

Министерство науки и высшего образования 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 подготовки: 09.04.04 – Программная инженерия
Направленность (профиль) образовательной программы
«Управление разработкой программного обеспечения»

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой




А.В. Бушманов

«10» июля 2020 г.


МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: компьютерная историко-топографическая реконструкция
Албазинского острога


Исполнитель
студент группы 8570м

 20.06.2020 С.Н. Бугаев
(подпись, дата)


Руководитель
профессор, доктор техн. наук

 20.06.2020 И.Е. Ерёмин
(подпись, дата)


Руководитель научного
содержания программы
магистратуры

 03.07.2020 И.Е. Ерёмин
(подпись, дата)

Нормоконтроль

 03.07.2020 В.В. Еремина
(подпись, дата)

Рецензент

 09.07.2020 А.Н. Рыбалев
(подпись, дата)

Благовещенск 2020

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

Факультет математики и информатики
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Зав. кафедрой



А.В. Бушманов

«03» февраля 2020 г.

ЗАДАНИЕ

К магистерской диссертации студента группы 857-ом Бугаева
Станислава Николаевича

1. Тема магистерской диссертации: Компьютерная историко-топографическая реконструкция Албазинского острога.

(Утверждено приказом от 30.04.2020 № 810-уч)

2. Срок сдачи студентом законченной работы (проекта) 20.06.2020.

3. Исходные данные к магистерской диссертации: Исторические летописи, спутниковые снимки, магнитометрическая съемка местности.

4. Содержание магистерской диссертации (перечень подлежащих разработке вопросов): историко-топографическая реконструкция, программное и техническое обеспечение, разработка компьютерной модели реконструкции.

5. Перечень материалов приложения: нет.

6. Рецензент магистерской диссертации: зав. каф. АППиЭ АмГУ, доцент, канд. техн. наук Рыбалёв А.Н.

7. Дата выдачи задания 01.09.2019.

Научный руководитель магистерской диссертации: профессор, доктор техн. наук Ерёмин И.Е.

РЕФЕРАТ

Магистерская диссертация содержит – 71 стр., 24рис., 1 приложение, 31 источник.

ВИРТУАЛЬНЫЕ РЕКОНСТРУКЦИИ, ЦИФРОВАЯ АРХЕОЛОГИЯ, 3D-ТЕХНОЛОГИИ В ИСТОРИИ

Целью данного исследования является максимально обоснованная реконструкция большого деревянного острога основных сооружений острога, имевших место к моменту завершения второго этапа его формирования.

Задачами данной выпускной квалификационной работы являются:

- проанализировать имеющуюся информацию по Албазинскому острогу и создать схему расположения строений;
- разработать детализированную двухмерную карту.
- разработать 3D-модели сооружений острога и рельеф местности.

СОДЕРЖАНИЕ

Введение	8
1. Историко-топографическая реконструкция	10
1.1 Общее понятие историко-топографической реконструкции	10
1.2 Обзор существующих подходов	11
1.3 Имеющаяся на данный момент информация об абазинском остроге	15
1.4 Цель и разработка алгоритма решения задачи	18
1.5 Реконструкция стен и башен острога	19
1.6 Выводы из главы	25
2. Программное и техническое обеспечение	26
2.1 Обоснование выбора программного обеспечения	26
2.1.1 Обзор и сравнение среды геометрического моделирования	26
2.1.2 Обзор и сравнение среды мультимедийной визуализации	37
2.2 Обзор и сравнение среды разработки кода	44
2.3 Аппаратное обеспечение	47
2.4 Выводы из главы	47
3. Разработка компьютерной историко-топографической реконструкции	48
3.1 Жизненный цикл программного обеспечения	48
3.2 Функциональная модель программного продукта	50
3.3 Разработка 3d-моделей ландшафта и сооружений	53
3.4 Реализация интерфейса	57
3.5 Разработка программного кода	62
3.6 Выводы из главы	64
Заключение	65

Библиографический список	66
Приложение 1	71

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 19.401-78 ЕСПД	Текст программы. Требования к содержанию и оформлению;
ГОСТ 2.111-68 ЕСКД	Нормоконтроль:
ГОСТ 2.105-95 ЕСКД	Общие требования к текстовым документам.

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЕ И СОКРАЩЕНИЯ

ПО – программное обеспечение;

ПК – персональный компьютер;

ГИС – геоинформационные системы;

GUI – (Graphical User Interface) графический интерфейс пользователя;

UMG – (Unreal Motion Graphics) дизайнер пользовательских интерфейсов;

UE4 – unreal engine 4

ВВЕДЕНИЕ

Компьютерная историческая реконструкция – воссоздание культуры той или иной исторической эпохи и региона с использованием археологических, изобразительных и письменных источников при помощи информационных технологий. На данный момент она остается достаточно актуальной проблемой. Термин «историческая реконструкция» может употребляться в следующих значениях: во-первых, как восстановление внешнего вида и конструкции объекта, которое основано на его сохранившихся фрагментах, остатках, и имеющейся исторической информации об этом объекте.

Исторические реконструкции процессов, событий и технологий определяются аналогично. Во-вторых, как собственно деятельность, которая направлена на восстановление различных аспектов исторических событий, объектов и т.д. В последние два десятилетия большинство стран охватило движение исторической реконструкции. По некоторым данным существует более 250 групп исторической реконструкции. Сегодня подобные 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.

На данный момент увеличилось внимание к восстановлению объектов историко-культурного наследия, что повлекло за собой возросший интерес не только к изучению историко-культурного наследия, но и к разработке методик и технологий его реконструкции. Наряду с физической реконструкцией нашли своё применение и виртуальные реконструкции.

Албазинский острог является ключевым памятником землепроходческого движения и русского освоения Приамурья в XVII в. в Амурской области. Албазинский острог является крупнейшим укреплением в центре России на Амуре во второй половине XVII века. С Албазиным связаны походы Ерофея Хабарова, экономическое и политическое развитие русского Приамурья, создание первого

в этом регионе воеводства, строительство первого на Амуре православного монастыря. Благодаря этому историко-топографическая реконструкция Албазинского острога является важной частью истории нашего края.

ИСТОРИКО-ТОПОГРАФИЧЕСКАЯ РЕКОНСТРУКЦИЯ

1.1 Общее понятие историко-топографической реконструкции

Историческое моделирование представляет собой создание трехмерных реалистичных моделей зданий или других объектов на основании известных данных о них. Возможности современных 3D редакторов обеспечивают максимальную реалистичность при работе даже над теми объектами, которые не сохранились в настоящее время, так что сравнение модели с реальностью не представляется возможным.

Где востребована историческая реконструкция

Трехмерная реконструкция объектов может быть использована в самых разных сферах:

Создание декораций для исторических фильмов. Здания, памятники и техника необходимы для того, чтобы в точности передать атмосферу определенной эпохи в фильме. 3D реконструкция позволяет создать детальные копии любого объекта на основании доступной информации о его внешнем виде.

Трехмерные модели для компьютерных игр. Если основой сюжета компьютерной игры являются какие-то исторические события, обстановка требует определенного антуража. Внимание большинства игроков, а также критиков виртуального мира часто обращено не только на основное действие, но и на детали окружения.

Виртуальная демонстрация. Так называемые виртуальные экскурсии, предлагающие возможность посетить интересные объекты, не отходя от монитора, позволяют рассмотреть в подробностях те места, в которых в реальности побывать не удастся. Эти экскурсии могут стать как развлекательным мероприятием, так и частью обучающего процесса.

Пространственный анализ для научных целей. Трехмерное моделирование было признано инструментом для пространственного исторического анализа и реконструкции ранее происходивших событий еще в конце XX века.

Как и в других отраслях, где востребованы трехмерные изображения, историческое моделирование подразумевает максимальное соответствие получившегося макета его реальному прототипу.

Отличительные черты исторического моделирования

Для того, чтобы создать реалистичную модель исторического объекта, в первую очередь необходимо найти максимум достоверных источников, на основании которых возможно построить виртуальное трехмерное изображение. Это могут быть гравюры, фотографии и другие изображения, сохранившаяся техническая документация и любая другая информация. Иногда создание трехмерных моделей осуществляется в сотрудничестве с учеными-историками и специалистами архивов.

Важным моментом, без соблюдения которого 3D реконструкция не будет полноценной, является внимание к системам измерений, указанным в тех или иных источниках об объекте. Перед тем, как начать работу над моделью, необходимо привести все количественные показатели к единой системе.

Состояние объекта также учитывается при разработке его трехмерной модели. Если речь идет о здании, необходимо принять во внимание, требуется ли воспроизвести его первоначальное состояние или же речь идет о реконструкции этого же объекта на определенном промежутке времени. То же самое касается техники: например, процесс создания простой модели танка будет отличаться от необходимости разработать тот же танк, неоднократно участвовавший в боях.

Историческое моделирование является очень интересным разделом создания трехмерных объектов, к тому же весьма актуальным на сегодняшний день.

1.2 Обзор существующих подходов

- Фотограмметрия – это научно-техническая область, ориентированная на разработку методов определения форм, размеров, пространственного положения и степени изменения во времени различных пространственных объектов по результатам измерений их фотографических изображений. Термин «фотограмметрия» имеет греческие корни: photos – свет, gramma – запись, metreo – измерение.

К предметам изучения фотограмметрии стоит отнести геометрические и физические свойства снимков, способы их получения и использования для определения количественных и качественных характеристик сфотографированных объектов, а также приборы и программные продукты, применяемые в процессе обработки. Характеристики объекта могут изучаться по его изображению на одиночном снимке или по паре перекрывающихся снимков, полученных из различных точек пространства.

С необходимостью анализа пространства посредством технологий фотограмметрии исследователь сталкивается при работе с изобразительными источниками, такими как фотографии или аэрофотоснимки. Нередко фотографии являются единственным историческим источником, характеризующим облик строения. Анализ перспективы фотографии, степени искажения пространства, выявление размеров строения невозможно без использования технологий фотограмметрии, осуществляемом в специализированном программном обеспечении, например, в пакете PhotoModeler Scanner и его аналогах.

В задачах построения виртуальной реконструкции технологии фотограмметрии играют не последнюю роль. Анализ материалов аэрофотоснимков и правка перспективы фотографии в большинстве случаев осуществляется не в графических редакторах, таких как Adobe Photoshop или GIMP, а в специализированном программном обеспечении; к нему можем отнести: PHOTOMOD 5, PHOTOMOD 5 GeoMosaic, MapEDIT PRO и др.

Среди программ, используемых в фотограмметрии для анализа фотографий и построений трёхмерных моделей на их основе, стоит отметить: PhotoSculpt Textures, 3DSOM, PhotoModeler и Autodesk 123D Catch (ранее Project Photofly).

Так как во времена существования Албазинского острога фото еще не существовало данный метод не может быть реализован в реконструкции Албазинского острога.

- Технология лазерного сканирования. Лазерный сканер (3D сканер) – это аппаратное устройство, анализирующее физический объект и на основе получен-

ных данных создающее его 3D модель. Трёхмерная модель сканируемого артефакта или строения обычно представляется в виде облака точек или готовой трёхмерной моделью. Отметим, что лазерные сканеры появились совсем недавно. Появление на рынке первых лазерных сканеров связано с деятельностью японской компании Cyra Technology. Основателем компании Cyra Technology стала семья Бена и Барбары Какура, благодаря деятельности которых в 1990-х на мировом рынке оборудования стал доступным первый лазерный сканер высокой четкости, который за последние 20 лет получил широкое распространение в среде как технических специалистов, так и гуманитарных (археологов, историков, музееведов и т.д.).

С целью популяризации внедрения лазерных сканеров в гуманитарные исследования для решения задач оцифровки, анализа объектов историко-культурного наследия в 2003 г. компания создала проект CyArk, имеющий целью создание депозитария цифровых копий результатов электронного сканирования объектов историко-культурного наследия, полученных с помощью разработанного фирмой лазерного сканера. Этот некоммерческий проект принёс фирме значительную известность и способствовал распространению практики внедрения аппаратов лазерного сканирования в гуманитарные науки, в частности, в археологии – для решения задач оцифровки археологических артефактов, архитектурных строений, а также рельефа.

Сегодня существует большой выбор разновидностей лазерных сканеров, отметим основные из них: сканеры фирмы CyArk, Optech ILRIS-3D laser scanner, Leica HDS6100, RIEGL LMS-Z390i, IMAGER5006, Topcon GLS-1000, Kreon (серии AQUILON, ZEPHYR и SOLANO), ZScanner и недорогие варианты, такие как, Roland LPX-250, Minolta VI-700, David Laser Skaner, самодельный лазерный сканер на базе технологий Kinect или обыкновенных web камер с программным алгоритмом анализа изображения.

Стоит отметить, что практика современных зарубежных археологических экспедиций в большинстве случаев не обходится без лазерного сканера. Как отмечает Д.С. Коробов, «трёхмерное лазерное сканирование осуществляется при

помощи специализированной и весьма дорогостоящей аппаратуры – 3D сканеров наземного и воздушного базирования». Несмотря на дороговизну, этот способ трёхмерного моделирования получает всё большее распространение в археологии за счёт максимального приближения результатов моделирования к исходному объекту. Причём речь идёт не только о создании трёхмерных изображений археологических находок, но и участков ландшафта в окрестностях памятников, а также видов раскопов и выявленных объектов.

В 2015г. Было произведено лазерное сканирование местности. Полученная модель позволила определять неровности земли: западины и провалы, которые могут быть неразличимы невооружённым взглядом. Однако эти данные не смогли полностью внести ясность в том, как выглядел Албазинский острог.

• Аэрофотосъёмка. Если использование ГИС в гуманитарных исследованиях началось относительно недавно – около 20 лет назад, то история применения аэрофотосъёмки насчитывает уже более 100 лет. «Мощный толчок в развитии аэрофотосъёмки произошёл в ходе Первой мировой войны, когда всеми воюющими странами она использовалась в разведочных целях. После окончания войны активизируется применение аэрофотосъёмки в археологии, проводившейся с самолётов».

В задачах виртуальной реконструкции данные аэрофотосъёмки позволяют создать трёхмерную модель ландшафта и выступить в качестве отправного материала плана территории, где фотография местности с воздуха позволяет уточнить место расположения объекта в пространстве.

В задаче построения виртуальной реконструкции, как правило, большую роль играют плановые аэрофотоснимки (vertical aerial photos), получаемые со спутников или специальных пилотируемых или беспилотных самолётов, вертолётов, гексо-, quadro- или ортокоптеров. Пространственное разрешение снимков зависит от возможностей камеры, а также от грузоподъёмности аппарата. Беспилотные летательные аппараты (Gauji 330x, Xaircraft x650, Dragonfly X4-X8, Microcopter, Conrad Quadrocopter и др.) позволяют исследователю получать фо-

тографии территории с любой высоты, благодаря вертикальному взлёту аппарата, его устойчивости, грузоподъёмности (до 3 кг), возможности удержания высоты по GPS и т.д.

Данный метод довольно дорогой и полученная 3D-модель не даст больших продвижений в ходе работы над реконструкцией острога. Поэтому данный метод не подходит для решения задачи.

1.3 Имеющаяся на данный момент информация об Абазинском остроге

Албазинский острог был крупнейшим на востоке России и являлся форпостом освоения бассейна Амура и был заложен в 1665г. Казачьим атаманом Никифором черниговским в районе городка Якса даурского князьца Албазы на месте зимовья, захваченного в 1651г.. В 1682 г. пришел указ об образовании в Приамурье Албазинского воеводства. В то же время было положено начало строительства большого деревянного острога. В 1685г. большой деревянный острог был разрушен маньчжурским войском. В 1686г. На месте сгоревшего острога был построен новый деревоземляной острог. Который простоял до тех пор, пока в 1689г. Головин подписал мирный договор с маньчжурами.

После подписания Нерчинского договора остатки албазинского гарнизона покинули крепость. Разрушением Албазина закончилась Амурская эпопея XVII столетия, память о которой навсегда сохранилась в сердцах наших земляков.

Сведений о внешнем облике укреплений Албазина времен Н.Р. Черниговского (1665г.-1682г.) немного. Они содержатся в "Росписном списке Албазинского острога, составленном сыном боярским Семеном Вешняковым при принятии его от прикащика Никифора Черниговского" в 1674 г. и в описании острога, составленном в 1684 г. воеводой А.Л. Толбузиным. Согласно этим данным в 1665-1682 гг. Албазинский острог был в плане прямоугольным и имел размеры 28х39 м. Его тыновые стены были усилены по углам с западной прибрежной стороны двумя башнями.

Еще одна башня находилась посередине восточной стены острога. К сожалению, следы этих укреплений при археологических исследованиях пока не

встречены. Ввиду этого реконструкция острога основана исключительно на данных письменных источников.

Информация по острогу (1682г.-1685г.) Сохранилась в росписи Албазинского острога 1684г.. Там сказано:

Албазинский острог, а по острогу строенья башня проезжая с полевою пристепенную сторону с вороты о трех мостах, а мерою в стенах четырех сажен печатных. Полевая стена рублена с подошвенного мосту до среднего в две стены, вышина до кровли той башни четыре сажени печатных и с розвалом, а от розвалу вверх до орла пять сажен. На третьем мосту четыре окна колодные. Крыта тесом, зубцы на кровле, чердак для караула, да на той же башне орел, обит железом белым.

Да на той же проезжей башне три пищали медные на колесах в станках. Пищаль весом 19 пуд. 20 гривенок, ядра в две гривенки, к ней 38 ядер. Другая пищаль весом 9 пуд 10 гривенок, ядром в 2 гривенки, 7 ядер. Третья пищаль весом шесть пуд 20 гривенок ядром гривенка, всех к ней 14 ядер.

