

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление 09.04.04 – Программная инженерия
Магистерская программа Управление разработкой программного обеспечения

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой

_____ А.В. Бушманов

« _____ » _____ 2017г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: Компьютерная реконструкция обороны Албазинского острога

Исполнитель магистрант группы 557-ом	_____	А.В. Вишневский
	(подпись, дата)	
Руководитель профессор, д-р техн. наук	_____	И.Е. Еремин
	(подпись, дата)	
Руководитель магистерской программы профессор, д-р техн. наук	_____	Е.Л. Еремин
	(подпись, дата)	
Нормоконтроль доцент, канд. физ.-мат. наук	_____	В.В. Еремина
	(подпись, дата)	
Рецензент ведущий науч. сотрудник, канд. техн. наук	_____	О.Г. Какаулин
	(подпись, дата)	
Рецензент доцент, канд. техн. наук	_____	А.Н. Семочкин
	(подпись, дата)	

Благовещенск 2017

РЕФЕРАТ

Магистерская диссертация содержит – 74 с., 75 рисунка, 1 приложение, 38 источников. Структура диссертации состоит из трех разделов: «Компьютерная историческая реконструкция», «Программное и техническое обеспечение», «Технологии реализации компьютерной симуляции».

КОМПЬЮТЕРНАЯ ИСТОРИЧЕСКАЯ РЕКОНСТРУКЦИЯ, 3D МОДЕЛИРОВАНИЕ, ПРОТОТИПИРОВАНИЕ, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, ТРЕХМЕРНАЯ ВИЗУАЛИЗАЦИЯ.

Актуальность исследования состоит в том, что в настоящее время намечается тенденция к взаимодействию и интеграции различных направлений исследований, также все более расширяется применение визуальных информационных технологий, которые применяются практически во всех сферах человеческой деятельности. Таким образом, компьютерные исторические реконструкции являются достаточно актуальной задачей.

Целью работы послужило создание программной среды, предназначенного для визуализации рассматриваемого исторического события.

Научная новизна основных результатов работы состоит в следующем: разработанный программный продукт реализует визуализацию воссозданного окружающего мира исторического конфликта, используя информационные технологии моделирования и визуализации, что существенно отличается от традиционного подхода к исторической реконструкции.

Практическая ценность диссертационной работы: акт о внедрении..

Основные результаты проведенного квалификационного исследования опубликованы в двух работах, среди которых одна журнальная статья, вторая – тезисы доклада на научной конференции.

					ВКР.155506.09.04.04.ПЗ			
Изм.	Лист	№ докум.	Подп.	Дата	КОМПЬЮТЕРНАЯ РЕКОНСТРУКЦИЯ ОБОРОНЫ АЛБАЗИНСКОГО ОСТРОГА	Лит.	Лист	Листов
Разраб.	Вишневский.А.В.					У	2	75
Пров.	Еремин И.Е.							
Н. контр.	Ермина В.В.							
Зав. каф.	Бушманов А.В.							
						АмГУ кафедра ИУС		

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 КОМПЬЮТЕРНАЯ ИСТОРИЧЕСКАЯ РЕКОНСТРУКЦИЯ	6
1.1 Общее понятие исторической реконструкции	6
1.2 Обзор существующих подходов	9
1.3 Историческая справка	15
2 ПРОГРАММНОЕ И ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	23
2.1 Основная среда разработки	23
2.2 Динамические модели и материалы	27
2.3 Физическая модель	30
3 ТЕХНОЛОГИИ РЕАЛИЗАЦИИ КОМПЬЮТЕРНОЙ СИМУЛЯЦИИ	31
3.1 Создание прототипа уровня	31
3.2 Реализация статической модели	36
3.2.1 Описание этапов моделирования	36
3.2.2 Создание трехмерной модели	36
3.2.3 Реализация текстур и материала модели	39
3.3 Реализация системы смены погодных эффектов	44
3.4 Реализация искусственного интеллекта персонажей	51
3.4.1. Общая теория игрового искусственного интеллекта	51
3.4.2 Постановка задач для искусственного интеллекта	53
3.4.3 Реализация искусственного интеллекта в среде Unreal Engine	54
3.5 Интерактивная часть реконструкции	63
Заключение	70
Библиографический список	71
Приложение А Изображения компьютерной реконструкции	75

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей дипломной работе использованы ссылки на следующие стандарты и нормативные документы:

ГОСТ 19.001-77 ЕСПД	Общие положения;
ГОСТ 19.004-80 ЕСПД	Термины и определения;
ГОСТ 19.101-77 ЕСПД	Виды программ и программных документов;
ГОСТ 19.102-77 ЕСПД	Стадии разработки;
ГОСТ 19.103-77 ЕСПД	Обозначение программ и программных документов;
ГОСТ 19.104-78 ЕСПД	Основные надписи;

ВВЕДЕНИЕ

Благодаря компьютерным технологиям появилась возможность создания трехмерных, интерактивных объектов, что позволяет использовать их в широком спектре задач, таких как историческая реконструкция. На данный момент она остается достаточно актуальной темой, поскольку благодаря современным типам медиа возможно использовать иные способы передачи информации в обучающем ключе, а также развлекательном, делая историю более интерактивной, давая возможность пользователю принять в ней непосредственное участие.

В последние несколько десятилетий множество стран охватило движение исторической компьютерной реконструкции. По различным данным, которые представлены на множестве тематических сайтах, посвященных реконструкциям, в настоящее время реализованы более 200 групп исторических реконструкций.

Большой интерес среди историков вызывают военно-исторические реконструкции. Большинство групп, занимающихся данной тематикой, рассматривают их как частный вид реконструкции. Данные работы не только обогащают событийный календарь территории, в частности Амурской области, но и формируют культурную и историческую составляющую ее имиджа.

Сегодня эти мероприятия зачастую становятся основой для развития событийного туризма в том или ином регионе, привлекая к себе внимание тысяч людей во всем мире, и в свою очередь, способствуя экономическому развитию регионов.

Объектом проводимых исследований является изучение исторических событий обороны Албазинского острога, с целью реализации реалистичной компьютерной реконструкции. В данной работе описываются основные методы и этапы разработки моделей окружения, системы погодных эффектов, компьютерного искусственного интеллекта, а также интерактивного модуля симуляции.

1 КОМПЬЮТЕРНАЯ ИСТОРИЧЕСКАЯ РЕКОНСТРУКЦИЯ

1.1 Общее понятие исторической реконструкции

Идея реконструкции (полной или частичной) исторического объекта, события или памятника культуры, разумеется, не нова. Большое количество реконструкций было сделано художниками эпохи Возрождения, например, художественная реконструкция Вавилонской башни Питера Брейгеля Старшего. В XVII – XIX веках, в связи с бурным развитием научных исследований и подходов, стали зарождаться попытки создания первых научных реконструкций. В конце этого периода традиционная тематика, связанная с легендарной историей, была дополнена сюжетами, связанными с вымершими животными и ископаемыми формами человека. Большинство из них, естественно, сегодня вызывают лишь улыбку (за исключением шедевров мирового искусства).

В прошлом веке методика создания разного рода реконструкций претерпела значительные изменения. Эпохальные открытия XIX – начала XX века обеспечили массовый интерес к истории и истории культуры, а появление принципиально новых способов создания изображения, таких как фотография и киносъемка, издание книг массовыми тиражами, возможности цветной высококачественной печати, сделали свое дело с технической точки зрения. Естественное стремление к максимальной точности и правдоподобию реконструкций (в сочетании с возможностью использования достижений прогресса) привело к появлению новых методик и приемов (например, использование фотографий специально выполненных макетов). Традиционные материалы (акварель, гуашь, тушь) были блестяще использованы чешским художником Захарием Бурианом при создании (под руководством ученого-палеонтолога Й. Августы) целой серии книг альбомов, рассказывающих о происхождении и развитии жизни на Земле. В свое время они стали бестселлерами. Вообще говоря, научно-(или псевдонаучно-) популярные книги, чаще всего связанные с историей и историей культуры, приобрели широчайшую популярность в XX веке, превратившись в специальную индустрию и став, с одной стороны, важным фактором, способствующим популяризации истории, с другой – са-

					ВКР.155506.09.04.04.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		6

мостоятельным (иногда – скандальным) специфическим элементом массовой культуры.

Принципиально новые возможности в задачу популяризации истории (и, в том числе, зримого воссоздания нашего прошлого) принес с собой появившийся в конце позапрошлого века кинематограф. Художники кино многократно воссоздавали историко-культурное пространство различных эпох и цивилизаций (реально существовавших, легендарных, или полностью вымышленных). Некоторые научно-(или псевдонаучно-) популярные фильмы стали заметными событиями в области массовой культуры, собирая аудиторию сравнимую с аудиторией фильмов экшн.

Среди множества технологий компьютерной графики, появившихся в последние десятилетия, можно выделить одну, оказавшую огромное влияние на процесс создания иллюстративного ряда для Интернета и современных электронных СМИ. Мы имеем в виду комплекс инструментов, предназначенных для создания трехмерной анимированной компьютерной графики (т. н. 3D-графика). Соответствующие приложения (самые популярные – 3Ds Max и Maya (отдельные версии приложений имели названия, незначительно отличающиеся от приведенных)) были разработаны в конце 80-х годов прошлого века.

Новые технические возможности появились в последние десятилетия. С одной стороны, они связаны с появлением различных цифровых методов создания изображения, с другой – с появлением принципиально новых средств телекоммуникаций, развитием традиционных и возникновением новых средств массовой информации. В то же время, во многих странах развитие инфраструктуры, в том числе транспортной системы в сочетании с некоторой стабилизацией внутривнутриполитической ситуации привело к развитию массового туризма. Памятники истории, археологии и культуры, которые 150 лет назад могли увидеть лишь некоторые специалисты и искатели приключений, сегодня стали обычным элементом семейного туризма. Естественно, это привело лишь к усилению интереса к нашему прошлому.

Трехмерную компьютерную графику можно упрощенно представить себе

					<i>ВКР.155506.09.04.04.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		7

следующим образом. Прежде всего художник создает т. н. *модели*. Моделирование представляет собой процесс создания так называемых *сцен*, состоящих из различных объектов (своего рода компьютерной скульптуры, виртуальных архитектурных объектов и т. д.), именуемых *моделями*, а также из виртуальных источников света, видеокамер и, иногда, некоторых других специальных объектов. Моделям может быть *назначен материал*. Это значит, что с каждым объектом сцены (прежде всего моделью) связывается некоторый пакет информации, определяющий свойство его поверхности - цвет, фактуру, (объект можно каким-то образом «покрасить»), прозрачность, возможность отражать другие объекты и т.д. Частью материала может быть графическое изображение в оцифрованном виде, которое с помощью специальных инструментов (компьютерных команд) «накладывается» на поверхность модели. Сцена может быть *визуализирована*, т. е, превращена в графический файл практически любого формата, и, следовательно, может быть представлена в распечатанном виде или просматриваться на каком-либо мониторе. С другой стороны, имея такую сцену, мы можем менять во времени положение объектов в пространстве, а также практически любые их свойства (выраженные в числовом виде). В результате мы получим т. н. *анимированную сцену*, которая может быть визуализирована в каком-либо из форматов видео и просматриваться зрителем как обычный фильм.

Появление такого инструментария дало возможность художникам, в том числе, создавать исторические реконструкции различной степени условности, от схематичных до т. н. «реалистических». Справедливости ради заметим, что реконструкции не являются, конечно, ни единственной, ни даже основной сферой применения этой техники. Наибольшее влияние на сегодняшний день она оказала на кинематограф, сделав возможным появление множества полнометражных художественных анимационных фильмов, таких как всем известные «Шрек», «Мадагаскар», «Ледниковый период» и «Наверх». С помощью инструментов 3D графики создаются самые разнообразные визуальные эффекты для игрового кино. Кроме того, существует множество примеров успешного применения 3D графики, без сомнения, очень интересной художественной техники вообще, как в

					ВКР.155506.09.04.04.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		8

различных видах искусства, так и в других областях (в инженерном и архитектурном проектировании и т. д.). Тем не менее, в жанре исторической (и не только) реконструкции за последние годы были достигнуты впечатляющие успехи.

Сегодня ни один серьезный проект в области исторического документального кино (например, «Древние империи» и другие сериалы BBC и других компаний) не обходится без выполненных в самой разнообразной стилистике компьютерных трехмерных реконструкций. Реконструкции, используемые в этих и подобных телефильмах, играют в них различную роль. Они могут быть информативны, т.е. давать зрителю подробное представление об исчезнувших памятниках и целых исторических ландшафтах, но могут также играть и сугубо декоративную роль. Широко известна масштабная реконструкция античного Рима, выполненная с высочайшей степенью детализации и реализованная как в электронной, так и в печатной версии. Великолепные реконструкции ископаемых животным обеспечили широчайшую популярность сериалам «Прогулки с динозаврами» и «Прогулки с чудовищами» телекомпании BBC. Различные трехмерные реконструкции публикуются на интернет-сайтах, как предназначенных для профессионалов, так и научно-популярных.

Другая область применения 3D графики – это возможность использовать реконструкции для публикации различных материалов в области истории, археологии и истории культуры, для введения их в научный обиход. Трехмерная компьютерная реконструкция сегодня – это, например, неотъемлемая часть публикации материалов археологических исследований. Несмотря на то, что сегодня эта деятельность не получила еще достаточного развития (может быть, просто в силу недостаточного финансирования науки), некоторые авторитетные ученые-гуманитарии уже сегодня считают, что 3D графика в XXI веке станет основной формой публикации материалов в соответствующих областях науки.

1.2 Обзор существующих подходов

Широкое и быстрое развитие компьютерных технологий способствовали улучшению всех возможных видов медиа, от фильмов до компьютерных игр. И если фильмы способны демонстрировать события прошлого в строгом ключе по-

					<i>ВКР.155506.09.04.04.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		9

вестования режиссера, то компьютерные исторические реконструкции способны внести интерактивный элемент для более детального изучения истории, а также, если это будет позволять реконструкция, изменить ход развития события для изучения альтернативных вариантов развития истории.

При формировании среды, в которой происходит действие компьютерной симуляции, делается попытка смоделировать ситуацию, царившую на тот момент в конкретном государстве. Игроку предоставляется возможность при помощи определенных инструментов участвовать в реконструкции, будь это дипломатия или военные действия. Такие исторические игры и фильмы обычно выделяются высоким уровнем детализации окружения: предметов обихода, одежды, оружия, а также природных эффектов, если это будет важно для корректной передачи истории. Стоит уточнить, что некоторые продукты могут рассказывать выдуманную историю, основываясь на исторической хронологии событий. К данным продуктам можно отнести серию игр *Assassin's Creed*, а также часть из них может быть основана полностью на событиях альтернативной истории, как серия *Red Alert*.

Assassin's Creed это серия мультиплатформенных компьютерных игр в жанре приключенческого боевика, разработанная и изданная компанией *Ubisoft*. Первая часть данного сериала была выпущена 14 ноября 2007 года. На данный момент было разработано и выпущено 9 частей основного сериала. Также для широкого продвижения своего продукта, студия *Ubisoft* использует и другие виды медиа: фильмы, телевидение, книги, комиксы.

Основной сюжет основан на многовековой войне между тамплиерами и асасинами. Благодаря введению фантастического элемента – машины Анимус, которая позволяет считывать память предков из ДНК, игрок вместе с главными героями игры может переживать события далекого прошлого. Благодаря данному сюжетному ходу, игроки могут принять участия в событиях 1191 года в эпоху Крестовых походов, в эпоху Ренессанса 1476-1499 годов в Италии, в период 1754-1783 годов времен Войны за независимость в Америке, в период Великой французской революции, а также в Викторианскую эпоху Лондона XIX века.

					<i>ВКР.155506.09.04.04.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		10



Рисунок 1 – Скриншот из проекта Assassin's Creed.



Рисунок 2 – Скриншот из проекта Assassin's Creed 2



Рисунок 3 – Скриншот из проекта Assassin's Creed 3

Правда о девятой роте – российская компьютерная игра в так называемом «документальном» жанре. Данный проект представляет собой интерактивную реконструкцию исторического боя, происходившего в ночь с 7 на 8 января 1988 года в южном Афганистане между бойцами девятой роты 345-го отдельного парашютно-десантного полка и отрядами душманов. В ходе специального расследования, стали известны детали военной операции «Магистраль» и реальные обстоятельства боя на высоте 3234. Игра воссоздаёт ту кровопролитную схватку. Горный ландшафт был смоделирован на основе спутниковой съёмки, а детали произошедшего уточнены у самих бойцов. Игроку, оказавшемуся в центре событий, предстоит самому увидеть битву и повторить подвиг тридцати девяти советских десантников, державших оборону против нескольких сотен душманов. Благодаря жанровой принадлежности к шутеру от первого лица, игра предполагает последовательное перевоплощение то в одного, то в другого участника боя, что позволит создать представление о случившемся. Разработкой занималась студия Extreme Developers, продюсировала проект студия KranX Productions, а издала игру фирма «1С».



Рисунок 4 – Скриншот из проекта Правда от 9 роты

Call of Duty - компьютерная серия игр в жанре шутер от первого лица. Первая и вторая части сериала были посвящены событиям Второй мировой войны. Игра была разработана компанией Infinity Ward и издана компанией Activision 29 октября 2003 года для платформы PC. В игре представлено 4 кампании, охватывающие несколько периодов войны с 1942 по 1945 годы. В каждой кампании по несколько миссий.



Рисунок 8 – Скриншот из проекта Call of Duty

1.3 Историческая справка

Суть конфликта при Албазино состояла в том, что установившаяся в Китае с середины XVII века маньчжурская императорская династия Цин не признавала присоединения к России приамурских земель, которые считала своими родовыми владениями, но до того фактически не контролировала. В 1682 году отношения между Россией и Китаем обострились, оба государства принимали меры по усилению своего присутствия на Амуре. Россия учредила отдельное Албазинское воеводство, направив на Амур воеводу А. Толбузина с отрядом служилых людей, организовывались новые поселения в бассейне реки Зея. В сентябре 1682 года китайский император дал поручение сановникам Лантаню и Пэнчуню оценить воз-

возможности вытеснения русских из Приамурья.

В ноябре Лантань с небольшим конным отрядом побывал вблизи Албазина, деревянные укрепления которого он оценил, как слабые.

Император Канси и Военный совет в марте 1683 года издали ряд распоряжений по подготовке военных действий на Амуре. Лантань основал в качестве передовой крепости на правом берегу Амура крепость Ай-гун с земляными укреплениями. В Западной Сибири был набран отряд служилых людей для усиления гарнизона Албазина, но он двигался по сибирским рекам и волокам к Амуре очень медленно.

Непосредственные события осады Албазина можно разделить на два этапа:

1. 1685 год – первый этап борьбы между Русским царством и Цинским Китаем за главный русский опорный пункт на реке Амур.

