры Brotherly ties

Рекомендуемые системные требования необходимые для запуска игры указаны в таблице 5.

Таблица 5 — Рекомендуемые системные требования игры

| Название | Значение |
|----------------------|---|
| Операционная система | Windows Vista/7/8/10 (только 64-битные) |
| Процессор | Intel Core i7-4720HQ CPU 2.6 ГГц |
| Видеокарта | NVIDIA GeForce 940M |
| Оперативная память | 8 Гб |
| Место на диске | 2,85 Гб |

2.5 Калькуляция проекта

В ходе выполнения выпускной квалификационной работы было использовано 5 программ: Archicad 21, Unreal Engine 4.17.2, 3Ds Max 2016, Adobe Photoshop CC, Windows Киностудия.

В игре реализовано 3 уровня.

В программе 3Ds Max проработано и сконвертировано более 190 3D моделей.

В программе Archicad 21 создана полноценная квартира и трёхэтажное здание.

В программе Unreal Engine 4 создано:

- 157 материалов;
- 58 Blueprint Файлов, 2 из которых анимационные и 5 Blueprint файлов уровней;
- 22 пользовательских интерфейса;
- 5 кат-сцен;
- 11 ассетов для настройки аудио файлов;
- 5 карт (3 карты уровней, карта главного меню, карта титров).

В программе Adobe Photoshop CC обработано и создано 233 текстуры.

В игре задействовано 6 анимаций врага, 11 анимаций главного героя, 6 анимаций второстепенного персонажа (использовались для производства кат-сцен).

Всего запаковок игры было 10, суммарное время запаковок 8 часов.

ЗАКЛЮЧЕНИЕ

Темпы развития рынка игровой индустрии наглядно показывают актуальность тематики данной работы. Видеоигры надежно закрепились в жизни современного человека будь то крупные проекты для компьютеров или небольшие головоломки для смартфонов. Благодаря доступности игровых движков и обилию информации в Интернете любой желающий может начать разрабатывать видеоигры, что только поспособствует развитию данного сегмента рынка цифрового контента.

В рамках выпускной квалификационной работы была разработана 3D игра в жанре survival horror с видом от первого лица на движке Unreal Engine 4. Помимо движка в работе были использованы программы Archicad 21 для моделирования зданий, используемых в локациях игры, 3Ds Max для доработки и конвертирования 3D моделей, Adobe Photoshop CC для создания карт нормалей текстур и Windows Киностудия для редактирования кат-сцен.

В результате работы выполнены следующие задачи:

- исследованы литературные и интернет-источники по вопросам разработки видеоигр, написания сценария;
- создан концепт игры, выбран жанр, вид;
- написан сценарий игры;
- проанализированы популярные игры в жанре survival horror, выпущенные за последние 10 лет;
- изучена документация, описывающая методы работы с движком Unreal Engine 4;
- полностью реализован запланированный функционал проекта в движке Unreal Engine 4.

Таким образом цель работы достигнута, а задачи полностью выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. «10 способов напугать» или «Как отличить клише от новаторства» [Электронный ресурс]. — Режим доступа: <http://mmoboom.ru/blog/76/post/5884-10-sposobov-napugat-ili-kak-otlichit-klishe-ot-novatorstva> (дата обращения: 23.11.2017).
2. 2017 Video Game Trends and Statistics — Who's Playing What and Why? [Электронный ресурс]. — Режим доступа: <https://www.bigfishgames.com/blog/2017-video-game-trends-and-statistics-whos-playing-what-and-why/> (дата обращения: 28.05.2018).
3. Бесплатный кембриджский словарь и тезаурус по английскому языку [Электронный ресурс]. — Режим доступа: <https://dictionary.cambridge.org/ru/> (дата обращения: 15.05.2018).
4. Введение в геймдизайн: Основные понятия и принципы проектирования игр [Электронный ресурс]. — Режим доступа: <https://vc.ru/10495-gamedev-challenges> (дата обращения: 20.05.2018).
5. Игромания [Электронный ресурс]. — Режим доступа: <https://www.igromania.ru/> (дата обращения: 16.05.2018).
6. Канобу [Электронный ресурс]. — Режим доступа: <https://kanobu.ru/> (дата обращения: 16.05.2018).
7. Компьютерные игры как искусство [Электронный ресурс]. — Режим доступа: <http://gamesisart.ru/> (дата обращения: 10.11.2017).
8. Общий толковый словарь Русского языка [Электронный ресурс]. — Режим доступа: <http://tolslovar.ru/j145.html> (дата обращения: 10.02.2018).
9. Основы сценария и диздока [Электронный ресурс]. — Режим доступа: https://gcup.ru/publ/gamedev/osnovy_scenarija_i_dizdoka/1-1-0-229 (дата обращения: 05.02.2018).

10. Почему нам нравится испытывать страх [Электронный ресурс]. — Режим доступа: <https://med.vesti.ru/articles/psihologiya/pochemu-nam-nravitsya-ispytyvat-strah/> (дата обращения: 20.05.2018).