А от той проезжей башни острожная стена до наугольной башни, мерою стена восемнадцать сажен, а наугольная башня мерою стена три сажени с полуаршина печатных. А в вышину три сажени до розвалу, розвал два аршина, и з бойницы, крыта тесом двойным, зупцы, а в башне мост.

А от тое наугольные башни к Амуру реке острожная стена до круглые проезжие башни мерою пятнадцать сажен.

А круглая проезжая башня рублена в восемь стен, а стены двойные. От земли до розвалу вышина четыре сажени, а розвал рублен брусчатой вышина два аршина, а на розвале мост, а кругом тое башни перила забраны в косяк и с окны и покрыты перила двойным тесом и с зубцами. Да на той же башне на розвале нарублена колокольня круглая в лапу до первых зупцов вышина четыре сажени и с розвалом, а на розвале шестнадцать столбов, а перила забраны в косяки кругом, а на столбах шатер. Розвал рублен брусчатой, вышина розвалу два аршина, а подволока – забрана в косяк. А с розвалу шатру вышина полпяты сажени печатных, а на шатре крест и с маковицей, а в той башне четыре моста.

А от круглой проезжей башни к Амуру реке до наугольной башни острожная стена длиной двадцать сажен с аршином.

А наугольная башня от Амура вышина от земли до розвалу три сажени два аршина, а розвал два аршина, а на розвале бойницы. Крыта двойным тесом с зубцами, а в башне мост.

А от наугольные башни по Амуру вверх острожная стена сорок четыре сажени.

А в той острожной стене старые две башни, поставленья Никнфорка Черниговского. А в вышину те башни по три сажени: и под теми башнями поделаны бойницы и избы для аманатов и покрыты тесом.

Да в той же острожной стене на углу государской двор на приезд воеводам и приказным людям.

А от Амура реки от государского двора острожная стена до наугольной башни сорок сажен.

А наугольная башня в вышину трех сажен до розвалу. А розвал два аршина, а стена у башни три сажени, а крыта башня тесом двойным с зупцами, а в башне два моста.

А от тое наугольные башни с полевую сторону острожная стена до проезжей же башни двадцать одна сажень с полусаженью.

А от государского двора и до наугольные полевые башни и от наугольные полевые башни до проезжей большой башни и до наугольной башни и от той наугольной башни к Амуру реке до проезжей круглой башни и до наугольной башни с трех сторон по острогу нарублены городни и покрыты двойным тесом с зупцами.

А с внутренюю сторону того острога плетен плетень. А от острогу поперег ширина в аршин и земли насыпан, а в вышину плетень по острогу кругом сажень печатная, полати поперег сажень и с полатай сквозь острог поделаны бойницы, а под полатами сквозь плетень по острогу бойницы же. И на те полати к верхнему бою на острог сделано семь лестниц. А вышина тому острогу до зупцов пол трети сажени.

А по заострожную сторону от Амура реки кругом того острога копан ров и частичк листвяшной бит в землю до Амура реки. А глубина тому рву пол пята аршина, а ширина тому рву три сажени. Да в том же остроге старой острог строения Никнфорка Черниговского мерою в длину осемнадцать сажен, попорег тринадцать сажен да башня с вороты. Да в том же старом остроге великих государей казенной анбар, а в том анбаре всякие великих государей казенные и десятинные и товарные казны прошлые 191 и нынешнего 192 году».

Перечислим основные метрические величины, имеющие место в указанном историческом источнике используя современную лексику:

Восточная проезжая башня имела квадратное сечение и ширину стен в четыре печатные сажени.

От нее до юго-восточной угловой башни отходила стена длиной 18 сажений.

От юго-восточной угловой башни до южной «круглой» проездной башни восьмиугольного сечения отходила стена длиной 15 сажений. От южной проездной башни до юго-западной угловой башни, расположенной на берегу Амура отходила стена в 20 сажений. От юго-западной угловой башни вверх по Амуру проходила стена длинной в 44 сажени.

В западной стене располагались две башни малого острога общей площадью 13 на 18 сажений и государственный гостевой дом. От гостевого двора до северо-восточной угловой башни (размеры не указаны, но были найдены и измерены остатки фундамента) отходила стена 40 сажений.

От северо-восточной угловой башни до восточной проездной башни отходила стена длиной в 21,5 сажени.

В настоящее время раскопками охвачено 15% территории крепости располагаясь.

1.4 Цель и разработка алгоритма решения задачи

Целью данного исследования является максимально обоснованная реконструкция большого деревянного острога основных сооружений острога, имевших место к моменту завершения второго этапа его формирования.

Из уже имеющихся информации и пониманием о нужном конечном продукте было разработан алгоритм решения задачи, состоящий из трех основных этапов.

Первый этап. На данном этапе разрабатывается схематичная двухмерная модель Албазинского острога. Данная схема разрабатывается путем анализа горизонтальных размеров фортификационных сооружений острога, «Роспись Албазинского острога», визуальные данные раскопок острога.

Второй этап. Разработка детализированной двухмерной модели, разработанной на основе ранее полученной схемы и визуальных данных раскопа, непосредственно привязываемых к топографической карте Албазинского городища с нанесенным на нее всеми схемами раскопок, проведенных на данный момент.

Третий этап. Разработка реалистичного рельефа и трех мерной модели острога, реализуемой на основе ранее полученных данных, но т.к. данных недостаточно для создание вертикальной проекции сооружений острога использовался внешний вид аналогичных архитектурных сооружений XVIIIв., а так же данные трехмерного сканирования рельефа Албазинского городища.

Пройдя все три этапа на выходе, получаем достоверную реконструкцию Албазинского острога.

1.5 Реконструкция стен и башен острога

Перед реализацией первого этапа разработки модели острога было обращено внимание на то, что в «Росписи Албазинского острога» помимо однозначно тактируемого аршина равного 72см используется две различно упоминаемые сажени. Например, к описанию плетня, усиливавшего острожные стены, относится следующий фрагмент текста: «... а в вышину плетень по острогу кругом сажень печатная, полати поперег сажень...». Аналогично выглядят измерительные данные, относящиеся и к другим элементам острога, а именно, как уже упоминалось выше, горизонтальные размеры проездной и угловой башен приведены в печатных сажнях, а длина всех внешних стен в обычных сажнях.

Необходимо отметить, что величину «печатной» сажени общепринято считать эквивалентной значению «казенной» сажени XVII века, состоявшей из трех

аршинов, т.е. равной 216 см. При этом во всех имеющихся научных работах, посвященных планиметрическому описанию общего комплекса сооружений Албазинского острога, применяется только указанное определение сажени. В свою очередь, в соответствующий исторический период весьма распространенной была более удобная в употреблении «маховая» сажень, равная 178 см.

С целью анализа возможности использования в практической реконструкции острога двух различных эталонных саженей было построено два макета как показано на рисунке 7, в схеме «А» был использован размер 1 сажень равный 216см а в схеме «Б» был использован размер 1 сажень равный 178см так же была проведена корреляция их значений с размерами северо-восточной угловой башни и отходящих участков острожных стен, обнаруженных в ее археологическом раскопе. Результаты измерений, проведенных на плане раскопа показанные на рисунке 2, показали, что ширина башни составляет более 6,2 м, т.е. три с лишним казенные сажени, а ширина плетня равна примерно 1,8 м, иными словами, одной маховой сажени.



Рисунок 1 – Проверка размеров острога

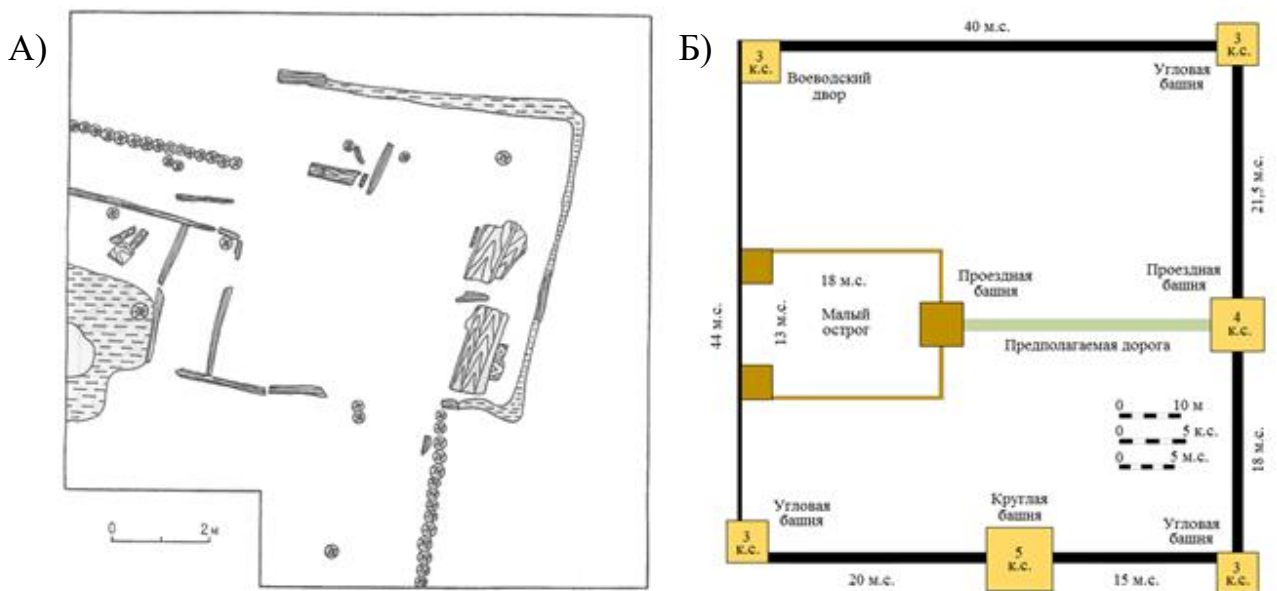


Рисунок 2 – Исходная репликация контура острога

Третья археологическая экспедиция, проводящая раскопки на территории Албазинского острога под руководством А.Н. Черкасова, дополнила традиционную технологию реконструкции его облика более современными инструментами данными аэрофотосъемки и магнитометрического сканирования показанный на рисунке 3А. При этом полученные ею результаты указывают на то, что внешний обвод стен острога имел форму параллелограмма, и совпадают с изначальными выводами первой экспедиции С.В. Глинского.

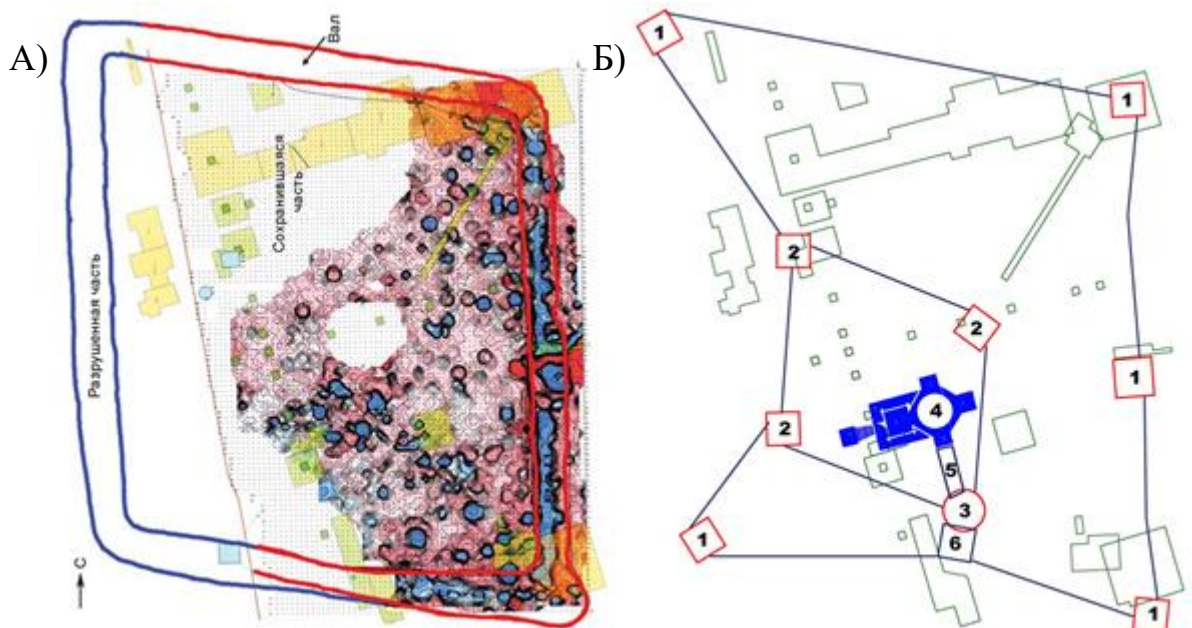


Рисунок 3 – Новейшие реконструкции внешнего контура Албазинского острога 1684 года

Ключевая проблема согласования общего содержания всего имеющегося массива исходных данных заключается в противоречии линейных величин физически наблюдаемого контура исследуемого архитектурного комплекса его габаритным размерам, указанным в «Росписи Албазинского острога». Причина названного обстоятельства объективно обусловлена общепринятой трактовкой величины печатной сажени эквивалентно длине казенной сажени XVII века, равной 2,16 м. Действительно, в этом случае габаритные размеры внешнего обвода крепостных стен, учитывая правильную геометрическую форму острога (прямоугольник или параллелограмм), выходят за физические пределы Албазинского городища. Для выхода из сложившейся ситуации могут быть разработаны репликации контура острога не правильной формы как показано на рисунке 3Б.

Необходимо отметить, что в рамках последней из рассмотренных реконструкций воспроизведен контур острожной Воскресенской церкви упоминаемой в «Росписи Албазинского острога» и встречающейся в других архивных документах по Албазинскому воеводству. При этом данное монументальное сооружение, существенно влияющее на облик острога, явно просматривается на китайском рисунке осады острога, показанного на рисунке 4



Рисунок 4 – Фрагмент рисунка из китайского атласа XVII века

Из данных, которые описанных выше было принято решение принять за основу маховую сажень для стен острога и казенную сажень для башен острога. Также имеет место фактическое соответствия размеров юго-восточной угловой башни, выраженных с помощью казенных и маховых саженей, параметрам раскопа северо-восточной угловой башни. Полагая, что все угловые башни, в том числе и воеводский двор, были однотипными, предполагается использовать выявленное обстоятельство на всех этапах реконструкции.

На рисунке 2Б представлена базовая схема контурной репликации Албазинского острога при составлении которой помимо учета казенной и маховой саженей был внесен следующий ряд авторских предположений:

Во-первых, для определения местоположения малого острога Никифора Черниговского, формально отсутствующего в используемом летописном документе, был обозначен предположительный контур дороги, напрямую связывающей внешнюю (восточную) и внутреннюю (старую) проезды башни.

Во-вторых, была изменена общепринятая ориентация малого (внутреннего) острога, с учетом расположения его наибольшей (продольной) стороны параллельно центральной оси предполагаемого участка проезды дороги.

В-третьих, горизонтальные размеры «круглой» проезды башни, последний этаж которой являлся церковной колокольней, рассматривались тождественными восьмиугольному фундаменту Воскресенской церкви и были условно обозначены квадратом со стороной в пять казенных саженей.

Объективная оценка разработанного схематического плана Албазинского острога, представляющего собой результат первого этапа решения рассматриваемой задачи, позволяет констатировать его достаточно высокую эффективность. А именно, общая совокупность величин горизонтальных параметров острожных стен и основных башен однозначно обеспечивает правильную форму внешнего контура острога в виде прямоугольника, сохраняя принципиальную функциональность всех фортификационных сооружений.

В рамках второго этапа решения задачи прямоугольный контур острога был трансформирован в виде параллелограмма с углами в 105 и 85 градусов,

обосновываемыми осевыми направлениями участков острожных стен, отходящих от северо-восточной угловой башни показанным на рисунке 2А, но при этом полностью сохраняющего линейные размеры исходной геометрической фигуры.

Отметим, что подобная трактовка общего плана Албазинского острога подтверждается данными топографической съемки его остатков, проведенной первой экспедицией С.В. Глинского в 1974 году, а также результатами современной магнитометрической съемки Албазинского городища, выполненной последней экспедицией, руководимой А.Н. Черкасовым показанным на рисунке 3А.

С помощью графического редактора Photoshop трансформированная описанным образом схема была масштабнo совмещена с фрагментом топографической карты городища с нанесенными на него контурами археологических раскопов, проведенных всеми тремя экспедициями. Показанные на рисунке 5.