В начале лета 1685 года цинская армия на кораблях речной флотилии двинулась от Айгуна вверх по Амуре. В Албазин было послано два казака из отряда Мыльника с указом Лифаньюаня (учреждения по связям с вассальными землями); от русских под угрозой смерти требовали немедленно уйти с Амура. 10 июня цинская флотилия появилась вблизи Албазина.

Два последующих дня цинские войска строили вокруг Албазина осадные сооружения, позиции для артиллерии. Напротив северной крепостной стены были установлены наиболее мощные «ломовые» пушки. Когда они открыли огонь, оказалось, что бревенчатые укрепления Албазина, предназначенные для защиты от туземных стрел, не выдерживают попаданий из тяжелых орудий. В результате вспыхнувших пожаров в Албазине сгорели хлебные амбары и церковь с колокольней. Была подбита одна из трех русских пушек. Жертвы артиллерийских обстрелов составляли около 100 человек.

В Нерчинске воевода Иван Власов подготовил отряд для похода на помощь Албазину (100 человек с 2 пушками). На помощь Албазину спешил и отряд из западносибирских острогов под командованием Афанасия Бейтона, однако в Забайкалье он оказался связан боями с монголами. Подкрепление опоздало. Ранним утром 16 июня цинские войска начали общий штурм. Защищавшие Албазин гар-

					<i>ВКР.155506.09.04.04.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		16

низон и люди «всякого звания» не давали маньчжурам преодолеть окружавшие крепость ров и вал и забраться на полуразрушенные укрепления. В 10 часов вечера маньчжуры отступили в свой лагерь. Лантань отдал приказ готовить новый штурм; маньчжуры заваливали крепостной ров хворостом, так что русские решили, что их готовятся сжечь вместе с городом. Отогнать противника от стен албазинцы не могли, так как у них подошли к концу запасы пороха.



Рисунок 9 – Иллюстрация воеводы Толбузина

Воевода Толбузин обратился к Лантаню с предложением вывести гарнизон и жителей из Албазина в Нерчинск. Цинское командование настаивало на уходе русских в Якутск, считая, что Нерчинск также находится на маньчжурских землях. Однако Толбузину удалось настоять на отступлении вверх по Амуру, что было для русских гораздо предпочтительнее, чем переход через горный хребет в Якутию. 26 июня 1685 года русские беспрепятственно покинули город и двинулись на запад. Лантань уничтожил строения Албазина и, считая свою миссию выполненной, отошел к Айгуну. Там был оставлен гарнизон в 500 человек, остальные цинские войска ушли по Сунгари на юг в Маньчжурию.

В конце июня в Нерчинске собрались и бывшие албазинцы, и опоздавшее к

ним подкрепление. Не желая себе «побежную славу из Албазина учинить», русские решили попытаться вернуть себе город. В середине июля из Нерчинска вниз по Амуру был послан русский разведывательный отряд. Узнав, что цинские войска покинули разрушенный Албазин и даже не сняли посева в окрестных деревнях, воевода Толбузин при поддержке нерчинского воеводы Власова немедленно организовал поход для восстановления главной русской крепости на Амуре. Заранее выслав вперед 200 конных казаков Бейтона, 27 августа 1685 года на пепелище Албазина прибыл на стругах сам Толбузин с войском из 514 служилых людей и 155 промысловиков и крестьян, которые до зимы отстроили заново город и несколько деревень. Таким образом, несмотря на военную победу Цинскому Китаю не удалось вытеснить русских из Приамурья, и в следующем году военные действия были продолжены новой обороной Албазина.

2. Осада Албазина 1686 года – второй этап борьбы за опорный пункт на Амуре.

Сразу после возвращения русских в августе 1685 года на место уничтоженного после первой обороны Албазина, началось строительство крепости, способной противостоять осадной цинской артиллерии. Вместо привычных для Сибири бревенчатых стен за глубоким рвом новый Албазинский острог окружили земляные валы, в центре которых находились засыпанные срубы. Толщина валов достигала 8,5 метров, высота была более 3 метров. На речной стороне для наблюдения построили традиционную бревенчатую башню. Для лучшего обстрела с укреплений их линия была сделана ломаной, с выступами бастионами.

17 апреля 1686 года цинский император Канси на аудиенции полководцу Лантаню дал указания по ведению военной кампании: взять Албазин, но на этот раз не разрушать его как в прошлом году, а укрепить как базу для дальнейшего наступления на Нерчинск. Для перевозки на Амур 5-тысячной цинской армии с артиллерией и провиантом потребовалось 150 речных судов.

7 июля 1686 года на Амуре у Албазина появился цинский флот; началась высадка войск на берег. Воевода Толбузин решил помешать ей и послал часть своих сил на вылазку. Атака казаков, поддержанная пальбой с крепостных валов,

					<i>ВКР.155506.09.04.04.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		18

внесла расстройство среди высаживавшихся цинских войск. Потребовалось личное вмешательство Лантаня, чтобы привести маньчжурские силы в порядок и отеснить русских обратно в крепость.

Лантань ожидал, как и в 1685 году, быстро сломить сопротивление защитников Албазина непрерывным артиллерийским обстрелом, но тот не давал результата; китайские ядра вязли в земляных валах. Однако от бомбардировки албазинцы несли потери, всего от обстрелов в городе погибло за лето 40 человек. Среди первых из них был воевода Алексей Толбузин. 12 июля он наблюдал за противником из башни, когда влетевшее ядро оторвало ему ногу; через четыре дня воевода скончался. Командование гарнизоном перешло к Афанасию Бейтону. В ночь на 14 июля цинские войска устроили общий штурм с приречной и северной стороны, однако албазинцы не только отбили приступ, но и сами сделали вылазку, дойдя до вражеского лагеря у речного берега.

Лантань принял решение готовиться к долгой осаде. Для цинского войска вокруг Албазина было устроено четыре осадных городка из землянок. Русскую крепость на расстоянии 400 м со всех сторон окружили траншеями и валами. За валами были устроены «раскаты» — насыпные возвышения, на которых устанавливались тяжелые орудия для обстрела внутрикрепостного пространства. Всего у маньчжур было 15 тяжелых «ломовых» орудий, способных простреливать весь Албазин. Защитники вынуждены были укрываться от их огня в подземных убежищах, все строения в городе были разрушены.

В октябре 1686 года маньчжуры устроили последний и самый ожесточенный штурм. К крепости двигали два «дровяных» вала, чтобы засыпать ими ров и поставить вровень с валами. Русские устроили вылазки и подожгли один вал, второй был взорван при помощи подкопа. Часть дров досталось русским, которые использовали их для обогрева. В результате боев к началу зимы погибло около ста русских. К декабрю в Албазине оставалось в живых всего 150 казаков, из которых только 45 могли нести караульные службу, остальные «оцынжали» и лежали больными. Общие потери цинских войск оцениваются в 2,5 тыс. человек.

					<i>ВКР.155506.09.04.04.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		19

на была пройти у Албазина, в случае отказа — достичь перемирия, после чего начать военные действия.

Спустя два месяца Головин прибыл в Тобольск, где набрал крестьян и ссыльных. В конце сентября того же года посольство добралось до Рыбного острога на Ангаре, где вынуждено было остаться на зиму. В это время пришло известие о согласии китайского императора заключить мир. В сентябре 1687 года Головин достиг Удинского острога, откуда через русского гонца просил китайскую сторону начать переговоры. В январе 1688 года монгольское войско начало войну, безуспешно осадило Селенгинский и Удинский остроги. Головин получил из Москвы разрешение оставить Албазин в случае заключения мира. В случае срыва переговоров Головин должен был передать китайской стороне, что «великие государи их царское величество изволяют быть границею Амуру реке и с сего времени их царского величества русские служилые и ясачные люди ни для каких промыслов за Амур реку ходить не будут, а на сю албазинскую сторону их китайских людей не пустят». Судя по присылаемым из Москвы инструкциям, царское правительство плохо представляло себе сложность ситуации на восточной границе, угрожая смертной казнью в случае ссоры с китайской стороной. Осенью 1688 года Головин прибыл в Нерчинск, где китайский император пожелал провести переговоры.

Ранее китайское правительство не вело переговоры с иностранными государствами. До столкновения с русскими император Канси рассчитывал на лёгкое овладение Приамурьем из-за малочисленности русских. В мае 1688 года было сформирована китайская делегация в составе трёх послов: князя Сонготу, дяди императора Тун Гуегана и Лантаня (руководил осадами Албазина); а также свиты из чиновников и военачальников, и иезуитов-переводчиков латыни — испанца Томаса Перейры и француза Жана Франсуа Жербильона. Переводчики к русским были настроены враждебно.

Император не спешил с заключением договора, пока не получил известие о возможном союзе русских с Джунгарским ханством. 20 июля 1689 года китайские послы на 76 военных судах доплыли до Нерчинска. В то же время к Нерчинску

					<i>ВКР.155506.09.04.04.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		21

подступила цинская армия с обозом в пять тысяч лошадей и четырёх тысяч верблюдов. Армия и флот насчитывали около 15 тысяч воинов. В Нерчинском гарнизоне находилось около 600 русских. 9 августа в Нерчинск прибыл Головин.

27 августа в 50 саженьях от надолбов Нерчинска состоялись последние переговоры. Здесь были поставлены шатры сторон. Переводчики зачитали текст договора. Русским, не имевшим переводчика маньчжурского языка, пришлось поверить иезуитам, уверявшим в аутентичности маньчжурского экземпляра договора. Здесь же было принято решение совместно подписать экземпляры договора на латыни, скрепив их печатями обоих государств. Головин и Соготу принесли клятву соблюдать заключённый договор, обменялись экземплярами и обняли друг друга. 30 августа китайские послы вместе с армией и флотом отправились в Пекин. Головин через Иркутск направился в Москву. Успевший застать русских по-слов голец сообщил им, что император был очень рад, узнав о заключении договора.

Позднее китайское правительство было разочаровано заключением договора и требовало от российских властей разграничения территорий в Монголии и походу Станового хребта до реки Уды. Российское правительство по приезде Головина было недовольно потерей Амура, участник посольства Степан Коровин был подвергнут допросу

					<i>ВКР.155506.09.04.04.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		22

2 ПРОГРАММНОЕ И ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

2.1 Основная среда разработки

Важным фактором при реализации компьютерной реконструкции является выбор среды, в которой будут происходить все основные действия симуляции. Для данного проекта выбор пал на технологию Unreal Engine от Epic Games. Unreal Engine – игровой движок, был разработан в 1998 году командой Epic Games специально для своего дебютного проекта – шутера от первого лица «Unreal».



Рисунок 11 – Скриншот из проекта Unreal на Unreal Engine 1

Данная версия объединяла ряд систем:

1. Рендеринг (англ. Rendering – «визуализация»), процесс получения изображения по модели, где модель – это описание любых объектов или явлений на строго определенном языке или в виде структуры данных. Описание может содержать геометрические данные, положение точек активного окна наблюдателя, информацию об освещении, степени наличия какого-то вещества, напряженность физического поля и прочее.

2. Обнаружение столкновений (англ. Collision detection) – вычисление пересечений между собой двух или более объектов. Данная тема чаще всего связана в

использовании физических движков, компьютерной анимации и робототехники. Системы обнаружения столкновений могут вычислить время воздействия и сообщить о наборах пересечениях точек, а также в зависимости от моделируемой задачи, сообщить, что должно происходить, когда столкновение обнаружено. Алгоритмы обнаружения столкновения являются одним из основных составляющих трехмерных компьютерных игр.

3. Искусственный интеллект (англ. Artificial intelligence) – набор программных методик, которые используются в компьютерных играх для создания иллюзии интеллекта в поведении персонажей, управляемых компьютером. Игровой искусственный интеллект, помимо методов традиционного искусственного интеллекта, включает также алгоритмы теории управления, робототехники, компьютерной графики и информатики в целом.

4. Работа с сетевыми компонентами (англ. Networking) – это принцип работы телекоммуникационных сетей, которая позволяет узлам совместно использовать ресурсы. В компьютерных сетях сетевые вычислительные устройства обмениваются данными друг с другом с использованием линии передачи данных. Соединения между узлами устанавливаются с помощью кабельных носителей или беспроводных носителей.

5. Сценарный язык (англ. Scripting language) – система описаний действий, выполняемых системой. В случае Unreal engine это система визуального программирования – Blueprints.

6. Управление файлами (англ. File management) – система пользовательского интерфейса для управления файлами и папками. Наиболее распространенные операции с файлами или группами файлов включают создание, открытие (например, просмотр, воспроизведение, редактирование или печать), переименование, перемещение или копирование, удаление и поиск файлов, а также изменение атрибутов файлов, свойств и прав доступа к файлам. Папки и файлы могут отображаться в иерархическом дереве на основе их структуры каталогов. Некоторые файловые менеджеры содержат функции, вдохновленные веб-браузерами, включая кнопки навигации вперед и назад.

7. Программный интерфейс приложения Glide API (англ. Application programming interface) для компьютерной графики, разработанный компанией 3dfx interactive для видеокарт на основе графических процессоров Voodoo Graphics. Изначально Glide была проприетарной (от Proprietary — частное, патентованное) технологией. Данное API было предназначено для увеличения производительности графической подсистемы в играх, поддерживающих геометрические и текстурные преобразование в основном в форматах данных, использующихся непосредственно в самих картах. На видеокартах VooDoo впервые действительно приемлемо выполнялись 3D-игры, в результате чего Glide получил широкое применение. В дальнейшем компания Epic Games стала использовать Direct3D от Microsoft и OpenGL, в связи их быстрого развития.

В 2002 году, вторая версия Unreal Engine дебютировала с выпуском компьютерной игры «America’s Army». Данная версия была во многом переписана и была заявлена поддержка игровых платформ GameCube и Xbox.



Рисунок 12 – Скриншот проекта America’s Army на Unreal Engine 2

В 2004 году впервые были опубликованы скриншоты третьей версии Unreal Engine. Хотя данная версия была в разработке в течении 18 месяцев, она ознаменовала значительные изменения, в том числе новую модель освещения, которая работала на попиксельной (англ. Per-pixel) основе, а не методом превертекса (англ. Prevertex), который использовался в предыдущих версиях движка. Данный метод

освещения позволил использовать технику Normal mapping, а также текстуры с высоким и низким разрешением. Предыдущие версии движка требовали непосредственного моделирования мельчайших деталей, а использование Normal mapping позволило сократить количество полигонов при сохранении высокого уровня детализации. Unreal Engine 3 также был разработан для использования как DirectX, так и OpenGL, что позволило расширить доступную пользовательскую базу систем до OS X, iOS, Android, Javascript и HTML5. В 2009 году была выпущена свободная версия Unreal Engine, как Unreal Development Kit (UDK).



Рисунок 13 – Скриншот проекта Batman Arkham Asylum на Unreal Engine 3



Рисунок 14 – Сравнение детализации персонажей

Последняя версия движка, Unreal Engine 4, была выпущена в 2014 году. В этой версии были отмечены некоторые из наиболее значительных изменений на сегодняшний день. К одному из основных улучшений относится удаление Kismet, визуальной скриптовой системы, присутствовавшей в ранних версиях движка. Она была заменена на более удобную для пользователей систему Blueprints. Система освещения снова была обновлена, чтобы использовать новый алгоритм для сокращения вычислительных затрат и улучшения общего качества изображения. Редактор Unreal Engine 4 был переработан, чтобы улучшить общее время сборки, а также увеличить скорость, с которой можно обрабатывать итерации проекта.

Для увеличения удобства при разработке кода программы на языке C++, была использована среда Visual Studio от компании Microsoft. Это линейка продуктов, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой Windows Form.

При совместном использовании Unreal Engine и Visual Studio у разработчиков появляется мощный инструмент, который позволяет проводить процессы рефакторинга и отладки кода без потери большого количества времени, увеличивая скорость. Но разработка возможна и без использования C++, путем использования встроенного языка визуального программирования Blueprints.

2.2 Динамические модели и материалы

В ходе работы над реалистичной окружающей средой были использованы следующие продукты программного обеспечения:

1. Autodesk 3ds Max – это профессиональная программа для работы с компьютерной графикой, создания 3D моделей, анимации, игр, а также визуализации изображений. Она был разработан компанией Yost Group и Autodesk Media and Entertainment в 1990 году. Программа располагает широкими средствами для создания разнообразных по форме и сложности трехмерных компьютерных моделей, реальных или фантастических объектов окружающего мира, с использованием разнообразных техник и механизмов. Самым распространённым способом моде-

					ВКР.155506.09.04.04.ПЗ	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		27

лирования для создания сложных и низкополигональных моделей для игр является метод полигонального моделирования, в которое входят техники редактируемой поверхности (англ. Editable mesh) и редактируемого полигона (англ. Editable poly).

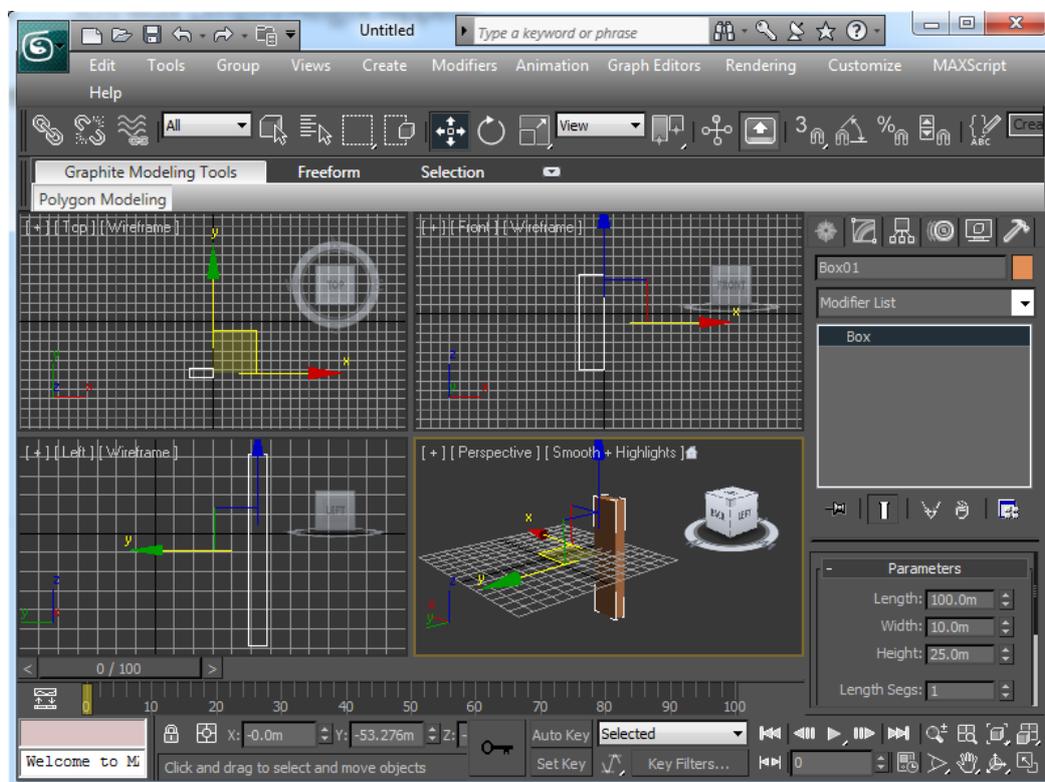


Рисунок 15 – Интерфейс Autodesk 3dsMax 2017

2. Substance Designer является мощным нодовым инструментом текстурирования для PBR рендера (Physically Based Rendering), который позволяет создавать substance files или bitmap текстуры. Он используется для текстурирования трехмерных моделей и для запекания различной информации о модели, например для запекания карт нормалей (англ. Normal map), дисплейсмента (англ. Displacement) и прочего. Программа содержит обширную библиотеку инструментов, материалов и процедурных эффектов, которые помогут достигать поставленных целей при разработке. Substance Designer широко используются в более чем 50 игровых проектах AAA-класса.