11. Приемы при проектировании архитектуры игр [Электронный ресурс]. — Режим доступа: <https://habr.com/post/255561/> (дата обращения: 16.04.2018).

12. Психотипы Бартла и балансировка аудитории [Электронный ресурс]. — Режим доступа: <https://habr.com/company/mailru/blog/263839/> (дата обращения: 05.04.2018).

13. Разработка игр — это просто: 12 этапов изучения геймдева [Электронный ресурс]. — Режим доступа: <https://proglib.io/p/gamedev-study-guide/> (дата обращения: 10.11.2017).

14. Русскоязычное сообщество Unreal Engine 4 [Электронный ресурс]. — Режим доступа: <https://uengine.ru/> (дата обращения: 13.05.2018).

15. Семь этапов создания игры: от концепта до релиза [Электронный ресурс]. — Режим доступа: <https://habr.com/company/miip/blog/308286/> (дата обращения: 17.04.2018).

16. Сценарий компьютерной игры: формат? [Электронный ресурс]. — Режим доступа: <http://narratorika.com/plotformat/> (дата обращения: 05.02.2018).

17. Четыре способа сделать из игры хороший Survival Horror [Электронный ресурс]. — Режим доступа: <http://gamehaze.ru/make-the-atmosfear/> (дата обращения: 21.11.2017).

18. App2Top [Электронный ресурс]. — Режим доступа: <http://app2top.ru/> (дата обращения: 30.05.2018).

19. Chris SolarSKI. The Aesthetics of Game Art and Game Design [Электронный ресурс]. — Режим доступа: https://www.gamasutra.com/view/feature/185676/the_aesthetics_of_game_art_and_.php?print=1 (дата обращения: 05.04.2018).

20. Greg Costikyan. I Have No Words & I Must Design: Toward a Critical Vocabulary for Games [Электронный ресурс]. — Режим доступа: <http://www.costik.com/nowords2002.pdf> (дата обращения: 05.04.2018).

21. JSON.TV [Электронный ресурс]. — Режим доступа: <http://json.tv/> (дата обращения: 30.05.2018).

22. StopGame [Электронный ресурс]. — Режим доступа: <https://stopgame.ru/> (дата обращения: 16.05.2018).

23. Technical Game Design [Электронный ресурс]. — Режим доступа: <http://technicalgamedesign.blogspot.com/2012/07/what-is-technical-game-designer.html> (дата обращения: 16.04.2018).

24. Unreal Engine 4 Documentation [Электронный ресурс]. — Режим доступа: <https://docs.unrealengine.com/en-us/> (дата обращения: 13.05.2018).

25. YouTube канал Гоша Дударь [Электронный ресурс]. — Режим доступа: <https://www.youtube.com/channel/UCvuY904el7JvBIPbdqbfguw> (дата обращения: 01.05.2018).

26. YouTube канал MrFantasticGhost [Электронный ресурс]. — Режим доступа: <https://www.youtube.com/channel/UC EzysQNG2c8JwZU3zcVrxFa> (дата обращения: 01.05.2018).

27. YouTube канал Titanic Games [Электронный ресурс]. — Режим доступа: https://www.youtube.com/channel/UCdoWGpMQK_L29bWxoDeIItw (дата обращения: 10.05.2018).

28. YouTube канал Virtus Learning Hub / Creative Tutorials [Электронный ресурс]. — Режим доступа: <https://www.youtube.com/channel/UCzeYJAUGSE-mqzKtit7m9g> (дата обращения: 10.04.2018).

29. YouTube канал Unreal Engine [Электронный ресурс]. — Режим доступа: <https://www.youtube.com/channel/UCBobmJyzsJ6Ll7UbfhI4iwQ> (дата обращения: 15.05.2018).

30. YouTube канал Unreal Engine Rus [Электронный ресурс]. — Режим доступа: <https://www.youtube.com/channel/UCLbkGIcYJxxL0tciH9RVebg> (дата обращения: 15.05.2018).

ПРИЛОЖЕНИЕ А

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»

Институт инженерно-педагогического образования
Кафедра информационных систем и технологий
Направление подготовки 09.03.02 Информационные системы и технологии
Профиль подготовки «Информационные технологии в медиаиндустрии»

УТВЕРЖДАЮ

Заведующий кафедрой

Н.С. Толстова

подпись

и.о. фамилия

« ____ » _____ 2018 г.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы бакалавра

студента (ки) 4 курса группы ИТм-402
Щепиной Татьяны Алексеевны
фамилия, имя, отчество полностью

1. Тема Художественно-техническое проектирование компьютерной 3D игры в жанре survival horror

утверждена распоряжением по институту от « ____ » _____ 20 ____ г. № ____

2. Руководитель Чернякова Татьяна Викторовна
фамилия, имя, отчество полностью

| | | | |
|---------------------------------|----------------------------------|---------------------------------------|------------------------------|
| <u>доцент</u> ученая степень | <u>к.пед.н.</u> ученое звание | <u>доцент кафедры ИС</u> должность | <u>РГППУ</u> место работы |
|---------------------------------|----------------------------------|---------------------------------------|------------------------------|

3. Место преддипломной практики РГППУ, ИПО, кафедра информационных систем и технологий

4. Исходные данные к ВКР Статьи на тему создания видеоигр, написания сценариев, 3D модели (пистолет, монстр, мусор и т.д.), звуковые эффекты, YouTube каналы, Посвященные работе с игровым движком Unreal Engine 4.