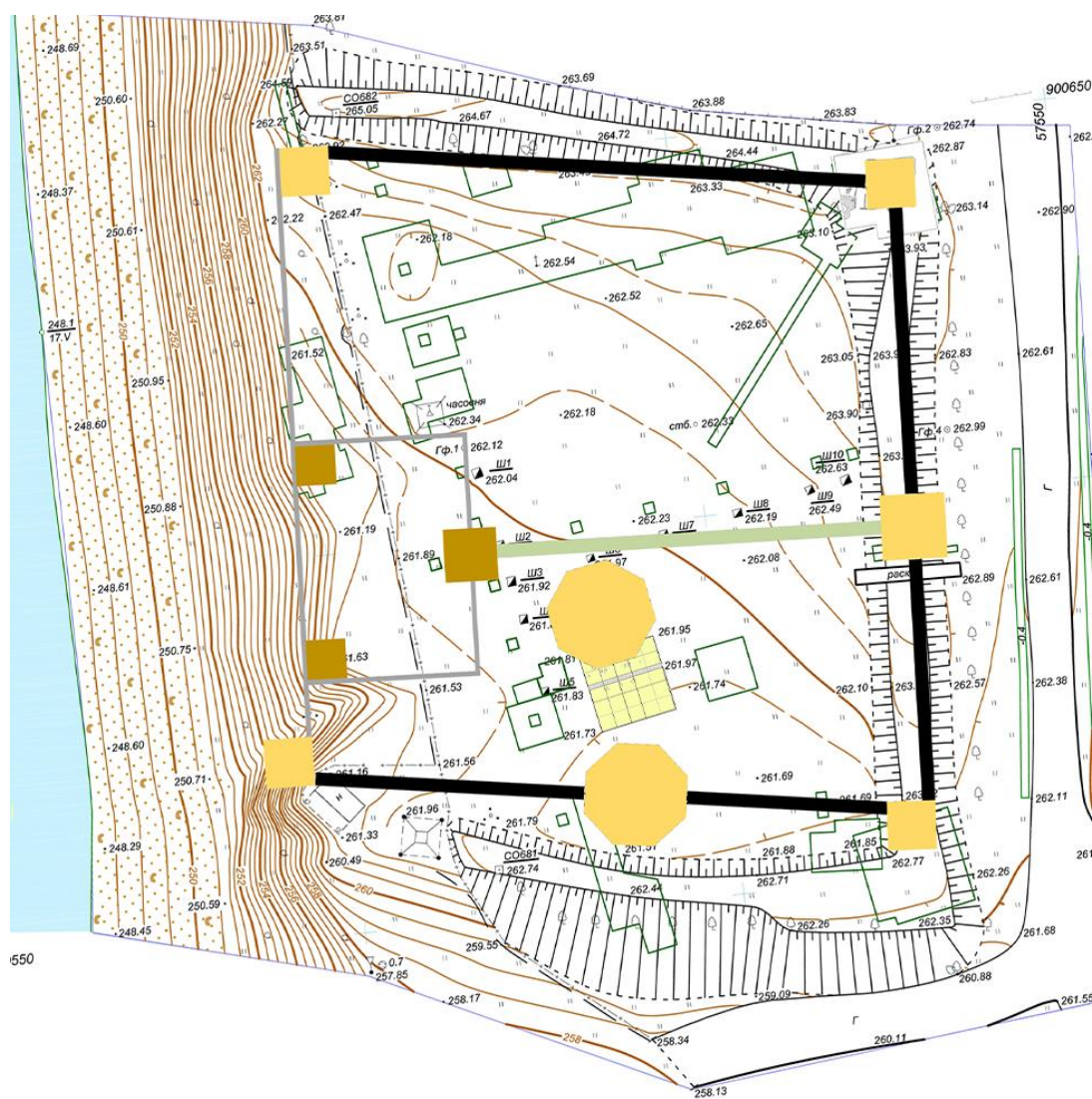


Рисунок 5 – Историко-топографические онтологические репликации острога

Оценивая полученный детализированный чертеж, реализующий итог второго этапа решения задачи, можно констатировать его определенную эффективность, т.е. пригодность для практической проверки корреляции предлагаемых вариантов структуры острога с общепринятыми теоретическими данными.

1.6 Выводы из главы

В первой главе были рассмотрены различные методики реконструкций строений и разработана собственная методика реконструкции Албазинского острога путем изучения исторической документации, совмещенной с использованием новых технологий.

ПРОГРАММНОЕ И ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

2.1 Обоснование выбора программного обеспечения

Обычно среда разработки включает в себя текстовый редактор, компилятор или интерпретатор, компоновщик, отладчик и справочную систему. Иногда также содержит систему управления версиями и разнообразные инструменты для упрощения конструирования графического интерфейса пользователя. Многие современные среды разработки также включают инспектор объектов, браузер классов и диаграмму иерархии классов, которые используются для объектно-ориентированной разработки ПО. Обычно среда разработки предназначается для одного определенного языка программирования, но существуют среды разработки, предназначенные для нескольких языков.

Для создания программного продукта отобрано наиболее удовлетворяющие потребности программное обеспечение. Данное программное обеспечение позволяет максимально эффективно и комфортно решать поставленные задачи.

Основные критерии выбора программного обеспечения:

1. Удобство использования.
2. Функциональность
3. Иметь системные требования к аппаратному обеспечению ПК доступные обычному пользователю.

2.1.1 Обзор и сравнение среды геометрического моделирования

Для разработки программного продукта был выбор программного обеспечения между Cinema4D, 3DS Max, Autodesk Maya, SketchUp, Blender.

Cinema 4D – это универсальная комплексная 3D программа, которая позволяет создавать и редактировать трёхмерные объекты и эффекты.

Программа Cinema 4D позволяет выполнять рендеринг объектов по методу Гуро. Поддерживает как высококачественный рендеринг, так и анимацию.

Главные достоинства программы Cinema 4D:

Максимально простой интерфейс по сравнению с аналогичными программами. Разработчики и вовсе «скромно» утверждают, что: «Cinema 4D – это самая

простая в обращении профессиональная 3D-программа». Люди, которые начинают изучать трехмерную графику именно в C4D, говорят, что программа довольно проста и понятна в освоении по сравнению с тем же 3ds Max. «Синьку» (так в простонародье называют Cinema 4D) очень любят начинающие и студенты. Хотя почитатели той же 3DS Max не всегда разделяют восторг любителей Cinema 4D;

Имеется встроенная поддержка русского языка;

Данные из Cinema 4D можно легко экспортировать в After Effects и наоборот;

Основная программа содержит инструменты для текстурирования, моделирования, рендера и анимации;

Имеются модули, которые открывают доступ к дополнительным специализированным функциям и инструментам программы;

Возможность выбрать определенную версию (так называемые Bundles и Editions), которая будет содержать модули и дополнения под Ваши конкретные задачи. К примеру, для начала можно иметь только основную программу, а потом, уже по мере необходимости, приобретать дополнительные модули;

Помимо основного рендера Cinema 4D может работать также и со сторонними рендерами. Они могут встраиваться как непосредственно в саму среду программы, так и при помощи коннекторов;

C4D позволяет не только импортировать 3D-геометрию в самом широком диапазоне файловых форматов, но также выбирать необходимую Вам модель из сотен имеющихся вариантов;

Многофункциональный набор различных инструментов моделирования Cinema 4D позволяет быстро создавать любые собственные объекты;

Можно анимировать любые параметры объекта;

Функция клонирования позволяет задать сложную анимацию для сотен объектов всего за несколько кликов;

В итоге хочется сказать, что Cinema 4D является универсальной комплексной программой для создания и редактирования двух и трехмерных эффектов и объектов.

Autodesk 3ds Max – профессиональное программное обеспечение для 3D-моделирования, анимации и визуализации при создании игр и проектировании. В настоящее время разрабатывается и издается компанией Autodesk.

Основная функция программы – создание и редактирование 3D графики. Остальные опции предназначены для дополнения созданных объектов и доведения их до реалистичного внешнего вида. Программа оснащена огромным количеством разнообразных модификаторов, инструментов для работы с моделями. 3Ds Max предлагает такие типы проектирования трехмерных объектов:

Полигональное моделирование. Самый распространенный вид 3D-моделирования, встречающийся во многих пакетах трехмерной графики. Может использоваться для разработки моделей различной сложности;

Моделирование на основе примитивов. 3Ds Max содержит встроенную библиотеку стандартных объектов, так называемых примитивов. Во многих случаях создание моделей начинается именно с них, ведь к таким примитивам применимы разнообразные модификаторы;

На основе сплайнов. Также один из базовых способов моделирования. Заключается в построении каркаса изделия из трехмерных кривых (сплайнов). На его основе генерируется сам 3D-объект;

На основе NURBS-кривых. NURBS, или неоднородный рациональный B-сплайн представляет собой особую технологию разработки 3D-моделей. Идеальный вариант для моделирования органики и объектов, имеющих гладкую поверхность;

На основе поверхностей Безье. Особый способ 3D моделирования на основе кривых Безье. Зачастую применяется к отдельным частям 3D модели, для которых создается сеть контрольных точек. С их помощью поверхность можно растягивать в любом направлении.

Autodesk 3Ds Max – это действительно мощная программа для визуализации, совместимая со многими модулями моделирования света, используемых материалов и различных эффектов. Приложение предоставляет возможность гибкого управления настройками, включая экспозицию, глубину резкости, и многое другое.

Окно Material Editor (редактор материалов) в 3Ds Max реализовано по новому принципу, т.е. каждая функция вынесена в отдельное диалоговое окно, за счет чего управлять материалами легко и удобно. В числе визуализаторов для программы такие модули, как Arnold, V-Ray, Mental Ray, RenderMan, FinalRender, Luxrender и многие другие.

Примечательно, что функция создания анимации применима практически ко всем объектам на сцене. В остальном, она реализована очень качественно. Анимации поддаются как целые объекты, так и отдельные их элементы. Присутствуют эффекты движения частиц (огонь, дым, брызги, снег), жидкостные эффекты. Также есть возможность детального моделирования траекторий движения объектов. Пользователь вплоть до малейших перемещений определяет путь передвижения моделей.

Вдобавок ко всему, инструменты анимации можно самостоятельно создавать и редактировать, а с их помощью формировать новые контроллеры анимации в среде создания графов. Не забывайте о динамике твердых и мягких тел, ткани, и многом другом. Помимо прочего программа предлагает возможность видеомонтажа анимационных сцен и множество фильтров изображений.

3Ds Max поддерживает «командную работу» и функционирует с файлами различных программ. Есть возможность экспорта и импорта проектов из других и в другие приложения. Также реализована функция совместной работы над одним проектом при помощи внешних ссылок. Для работы с плагинами, текстурами, изображениями, 3D-моделями, созданными в других приложениях пакет, поддерживает огромное количество типов файлов.

Для представленной программы разработано немереное количество плагинов и дополнений. Каждый из них отвечает за конкретные задачи. В частности,

существует много плагинов для моделирования фотореалистичных эффектов дыма, огня, взрывов, салютов, брызг и т.д. Разработаны также отдельные дополнения, помогающие моделировать ландшафты, звездное небо. Некоторые модули направлены на более удобное создание органики (растений и прочего).

Многие из опций, которые будут описаны ниже, отсутствовали в ранних версиях приложения и появились в числе возможностей лишь некоторое время назад. Предназначение всех этих функций – упрощение работы с 3D-объектами.

Модуль HairandFur. Предназначен для простого моделирования шерсти, волос, травы и прочих подобных элементов. Очень удобный и функциональный инструмент;

Текстурирование и UV-маппинг. Накладывание текстур в последних версиях 3Ds Max значительно упрощено и обеспечивает высокую гибкость;

Моделирование твердотельных объектов. Autodesk 3Ds Max – чрезвычайно удобная программа для этой цели. Пакет оснащен всеми необходимыми инструментами для 3D моделирования Hard Surfaces;

Булевы операции (Booleans). Предназначены для простого и быстрого добавления/вычитания одного объекта из другого. Очень популярная опция в среде 3D моделирования;

Система частиц. Благодаря этой системе, реализованной очень качественно, появляется возможность разработки абстрактных компонентов – дыма, капель дождя, брызг фонтана и т.д.

Конечно, это не все особенности приложения, но в зависимости от версии программы они могут варьироваться. Потому мы сочли нужным рассказать лишь о базовых опциях, доступных практически во всех современных модификациях 3Ds Max.

Безусловно, обзор 3Ds Max не будет полноценным, если не рассказать о сфере применения программы. Ниже мы перечислим сферы, в которых приложение окажется наиболее полезным:

3D моделирование и визуализация архитектурных объектов;
визуализация и дизайн интерьера;

3D моделирование для компьютерных игр;
многопрофильный дизайн;
рекламная анимация;
художественная анимация и создание спецэффектов;
WEB-дизайн и компьютерная графика.

Такая многопрофильная программа, как 3Ds Max может использоваться в самых различных целях и не ограничена перечисленными сферами. Тем не менее, это самые распространенные области, в которых приложение не только чаще всего используется, но и для которых лучше всего подходит.

Autodesk Maya – редактор трёхмерной графики, доступный на Windows, macOS и Linux. Maya обладает широкой функциональностью 3D-анимации, моделирования и визуализации. Программу используют для создания анимации, сред, графики движения, виртуальной реальности и персонажей. Широко применяется в кинематографии, телевидении и игровой индустрии. Изначально разработан Alias Systems Corporation, а затем выкуплен и поддерживается в настоящее время Autodesk, Inc.

Maya названа в честь Санскритского слова मया māyā, майя, означающего «иллюзия». Maya существовала в трёх версиях:

Maya Unlimited – самый полный и самый дорогостоящий пакет. Содержит расширения Hair, Fur, Maya Muscule, Fluid Effects, Cloth и другие.

Maya Complete – базовая версия пакета, в которой присутствуют полноценные блок моделирования и анимации, но отсутствуют модули физической симуляции.

Maya Personal Learning Edition – бесплатный пакет для некоммерческого использования. Есть функциональные ограничения, ограничение на размер визуализированного изображения, пометка водяными знаками финальных изображений.

Однако на выставке «SIGGRAPH 2009» компания Autodesk представила новую версию своей 3D-системы «Autodesk Maya» 2010. Начиная с этого релиза, разработчики отказались от деления программы на «Maya Complete» и «Maya

Unlimited», – теперь «Maya» предлагается как единый продукт. В частности «Maya 2010» содержит всю функциональность «Maya Unlimited 2009» и «Maya Complete 2009», включая «Maya Nucleus Unified Simulation Framework», «Maya nCloth», «Maya nParticles», «Maya Fluid Effects», «Maya Hair», «Maya Fur». Начиная с той же «Maya 2010» в «Maya» включена система композиции «Maya Composite», основанная на программе «Autodesk Toxic», которая более не будет доступна в виде отдельной программы. Кроме этого, также начиная с «Maya 2010», в «Maya» включена система «Autodesk MatchMover», менеджер для составления заданий сетевой визуализации «Autodesk Backburner», пять узлов визуализации для пакетного рендеринга средствами «Mental Ray».

Изначально «Maya» была разработана «Alias Systems Corporation» и выпущена для операционных систем Linux, IRIX, Mac OS X и Microsoft Windows. (На платформе IRIX версия 6.5 стала последней в связи с уменьшающейся популярностью ОС Irix в последние годы.) В октябре 2005 года компания Alias волилась в Autodesk. В сентябре 2007 года – теперь уже Autodesk – выпустила новую версию, получившую имя «Maya 2008». Представители компании в различных интервью подтвердили, что не будут сливать Maya и 3ds Max в один продукт.

Важная особенность Maya – её открытость для сторонних разработчиков, которые могут преобразовать её в версию, оптимальную для каждой студии, предпочитающей писать код, специфичный для своих нужд. Даже невзирая на присущую Maya мощь и гибкость, эта особенность достаточна для того, чтобы повлиять на выбор пользователя.

В Maya встроен мощный интерпретируемый платформенно-независимый язык: Maya Embedded Language (MEL), очень похожий на Tcl и C. Это не просто скриптовый язык, – это средство и способ настроить и доработать основную функциональность Maya (большая часть окружения Maya и сопутствующих инструментов написана на MEL). В частности, пользователь может записать свои действия как скрипт на MEL, из которого можно быстро сделать удобный макрос. Так аниматоры могут дополнять Maya созданной ими функциональностью даже не

Владея языком MEL, оставляя при необходимости такую возможность. Для написания внешних расширений на языке C++ имеется подробно документированный C++ API. (Собственно внешние расширения Maya можно писать на любом компилируемом языке программирования, но наиболее удобен для этого именно C++.) Также для разработчиков теперь имеется возможность написания дополнений на языке Python. Язык MEL не привязан к платформе, поэтому код, написанный на нём, будет исполняться в любой операционной системе, в которой работает Maya.

Файлы проектов, включая все данные о геометрии и анимации, сохраняются как последовательности операций MEL. Эти файлы могут быть сохранены в текстовом файле (.ma – Maya ASCII), который может быть отредактирован в любом текстовом редакторе. Это обеспечивает непревзойдённый уровень гибкости при работе с внешними инструментами. (Похожие продукты Autodesk 3ds Max)

Визуализация в Maya реализована четырьмя встроенными визуализаторами: Maya Software, Maya Hardware, Maya Vector Render и Arnold. Также существует ряд визуализаторов от сторонних разработчиков, в которых включена поддержка Maya. Основные из них: Corona Renderer, V-Ray, Arnold, RenderMan, finalRender, 3Delight, Gelato, Turtle, Maxwell Render, Fryrender, Indigo Renderer, Brazil R/S, mental ray, Redshift, Octane

С самых ранних версий Maya зарекомендовала себя в сфере киноискусства и анимационного кино, в частности с её помощью были реализованы такие кино- и анимационные персонажи, как Стюарт Литтл, Человек-невидимка, Шрек, ВАЛЛ-И, Голлум (Властелин колец), Халк, Дейви Джонс (Пираты Карибского моря) и другие. Программа использована для создания анимационного фильма «Последняя фантазия: Духи внутри», мультфильмов «Южный парк», «Смешарики», «Фиксики», «Фиксики: Большой секрет» и «Маша и Медведь». Неоднократно студии, использующие Maya в производстве визуальных эффектов, были отмечены Американской Академией кинематографических искусств и наук.

Среди них Оскар за визуальные эффекты получили фильмы: «Матрица», трилогия «Властелин колец», «Человек-паук 2», «Кинг-Конг», «Пираты Карибского моря: Сундук мертвеца», «Золотой компас». Оскар в учрежденной в 2001 году номинации «За лучший анимационный фильм» получил и мультфильм «Шрек». Также программа использовалась для создания анимации персонажей из мультфильма Зверополис, однако она была дополнена.