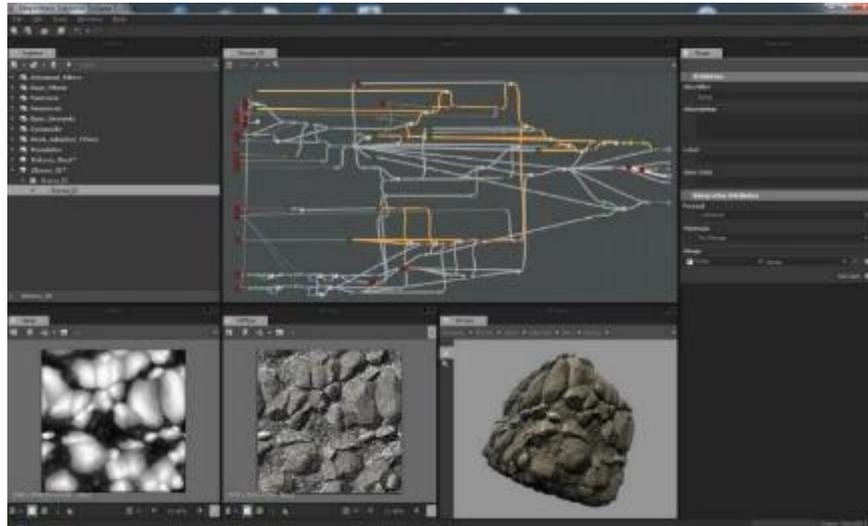


Рисунок 16 – Интерфейс программы Substance Designer

3. Substance Painter – программа, позволяющая создавать текстуры на трехмерной модели путем рисования кистью на полигоне. Данный метод позволяет контролировать каждую итерацию и ускоряет работу при текстурировании.

4. SpeedTree – технология для визуализации растений с помощью трехмерной графики. При использовании данного пакета, существенно упрощает создание растений и их анимаций для дизайнеров, и дополнительно позволяет снизить нагрузку на ресурсы компьютера, в сравнении с использованием стандартных средств игрового движка Unreal Engine. В последних релизах программы имеется возможность модульного подключения в виде плагина.



Рисунок 17 – Интерфейс программы SpeedTree

2.3 Физическая модель

Для реализации реалистичной физической системы взаимодействия объектов было принято решение использовать APEX SDK.

APEX – это мультиплатформенная, масштабируемая система. С ее помощью художники могут создавать физические эффекты при минимальном участии программистов. Вместо низкоуровневого API PhysX (предыдущая версия APEX), разработчику предоставляется набор инструментов для создания определенных физических эффектов на базе готовых модулей.

APEX состоит из двух основных компонентов:

Авторский компонент, обеспечивает высокоуровневое создание динамических систем через подключаемые модули DCC (3dsMax/Maya), автономные инструменты или плагины для игровых движков.

Компонент Runtime, состоит из модульного SDK, который включает в себя инфраструктуру, которая поддерживает все модули APEX, и предоставляет интерфейс визуализации, позволяющий упростить процесс интеграции в игровые движки. Поскольку APEX масштабируется на широком спектре аппаратных платформ, художник способен создавать компоненты, которые можно легко адаптировать для вычислительных мощностей любых целевых платформ.

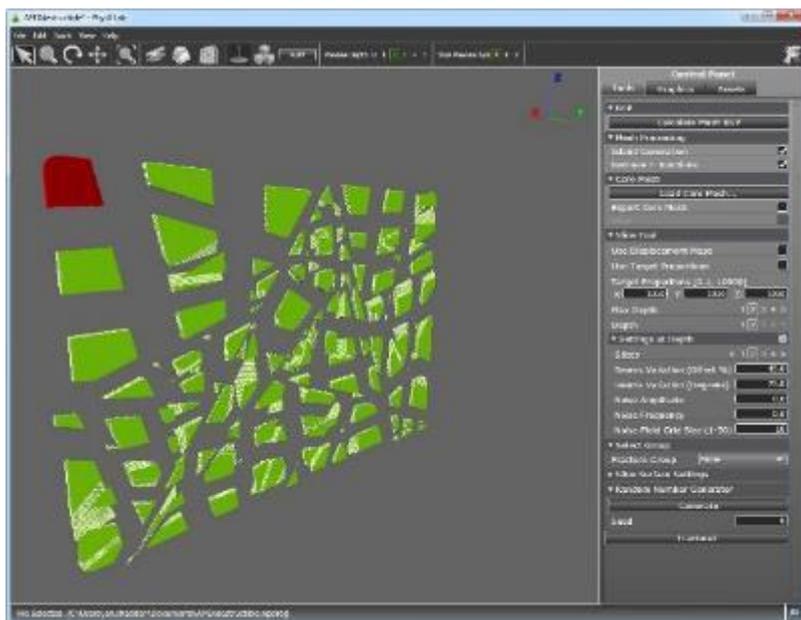


Рисунок 18 – Модификатор Apex в программе 3dsMax

Изм.	Лист	№ докум.	Подп.	Дата

3 ТЕХНОЛОГИИ РЕАЛИЗАЦИИ КОМПЬЮТЕРНОЙ СИМУЛЯЦИИ

3.1 Создание прототипа уровня

Создание уровня начинается с проработки его прототипа на основе примеров (референсов) из фотографий, художественного описания и рисунков художников. При создании исторической реконструкции основной проблемой может являться поиск данных примеров.



Рисунок 19 – Изображение острога художником иллюстратором

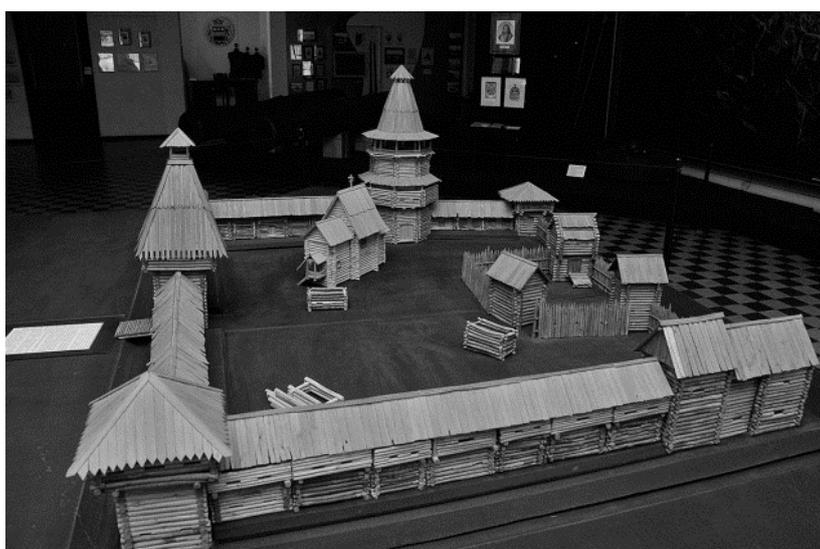


Рисунок 20 – Фотография макета острога в краеведческом музее

Если говорить в контексте дизайна уровней, работа условно будет делиться на четыре этапа:

1. «Greybox» - Уровень собирается буквально из серых кубов. Здесь закладывается фундамент всей последующей работы и выясняется размер уровня, требуемые графические и аудио ассеты.

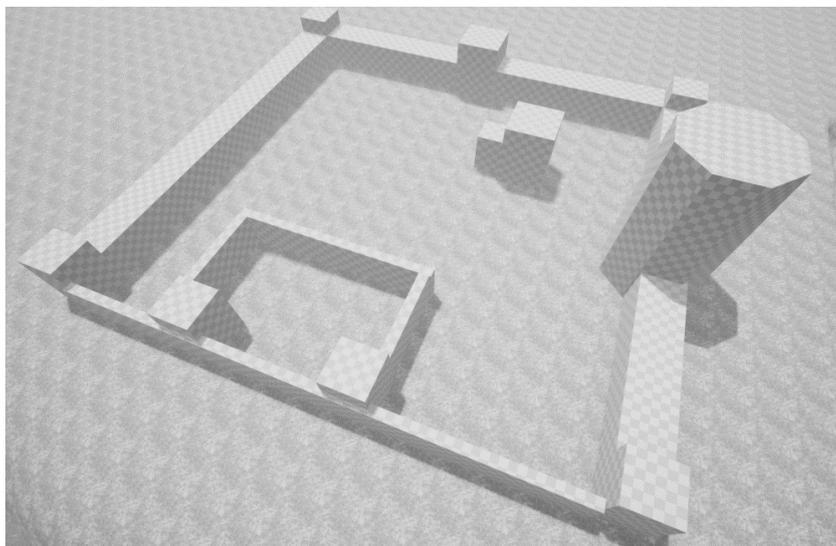


Рисунок 21 – Скриншот начальной стадии уровня

2. «Whitebox» - Геометрия уровня уточняется, по возможности добавляются новые игровые механики, диалоги, заготовки роликов, звук и так далее. На данном этапе лучше оттачиваются игровые механики и поведение скриптовых элементов.



Рисунок 22 – Скриншот одной из итерации геометрии объектов

3. Графический пасс – К моменту работы над этим этапом необходимо быть уверенным, что уровень вас устраивает с точки зрения игровых механик и особенно, с точки зрения соблюденного исторического контекста.



Рисунок 23 – Скриншот одной из последних итераций геометрии объектов

4. Завершающий пасс – Наносятся финальные штрихи: добавляются эффекты, позволяющие отображать среду более правдоподобно. К таким эффектам относятся: симуляция природных и погодных эффектов, звуков, добавление птиц и т.д.



Рисунок 24 – Скриншот одной из последних итераций геометрии объектов



Рисунок 25 – Скриншот внутреннего убранства землянки

5. Полировка – Этап, как правило происходящий уже после того, как работа над уровнем была завершена. Обычно включает в себя правку ошибок, пропущенных ранее, а также изменения по результатам новых тестирований и собранной статистики.



Рисунок 26 – Тестирование эффектов огня

На любой итерации важно учитывать историческую достоверность, а также для упрощения работы с природными элементами, была проведена обязательная работа по исследованию географической области реконструируемого объекта. Ал-

базинский острог располагался на территории нынешней Амурской области вблизи реки Амур.



Рисунок 27 – Современная карта, красным крестом выделено место расположения острога



Рисунок 28 – Ландшафт прототипа в среде Unreal Engine

Климат Амурской области переходный от резко континентального на северо-западе к муссонному на юго-востоке. Данная информация позволяет более точно подойти к дальнейшей работе по созданию растительности, а также погодных эффектов. В следующих этапах мы рассмотрим создание каждого элемента окружения более детально.

3.2 Реализация статической модели

3.2.1 Описание этапов моделирования

Создание любого вида трехмерной модели происходит по следующим этапам:

1. Сбор описательной информации. Для этого используются рисунки, фотографии, а также текстовое описание;
2. Создание высокополигональной модели объекта. Данный процесс осуществляется в программах компьютерного моделирования;
3. Создание низкополигональной модели объекта. На основе высокополигональной модели осуществляется построение новой более простой геометрии;
4. Создание UV развертки. На данном этапе необходимо провести соответствие между координатами поверхности трехмерного объекта (X, Y, Z) и координатами на текстуре (U, V)
5. Создание текстур и материалов для их дальнейшей визуализации в среде Unreal Engine.

3.2.2 Создание трехмерной модели

Реализация любой трехмерной компьютерной симуляции подразумевает создание 3D моделей, так как без них невозможно визуализировать среду в полной мере. Разберем данный процесс на создании одного из объектов реконструкции – башня острога.

Для начала необходимо найти необходимый референс из любых удобных для вас источников, в нашем случае это был интернет и информация из краеведческого музея.



Рисунок 29 – Фотография башни острога

Желательно, чтобы примеров для воссоздания объекта было больше чем один, так как важно передать точное представление об объекте на созданной модели. Очень помогли при моделировании башен острога различные схемы, которые продемонстрировали объект в нескольких проекциях.

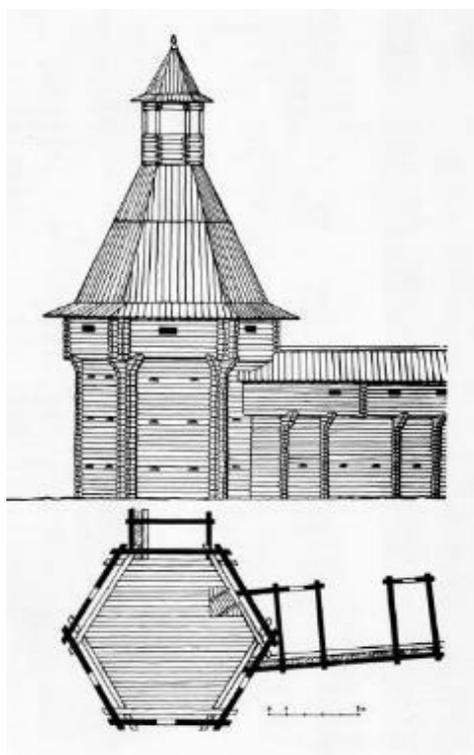


Рисунок 30 – Схема башни острога

Изм.	Лист	№ докум.	Подп.	Дата

ВКР.155506.09.04.04.ПЗ

Лист

37

После нахождения нужных референсов, приступаем к созданию высокополигональной модели в программе 3dsMax. Для оптимизации процесса, желательно использовать модульный метод моделирования, когда один сложный объект разделяют на несколько менее сложных. В дальнейшем возможно использование модели в отрыве от общего объекта реконструкции, смоделированная часть башни может быть использована для реализации стен острога, что опять-таки сокращает время разработки. На этапе создания высокополигональной модели важно передать все необходимые детали, для того, чтобы модель легко считывалась с визуальной точки зрения. После приступаем к созданию низкополигональной модели.

Основной целью создания низкополигональной (Low-poly от англ. Low-низко и polygon – полигон) модели является реализация объекта с небольшим количеством полигонов для уменьшения нагрузки системы и общей оптимизации проекта. Также низкополигональная модель может использоваться для системы LOD, что позволяет реализовать динамическую подгрузку и выгрузку высокополигональной модели на низкополигональную и обратно. Такой принцип часто используют при создании больших пространств окружающей среды, и если во время рендеринга сцены некоторые объекты, будут сильно удалены от игрока, то будет целесообразнее выгружать высокополигональную модель и заменять ее на низкополигональный аналог, во избежание проблем с производительностью.

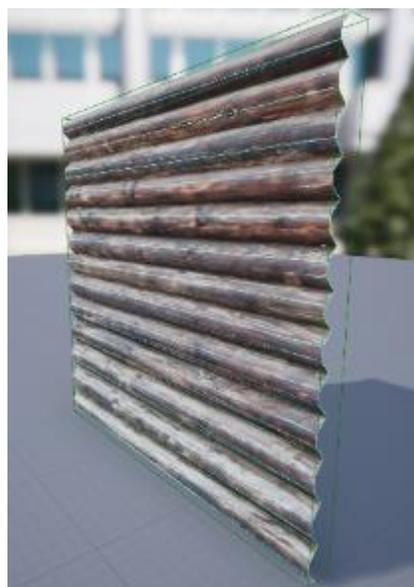


Рисунок 31 – Высокополигональная модели стены с выделением всех граней

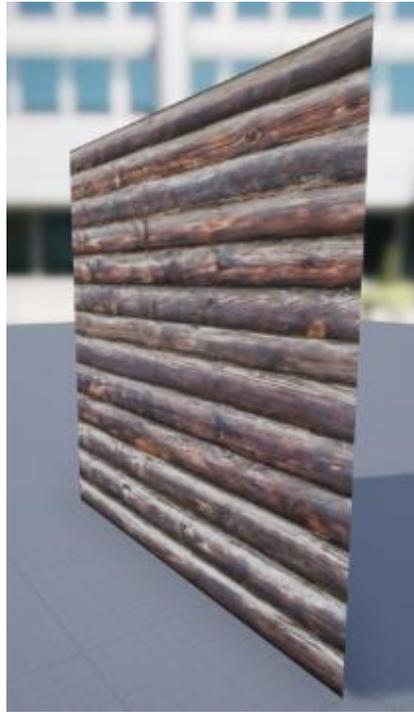


Рисунок 32 – Низкополигональная модель стены

3.2.3 Реализация текстур и материала модели

Для текстурирования объекта, нужно создать UV развертку на основе его низкополигональной модели. Сложнее дело обстоит с объектами, форма которых гораздо сложнее кубика или шара. Развертка поверхности – фигура, получающаяся в плоскости при таком совмещении точек данной поверхности с этой плоскостью, при котором длины линий остаются неизменными. Для создания развертки каждой модели, используем в 3dsMax модификатор под названием – UnwrapUVW. Существуют разные способы создания текстурных разверток. В нашем случае мы имеем объект, который можно разделить на несколько частей и разложить их по отдельности не «разрывая» самого объекта. Деление объекта происходит путем выделения граней нужного контура с дальнейшим применением метода Flatten Mapping для каждой части модели. После сохранения UV развертки приступаем к созданию необходимых текстур и материалов с помощью инструментов Substance Designer и Substance Painter. В то время как Substance Designer позволяет создавать процедурные и настраиваемые текстуры в обмен на небольшое количество технических знаний, Substance Painter делает это спонтанно и интуитивно – путем рисования кистью по модели.