5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов)
Разработка концепта игры, написание сценария, разработка моделей главного и второстепенного персонажей, разработка модели врага, моделирование локаций, подбор текстур и материалов, построение уровней, подбор головоломок, программирование

основных элементов игры, создание внутриигровых роликов, тестирование, упаковка, выпуск.

6. Перечень демонстрационных материалов презентация выполненная в MS Power Point, видеоролик с игровым процессом, компьютерная игра

7. Календарный план выполнения выпускной квалификационной работы

| № п/п | Наименование этапа дипломной работы | Срок выполнения этапа | Процент выполнения ВКР | Отметка руководителя о выполнении |
|-------|---|-----------------------|------------------------|-----------------------------------|
| 1 | Сбор информации по выпускной квалификационной работе | 23.04.2018 | 10% | подпись |
| 2 | Выполнение работ по разрабатываемым вопросам и их изложение в пояснительной записке: | 03.05.2018 | 60% | подпись |
| 2.1 | Анализ существующих разработок, выявление их достоинств, недостатков и функциональных элементов | 03.05.2018 | 5% | подпись |
| 2.2 | Анализ средств разработки | 06.05.2018 | 5% | подпись |
| 2.3 | Определение функционала и формулировка требований к разрабатываемому продукту | 10.05.2018 | 5% | подпись |
| 2.4 | Разработка игры | 12.05.2018 | 25% | подпись |
| 2.5 | Тестирование и упаковка продукта | 26.05.2018 | 15% | подпись |
| 3 | Оформление текстовой части ВКР | 27.05.2018 | 10% | подпись |
| 4 | Выполнение демонстрационных материалов к ВКР | 01.06.2018 | 10% | подпись |
| 5 | Нормоконтроль | 06.06.2018 | 5% | подпись |
| 6 | Подготовка доклада к защите в ГЭК | 13.06.2018 | 5% | подпись |

8. Консультанты по разделам выпускной квалификационной работы

| Наименование раздела | Консультант | Задание выдал | | Задание принял | |
|----------------------|-------------|---------------|------|----------------|------|
| | | подпись | дата | подпись | дата |

Руководитель _____
подпись дата

Задание получил _____
подпись студента дата

9. Дипломная работа и все материалы проанализированы.

Считаю возможным допустить _____ Щепину Т.А. к защите выпускной квалификационной работы в государственной экзаменационной комиссии.

Руководитель _____
подпись дата

10. Допустить _____ Щепину Т.А. к защите выпускной квалификационной работы
фамилия и. о. студента

в государственной экзаменационной комиссии (протокол заседания кафедры от «__» _____ 20__ г., № _____)

Заведующий кафедрой _____
подпись дата

ПРИЛОЖЕНИЕ Б

Сценарий игры *Brotherly ties* (рус. Братские узы).

Действующие лица: Мэтт — главный герой мужчина 25 лет, Стэн — младший брат героя пропал несколько лет назад (погиб), антропоморфный монстр, плод воображения главного героя.

Место действия: окрестности Нью-Йорка, Соединённые Штаты Америки.

| № уровня | Описание локации | Начало уровня | Ключевые события задания | Конец уровня |
|----------|------------------------|---|--|---|
| 1 | Лабиринт | Кат-сцена: главный герой видит убегающего брата | Найти выход из лабиринта попутно уничтожая встретившихся монстров | Кат-сцена: брата съедают монстры, персонаж перемещается на второй уровень |
| 2 | Двухкомнатная квартира | Кат-сцена: главный герой слышит стук в дверь, от которого просыпается | Проверить, кто постучал в дверь, обнаружить записку, найти на локации таблетки, фонари и пистолет (код от сейфа с пистолетом в записной книжке) | Подойти к двери, персонаж перемещается на третий уровень |
| 3 | Заброшенный приют | Кат-сцена: главный герой осматривает приют с улицы | Найти ключ от главной двери приюта, добраться до спальни мальчиков на третьем этаже, для этого решить головоломку на первом этаже и найти ключ от второго, решить головоломку на втором этаже и найти ключ от третьего, решить головоломку на третьем этаже и попасть в комнату мальчиков попутно уничтожая встретившихся монстров | Кат-сцена: главный герой встречается с братом, конец игры, титры |

ПРИЛОЖЕНИЕ В

Скачанные модели