SketchUp (Google SketchUp) – программа для быстрого создания и редактирования трехмерной графики, разрабатываемая компанией @Last Software с 1999 г. и поглощенной в 2006 г. Основной идеей SketchUp является простота интерфейса, что позволяет освоить работу с программой даже непрофессиональному пользователю. Программа реализует концепцию прямого моделирования геометрии, в рамках которой пользователь сначала стоит плоский контур из имеющихся примитивов, затем вытягивает его с целью создания или вычитания объема, после чего придает модели нужную форму посредством перетаскивания ее элементов (вершин, ребер и граней) с помощью указателя мыши.

SketchUp поддерживает экспорт и импорт различных форматов трехмерной и растровой графики. В программе имеются библиотеки компонентов, которые можно пополнять своими элементами, и библиотека материалов. Поставляется в двух вариантах – бесплатном SketchUp и коммерческом SketchUp Pro (\$495). Бесплатная версия входит также в комплект пакета геоинформационной системы Google Earth.

Профессионалы и любители используют SketchUp для архитектурно-строительного проектирования, дизайна интерьеров, проектирования мебели, разработки игр и трехмерной визуализации.

Программа SketchUp может выполняться как на Windows XP/Vista, так и на Mac OS X. Имеет локализации для дюжины языков, включая русский.

Последняя версия SketchUp (8.0) была выпущена 1 сентября 2010 г. [1]

Компания Google поддерживает библиотеку трехмерных моделей (зданий, мостов, машин, мебели, людей, животных, вымышленных персонажей и проч.) с бесплатным доступом к поиску и добавлению новых моделей.

Для программы SketchUp существует огромное количество подключаемых модулей (плагинов) от других разработчиков, созданных с помощью программного интерфейса на языке Ruby. Программы на Ruby являются платформенно-независимыми и интерпретируются во время исполнения. Существует возможность интеграции в них бинарного кода, откомпилированного для конкретной целевой платформы.

Blender – служит для создания трехмерной компьютерной графики. Она используется для 3D моделирования и визуализации – они необходимы для различных сфер деятельности.

В первую очередь, они необходимы для 3D-модельеров – они создают 3D модели зданий, оборудования животных для игр, и менее часто в киноиндустрии.

Во вторую очередь это необходимо для профессии visualizer. Визуализация интерьеров комнат, внешности, выставочных стендов.

В третью очередность это дизайнеры. С целью формирования внешней рекламы, печатной продукции, а кроме того дизайна веб-сайтов. Нередко значительно легче и быстрее создать объект в необходимом ракурсе, нежели находить его и подбирать ракурс либо рисовать, и выходит значительно реалистичнее, таким образом присутствие визуализации объекта предусматривается физические характерные черты объектов. А в случае, если необходимо ввести в дизайн объекты, которые никак не существуют в настоящем мире, в таком случае их возможно только смоделировать либо нарисовать. Таким образом, как программа считается векторной, у нас выйдут весьма высококачественные картинки в итоге. В том числе и в баннерах согласно качеству, они станут выигрывать по сравнению с фотографиями.

Кроме того, Blender станет полезен с целью анимации и эффектов – применяется в кинопромышленности с целью формирования полнометражных и малометражных мультфильмах, в маркетинговой продукции (рекламирование по tv), с целью красивой демонстрации, например, при постройке квартирного комплекса либо предстоящего ремонта в квартире. А кроме того, при применении в

постобработке, формирования разных эффектов в кинофильмах и их монтаж, сочетание 3D и видеоряда, анимации героев в видеоиграх.

Создание интерактивных игр используя движок Блендер. Но чаще всего используют сторонний игровой движок.

Что же умеет этот пакет 3D графики и чем он примечателен:

- во-первых, он является свободным движком 3D-моделирования – это говорит о том, что его можно использовать как в своих, так и в коммерческих целях. А также у него есть открытый исходный код, что позволяет при умении программировать изменять эту программу так как хочется пользователю;

- во-вторых, это вес дистрибутива он составляет примерно 80мб;

- в-третьих, в него интегрировано 6 движков рендеринга, по умолчанию работают 2 Cycles Render и Blender Render;

- в-четвертых, вы можете активировать сторонние плагины как платные, так и бесплатные движки рендеринга такие как V-Ray;

- в-пятых, можно использовать скульптур и ощутить себя профессиональным скульптором. На вооружении в blender имеется большое количество инструментов;

- в-шестых, при сохранении проекта возможно сохранить все текстуры и ресурсы в одном единственном файле, и они никогда не потеряются, как это происходит в других 3D редакторах;

- в-седьмых, он имеет поддержку множества языков интерфейса, и быстрых уведомлений;

- в-восьмых, можно создавать анимацию (видеоролики, полнометражные и короткометражные фильмы).

Работать с редактором видео (редактировать видео и совмещать с объектами 3D сцены и анимацией).

Возможность постобработки (добавление эффектов уже к готовому видео).

Для того чтобы выбрать наиболее подходящее программное обеспечение для реализации проекта была разработана таблица 1.

Таблица 1 – сравнение 3D-редакторов

	Cinema4D	3DS Max	Autodesk Maya	SketchUp	Blender
Система рас- простране- ния	Платная	Платная	Платная	Платная / Ограниченна бесплатная	Бесплатная
Системные требования	ОС: 64х ОЗУ: 4гб ГПУ: 4гб с поддержкой OpenGL 4.1	ОС: 64х ОЗУ: 4гб ГПУ: 4гб	ОС: 64х ОЗУ: 8гб	ОС: 64х ОЗУ: 3гб ГПУ:512мб	ОС: 64х ОЗУ: 2гб ГПУ: 1гб с поддержкой OpenGL 3.3
Простота ин- терфейса	+	-	-	+	+
Русифициро- ванная вер- сия	+	-	-	+	+
Возможность импорта и экспорта	+	+	+	Доступно в платной вер- сии	+

Изучив плюсы и минусы 3D-редакторов выбор программного обеспечения остановился на Blender. Т.к. данное программное обеспечение не требует мощное и дорогое оборудование для работы, легко в освоении и бесплатное.

2.1.2 Обзор и сравнение среды мультимедийной визуализации

Для разработки программного продукта был выбор программного обеспечения между Unity и Unreal Engine.

Для того чтобы сравнить Unity и Unreal было создано для сравнения два идентичных тестовых приложения и охарактеризовавших их показатели в соответствии с системой контрольных точек (эталонное тестирование).

Написав два аналогичных приложения, появилась возможность протестировать оба приложения на ровном игровом поле и узнать какая версия работает, лучше подчеркнув плюсы и минусы при работе в той или иной среде разработки

Для тестирования была создана 3D-сцена, где пользователю предлагалось самостоятельно расставлять объекты на карте. Размеры сетки составляли 11x16,

она вмещала до 176 зданий. Каждый квадрат сетки поддерживал до 6 деревьев, таким образом, в сцене могло быть свыше 1000 деревьев. Был добавлен простой пользовательский интерфейс, где можно было добавить в сцену указанное количество деревьев и зданий. Также добавили функцию добавления зданий в конкретных местах – для этого нужно было щелкнуть по карте в желаемой точке. Что касается организации программы, была построена сетка на плоскости, а просмотр сцены сделали через камеру, расположенную «над головой» пользователя. Также добавлена специальная функция камеры, чтобы пользователь мог масштабировать и панорамировать сцену. Поскольку этот прототип создавался, чтобы определиться с движком для разработки, сделано все возможное, чтобы в обоих вариантах сцена выглядела одинаково. В противном случае было бы невозможно сравнить визуальное качество первого и второго варианта. Для этого пришлось повозиться, поскольку некоторые вещи обрабатываются в Unreal и в Unity по-разному. В итоге получились две очень похожие сцены.

Чтобы унифицировать тестирование производительности, было решено применить в обеих системах идентичные модели деревьев и зданий. Для деревьев использовалась мобильная модель SpeedTree, включавшая как раз около 1000 многоугольников и позволяла хорошо оценить, насколько мелкие инкременты в отображаемых треугольниках сказываются на кадровой частоте. Что касается анимированных зданий, не удалось найти для них такую модель, которая работала бы с обоими движками, поэтому применилось две разные модели. Обе были рассчитаны чуть более чем на 16 000 многоугольников каждая, у них были практически идентичные настройки материалов. Были полностью отключенные уровни детализации (LOD), чтобы в обоих вариантах на любом расстоянии от камеры отображалось одинаковое количество треугольников. Тест проектировался не только с целью отразить различия производительности между двумя движками, но и чтобы показать разницу в качестве рендеринга. Кроме того, приходилось внимательно следить за стандартными шейдерами Unreal Engine. Заметив, что Unreal выглядит явственно лучше, было обнаружено, что в камере действует ряд шейдеров, затратных с точки зрения производительности. После их

отключения сцена визуально почти не изменилась. Освещение представляло совсем другую проблему, поэтому понадобилось некоторое время, чтобы довести его до ума.

Проект Unity плюсы:

Основные элементы Unity – это объекты («GameObject») и компоненты («MonoBehaviour»). Освоив эту концепцию, уже можно работать с Unity. Если правильно пользоваться этой системой, она позволяет значительно улучшить организацию проекта.

В Unity включено много компонентов, обеспечивающих всем необходимым для создания игры – кроме игровой логики как таковой. Как было указано выше, компонент может быть таким маленьким, как Plane (в Unreal отсутствует), который использовался для построения сетки. Новейшие дополнения движка – компоненты «UI» и «Layout», обеспечивающие создание мощных и масштабируемых графических пользовательских интерфейсов.

Редактор можно расширять собственными сценариями, кроме того, в Asset Store доступна масса ресурсов на все случаи жизни. Хранилище Asset Store также содержит множество полезных сценариев, моделей материалов и пр. Они будут особенно кстати при прототипировании – можете просто загрузить все необходимое в виде временных ресурсов и пользоваться этим как подручными имитационными моделями.

Unity был одним из первых общедоступных движков, поддерживавших мобильную разработку. Поэтому он очень удобен при развертывании в мобильной среде, выглядит и действует там практически так же, как и в редакторе. Система постоянно совершенствуется, и развертывание протекает очень гладко. Это был существенный фактор, благодаря которому мы решили делать мобильный прототип.

Пожалуй, Unity может похвастаться самым широким сообществом специалистов среди всех игровых движков, поэтому если возникнет вопрос – скорее всего, ответ на него найдется. Пусть Unity и поддерживает множество языков для

написания сценариев, документация по каждому из них очень основательная. Более того, даже найти ответ, касающийся другого языка, логика этого ответа будет понятна, и будет возможность адаптировать ее для решения своей проблемы.

В Unity проделана огромная работа по оптимизации рендеринга для множества однотипных объектов. Чтобы добиться сопоставимой производительности в Unreal, пришлось бы задействовать Instanced Rendering, а этот механизм обычно менее гибок, чем рендеринг в Unity.

Проект Unity минусы:

Исходный код движка закрыт. Если обсуждать с Unity цену исходного кода, то с этим придется смириться. Поэтому возможны проблемы.

Новая система UI вполне хороша. В ней отсутствует специальный редактор, все изменения вносятся прямо в сцене – а сцена-то очень большая. Когда открывается сцена и хочется отредактировать UI, сначала придется изрядно увеличить масштаб интересующей области.

Проблема с пользовательским интерфейсом Unity связана с масштабированием под дисплеи различного размера. В принципе, у объекта холста есть опции масштабирования с некоторыми параметрами. Но организовать их предпросмотр было очень сложно, пришлось несколько раз развертывать приложение на устройстве, пока не была подобрана оптимальная конфигурация.

Проект Unreal плюсы:

Пробная версия Unreal совершенно бесплатная. В ней получаете полнофункциональный редактор. В Unity также есть бесплатная версия, но переход на Pro обойдется в кругленькую сумму.

В Unreal есть мощный редактор, заключающий в себе несколько узкоспециальных редакторов. Если вы знакомы с этими «вложенными» редакторами, то они очень помогут вам в разработке, а зачастую предоставят и такую информацию, которой в Unity можно не увидеть. Есть редакторы, которые даже могут послужить полноценной заменой некоторым программам. Взаимодействие всех этих подсистем – просто шедевр.

Движок поставляется вместе со всем исходным кодом. Поэтому в нем можно покопаться и понять, как функционируют отдельные детали. Более того, даже можно исправлять баги в движке или самостоятельно дополнять его функционал.

Визуализация в редакторе великолепна. Просто глаза разбегаются от избытка опций рендеринга (связанных, например, с освещением или со сложностью шейдеров). Здесь можно найти массу ультрасовременных шейдеров, которые также поставляются вместе с движком. В принципе, Unreal предлагает наилучший механизм рендеринга на рынке. Можно создавать удивительно красивые сцены.

Чертежи (blueprints) удобны для того, чтобы быстро создать что-нибудь простое и реализовать базовую игровую логику. Они превосходно интегрируются с C++, и такое решение принято неслучайно: оно не только открывает широчайшие возможности как для начинающих, так и для опытных разработчиков, но и позволяет им взаимодействовать друг с другом.

Общая интеграция с C++ великолепна. Как и горячая перезагрузка.

Проект Unreal минусы:

При разработке на Unreal Engine сложно набрать темп. Даже если отлично знаешь C++, потребуется немало времени для изучения различных макросов и функций UE4. Это может быть очень сложно для тех, кто одновременно занимается изучением C++.

В чертежах можно очень быстро запутаться. Когда логика включает десятки узлов, в каждом из которых находится чертеж, то ее иногда удается упростить до пары строк кода, написанных на обычном C++. Обычно это не проблема, поскольку вполне можно работать и с C++, но с некоторыми вещами, например, «UMG» (система UI) использовать чертежи необходимо, поэтому и возможна путаница.

Хотя Unreal и обладает большим сообществом разработчиков, удается редко получать ответы на вопросы. Unreal Engine 4 активно наращивает сообщество, это уже удается неплохо, складывается ощущение, что специалисты стремятся развиваться и помогать. Но сообщество Unity все-таки лучше.

Серьезно не хватает документации по C++. Онлайн-справочный материал по классам C++ неудобен. Кроме того, из-за постоянных обновлений многие возможности быстро устаревают. Надо быть внимательным, просматривая справочные видеоролики, поскольку там может описываться неактуальная версия движка и функции, которые больше не используются.

Работая с GUI, использовали инновационную систему «UMG». Она основана на чертежах и может быть очень полезна. Однако система по-прежнему сырая, в ней недостает некоторых элементов управления, например, кнопок-переключателей. Кроме того, соответствующая документация по C++ практически отсутствует. Редактор несколько раз отваливался, пока разрабатывали UI. Неожиданные отказы могут стоить целых часов рабочего времени.

Результаты:

Удивительно, но при тестировании кадровой частоты (FPS) на разных устройствах мы получили очень несхожие результаты. На некоторых устройствах Unity выигрывал при любой конфигурации. В других случаях Unreal обставлял Unity в тех тестах, где было много зданий. В принципе, Unreal выиграл, но дорогой ценой. Качество изображения и согласованность в Unity были существенно лучше. Текстуры Unreal на некоторых устройствах выглядели расплывчато, деревья получались значительно хуже. Эта проблема была особенно очевидна, если говорить об освещении сцены. Было очень сложно подобрать настройки так, чтобы правильно настроить свет, весь сеттинг зачастую выглядел затемнено. В этом отношении Unity оказался гораздо последовательнее, изображение на любых смартфонах было таким же, как и при предварительном просмотре в редакторе.

Результаты для обоих движков показаны на рисунке 6.

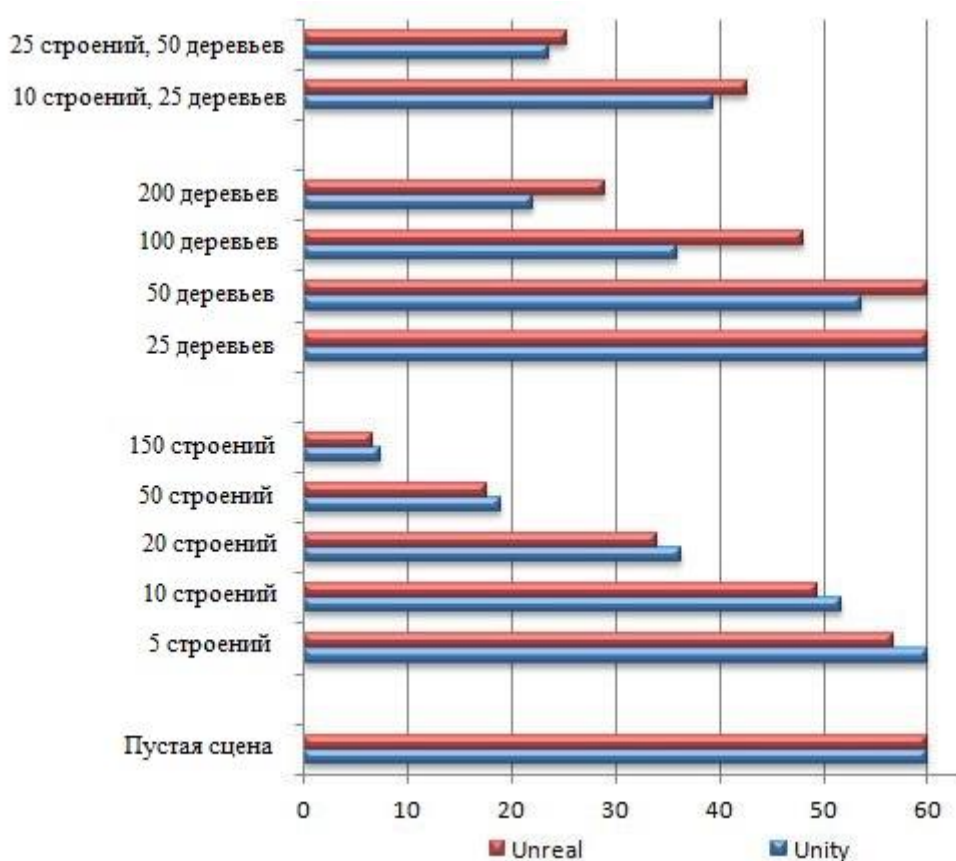


Рисунок 6 – Сравнение unity и unreal

Оба движка примерно с одинаковым успехом отображали анимированные модели. Однако тесты с деревьями оказались безрезультатными, поскольку Unreal не отобразил никаких текстур на моделях деревьев.