Substance Painter – отличный инструмент для быстрого текстурирования объектов или персонажей, рабочий процесс в нем базируется на слоях – как в Photoshop. Но особенно он привлекателен тем, что можно использовать Substance материалы, чтобы рисовать прямо по мешу по всем каналам (diffuse, specular и т.д.) одновременно или создавать полностью настраиваемые эффекты, например, эрозию или грязь. Unreal engine начиная с версии 3.5 поддерживает технологию физически корректной визуализации (англ. Physically-based rendering, PBR). Данная технология представляет собой набор техник визуализации реалистичных материалов, максимально приближая их внешний вид к аналогам из реального мира. Все техники PBR основаны на теории микрогранул (англ. Microfacers). Данная теория описывает любую поверхность в микроскопическом масштабе как набор небольших идеально отражающих зеркал, которые и называются микрогранулами. В зависимости от шероховатости поверхности, расположение этих крошечных зеркал может сильно различаться:

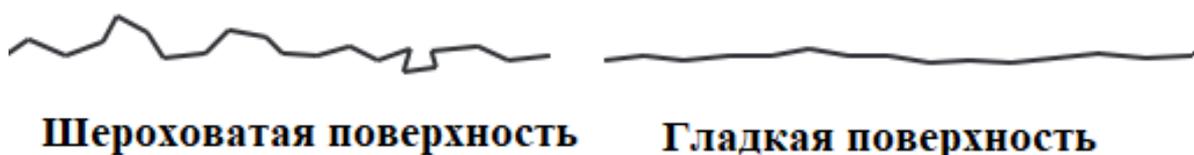


Рисунок 33 – Примеры типов поверхностей

Чем грубее поверхность, тем более хаотичнее выстроена каждая микрогранула по поверхности. Эффект этих маленьких зеркал заключается в том, что при разговоре о зеркальном освещении или отражении, входящие световые лучи с большей вероятностью рассеиваются по совершенно разным направлениям на более шероховатых поверхностях, что приводит к более широкому распространению зеркальных отражений. Напротив, на гладкой поверхности световые лучи отражаются примерно в одном и том же направлении, что дает нам более мелкие и более резкие отражения:



Шероховатая поверхность

Гладкая поверхность

Рисунок 34 – Отражение световых лучей на разных поверхностях

На микроскопическом уровне поверхность не является полностью гладкой, но, так как эти микрогранулы достаточно малы, что мы не можем различить их по попиксельной основе, мы статистически аппроксимируем шероховатость поверхности с учетом параметра неровности. Основываясь на неровности поверхности, мы можем вычислить отношение микрогранул, грубо ориентированных к некоторому вектору h по формуле (1).

$$h = \frac{l + v}{\|l + v\|}, \quad (1)$$

где h – параметр шероховатости исследуемого материала; l – вектор пучка света, падающего на поверхность; v – вектор расстояния взгляда до поверхности.

Чем больше микрогранул выровнено по отношению вектора h , тем резче и сильнее зеркальное отражение. Вместе с параметром шероховатости, который изменяется между нулем и единицей, мы можем статистически аппроксимировать выравнивание микрогранул. Мы можем видеть, что более высокие значения шероховатости показывают большую зеркальную форму отражения, в отличие от меньшего значения, что наоборот, приводит к уменьшению отражения, но также к увеличению резкости его границ.

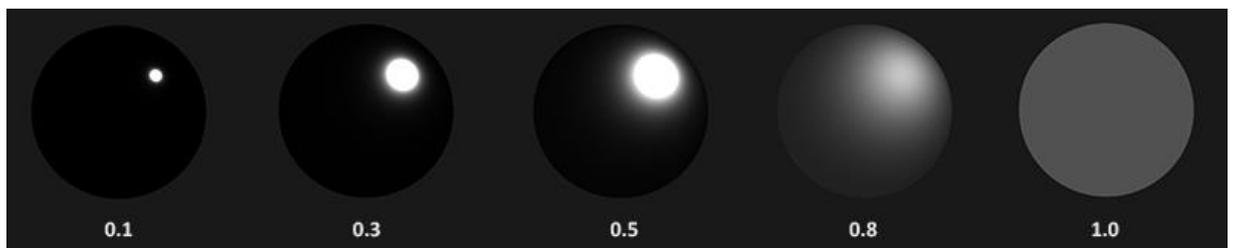


Рисунок 35 – отображение зеркальных поверхностей

У вышеописанной технологии физически корректной визуализации есть следующие преимущества:

1. Намного легче создавать реалистичные модели, так как отпадает необходимость в бесконечном переборе множества вторичных параметров материала. Пользователь задает лишь несколько фундаментальных характеристик – все остальное вычисляется на основе точных физических формул и алгоритмов.
2. Модели выглядят одинаково корректно во всех условиях освещения.
3. Сам процесс разработки более логичный, интуитивный и линейный, не смотря на широчайший спектр возможных реализаций.

Для создания текстур необходимо произвести процесс снятия с высокополигональной модели мелкие детали. Данный процесс называется запекание, он используется для последующей имитации разных объектов на поверхности, которые фактически будут отсутствовать на геометрии. С помощью Substance Painter производим запекание модели для получения карты нормалей.

Карта нормалей – это текстура, содержащая информацию об отклонении нормалей от поверхности, и имеет в основном фиолетовый цвет с градиентными переходами до красного цвета и демонстрирующая, на сколько должна быть искривлена поверхность, на сколько она вогнутая или выпуклая и какие дополнительные детали на ней присутствуют.

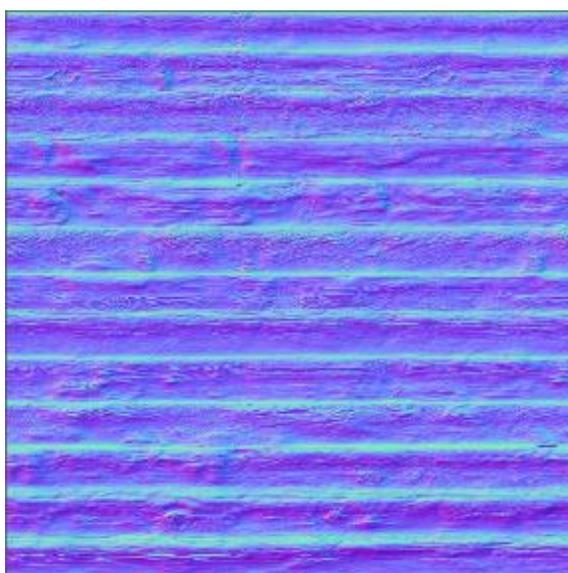


Рисунок 36 – Карта нормалей поверхности стены

Использование карт нормалей является самым эффективным способом для создания реалистичных детализированных поверхностей без увеличения количества полигонов на модели.

Следующим этапом, после того как созданы низкополигональная модель и карты нормалей, можно приступить к созданию текстурных карт. Данный процесс легко реализуем в среде Substance Painter, при загрузке модели в формате fbx, а также ее uv карты мы генерируем остальные текстурные карты для создания основного цвета, передачи шероховатостей, а также затенения объекта (англ. Ambient occlusion).

Ambient occlusion — модель затенения, используемая в трёхмерной графике и позволяющая добавить реалистичности изображению за счёт вычисления интенсивности света, доходящего до точки поверхности. Для реализации затенения и шероховатостей объекта, мы использовали текстуру – Linear Color.

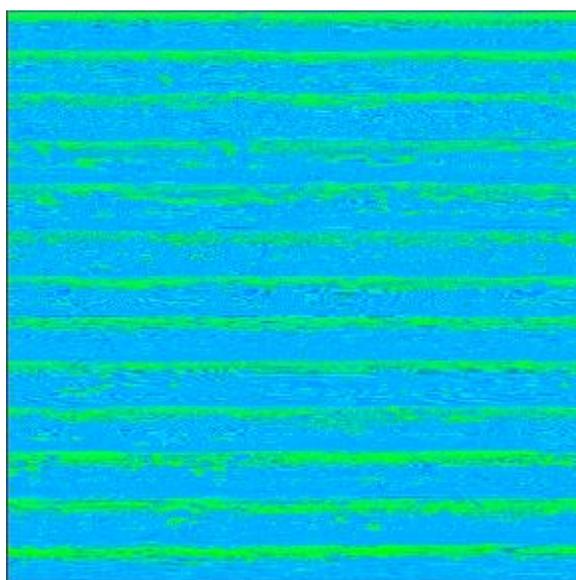


Рисунок 37 – Linear Color текстура

На завершающем этапе мы создаем новый материал и экспортируем его в среду Unreal Engine. Благодаря мощному редактору создания материалов, создаем общий нод материала – LogWall_mat и соединяем созданные ранее текстуры в нужные нам блоки. Все изменения материала, мы можем наблюдать в реальном времени в поле Rewiew.

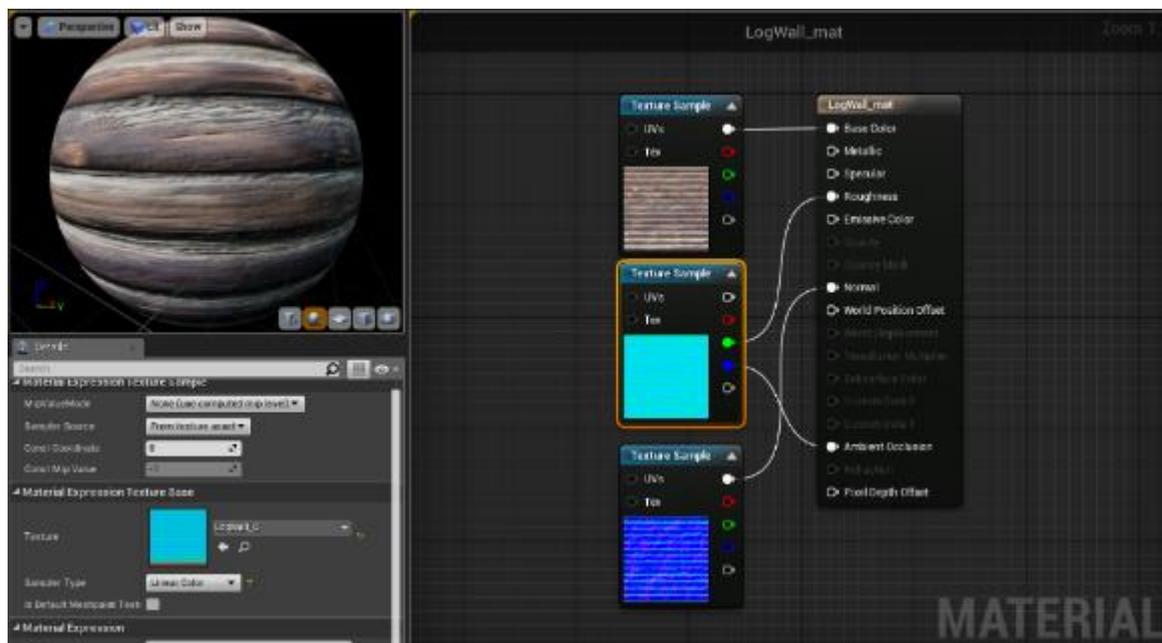


Рисунок 38 – Настроенный материал бревенчатой стены



Рисунок 39 – Готовый результат в среде Unreal Engine

3.3 Реализация системы смены погодных эффектов

В современных компьютерных реконструкциях очень важно придать реалистичности изображению. Одним из таких способов является передача погодных эффектов для улучшения погружения игроком или зрителем. Среда Unreal Engine не имеет готовую систему смены дня и ночи, а также других погодных эффектов.

Поэтому было принято решение реализовать данную систему самостоятельно для придания реконструкции более «живого» внешнего вида.

Для начала создаем класс нужного нам объекта и приступаем к его формальному описанию. Основой нашей системы является статичная модель – сфера. Непосредственно симуляция времени суток будет реализована путем создания динамического материала данной сферы с множеством переменных и функций. Unreal engine уже имеет ряд компонентов необходимых для реализации солнечного и лунного света, а также атмосферы окружающей среды.

Direction light (с англ. Направленный свет) – компонент имитирующий свет от излучаемого источника, находящимся бесконечно далеко. Это означает, что все тени, излучаемые этим светом, будут параллельны, что делает данный компонент идеальным выбором для имитации солнечного света. Также важным параметром для нас при создании реалистичного солнца является возможность его динамической трансформации. Данный параметр также является важным и для лунного света.

Exponential Height Fog (с англ. Экспоненциальная функция высоты тумана) – компонент, с помощью которого возможна реализация реалистичного тумана, путем создания большой плотности в низких местах карты и меньшей плотности в высоких местах.

Sky Lights (с англ. Небесный свет) – компонент для автоматизации установки цвета неба и общего освещения сцены.

Для базового контроля данной системы создаем переменные:

1. Time of day (float) – параметр для изменения времени суток
2. Saturation (float) – параметр изменяющий насыщенность цвета неба
3. Overall intensity (float) – параметр общей интенсивности неба
4. Night Brightness (float) – параметр яркости ночного неба

Так как нашей задачей является создание автоматизированного модуля, с которым будет удобно работать, необходимо инициализировать функции в которых будут происходить основные процессы по работе с компонентами и материалами.

Для реализации перемещения источников света (солнца и луны), создаем функцию Set Solar Angle, где описываем все перемещения объектов Direction light.

Схема реализации установки вращения солнца

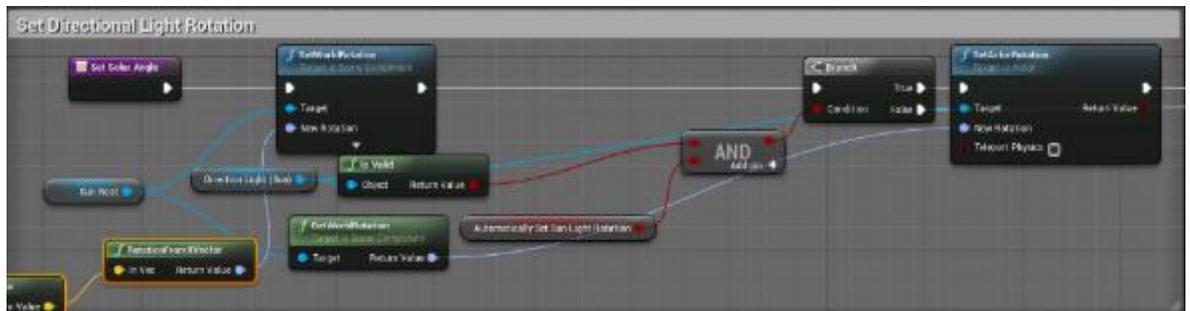


Рисунок 40 – Схема вращения солнца

Для создания лунного света, было важно получать значения координат солнца и на их основе корректировать угол перемещения луны. Схема данной реализации

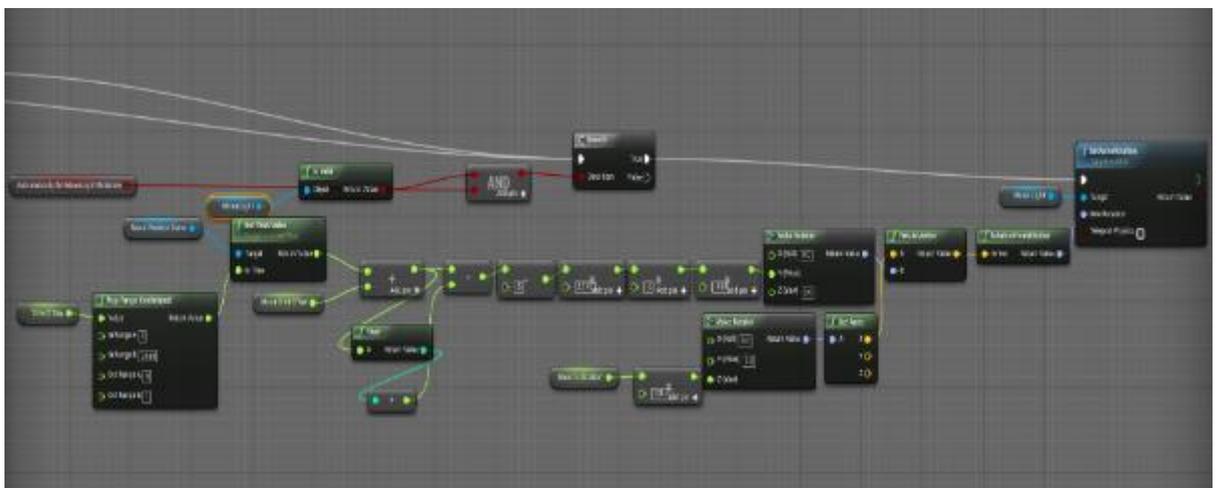


Рисунок 41 – Схема вращения луны

Сохраняем созданную функцию и переходим к реализации «умного» материала, для небесной поверхности. В его основе были использованы комбинированные текстуры, что позволило сократить время разработки. Для реализации движения, пористости и объемности облаков был создан материал, использующий только одну текстуру, изображенную на рисунке

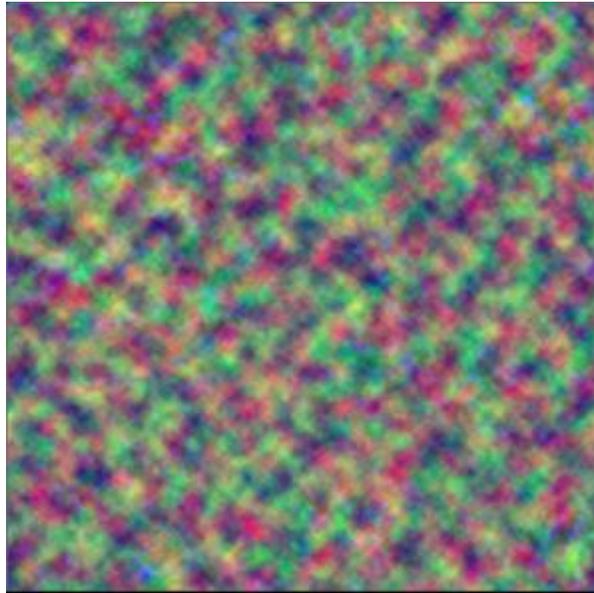


Рисунок 42 – RGB текстура для симуляции облаков

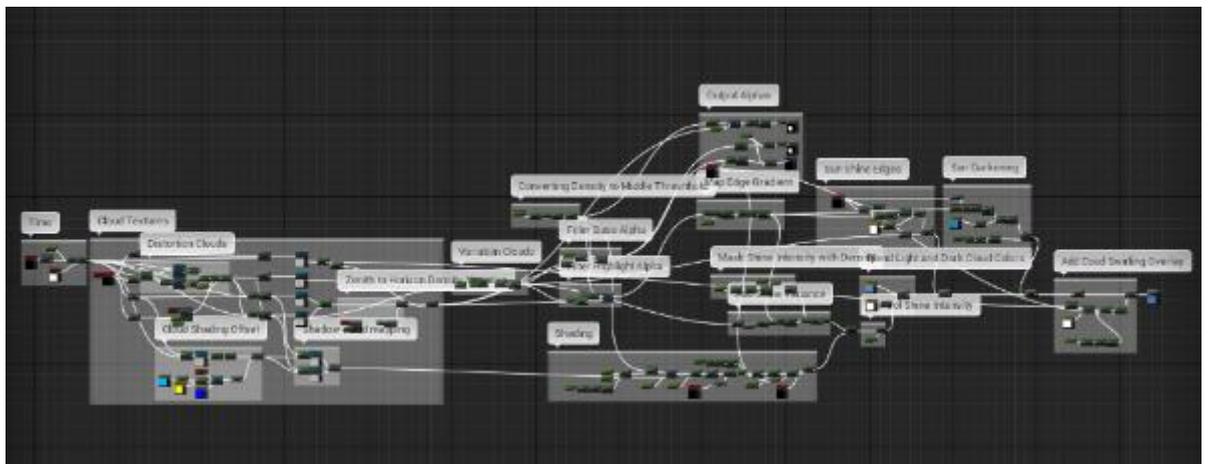


Рисунок 43 – Логическая схема реализации слоя облаков

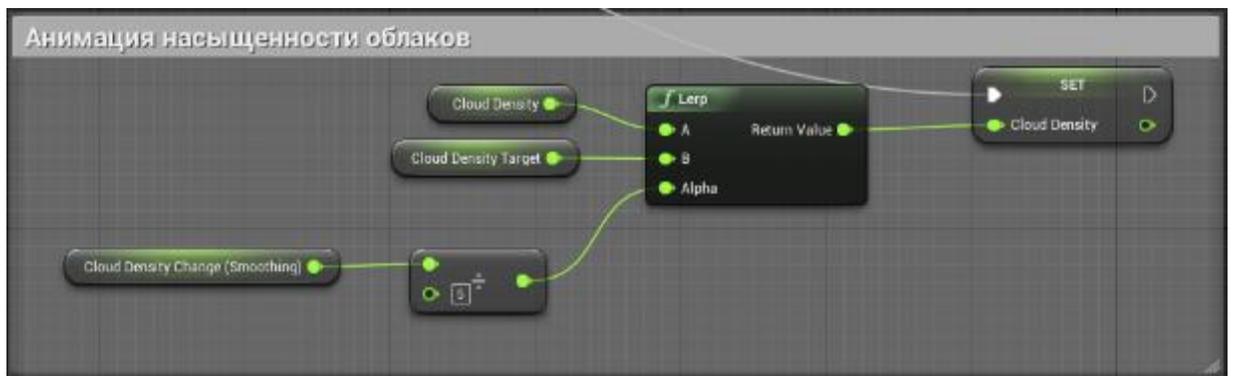


Рисунок 44 – Схема анимации плотности облаков



Рисунок 45 – Плотность облаков

Реализованная система цикла дня и ночи в дальнейшем будет подключена к модулю искусственного интеллекта для реалистичного поведения игровых агентов. Они смогут ориентироваться на время суток и приступать к различным задачам исходя из этих данных. Реализация искусственного интеллекта будет отображена более подробно в дальнейшем.