В конечном итоге, эталонное тестирование лишь показало, что два движка почти равны по силе. Поэтому особенно важно обращать внимание на конкретные достоинства и недостатки каждого из них. В конце концов, если производительность настолько близка, то на первый план выходит удобство и скорость разработки для каждого из этих движков.

По результатам исследования многое понравилось в обоих движках. Кроме того, обнаружено много областей, где эти инструменты можно улучшить, чтобы программисту было удобнее работать. Ни один из движков не имел существенного перевеса над другим, учитывая, как быстро изменяются их возможности и поддержка. Тестирование рендеринга показало, что оба продукта выжимают из устройства максимум и, в принципе, вполне сопоставимы. Опять же, в этой си-

туации на первый план выходит удобство использования и интуитивная понятность разработки. Учитывая все, что узнали об этих движках, и с какими проблемами столкнулись при разработке, оказался перед непростым выбором, но в итоге отдал предпочтение Unity.

Вот основные доводы в пользу нашего решения:

Unity визуально выглядит более согласованно на всех устройствах, кроме того, быстро разворачивается «одним щелчком» на любой платформе.

Unity занимает на устройстве гораздо меньше места, меньше сказывается на работе конечного пользователя.

Unity гораздо проще изучить и понять. Вооружившись Unity, неопытный пользователь может приступить к работе быстрее и создавать продукты, поддержку которых гарантирует большое сообщество специалистов.

Длительность итерации в Unity гораздо меньше (развертывание и компиляция исходного кода происходит быстрее, шейдеры компилируются почти мгновенно)

2.2 Обзор и сравнение среды разработки кода

Для реализации проекта язык программирования выбирался исходя от использования среды мультимедийной разработки т.к. в данном проекте средой мультимедийной разработки является Unity то выбор среды разработки кода ограничился с# и java.

C# – объектно-ориентированный язык программирования. Разработан в 1998-2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного

и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Название «Си шарп» (от англ. Sharp – диез) происходит от буквенной музыкальной нотации, где латинской букве C соответствует нота До, а знак диез (англ. Sharp) означает повышение соответствующего ноте звука на полутон, что аналогично названию языка C++, где «++» обозначает инкремент переменной. Название также является игрой с цепочкой $C \rightarrow C++ \rightarrow C++++(C\#)$, так как символ «#» можно представить состоящим из 4 знаков «+».

Из-за технических ограничений на отображение (стандартные шрифты, браузеры и т. Д.) и того, что знак диеза # не представлен на стандартной клавиатуре компьютера, при записи имени языка программирования используют знак решётки (#). Это соглашение отражено в Спецификации языка C# ECMA-334. Тем не менее, на практике (например, при размещении рекламы и коробочном дизайне), «Майкрософт» использует знак диеза.

Названия языков программирования не принято переводить, поэтому язык называют, используя транскрипцию, – «Си шарп».

C# разрабатывался как язык программирования прикладного уровня для CLR и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает BCL. Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем (однако, эта закономерность была нарушена с выходом C# 3.0, представляющего собой расширения языка, не опирающиеся на расширения платформы .NET). CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, сборка мусора не реализована в самом C#, а производится

CLR для программ, написанных на C# точно так же, как это делается для программ на VB.NET, J# и др.

JavaScript (/ˈdʒɑːvɑːskript/; аббр. JS /ˈdʒeɪ.ɛs./) – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией стандарта ECMAScript (стандарт ECMA-262).

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

Название «JavaScript» является зарегистрированным товарным знаком корпорации Oracle в США.

JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам – функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания – что придаёт языку дополнительную гибкость.

Несмотря на схожий с Си синтаксис, JavaScript по сравнению с языком Си имеет коренные отличия:

- объекты с возможностью интроспекции;
- функции как объекты первого класса;
- автоматическое приведение типов;
- автоматическая сборка мусора;

анонимные функции.

В языке отсутствуют такие полезные вещи, как стандартная библиотека в частности, отсутствует интерфейс программирования приложений по работе с файловой системой, управлению потоками ввода-вывода, базовых типов для бинарных данных, стандартные интерфейсы к веб-серверам и базам данных, система управления пакетами, которая бы отслеживала зависимости и автоматически устанавливала их.

2.3 Аппаратное обеспечение

ПК который должен удовлетворить потребности пользователя для решения их функциональных задач должны обладать следующими минимальными характеристиками:

Процессор: x64, от 2ггц;

Видеокарта: 1гб;

Оперативная память объемом от 2гб;

Жесткий диск объемом 30гб.

На ПК должно стоять программное обеспечение Windows XP и выше, библиотека Direct 9.

2.4 Выводы из главы

Во второй главе произведен анализ ПО и выбрано наиболее актуальное ПО для разработки виртуальной реконструкции. Для данного проекта было выбрано следующие ПО – среда геометрического моделирования Blender, среда разработки кода Visual studio, среда растровой графики Adobe Photoshop, среда мультимедийной визуализации Unity 3D.

РАЗРАБОТКА КОМПЬЮТЕРНОЙ ИСТОРИКО-ТОПОГРАФИЧЕСКОЙ РЕКОНСТРУКЦИИ

3.1 Жизненный цикл программного обеспечения

Знание различных моделей жизненного цикла и умение их применять на практике необходимы любому руководителю проекта. Правильный выбор модели позволяет грамотно планировать объемы финансирования, сроки и ресурсы, необходимые для выполнения работ, сократить риски как разработчика, так и заказчика. Это способствует повышению авторитета разработчиков в глазах заказчика и в свою очередь оказывает влияние на перспективу дальнейшего сотрудничества с ним и другими заказчиками. Считать, что спиральная модель лучше остальных, неверно. Ведь на каждый проект заключается отдельный договор с определенной стоимостью. Заключать договор на большую сумму с неопределенным итоговым результатом заказчик никогда не будет. В этом случае он предложит вложить вначале небольшую сумму в проект и уже по результатам первой версии (итерации) будет решать вопрос о заключении дополнительного договора на развитие системы.

Каждая из моделей имеет свои достоинства и недостатки, а также сферы применения в зависимости от специфики разрабатываемой системы, возможностей заказчика и разработчика. В таблице 2 приводится сравнительная характеристика рассмотренных выше моделей, которая должна помочь в выборе стратегии для конкретного проекта.

В таблице 2 не стоит рассматривать значения «Да» и «Нет» как жесткие требования. Например, незначительное изменение требований по мере развития проекта при использовании каскадной модели (например, добавление некоторых непредусмотренных сервисных функций) встречается не так уж редко и в случае их реализации способствует улучшению взаимоотношений между сторонами. Аналогично распространение промежуточного программного обеспечения при спиральной модели необязательно, а иногда даже вредно отражается на процессах внедрения и опытной эксплуатации системы.

Таблица 2 – Сравнение моделей жизненного цикла

Характеристика проекта	Модель		
	<u>Каскадная</u>	<u>Инкрементная</u>	<u>Спиральная</u>
Новизна разработки и обеспеченность ресурсами	Типовой. Хорошо проработаны технология и методы решения задачи		Нетиповой (новаторский). Нетрадиционный для разработчика
	Ресурсов заказчика и разработчика хватает для реализации проекта в сжатые сроки	Ресурсов заказчика или разработчика не хватает для реализации проекта в сжатые сроки	
Масштаб проекта	Малые и средние проекты	Средние и крупные проекты	Любые проекты
Сроки выполнения проекта	До года	До нескольких лет. Разработка одной версии может занимать срок от нескольких недель до года	
Заключение отдельных договоров на отдельные версии	Заключается один договор. Версия и есть итоговый результат проекта	На отдельную версию или несколько последовательных версий обычно заключается отдельный договор	
Определение основных требований в начале проекта	Да	Да	Нет
Изменение требований по мере развития проекта	Нет	Незначительное	Да
Разработка итерациями (версиями)	Нет	Да	Да
Распространение промежуточного ПО	Нет	Может быть	Да

Спиральная модель ЖЦ делающая упор на начальные этапы ЖЦ: анализ и проектирование. На этих этапах реализуемость технических решений проверяется путем создания прототипов. Каждый виток спирали соответствует созданию фрагмента или версии ПО, на нем уточняются цели и характеристики проекта, определяется его качество и планируются работы следующего витка спирали. Таким образом углубляются и последовательно конкретизируются детали проекта.

Разработка итерациями отражает объективно существующий спиральный цикл создания системы. Неполное завершение работ на каждом этапе позволяет переходить на следующий этап, не дожидаясь полного завершения работы на текущем, как и показано на Рисунке 7. При итеративном способе разработки недостающую работу можно будет выполнить на следующей итерации. Главная же задача – как можно быстрее показать пользователям системы работоспособный продукт, тем самым активизируя процесс уточнения и дополнения требований.

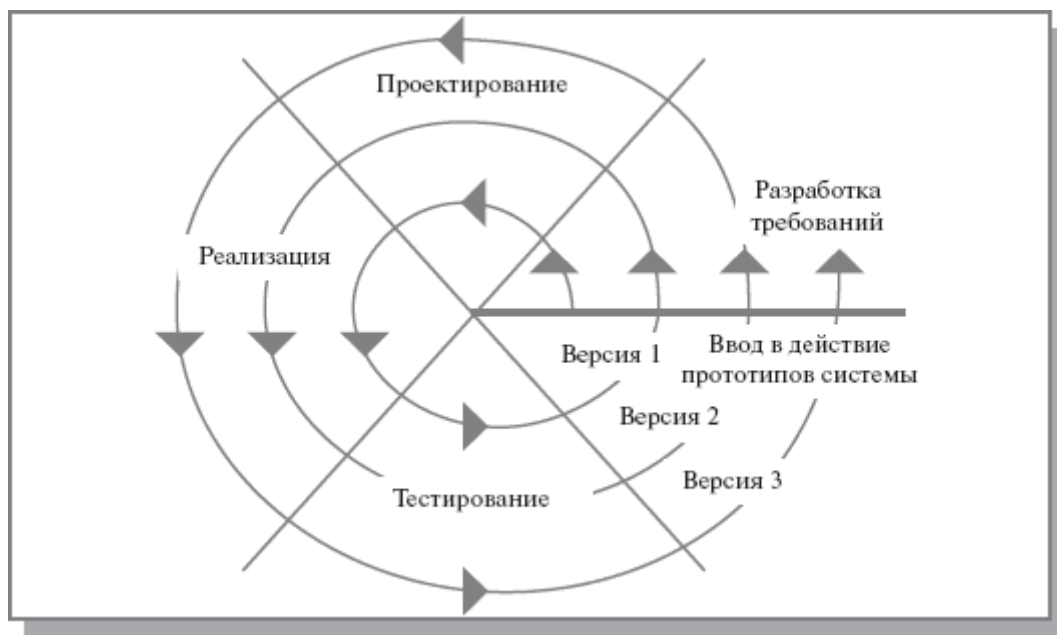


Рисунок 7 – Спиральная модель жизненного цикла

Так как при разработке реконструкции Албазинского острога новые входные данные могут сильно повлиять на ход разработки реконструкции, необходимо не завершая один цикл начинать новый учитывая прошлый опыт работы, кроме того, в уже разработанное программное обеспечение могут вноситься изменения по мере развития. Данному требованию полностью удовлетворяет спиральная модель, показанная на рисунке 7.

3.2 Функциональная модель программного продукта

Для создания функциональной модели была использована диаграмма IDEF0 она предназначена для моделирования решений, действий и деятельности организации или системы и была получена из установленного языка графического моделирования структурированного анализа и проектирования Методика (ТСУР) , разработанной Дуглас Т. Росс и Softech, Inc. В своем первоначальном

виде, IDEF0 включает в себя как определение графического языка моделирования (синтаксис и семантика) и А описание комплексной методологии для разработки моделей. ВВС США по заказу разработчиков SADT «разработать функции модели метод для анализа и передачи функциональной точки зрения системы. IDEF0 должна помочь в организации системного анализа и способствовать эффективной коммуникации между аналитиком и клиентом через упрощенных графических устройств».

В случае, если функциональный блок – схема используется для отображения функционального потока в продукте, IDEF0 используются для отображения потока данных, управления системой, а также функционального потока жизненного цикла процессов. IDEF0 способен графически представляющих широкий спектр бизнеса, производства и других видов деятельности предприятия на любом уровне детализации. Это обеспечивает строгое и точное описание, а также способствует согласованности использования и интерпретации. Это хорошо проверено и доказано на протяжении многих лет использования со стороны правительства и частного сектора промышленности. Он может быть создан с помощью различных компьютерных графических инструментов. Многие коммерческие продукты, специально для поддержки разработки и анализа IDEF0 диаграмм и моделей.

Связанная методика, интеграция Определение для информационного моделирования (IDEF1x), используется в дополнение к IDEF0 для систем с большими объемами данных. Стандарт IDEF0, стандарты Federal Information Processing Publication 183 (FIPS 183), а также стандарт IDEF1x (FIPS 184) поддерживаются Национальным институтом стандартов и технологий (NIST).

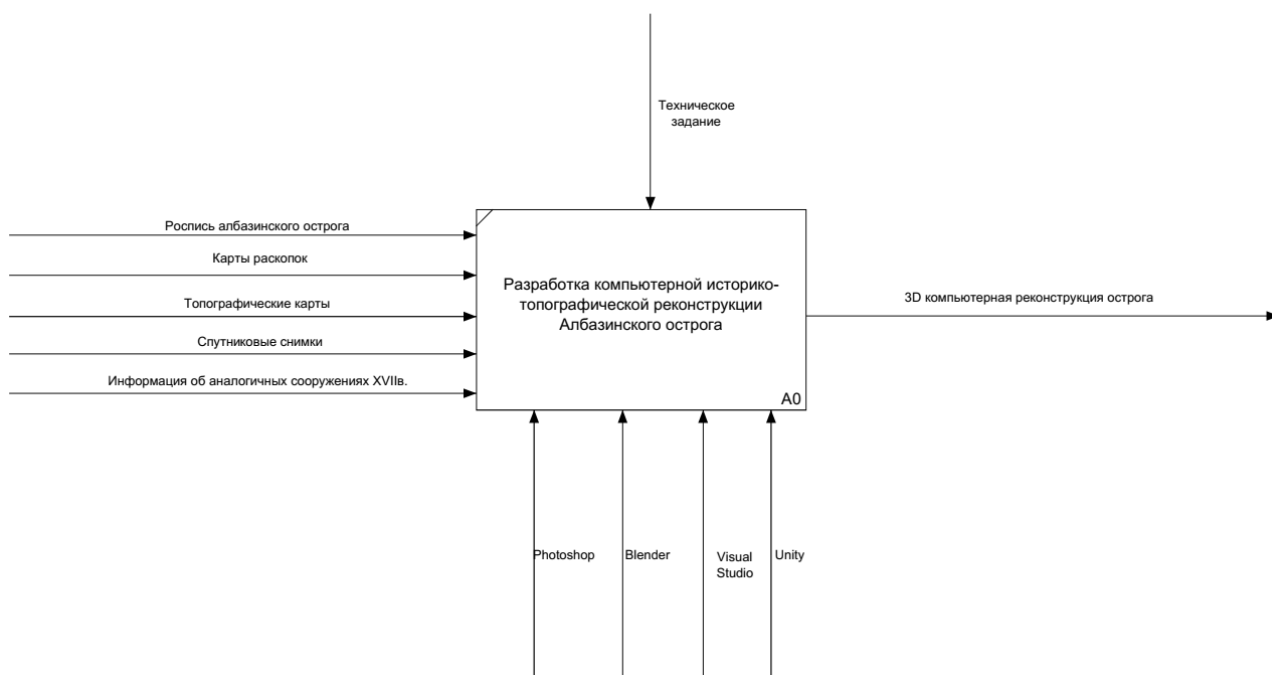


Рисунок 7 – Функциональная модель разработки ПО

На рисунке 7 видно, что для разработки компьютерной историко-топографической реконструкции управляющим механизмом является техническое задание. Для работы требуется такое ПО как Photoshop, Blende, Visual studio, Unity. На вход получаем роспись Албазинского острога, карты раскопок, технические карты, ситниковые списки, информация об аналогичных строения 17в.. На выходе получаем готовую 3D компьютерную реконструкцию.