На рисунках 1 изображена работа цикла дня и ночи.

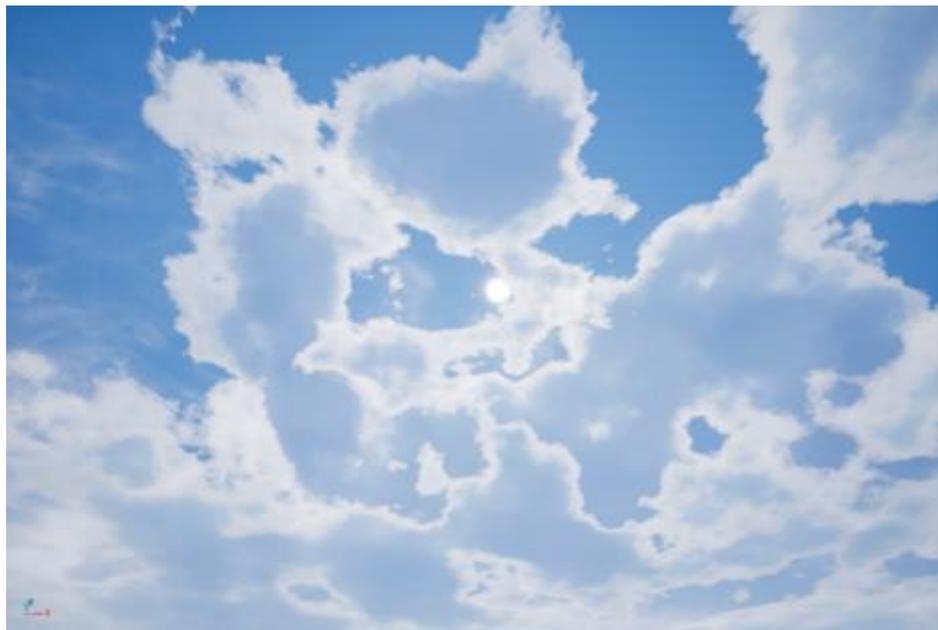


Рисунок 46 – Полдень

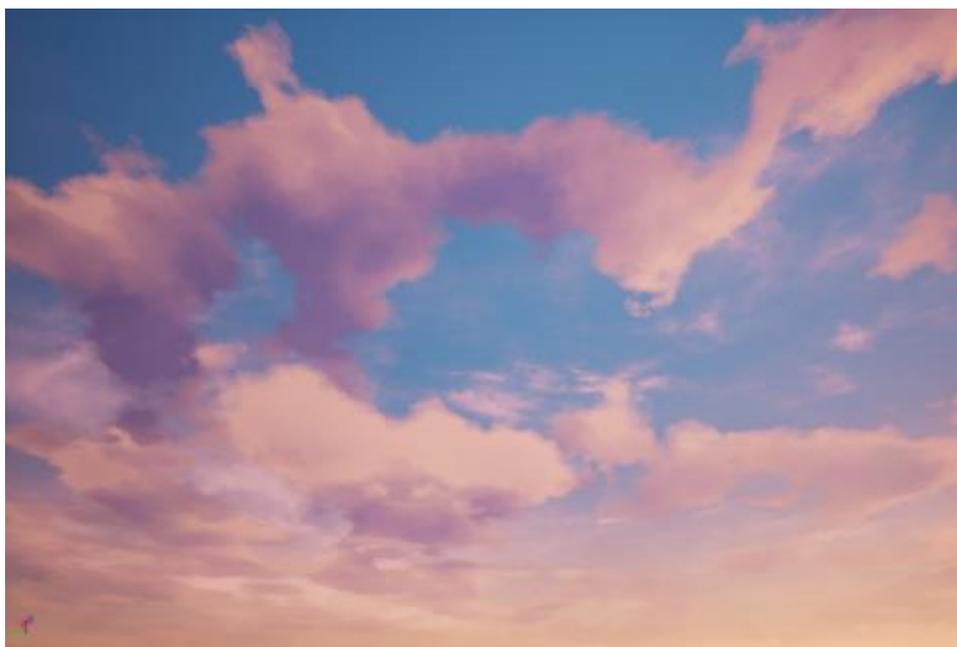


Рисунок 47 – Закат

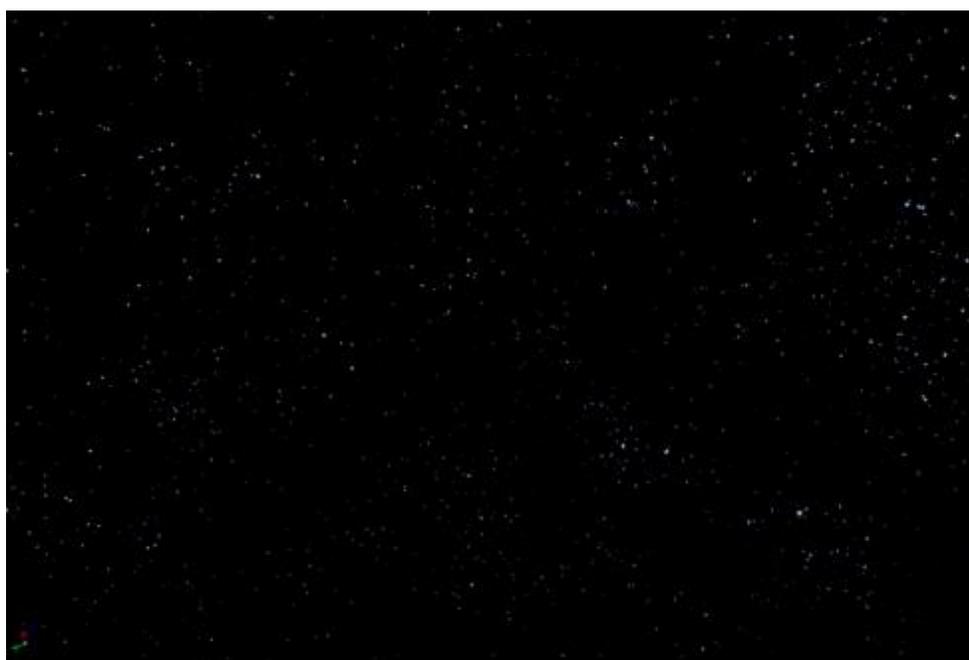


Рисунок 48 – Ночь

Также для упрощения дальнейшей работы с погодными эффектами, была реализована система симуляции лунных фаз. Принцип ее работы заключается в подключении заранее созданной текстуры, на которой были представлены все возможные фазы внешнего вида космического объекта. Путем подключения дан-

ной текстуры мы распознаем контуры объекта, чтобы достичь эффект анимации путем постепенной смены изображений. Данный процесс автоматизирован и легко управляем благодаря реализованной логической схеме, представленной на рисунке 49.

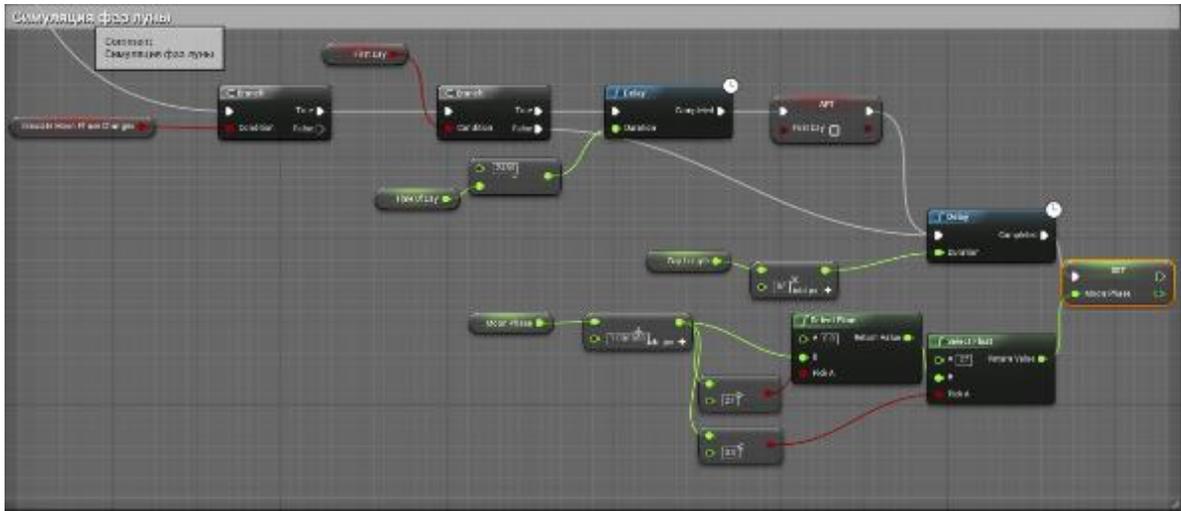
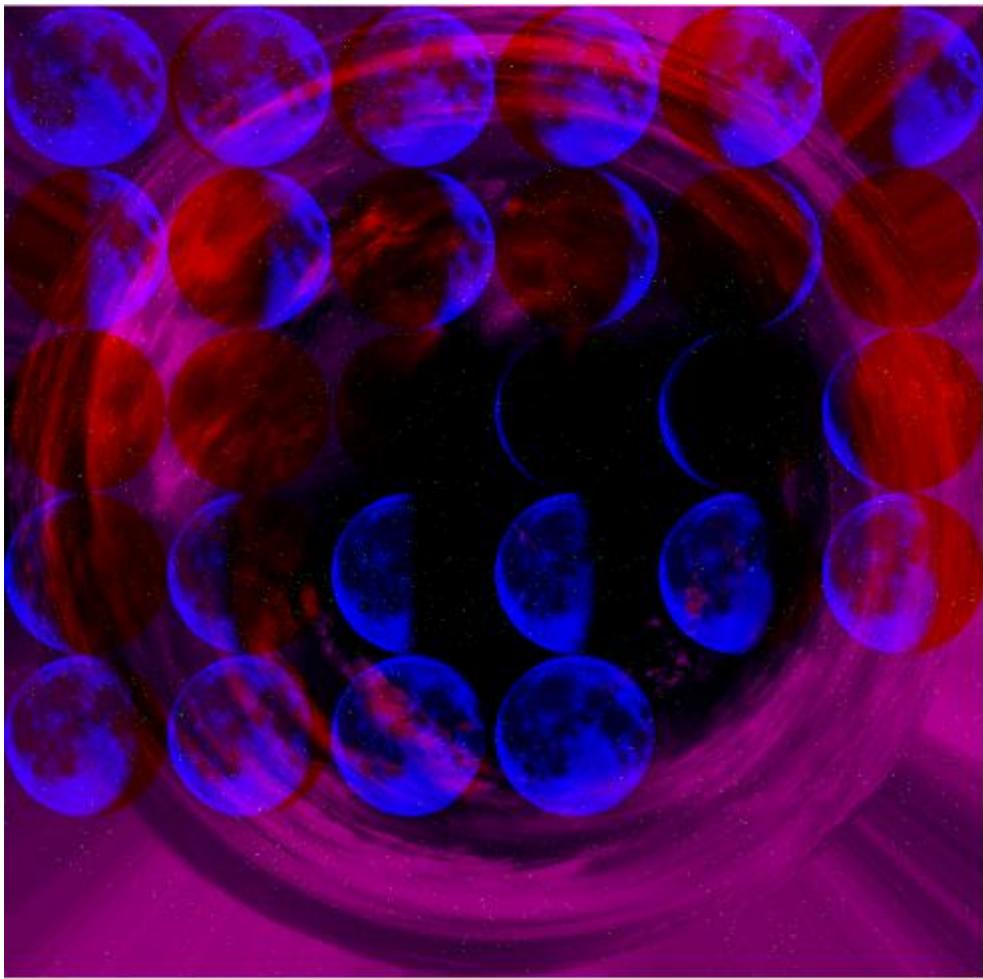


Рисунок 49 – Схема симуляции фаз лунного тела



Изм.	Лист	№ докум.	Подп.	Дата

ВКР.155506.09.04.04.ПЗ

Рисунок – Универсальная текстура лунных фаз

Для полной автоматизации всех процессов, все переменные и функции были выведены с помощью интерфейса. Используя его, мы можем настраивать окружающую среду под нужную нам задачу.



Рисунок 50 – Внешний вид настроек базовых параметров

3.4 Реализация искусственного интеллекта персонажей

3.4.1. Общая теория игрового искусственного интеллекта

В зависимости от сложности игры и компьютерной симуляции, система игрового искусственного интеллекта может быть крайне простой и представлять собой лишь набор правил или иметь достаточно сложные алгоритмы работы (например, если стоит задача по управлению многотысячной армией в ходе сражения). Рассмотрим некоторые базовые основы, которыми руководствуются разработчики при создании игры с элементами искусственного элемента.

Одной из самых простых форм игрового искусственного интеллекта является система на основе правил, в которой поведение игровых агентов определяется заранее заданными алгоритмами. Такие системы наиболее далеки от традиционного понятия искусственного интеллекта. Правила могут как задаваться жестко (например, игровой агент всегда поворачивает в одну сторону), так и основываясь на анализе текущего состояния объектов (например, игровой агент всегда движется за игроком).

Также одним из возможных методов построения искусственного интеллекта является использование конечных автоматов. Они используются для создания иг-

ровых агентов, которые в течении игры могут обладать различным (конечным) числом состояний. В качестве таких состояний можно задать физические условия, в которых находится объект, его эмоции и т.д. Данные состояния в зависимости от условий игры могут меняться. Примеры возможных состояний игрового агента представлены на рисунке

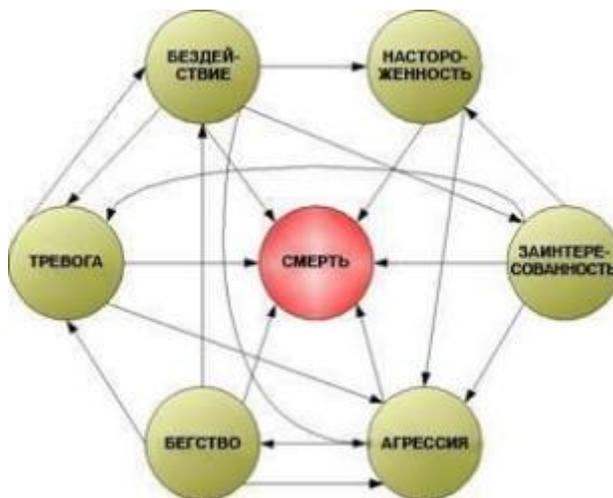


Рисунок 51 – Возможные состояния игрового агента

Стоит отметить, что под эмоциональным состоянием игрового агента понимается его алгоритм поведения в той или иной ситуации. Эмоциональное состояние не имеет никакого отношения к эмоциям искусственного интеллекта.

Для того, чтобы внести разнообразие в игру и создать более сильного и динамического противника, разработчикам необходимо наделить игровых агентов способностью развиваться, приспосабливаться и адаптироваться. Такое правило активно применяется в боевых и стратегических играх, которые имеют сложную механику и предоставляют большое количество разнообразных возможностей для игровых агентов.

Адаптивный искусственный интеллект должен обладать достаточно важным свойством – уметь предсказывать следующий ход противника. Для этого могут применяться различные методы (выявление закономерности прошлых ходов игрока, использование случайных догадок и т.д.) В настоящее время данные системы искусственного интеллекта реализуют с помощью нейронных сетей. Благодаря им

система способна заниматься самообучением, после каждой итерации симуляции (например, сражение с игроком), что позволит искусственному интеллекту запомнить паттерны поведения игрока и сопоставить ему свои новые действия.

3.4.2 Постановка задач для искусственного интеллекта

Для реализации искусственного интеллекта было принято решение использовать систему конечного автомата. Все игровые агенты должны иметь возможность воспринимать окружающую среду и ориентироваться в ней при перемещении. Для этого игровой искусственный интеллект должен обладать рядом специфических возможностей.

Первая возможность — это зрение. Чтобы принимать взвешенные решения, игровому агенту нужно знать окружающую его обстановку. В робототехнике для этого активно применяются различные технологии компьютерного зрения, позволяющие машинам получать трёхмерные объемные изображения. В системах игрового искусственного интеллекта такая детализация, безусловно, избыточна. Компьютерный мир здесь имеет одно большое преимущество: разработчикам уже заранее известны все объекты, с которыми машина может столкнуться.

Вторая возможность – это слух. Для создания более интеллектуальных игровых агентов можно наделить их возможностью слышать передвижение игроков. Поскольку каждые действия в компьютерном мире могут сопровождаться определенным уровнем звука, то можно заранее задать тот уровень, по достижению которого игровой агент будет способен обнаружить игрока.

Третья возможность – это взаимодействие с временными объектами. При наделении игровых агентов возможностью «видеть» окружающий мир, можно несколько расширить их восприятие, предоставив им возможность взаимодействовать с временными объектами. Такие объекты не являются значимой частью компьютерного мира, а предназначены лишь для создания визуальных эффектов, такие как следы от пуль, дым, искры, отпечатки подошв на земле и т.д. Некоторые из временных объектов могут указывать на следы недавнего сражения и привести к потенциальной цели, поэтому предоставив возможность игровым агентам использовать такие подсказки, можно значительно расширить их интеллектуаль-

ность.

Четвертая возможность – это укрытие. В случае беспорядочного огня противника игровых агентов можно наделить способностью прятаться за укрытия. Чтобы это реализовать, агент сначала должен определить, сможет ли он спрятаться за окружающие его объекты, сравнив свои размеры с размерами укрытия, и далее, определив свободно ли выбранное место, переместиться туда.

И последняя, пятая возможность – это навигация искусственного интеллекта. Оценив окружающую обстановку и приняв какое-либо решение, игровому агенту нужно понять, каким образом ему двигаться к заданной точке. Для этого можно использовать несколько подходов в зависимости от характера и уровня игры. Наиболее простым является алгоритм «Столкнуться и повернуть», когда агент движется до тех пор, пока не столкнется с каким-либо препятствием. Далее движение осуществляется по направлению, используя который агент окажется ближе всего к заданной цели. Такой подход чаще всего применяется в несложных играх. В более сложных играх данный подход несколько модифицируется: игровых агентов можно снабдить памятью, что позволяет им быстрее найти путь к цели.

В ряде стратегических игр у игроков нет времени, пока их войска разбредутся и достигнут цели, используя алгоритм «Столкнуться и повернуть». Здесь активно используются технологии предварительного поиска наиболее оптимальных путей с помощью карты. В случае нашей компьютерной реконструкции данной картой является компонент NavMesh.

3.4.3 Реализация искусственного интеллекта в среде Unreal Engine

Чтобы сократить время разработки, но в тоже время увеличить универсальность искусственного интеллекта, было принято решение разработать общий конечный автомат, который будет включать в себя все возможные состояния для всех игровых агентов. Данная система интеллекта позволит запускать симуляцию отдельно от игрока, если он захочет просто наблюдать за действиями игровых агентов. Для компьютерной реконструкции важно, чтобы персонажи могли совершать следующие действия:

1. Перемещаться по карте в зависимости от эмоционального состояния и по-

ставленной перед ними задачи (например, патрулирование местности)

2. Адекватно реагировать на внешние воздействия (например, поиск укрытия при ведении огня)

3. Реагировать на действия игрока (например, разговор или же столкновение с ним)

В среде Unreal Engine для построения искусственного интеллекта используется компонент Behavior Tree. Это очень мощный инструмент, благодаря которому возможна реализация поведения игрового агента через систему визуального программирования. Его работа состоит из нескольких элементов:

1) Систему Task (от англ. Запрос) – с ее помощью мы указываем, что делать искусственному интеллекту.

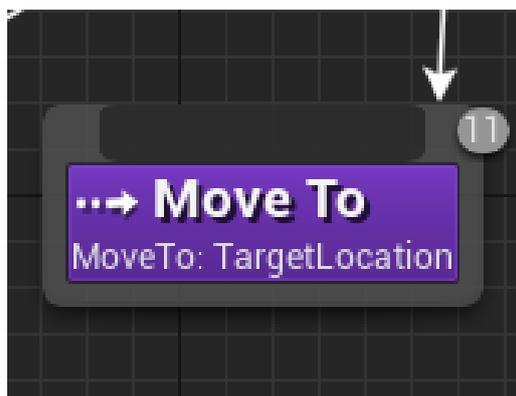


Рисунок 52 – Указание на перемещение в нужную точку

2) Систему Decorator (от англ. Окружение) – с ее помощью искусственный интеллект анализирует свое состояние (например, перемещение)

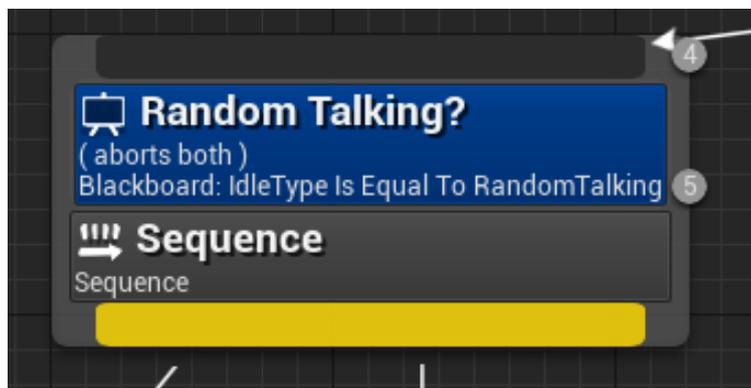


Рисунок 53 – Состояние разговора игрового агента

3) Систему Keys (от англ. Ключей) – с ее помощью мы можем объявлять различные переменные с разными типами, для внедрения в нашу систему нужных объектов симуляции (например, маркер точки перемещения)

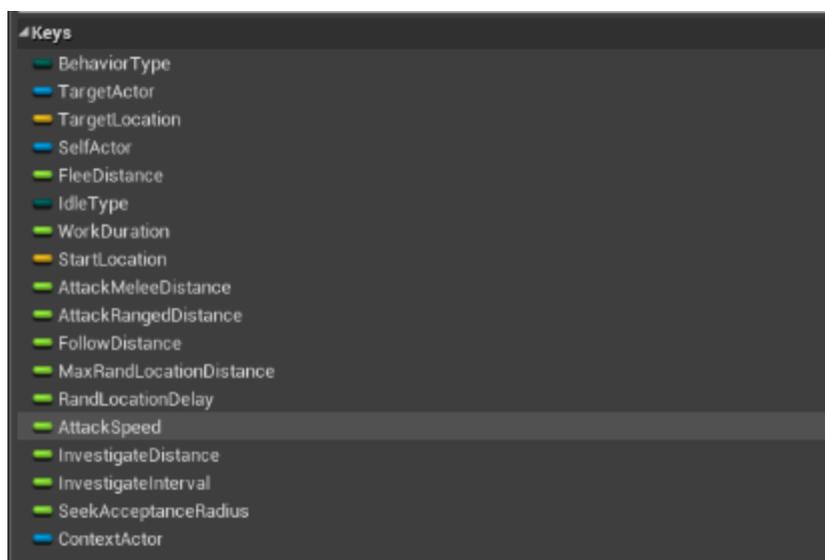


Рисунок 54 – Объявленные переменные, доступные для анализа искусственным интеллектном

Для начала работы с искусственным интеллектном, необходимо объявить родительский класс и на его основе создавать производные классы для каждого типа агентов. Для более удобной работы структурируем возможные модели поведения:

1) NPC_Civilian – модель поведения простого жителя. Способен перемещаться по доступной игровой зоне, а также взаимодействовать с элементами окружения и персонажами. В случае опасности от игрока или враждебного искусственного интеллекта – пытается спрятаться в безопасной зоне.

2) NPC_Guard_Melle – модель поведения защитника острога, вооруженным холодным оружием. Главной задачей является патрулирование указанных ему зон, при расширении задач, способен передвигаться между несколькими зонами патрулирования. В случае опасности от игрока или враждебного искусственного интеллекта – вступает в бой.

3) NPC_Guard_Ranged – модель поведения защитника острога, вооруженным огнестрельным оружием. Главной задачей является патрулирование указанных ему зон, при расширении задач, способен передвигаться между несколькими зонами патрулирования. В случае опасности от игрока или враждебного искусственного интеллекта – вступает в бой.

4) NPC_Animal – модель поведения домашних животных (например, куры), находящихся на территории острога. При столкновении с игроком – разбегаются в разные стороны.

5) NPC_Animal_Aggressive – модель поведения диких животных (например, медведь), находящихся в лесной территории. При обнаружении игрока или других типов NPC – преследует и атакует.

Каждая модель поведения была подключена к модулю симуляции погодных эффектов. Основной принцип такой способности искусственного интеллекта был реализован системой вопросов и принятием решений. Игровой агент каждую секунду анализирует окружающий его мир и находится в определенном состоянии.

При подключении определенного класса модели поведения, игровой агент проверяет свои возможные состояния и все действия, которые он может совершить. Каждое действие включает в себя определенную цель, это может быть либо определенный временной промежуток, либо достижение физической точки в окружающей обстановке. Разберем данную систему на примере охранника, который патрулирует территорию:

1. Агент получает сигнал о патрулировании местности;
2. Он сверяется с основными контрольными пунктами, которых он может достичь;
3. Он начинает свое движение от одной точки на карте до следующей;

4. При достижении точки, у агента включается таймер, по истечению которого, он возвращается к шагу 3;

Универсальность разработанной системы искусственного интеллекта позволяет организовывать симуляции как для массы игровых агентов, а также для отдельных объектов. Также предоставляется возможность легкой модификации поведения, путем добавления новых запросов на выполнение. Таким способом была дополнена заранее реализованная система патрулирования, путем добавления дополнительных правил поведения в зависимости от времени суток. Если на улице будет день, то искусственный интеллект даст сигнал к патрулированию, после наступления ночи, определенная группа агентов сменятся для симуляции процесса отдыха.

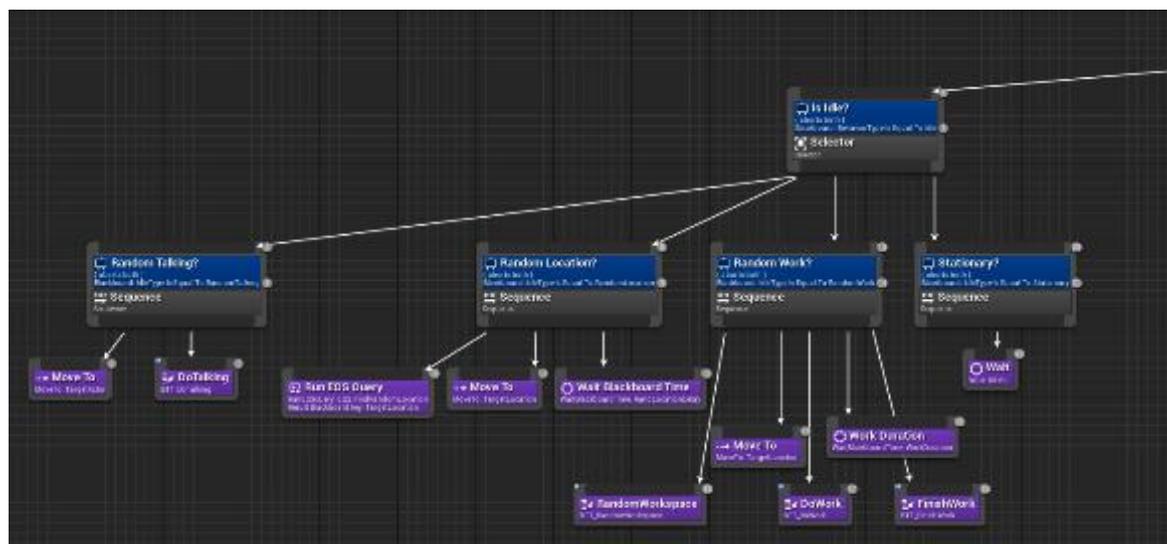


Рисунок 55 – Ветка решений при спокойном перемещении и бездействии.