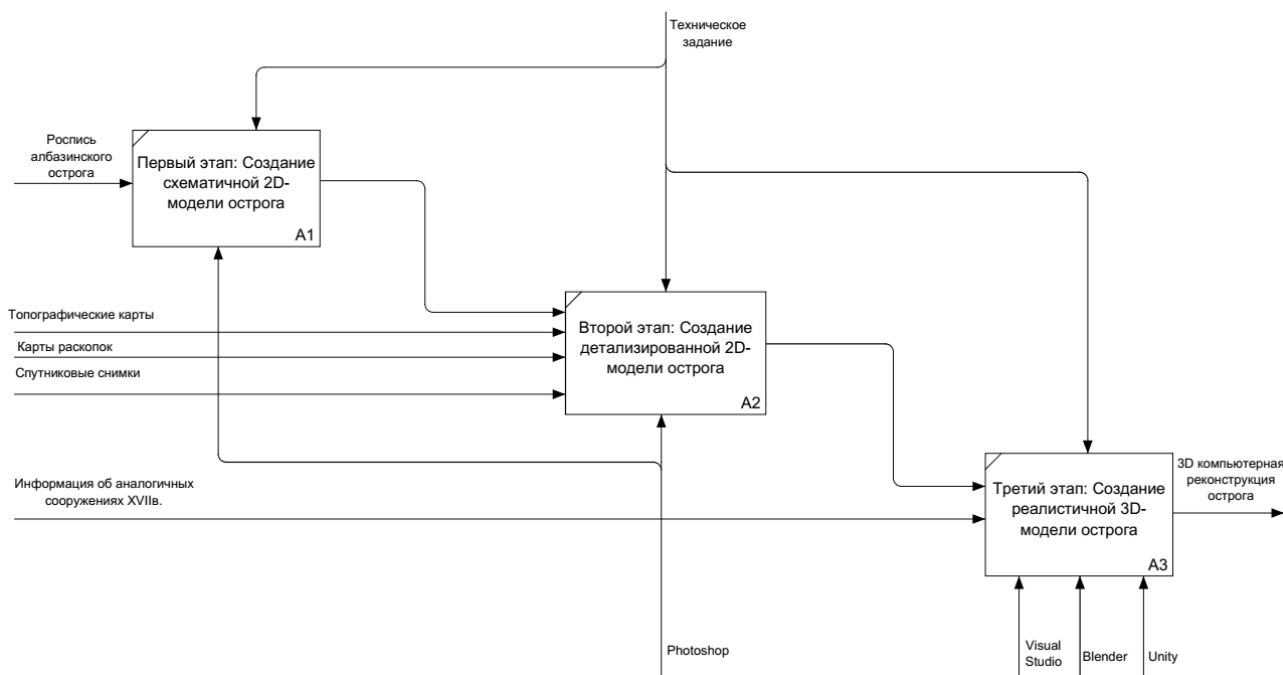


Рисунок 8 – Декомпозиция функциональной модели разработки ПО

На декомпозиции функциональной модели, показанной на рисунке 8 видно, что на первом этапе создания схематичной двухмерной модели острога используется роспись Албазинского острога и программа photoshop для создания схематичной модели острога. На втором этапе используются схематичная модель Албазинского острога, полученная на первом этапе и данная схема дополняется топографическим планом, картой раскопок, спутниковые снимки получая на выходе детализированную карту раскопок. На третьем этапе используется детализированная карта, полученная на втором этапе разработки и информация об аналогичных строениях 17в. с помощью этих данных и программного обеспечение такого как visual studio, blender, unity создается реалистичная 3D-модель Албазинского острога.

3.3 Разработка 3D-моделей ландшафта и сооружений

Для разработки ландшафта изначально был выбран метод выгрузки карты высот из открытых источников, но как оказалось в рамках реконструкции острога данные карты имеют крайне плохую детализацию. Пример такой карты показан на рисунке 9.

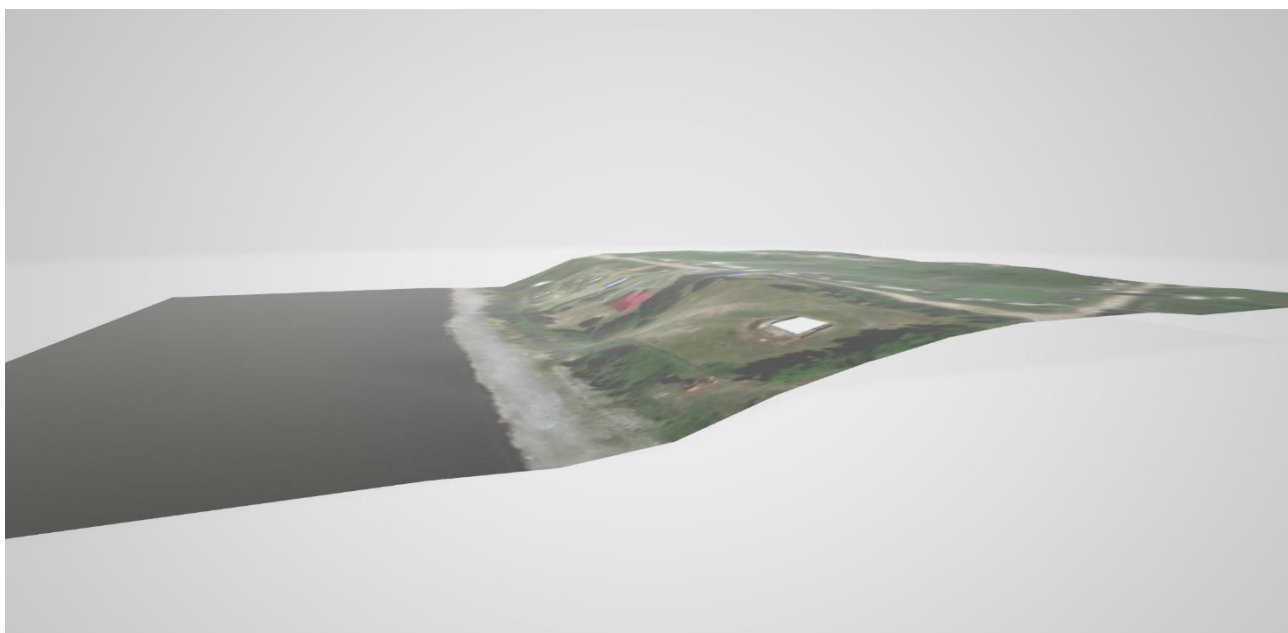


Рисунок 9 – Карта из открытых источников

Осознав, что выгрузка из открытых источников не подходит было принято решение использовалась карту местности с указанием высоты. Тогда была найдена карта, где на определенном расстоянии друг от друга были указаны

точки и ее высота. Следующем этапом стало разбиение плоскости с текстурой карты на маленькие полигоны, где вершины полигонов являлись точки с указанием высот как показано на рисунке 10.

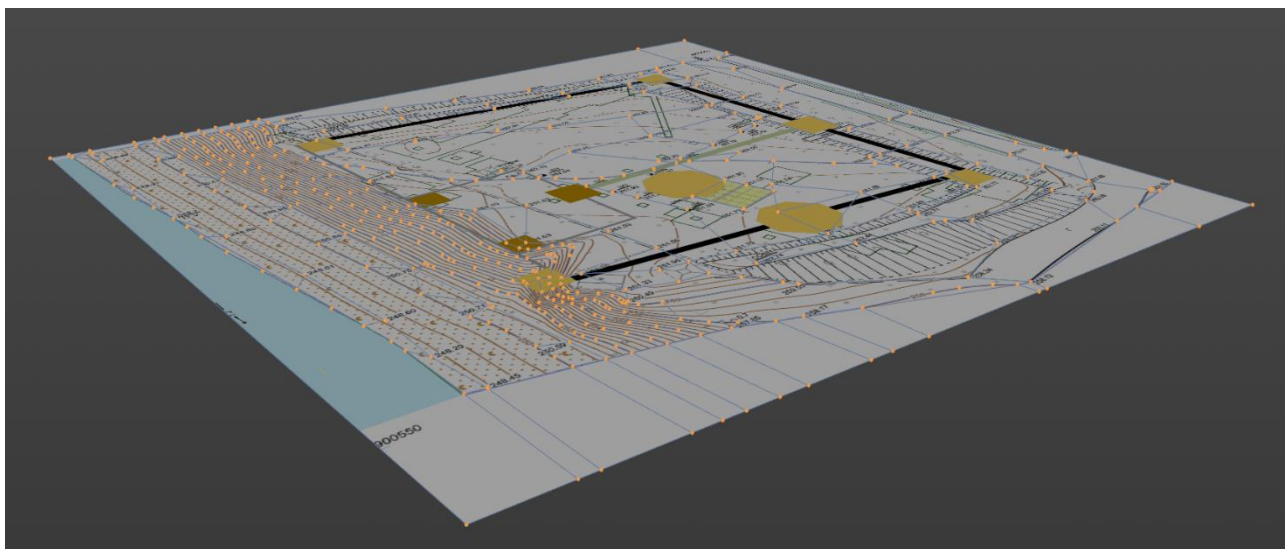


Рисунок 10 – Плоскость, разбитая на полигоны

После того как плоскость была разбита на полигоны каждой точке (вершине полигона) была задана высота. Но из-за недостаточного количества точек на карте углы были слишком резкие и в некоторых местах полигоны располагались неправильно как показано на рисунке 11.

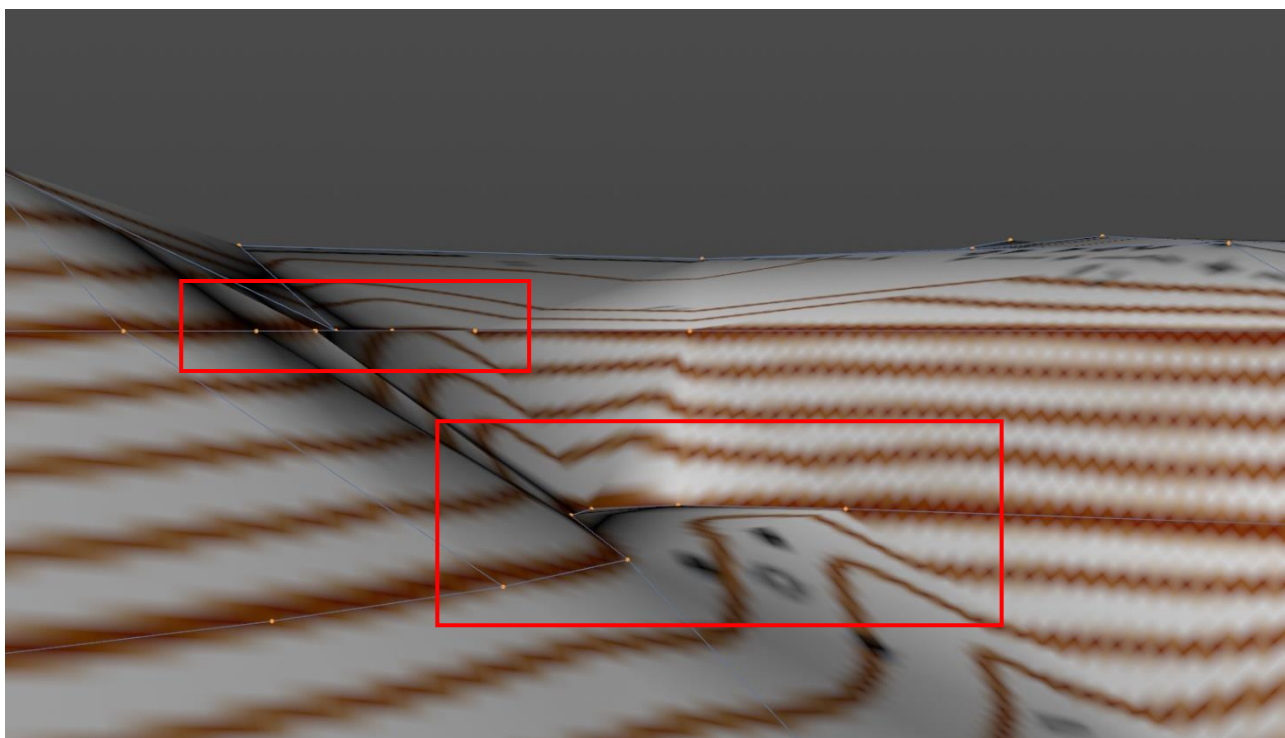
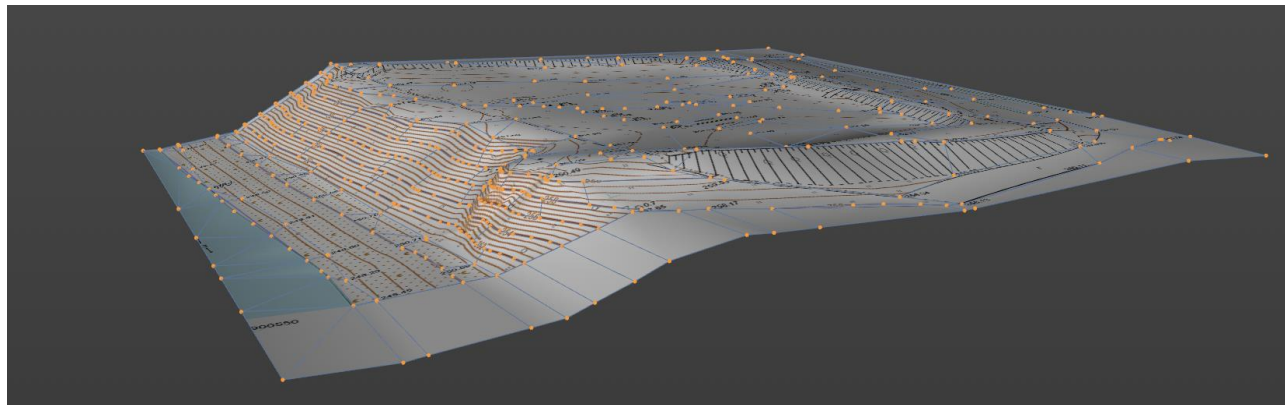


Рисунок 11 – Аномалии на плоскости.

Чтобы исправить данную ситуацию стало необходимо вручную добавить точки и вручную отредактировать их высоту. Разницу в количестве точек можно увидеть на рисунке 12.

До:



После:

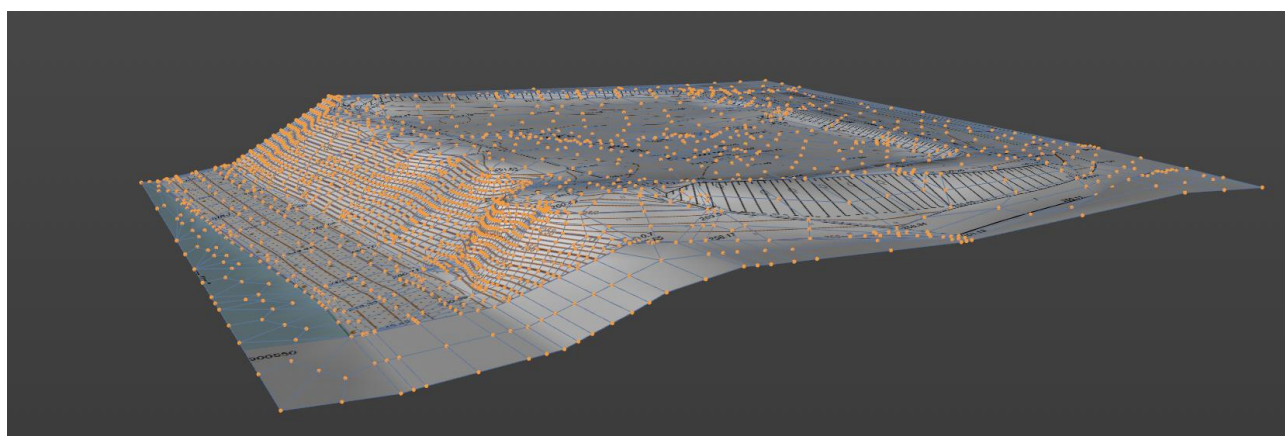


Рисунок 12 – Соотношения количества точек на карте.

После выравнивания всех точек вручную получился реалистичный рельеф местности. Показанный на рисунке 13.

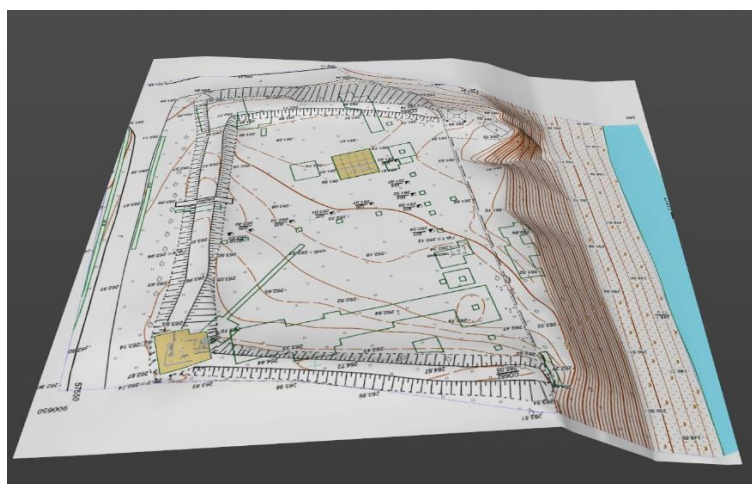


Рисунок 13 – Реалистичный ландшафт местности.

Т.к. основная задача состоит в том, чтобы точно определить нахождение сооружения на карте, а не детализированное отображение сооружений было принято решение для создания моделей сооружений Албазинского острога использовать полигональное моделирование – это самая первая разновидность трёхмерного моделирования, которая появилась в те времена, когда для определения точек в трёхмерном пространстве приходилось вводить вручную с клавиатуры координаты X, Y и Z. Как известно, если три или более точек координат заданы в качестве вершин и соединены рёбрами, то они формируют многоугольник (полигон), который может иметь цвет и текстуру. Соединение группы таких полигонов позволяет смоделировать практически любой объект. Недостаток полигонального моделирования состоит в том, что все объекты должны состоять из крошечных плоских поверхностей, а полигоны должны иметь очень малый размер, иначе края объекта будут иметь огранённый вид. Это означает, что если для объекта на сцене предполагается увеличение, его необходимо моделировать с большим количеством полигонов даже несмотря на то, что большинство из них будут лишними при удалении от объекта. И типом моделирования модели low-poly, то есть модели с малым количеством полигонов, другими словами, которые состоят из минимального числа полигонов. При этом их достаточное количество для визуального восприятия получаемого объекта. Широкое распространение имеют на мобильных платформах в игровой индустрии в связи с ограничением производительности. То есть такие модели используются, когда в силу каких-либо обстоятельств не требуется высокая детализация.