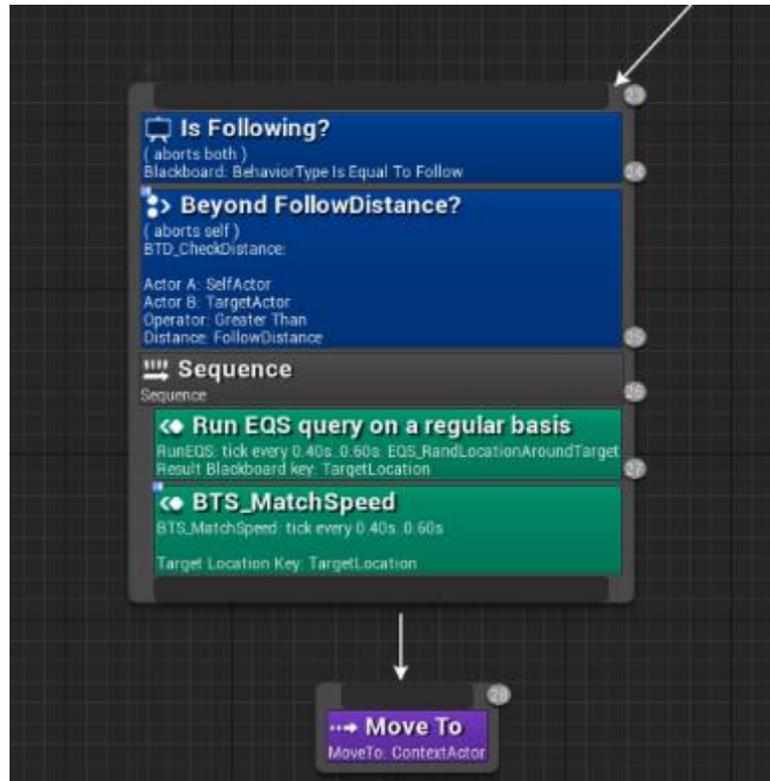


Рисунок 56 – Ветка принятия решений для компаньонов, которые следуют за игроком

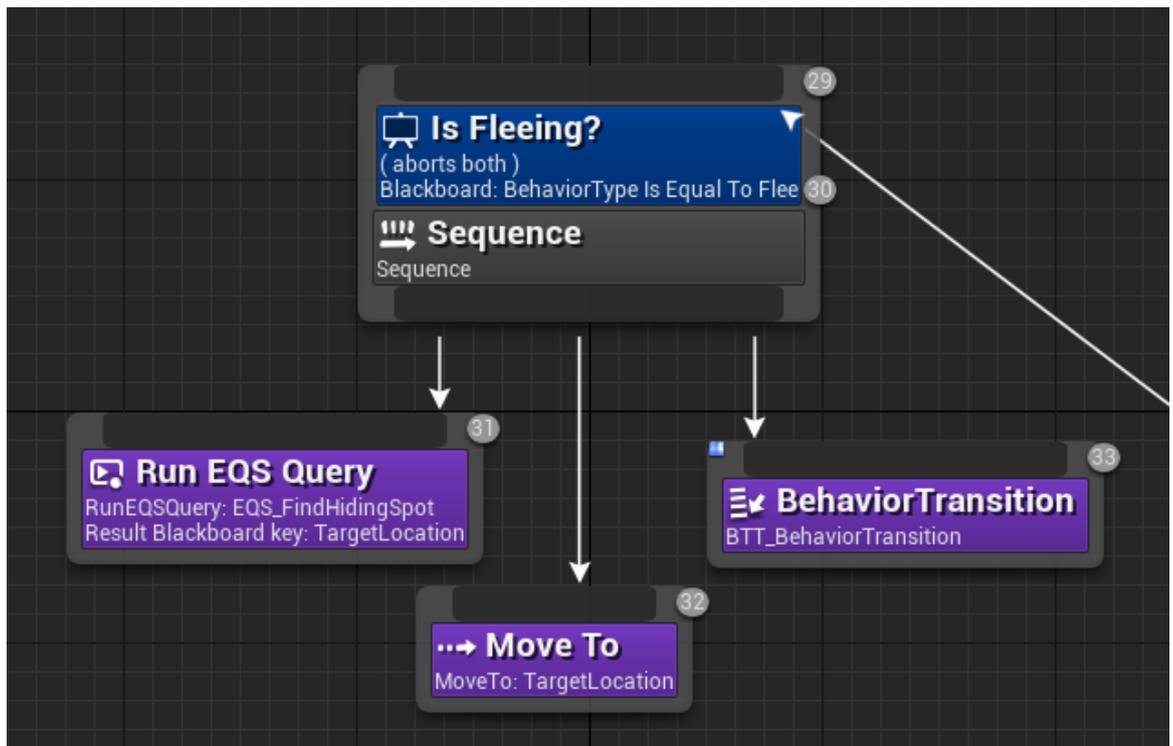


Рисунок 57 – Ветка решения «Убегать» для мирных агентов, в случае опасности

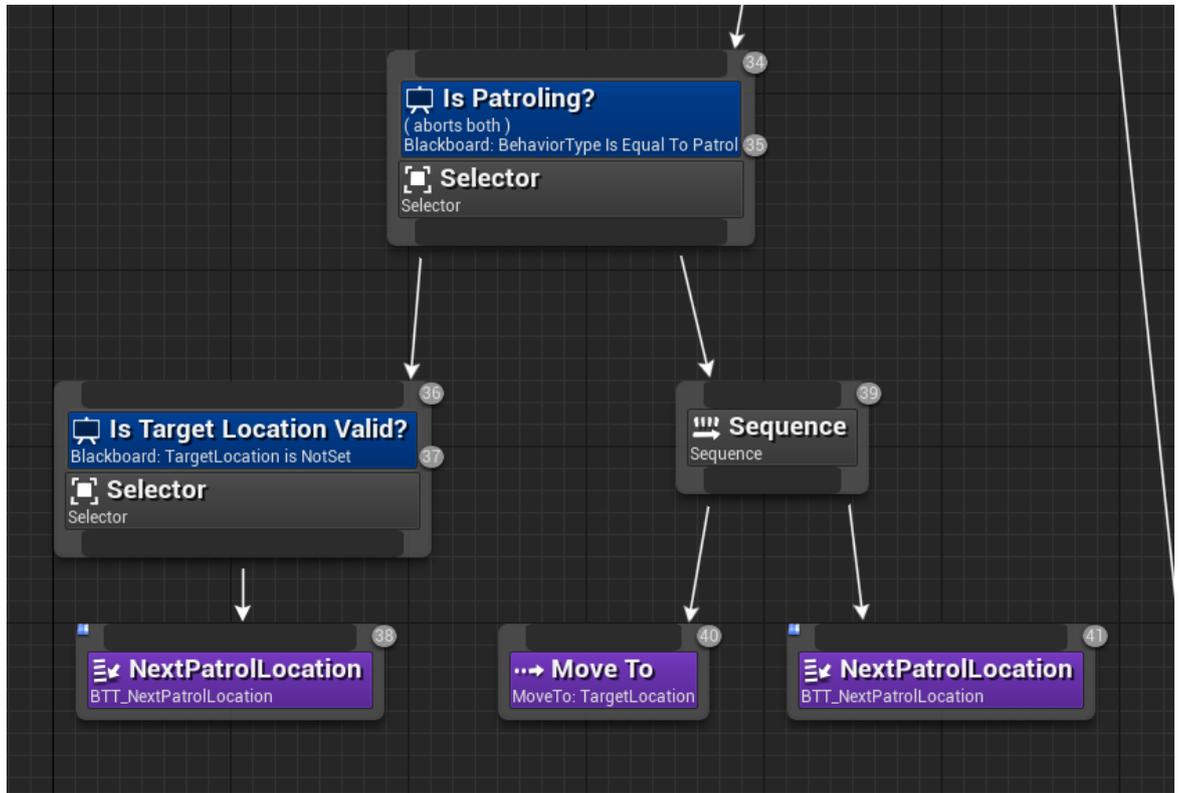


Рисунок 58 – Модель поведения агента при патрулировании территории

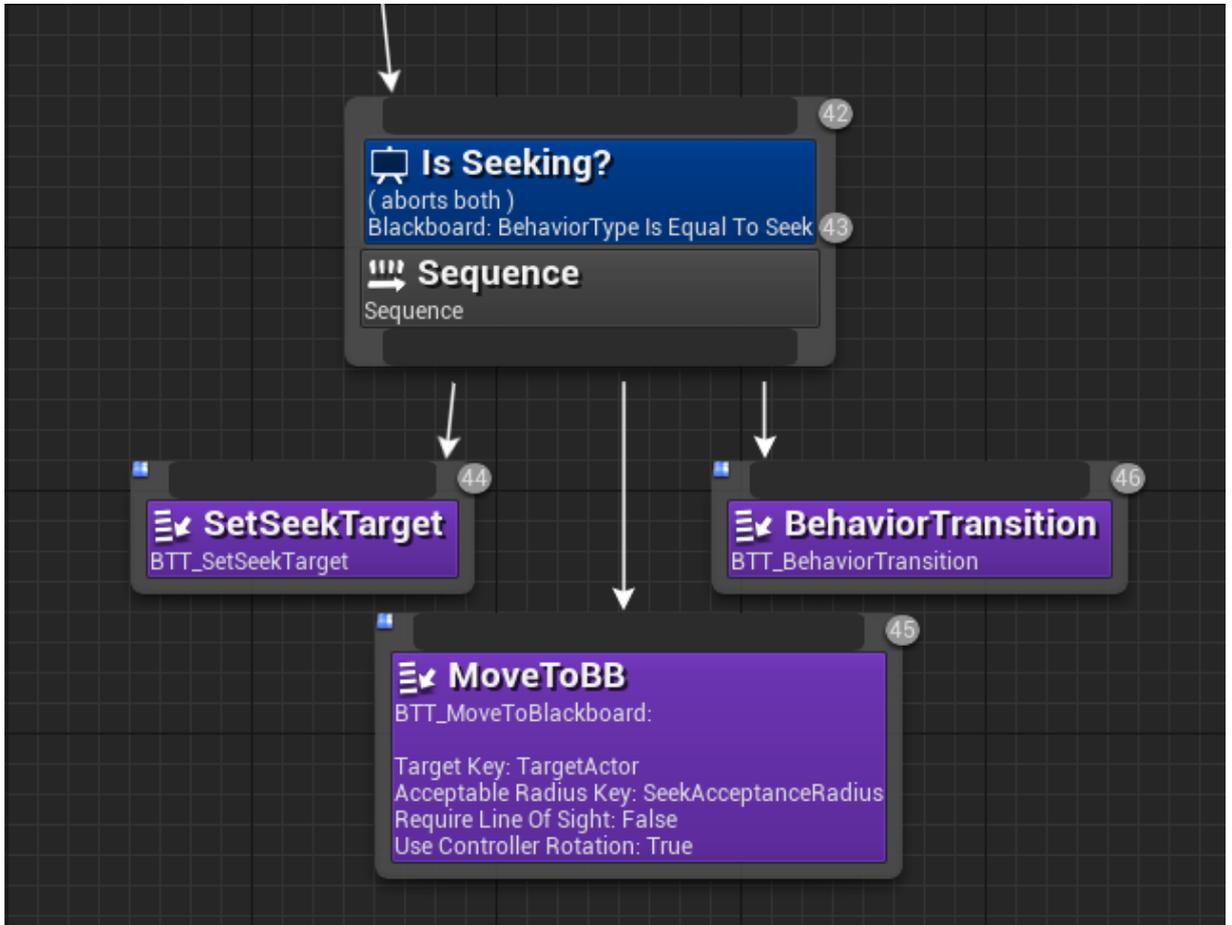


Рисунок 59 – Модель поведения агента во время поиска цели

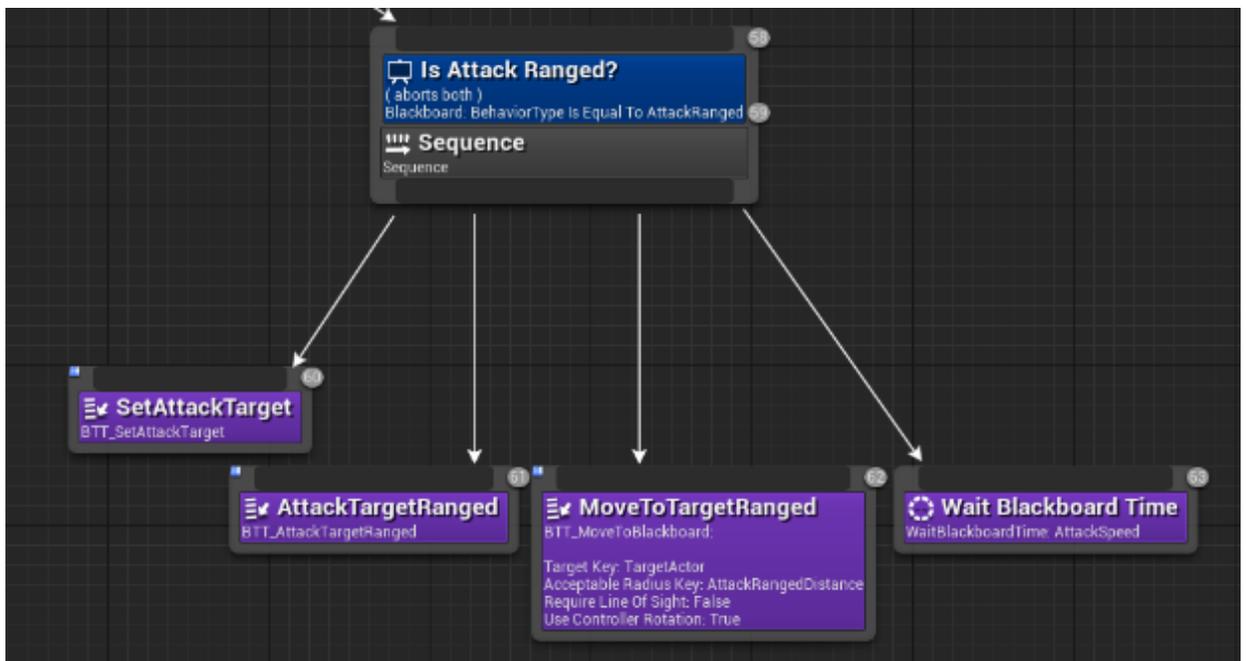


Рисунок 60 – Модель поведения агента при стрельбе из оружия

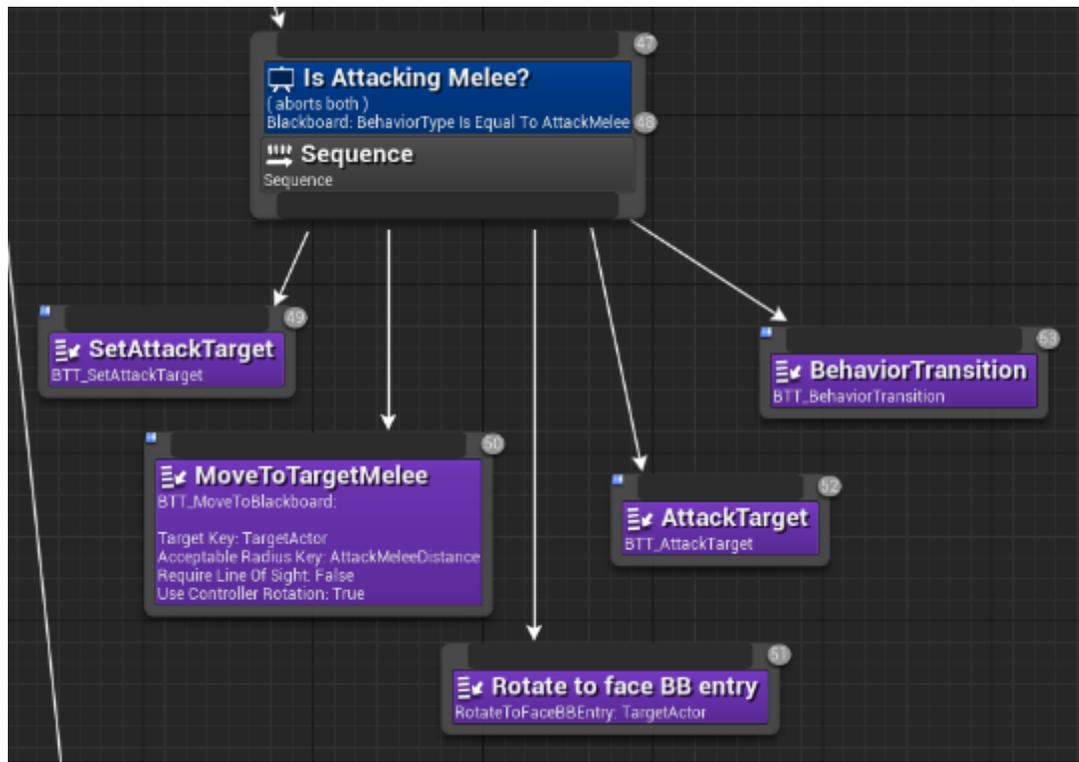


Рисунок 61 – Поведение агента при использовании холодного оружия

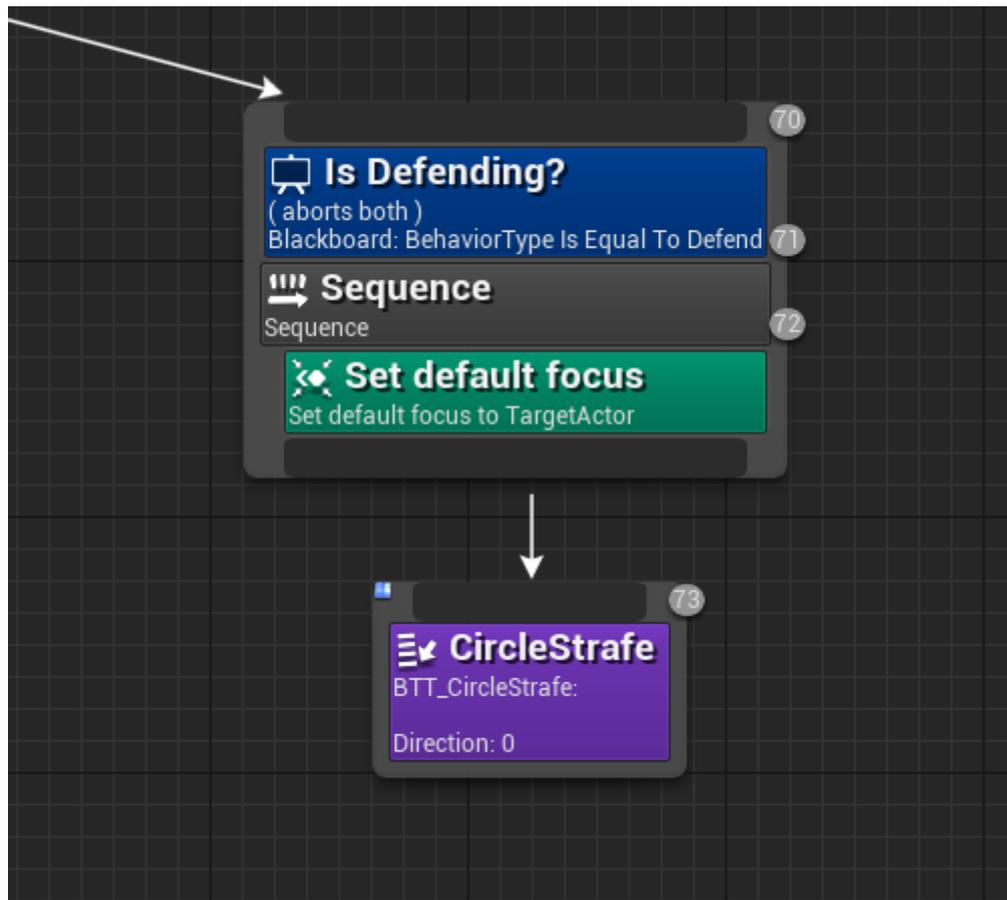


Рисунок 62 – Поведение агента при защите от атаки

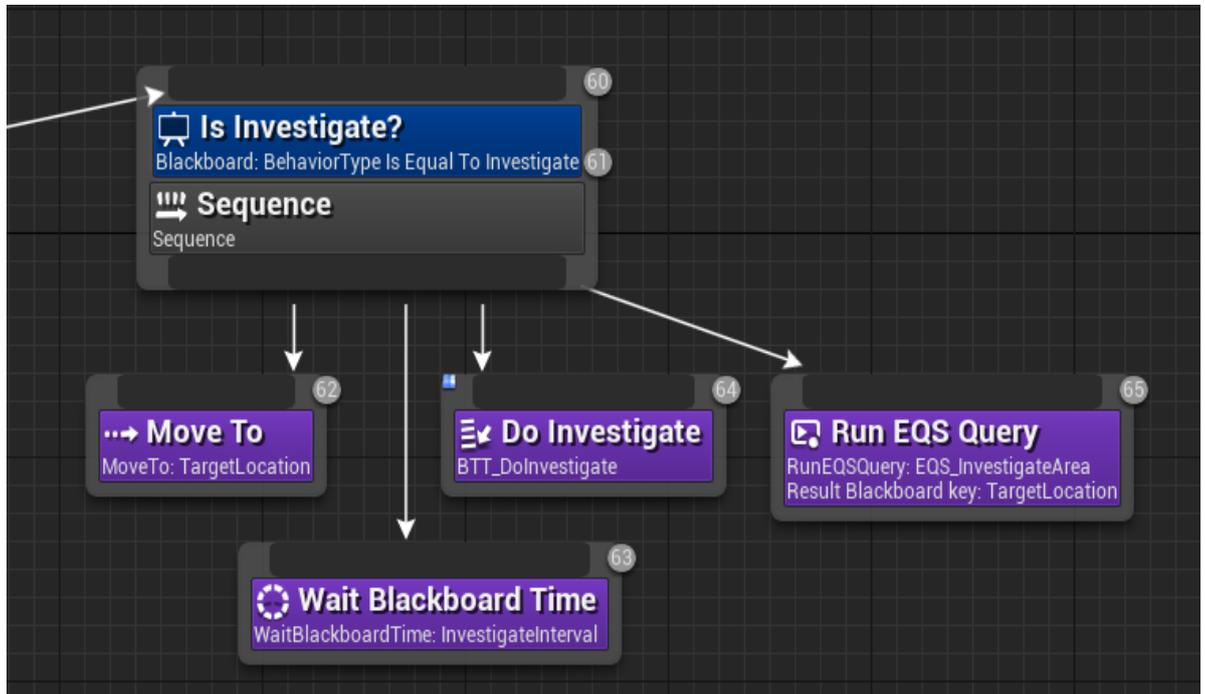


Рисунок 63 – Поведение агента при исследовании территории

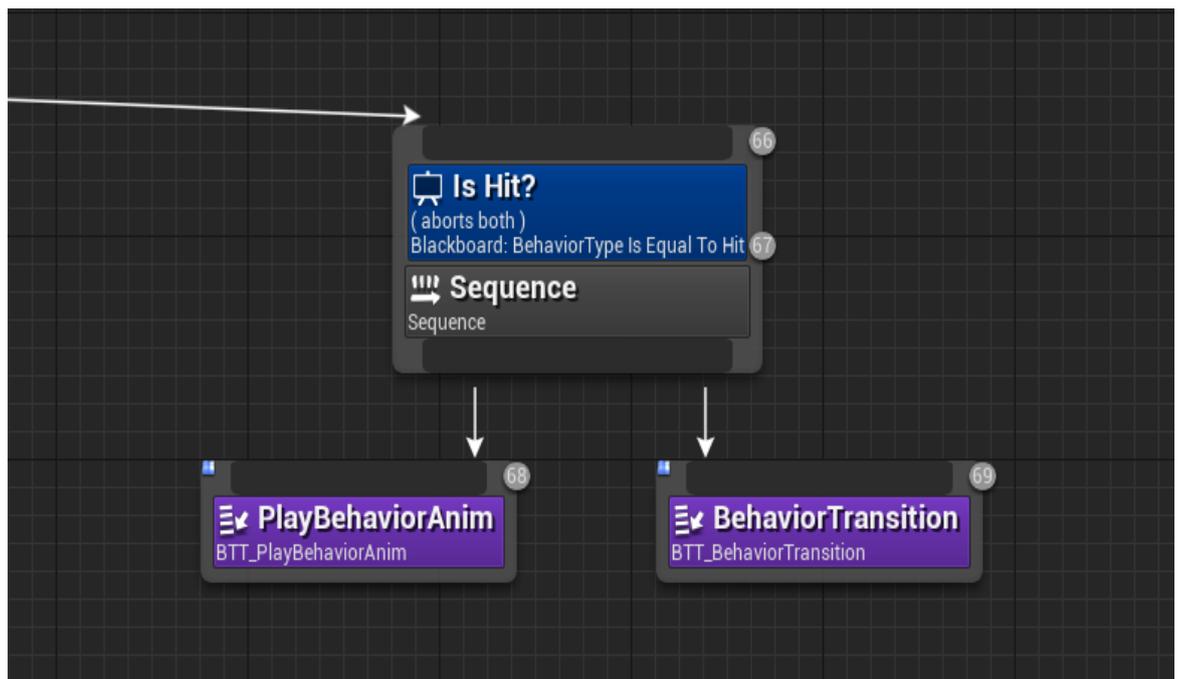


Рисунок 64 – Поведение агента при поражении (смерть)

3.5 Интерактивная часть реконструкции

Для реализации интерактивности компьютерной реконструкции, важно правильно организовать процесс вовлечения игрока. Чтобы ему не было скучно и все было интуитивно и понятно. В среде Unreal Engine основная структура реализа-

ции игрового процесса представлена на рисунке 65.