Каждое отдельное сооружение создавалось из простого объекта, например такого как куб. Blender позволяет из простого объекта путем разрезания его полигонов и выдавливания нужных полигонов довольно за короткий промежуток времени удалось создать 3D-модель Албазинского острога показанную на рисунке 12.

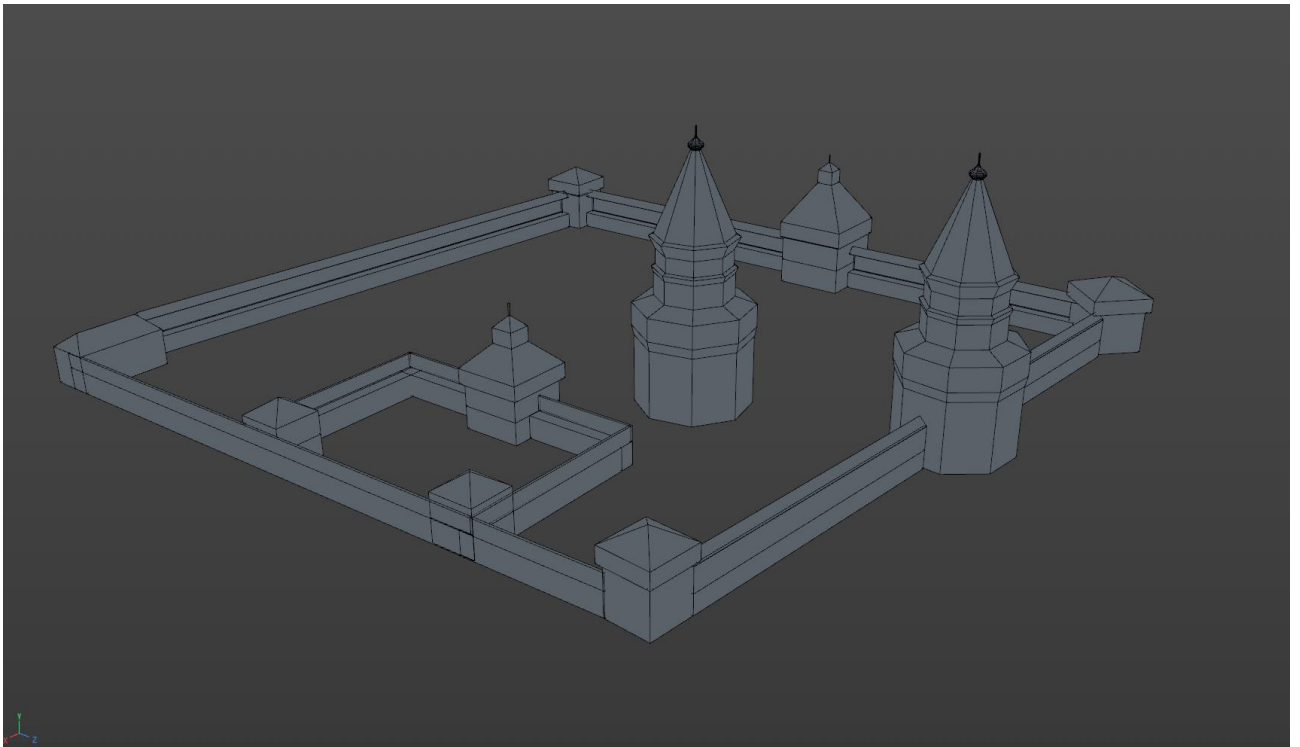


Рисунок 12 – 3D модель Албазинского острога

После того как ландшафт и 3D-модель Албазинского острога готовы осталось только совместить их как показано на рисунке 13.

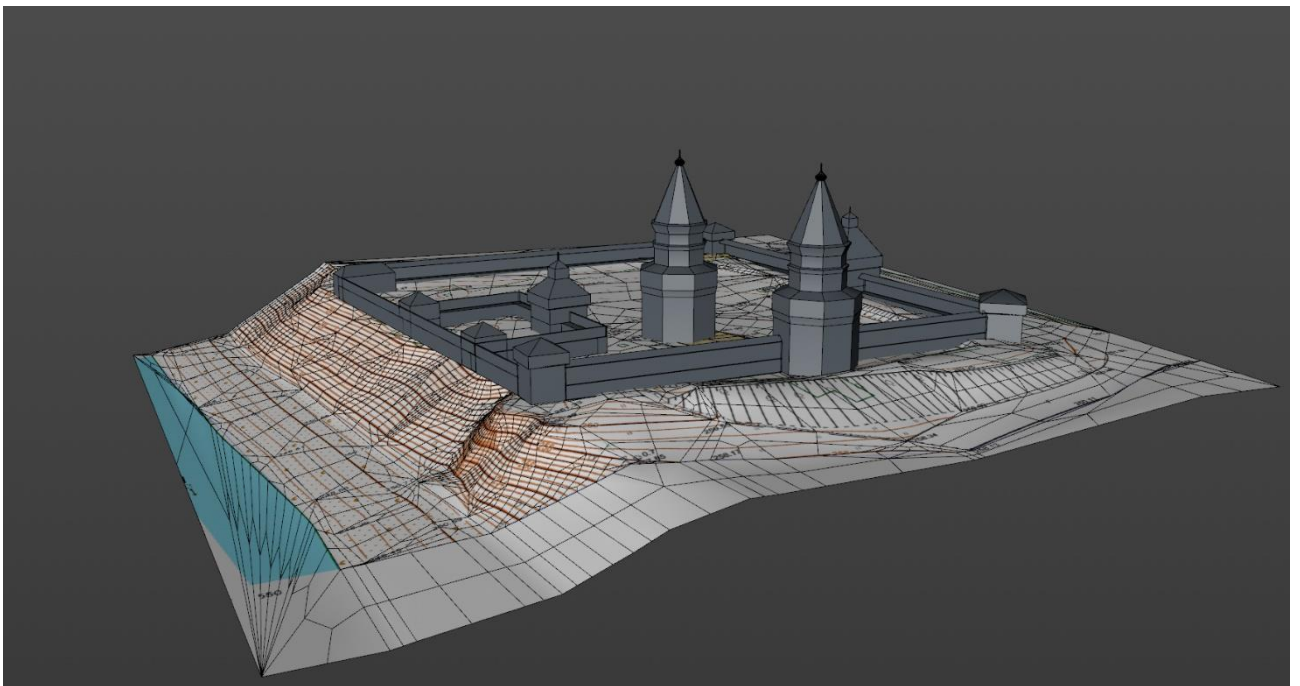


Рисунок 13 – Совмещенный ландшафт и 3D-модель острога

В итоге получаем реалистичную репликацию Албазинского острога.

3.4 Реализация интерфейса

При создании пользовательского интерфейса использовались библиотеки

Unity UI – интерфейс пользователя (User Interface). Основными компонентами пользовательского интерфейса в Unity 3D выступают:

1. Canvas – область, включающая в себя все элементы UI, в тоже время является объектом сцены. Все элементы UI должны быть дочерними объектами относительно Canvas.

2. EventSystem – объект, включающий в себя компоненты обработчика нажатий на элементы UI, настройки клавиш активации окон и мульти нажатий.

3. Элементы UI – кнопки, области ввода текста, текстовые панели и так далее.

В начале проектируется визуальная часть интерфейса, которая программируется в зависимости от конечного функционала приложения.

Для работы приложения было разработано 3 различных меню список которых показан на рисунке 14.

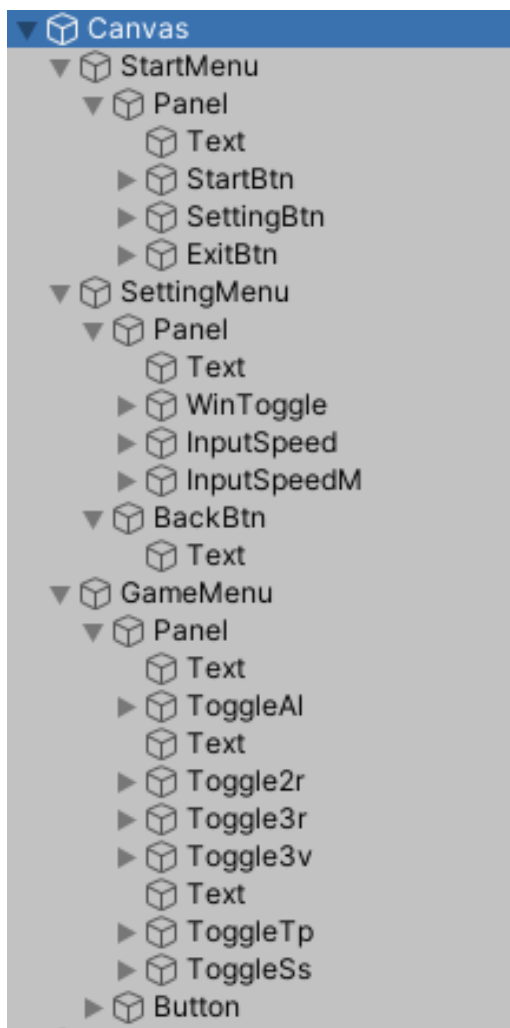


Рисунок 14 – Список меню

StartMenu это стартовое меню которое запускается сразу после запуска приложения. Данное меню состоит из панели, на которой размещено 3 кнопки. Кнопка StartBtn отвечает за запуск приложения кроме этой кнопки можно воспользоваться физической кнопкой на клавиатуре «ESC» она позволяет одним нажатием открывать и закрывать стартовое меню. Кнопка SettingBtn отвечает за включение настроек, кнопка ExitBtn отвечает за выход из приложения. Более наглядно данное меню можно увидеть на рисунке 15

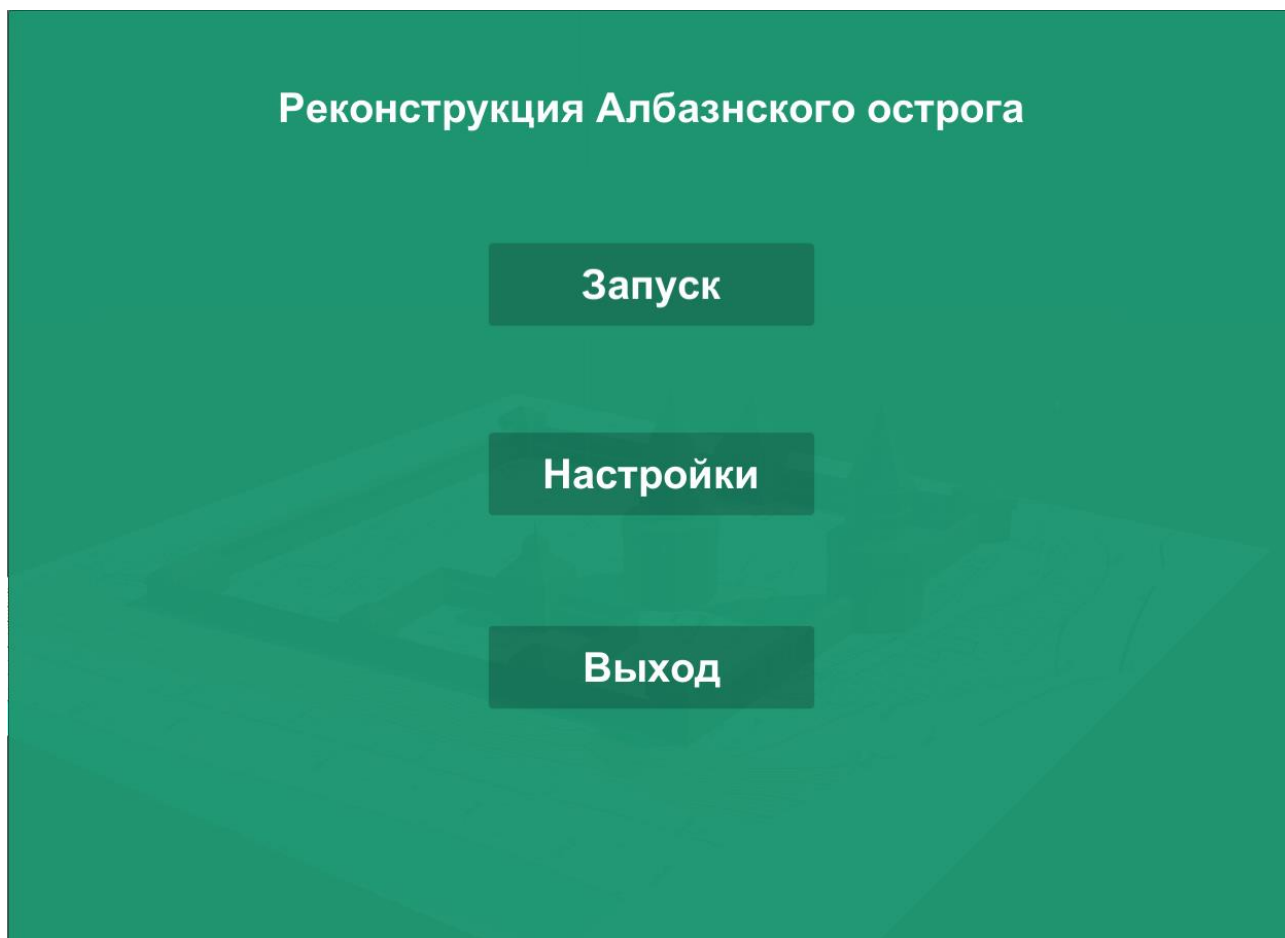


Рисунок 15 – Главное меню

Меню настроек «SettingMenu» состоит из панели и закрепленного на нее winToggle Он позволяет в активном режиме переводить приложение в оконный режим. Поля для ввода «InpitSpeed» и «InputSpeedM» позволяют управлять скоростью перемещения и скоростью вращения камеры, чем выше цифра, тем больше скорость. Данное меню можно увидеть на рисунке 16.

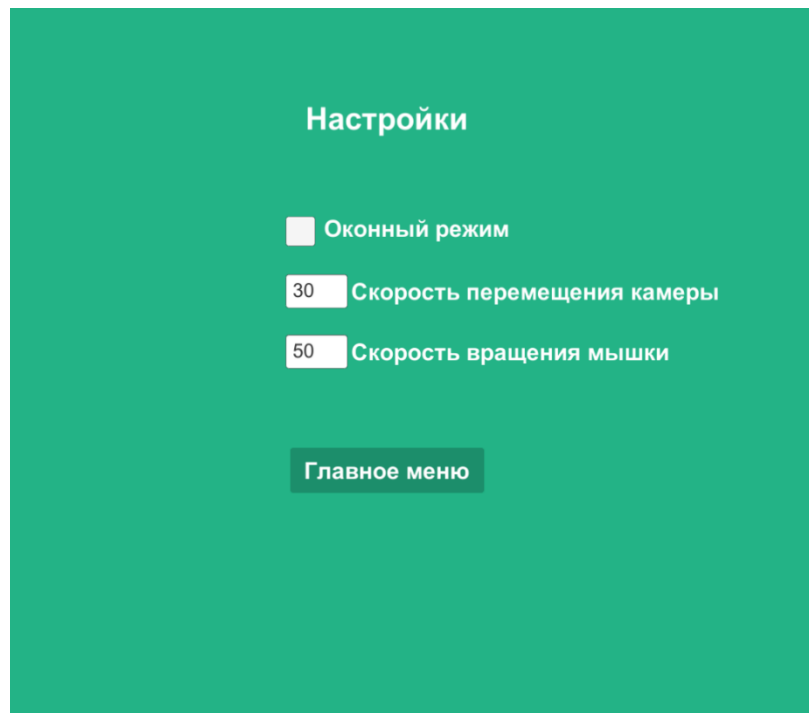


Рисунок 16 – меню настроек

Игровое меню «GameMenu» состоит из панели и помещенных на нее тугл. Тугл «ToogleAI» позволяет показать или скрыть Албазинский острог. Тугл «Toogle2r» позволяет показать или скрыть 2D карту ландшафта. Тугл «Toogle3r» позволяет показать или скрыть 3D карту ландшафта. Тугл «Toogle3v» позволяет показать или скрыть 3D карту ландшафта. Тугл «ToogleTr» позволяет при активном тугл «Toogle2r» или «Toogle3r» показать на карте топографическую карту.

Тугл «ToogleSS» позволяет при активном тугл «Toogle2r» или «Toogle3r» показать на карте спутниковый снимок

Как выглядит данная настройка можно увидеть на рисунке 17.

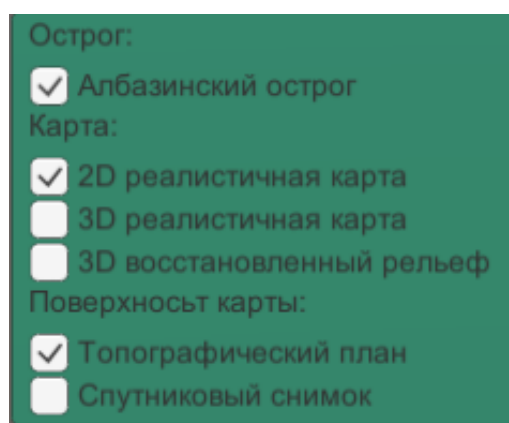


Рисунок 17 – Основное меню

Выбором различных тугл можно увидеть нужную информацию на данный момент настройки позволяют увидеть 10 различных вариаций отображения Ал-базинского острога. Примером вывода нужной информации можно увидеть на рисунках 18,19, 20.