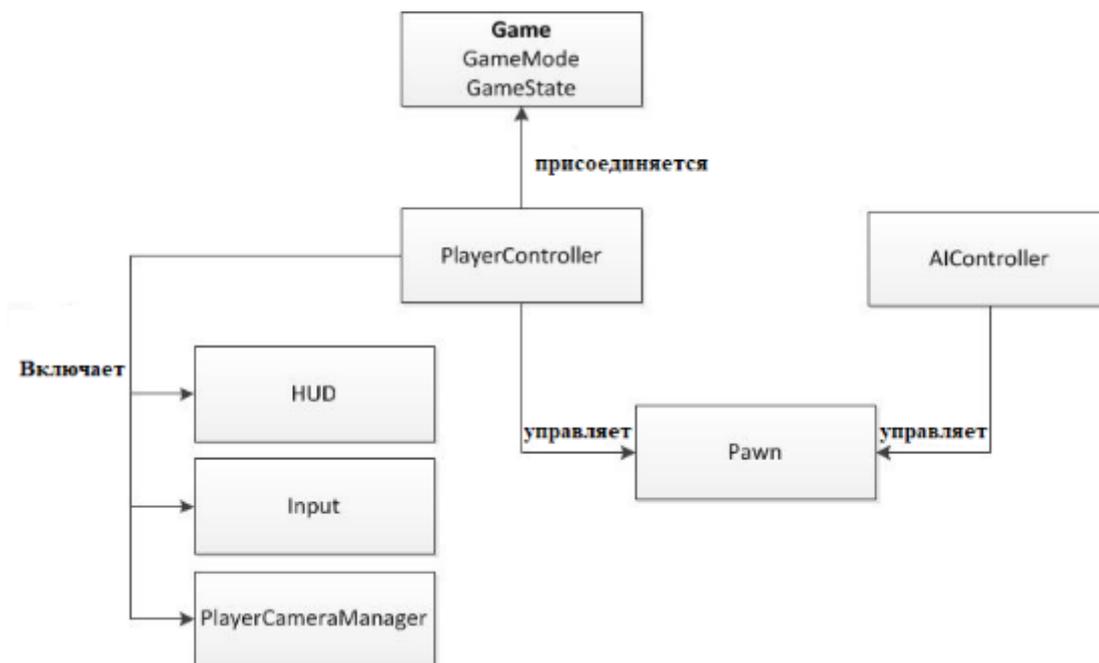


Рисунок 65 – Структура взаимодействия объектов в среде Unreal Engine

Существует два основных класса, которые обрабатывают информацию об игре: Game Mode (с англ. Режим игры) и Game State (с англ. Состояние игры). Даже самая открытая игра имеет основу правил, и эти правила и составляют Game Mode. На самом базовом уровне эти правила включают:

1. Количество присутствующих игроков и зрителей, а также максимальное количество игроков и зрителей.
2. Как игроки входят в игру, также может включать в себя правила для выбора мест появления персонажа.
3. Может ли игра быть приостановлена или нет, и как выполняется приостановка игры.
4. Переходы между уровнями, включая начало или конец игры в кинематографическом режиме.

Когда связанные с правилами события в игре происходят, они должны отслеживаться и передаваться всем игрокам. Для этого информация сохраняется и синхронизируется через Game State. Эта информация включает:

1. Как долго игра была запущена (включая время подключения локального игрока)
2. Когда каждый отдельный игрок присоединился к игре, и текущее состояние этого игрока.
3. Базовый класс текущего игрового режима
4. Началась игра или нет.

Чтобы создать многогранную реконструкцию было принято решение разработать несколько правил ее симуляции. Первым правилом является возможность мирного изучения окружающей среды, взаимодействие с персонажами, выполнение мирных заданий. Второе правило включает в себя симуляцию боевых действий.

Класс Pawn (с англ. Пешка), является базовым классом для всех Actors (с англ. Актеров), которые могут контролироваться игроками или же искусственным интеллектом. Pawn – это физическое представление игрока или объекта искусственного интеллекта в мире симуляции. Это означает, что данный класс будет определять, как визуально будет выглядеть участник симуляции и окружающие его игровые агенты, а также как они взаимодействуют с миром с точки зрения столкновений и других физических отношений.



Рисунок 66 – Персонаж доступный для управления игроком



Рисунок 67 – Персонаж под управлением искусственного интеллекта

Для непосредственного управления персонажем была реализована логическая схема из переменных и функций, части которой представлены на рисунках

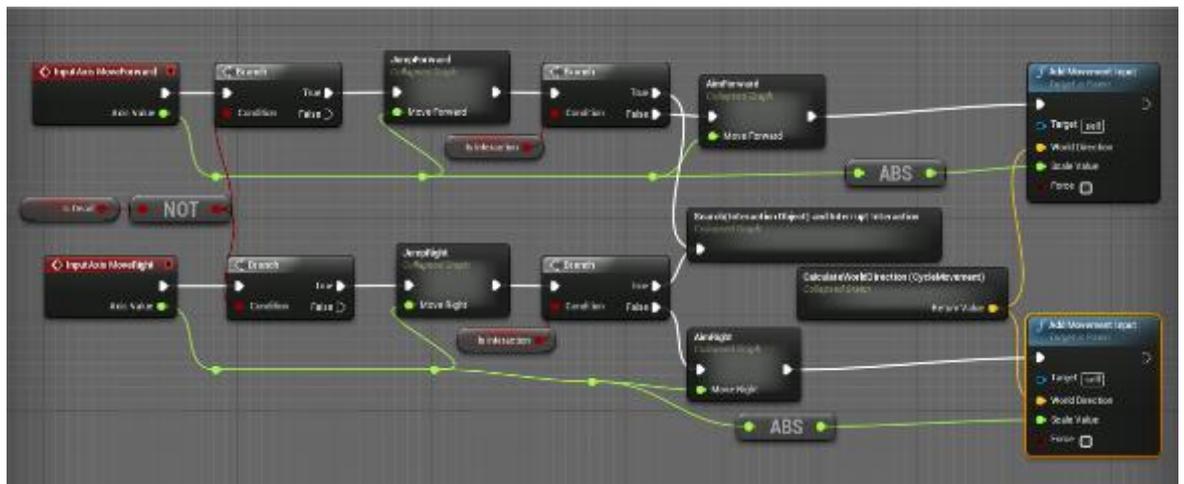


Рисунок 68 – Ввод движения персонажа

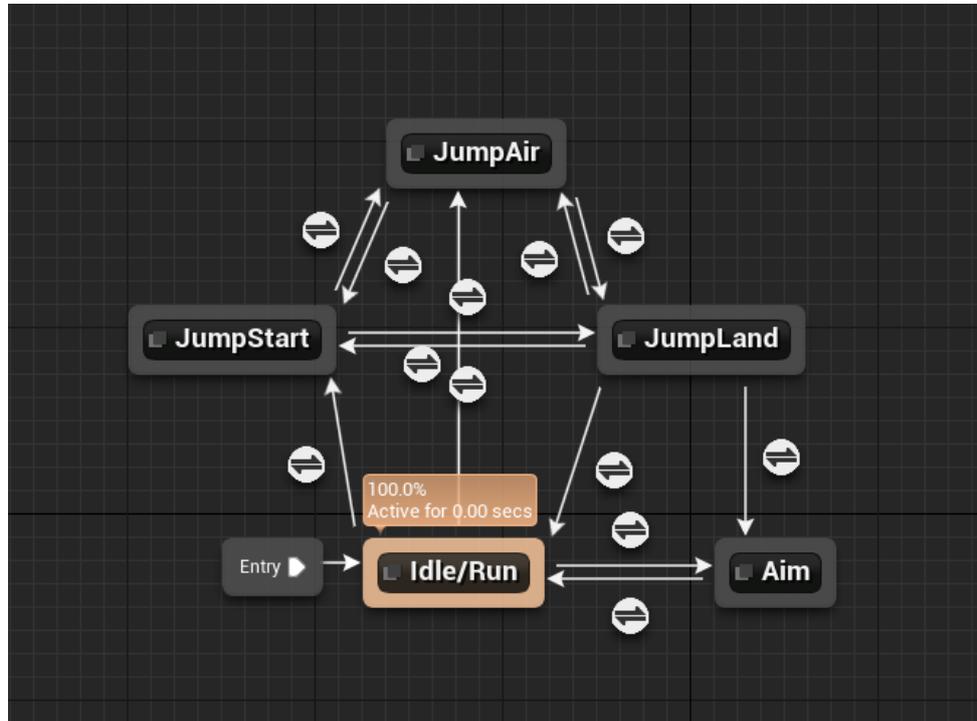


Рисунок 72 – Блок взаимодействий анимаций

При разработке программного обеспечения одной из важных частей является пользовательский интерфейс. Благодаря ему пользователю будет гораздо удобнее ориентироваться в реализованной реконструкции. В среде Unreal Engine для создания интерфейса, необходимо инициализировать класс HUD. Затем с помощью инструмента UMG UI Designer, была произведена работа по развертке интерфейса. При создании сетки интерфейса, необходимо учитывать такие нюансы, как разные соотношения сторон экранов у пользователей. Хотя в настоящий момент особо распространены мониторы с соотношением 16 к 9, но все же было необходимо учитывать и другой сегмент пользователей с соотношением экранов 4 к 3. Все элементы интерфейса на ранней стадии работы были реализованы с помощью внутренних инструментов UMG UI Designer, что позволило ускорить основную разработку остальных элементов. После тестирования, каждый элемент пользовательского интерфейса нужно было переработать в соответствии арт-дизайна компьютерной реконструкции. Для этого был выбран необходимый шрифт, который был интегрирован в среду разработки Unreal Engine. При возможной локализации и дальнейшего распространения для иностранных игроков, также были предусмотрены символы английского алфавита, которые были встроены отдельным мо-

ЗАКЛЮЧЕНИЕ

В соответствии с поставленными задачами в ходе данной работы были выполнены следующие этапы:

1. Выполнен анализ предметной области;
2. Выбраны программные средства для реализации проекта;
3. На основе полученных данных из исследованной предметной области были разработаны компьютерные модели окружения Албазинского острога, а также система погодных эффектов, компьютерный искусственный интеллект, а также интерактивный модуль симуляции.

В ходе написания работы были исследованы исторические факты о событиях 1685 года на Дальневосточной территории Российского царства. Также был выполнен анализ технологии трехмерного моделирования различных по сложности объектов, текстурирования и создания реалистичных материалов. Были рассмотрены основные методы разработки искусственного интеллекта и интерактивных модулей симуляции. Осуществлена непосредственно сама разработка трехмерных моделей, материалов, искусственного интеллекта и модуля симуляции. Была оценена перспективность внедрения информационных технологий в такие направления, как историческая реконструкция.

В результате исследования были получены следующие результаты:

1. Демонстрационные видеозаписи рассматриваемого исторического события.
2. Программа симуляции исторического события.
3. Акт внедрения в учебный процесс лица Амурского государственного университета

					ВКР.155506.09.04.04.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		70

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Макаров, О. Назад из виртуальности / О. Макаров // Популярная механика. – 2011. – № 10 (108) – С. 54-60.
- 2 Михеенко, М. Микеланджело в цифре: моделирование / М. Михеенко // Популярная механика. – 2009. – № 5 (79) – С. 70-74.
- 3 Меженин, А. В. Технологии 3D моделирования для создания образовательных ресурсов / А. В. Меженин. – СПб.: СПбГУ ИТМО, 2008. – 112 с.
- 4 Флеминг, Б. Текстурирование трехмерных объектов / Б. Флеминг. – М.: Изд-во ДМК-Пресс, 2008. – 240 с.
- 5 Ратнер, П. Трехмерное моделирование и анимация человека / П. Ратнер. – М.: Вильямс, 2005. – 277 с.
- 6 Большаков, В.П. Основы 3d моделирования / В. П. Большаков, А. Л. Бочков, А. А. Сергеев. – СПб. : Питер, 2012. – 100 с.
- 7 Дегтярев, В.М. Инженерная и компьютерная графика / В. М. Дегтярев, В. П. Затыльников. – М.: Академия, 2012. – 240 с.
- 8 Грек, А. Нарисованный флот / А. Грек // Популярная механика. – 2017. - №2(172) – С. 36-41.
- 9 Шампандар, А. Искусственный интеллект в компьютерных играх / А. Шампандар. – М.: Вильямс, 2016. – 768 с.
- 10 Келлер, Э. Введение в ZBrush 4 / Э. Келлер. – М.: ДМК Пресс, 2012. – 768 с.
- 11 Гук Д. Ю. Исторические реконструкции: от идеи к виртуальному воплощению / Д. Ю. Гук // Труды IV (XX) Всероссийского археологического съезда в Казани. – Казань: Отечество, 2014. – Т. 4 – С. 294-296.
- 12 Тимофеев, С. М. 3ds max 2014 в подлиннике / С. М. Тимофеев. – СПб.: Изд-во БХВ-Петербург, 2014. – 512 с.
- 13 Что такое PBR: физически корректный рендеринг и шейдинг [Электронный ресурс]. URL: <http://3dyuriki.com/2015/02/26/что-такое-pbr-fizicheski-korrektnyj-rendering-i-shejding/> (дата обращения: 25.02.2017).

					<i>ВКР.155506.09.04.04.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		71

14 Что такое PBR: физически корректный рендеринг и шейдинг 2 [Электронный ресурс]. URL: <http://3dyuriki.com/2015/02/26/chto-takoe-pbr-fizicheski-korrektnyj-rendering-i-shejding-2/> (дата обращения: 26.02.2017).

15 Как создавать PBR текстуры: практический курс [Электронный ресурс]. URL: <http://3dyuriki.com/2015/03/25/kak-sozdavat-pbr-tekstury-prakticheskij-kurs/> (дата обращения: 01.03.2017).

16 Черников, И. И. Энциклопедия мониторов. Защитники речных границ России / И. И. Черников. – СПб.: Изд-во Судостроение, 2007. – 695 с.

17 Попов, И. М. Россия и Китай: 300 лет на грани войны / И. М. Попов. – М.: Изд-во Астрель, 2004. – 547 с.

18 Шведов, В.Г. Оборона албазина: историко-географический анализ / В. Г. Шведов, А. А. Агжитов // Естественно-географическое образование на Дальнем Востоке: мат. IV рег. науч.-пр. конф. – Благовещенск, 2013. – С. 18–26.

19 Топология для анимации персонажей [Электронный ресурс]. URL: http://skif3d.blogspot.ru/2010/05/blog-post_19.html (дата обращения: 02.12.2016).

20 Звягин, К. Трехмерное моделирование и анимация на примерах / К. Звягин, А. Меженин, В. Тозик. – СПб.: БХВ-Петербург, 2008. – 880 с.

21 Уроки по 3d печати [Электронный ресурс]. URL: <http://can-touch.ru/3d-tutorials/> (дата обращения: 15.11.2016).

22 Бондаренко, М.Ю. Плагины для 3ds MAX в примерах/ М. Ю. Бондаренко, С. В. Бондаренко. – М.: КомБук, 2004. – 608с.

23 Мильчин, Ф. М. 3D Studio Max 7.0. Все, что Вы хотели знать, но боялись спросить/ Ф. М. Мильчин. – М.: Бук-Пресс и К, 2005. – 368 с.

24 Верстак, В. А. 3ds Max 8. Секреты мастерства/ В. А. Верстак. – М.: ПИТЕР, 2006. – 681 с.

25 Багрин, Е. А. Комплекс вооружения русских служилых людей в Забайкалье и Приамурье в третьей четверти XVII – начале XVIII вв. (по материалам письменных источников) / Е. А. Багрин // Вестник Бурятского государственного университета. История. – Улан-Удэ: Изд-во Бурятского госуниверситета, 2011. – Вып. 7. – С. 36-39.

26 Бобров, Л.А. Опыт изучения защитного вооружения русских воинов XVI – XVII в. Сибири / Л. А. Бобров, Ю. С. Худяков // Культура русских в археологических исследованиях: сб. науч. тр. – Омск: Изд-во ОмГУ, 2005. – С. 268-275.

27 Багрин, Е. А. Боезапас русских служилых в Сибири и на Дальнем Востоке в XVII веке / Е. А. Багрин // Культура русских в археологических исследованиях: сб. науч. тр. – Омск: Апельсин, 2008. – С. 283-293.

28 Глушаков, С.В. Компьютерная графика / С. В. Глушаков, Г. А. Кнабе. – М.: Харьков: Фолио, 2002. – 500 с.

29 Шикин, Е.В. Компьютерная графика / Е. В. Шикин, А. В. Боресков. – М.: Диалог-Мифи, 2005. – 464 с.

30 Гейлер, Э. М. Photoshop. Полный курс / Э. М. Гейлер. – М.: НТ Пресс, 2009. – 496 с.

31 Мелихов, Г. В. Манчжуры на Северо-Востоке (XVII в.) / Г. В. Мелихов. – М.: Наука, 1974. – 101 с.

32 Миронов, Д. Ф. Компьютерная графика в дизайне / Д. Ф. Миронов. – М.: БХВ-Петербург, 2008. – 560 с.

33 Шлыкова, О. В. Культурная политика региона в контексте глобализации и информатизации общества / О. В. Шлыкова. – М.: РГГУ, 2007. – 843 с.

34 Курс уроков по созданию персонажа [Электронный ресурс] – <http://soohar.ru/sozдание-personazha-the-warrior-chast-1/> (дата обращения: 29.03.2016).

35 Процесс текстурирования персонажа [Электронный ресурс] – <https://www.allegorithmic.com/blog/pounn-character-making> (дата обращения: 29.03.2016).

36 Беспрозванных, Е. Л. Приамурье в системе русско-китайских отношений / Е. Л. Беспрозванных. – Хабаровск: Хабаровское книжное издательство, 1986. – 336 с.

37 Боднарюк, М.К. Компьютерная реконструкция персонажей обороны Албазина // Молодежь XXI века: шаг в будущее: мат. XVII рег. науч.-пр. конф. – Благовещенск: БГПУ, 2016. – Т. 3.– С. 201-202.

					<i>ВКР.155506.09.04.04.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		73

38 Боднарюк, М.К. Компьютерная историческая реконструкция / М. К. Боднарюк, А. В. Вишневский, И. Е. Еремин, А. Н. Черкасов // Ученые заметки Тихоокеанского государственного университета. – 2016. – Т.7 – № 3 – С. 111-116.

					<i>ВКР.155506.09.04.04.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		74

ПРИЛОЖЕНИЕ А

Изображения компьютерной реконструкции



Рисунок А.1 – Южная сторона острога



Рисунок А.2 – Общий план острога

					ВКР.155506.09.04.04.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		75

Продолжение ПРИЛОЖЕНИЯ А



Рисунок А.3 – Крутой берег



Рисунок А.4 – Пушка с ядром

Продолжение ПРИЛОЖЕНИЯ А



Рисунок А.5 – Корабль Цинского Китая на реке Амур



Рисунок А.6 – Модели персонажей реконструкции