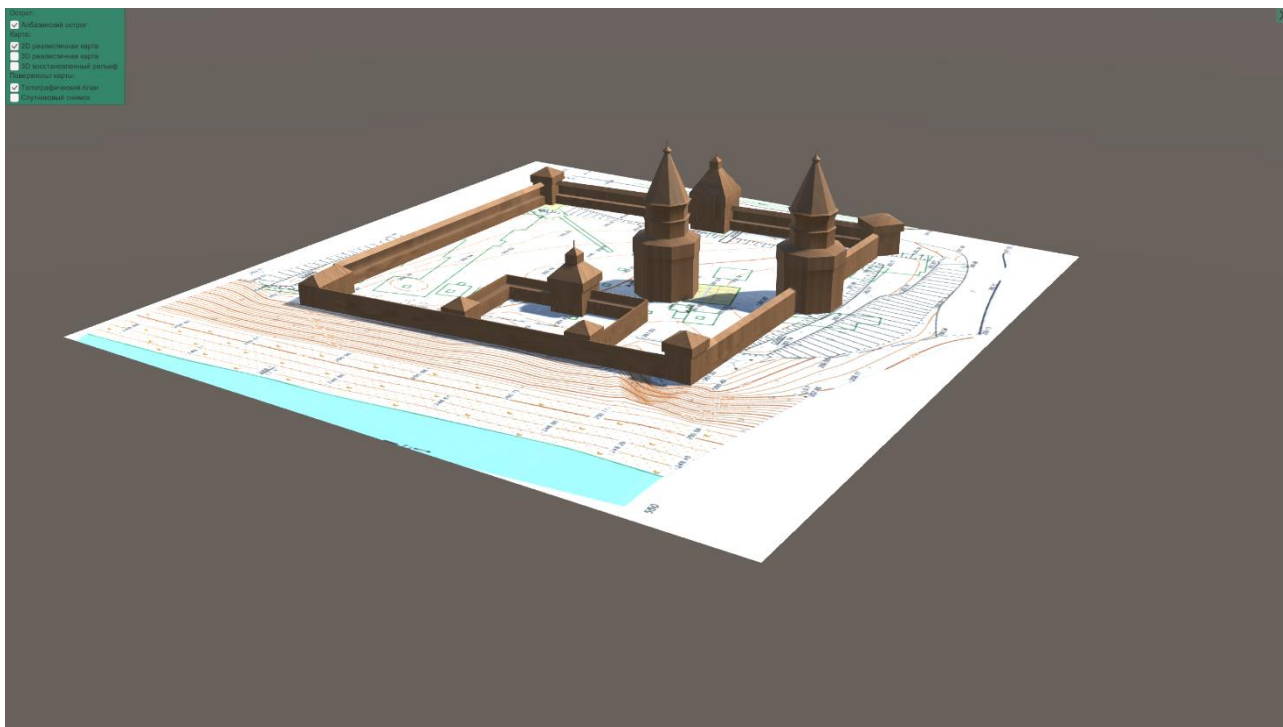


Рисунок 18 – Острог и 2D карта

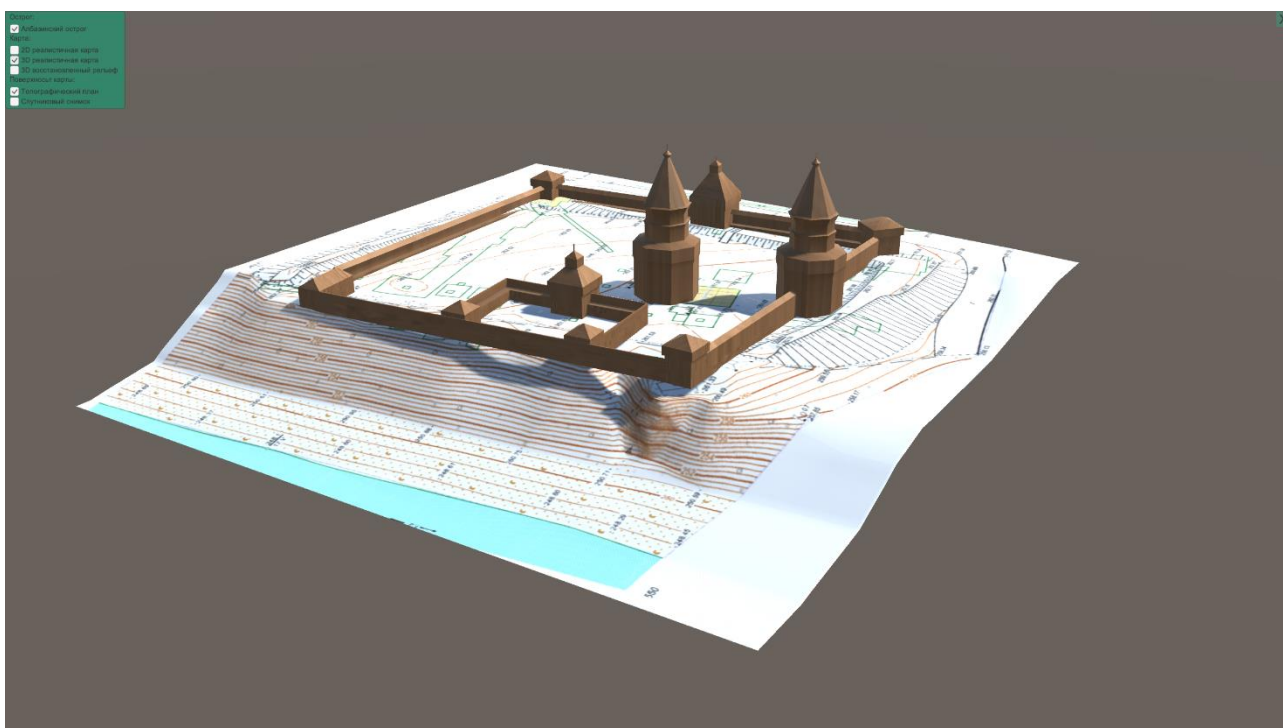


Рисунок 19 – Острог и 3D карта

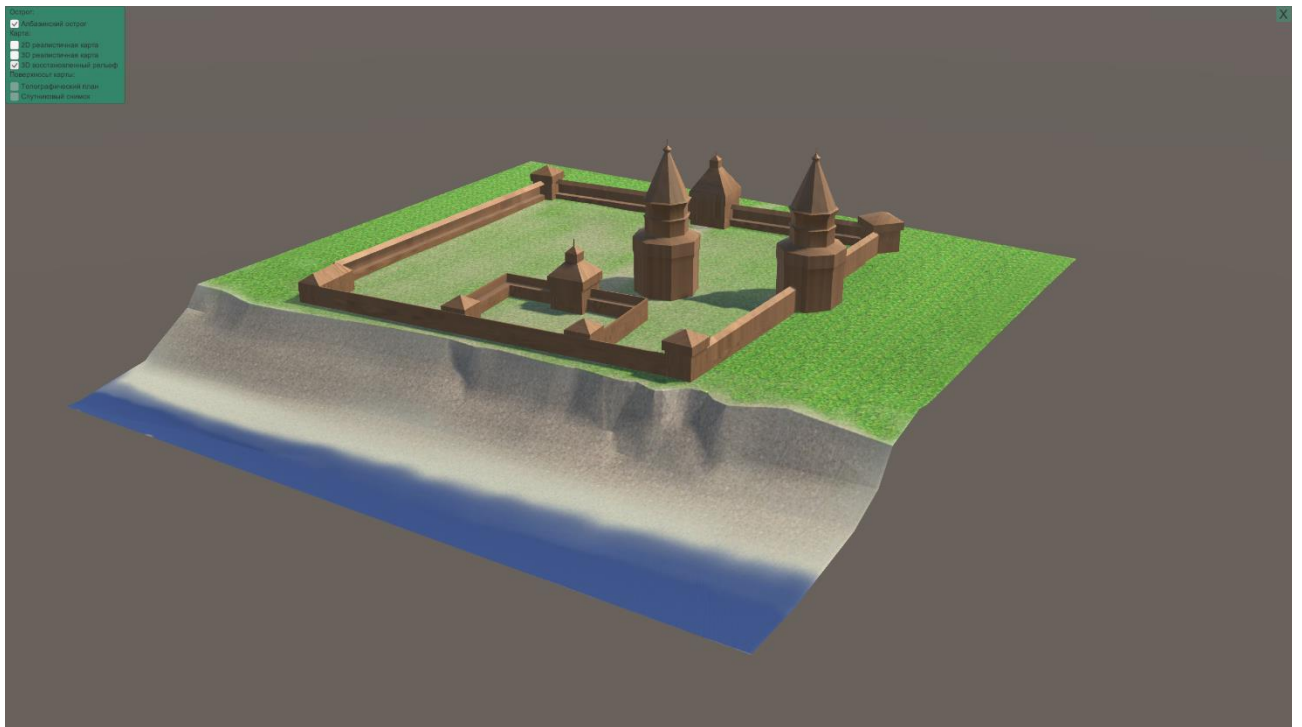


Рисунок 20 – Острог и восстановленный рельеф

На рисунках отображена лишь часть возможных репликации острога. Чередую выбор тугл в основном меню, можно увидеть много различных моделей.

3.5 Разработка программного кода

Для корректной работы программы требовалось написать программный код. Один из ключевых скриптов приложения является камера. Для вращения камеры изначально был написан код, показанный на рисунке 21.

```
if (Input.GetMouseButton(1))
{
    xRotation -= Input.GetAxis("Mouse Y");
    yRotation += Input.GetAxis("Mouse X");

    transform.rotation = Quaternion.Slerp(transform.rotation, Quaternion.Euler(xRotation, yRotation, 0), speedMouse);
}
```

Рисунок 21 – Скрипт камеры

Но входе тестирования выяснилось, что данный код не может обеспечить плавное перемещения камеры т.к. время вывода кадров не постоянно, из-за этого вывод кадров через фиксированную величину не дает плавную картинку, например, как показано на рисунке 22.

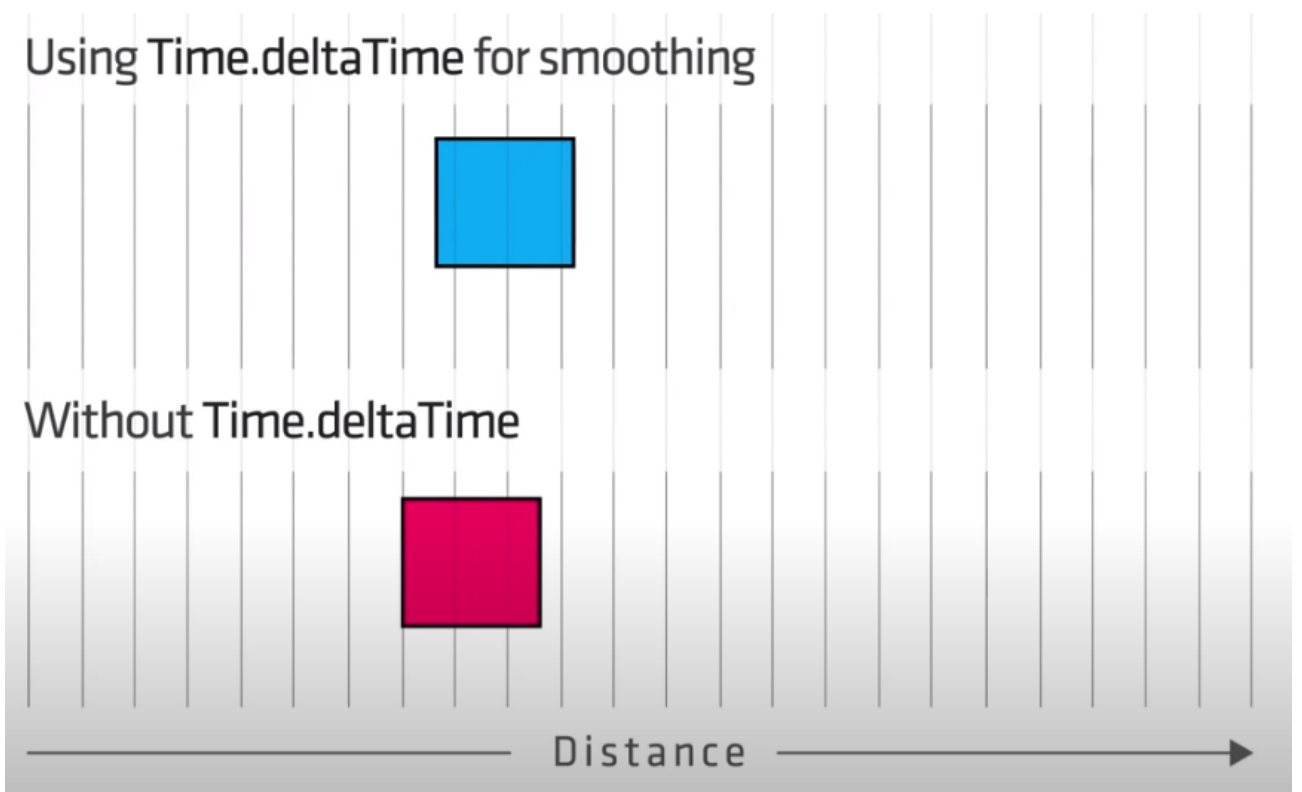


Рисунок 22 – Схематичное отображение вывода кадров

На рисунке 22 видно, как при выводе кадров на экран ведут себя различные методы вывода кадров на экран. Итогом стала модернизация скрипта вращения камеры до вида, показанного на рисунке 23.

```
if (Input.GetMouseButton(1))
{
    xRotation -= Input.GetAxis("Mouse Y");
    yRotation += Input.GetAxis("Mouse X");

    transform.rotation = Quaternion.Slerp(transform.rotation, Quaternion.Euler(xRotation, yRotation, 0), Time.deltaTime * speedMouse);
}
```

Рисунок 23 – Измененный скрипт вращения камеры

При запуске приложения с помощью функции «void Start» вызывается метод «StartMenu()» который в свою очередь отображает главное меню на экране из этого меню можно начать использование приложения вызвав метод «StartGame()» или открыть настройки вызвав метод «StartSetting()» показанные на рисунке 24.


```

Ссылка: 2
public void StartMenu ()
{
    startm.SetActive(true);
    settingm.SetActive(false);
    gamem.SetActive(false);
}

ссылка: 1
public void StartGame ()
{
    startm.SetActive(false);
    settingm.SetActive(false);
    gamem.SetActive(true);
}

Ссылка: 0
public void StartSetting ()
{
    startm.SetActive(false);
    settingm.SetActive(true);
    gamem.SetActive(false);
}

```

Рисунок 24 – Вывод экранных форм

Не посредство в самом приложении при первом запуске происходит вызов метода «Toggle2rf()» Он позволяет вывести на экран Албазинский острог и 3D-карту местности. После открытия основного меню приложения можно вызвать другой метод для отображения нужной информации на экране.

3.6 Выводы из главы

В третьей главе были разработана 3D-модель реалистичного рельефа и модели сооружений острога. Разработан программный продукт после чего произведено тестирование и отладка приложения.

ЗАКЛЮЧЕНИЕ

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

1. Выполнен анализ предметной области;
2. Выбраны программные средства для создания ПО;
3. Используя выбранное программное обеспечение и проанализированные данные в предметной области разработана 3D историко-топографическая реконструкция Албазинского острога.

В ходе написания работы были рассмотрены различные методики реконструкций строений и разработана собственная методика реконструкции Албазинского острога путем изучения исторической документации, совмещенной с использованием новых технологий.

Произведен анализ ПО и выбрано наиболее актуальное ПО для разработки виртуальной реконструкции. Для данного проекта было выбрано следующие ПО – среда геометрического моделирования Blender, среда разработки кода Visual studio, среда растровой графики Adobe Photoshop, среда мультимедийной визуализации Unity 3D.

Была разработана 3D-модель реалистичного рельефа и модели сооружений острога. Разработан программный продукт после чего произведено тестирование и отладка приложения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Трухин В.И., Крюков В.В. Албазинское воеводство (сборник документов). – Хабаровск: Библиотека дальневосточного казачества, 2016.
2. Новиков-Даурский Г.С. Историко-археологические очерки, статьи, воспоминания. – Благовещенск: Амур. кн. изд-во, 1961.
3. Трухин В.И., Багрин Е.А. Албазинский острог в 1665/1666-1689 гг.: фортификация и защитники – опыт исторической реконструкции // История военного дела: исследования и источники. – 2019. – Т. X. – С. 385-431.
4. Нестеров С.П. Город Албазин на Амуре: численность жителей в последней четверти XVII века // Археология, этнография и антропология Евразии. – 2017. – Т. 45, № 2. – С. 113-122.
5. Кочедамов В.И. Первые русские города Сибири. – М.: Стройиздат, 1978.
6. Друковский, А.В. 3ds Max 2014. Лучший самоучитель / А.В. Друковский, О.В. Якунина. – М. : Астрель, 2015. – 352 с.
7. Линовес Джонатан, Виртуальная реальность в Unity / Джонатан Линовес. – М. : ДМК записываются ПРЕССходимых, 2016. – 316 с.
8. Пахмурин, А.Г. ГИА "Иннотер" 3D-моделирование картографической информации в городской среде (на примере г. Юбилейного Московской области) [Электронный ресурс] – Режим доступа : <https://innoter.com/scientific-articles/990> –10.06.2017
9. Хокинг, Д. Unity в действии. Мультиплатформенная разработка на C# / Д. Хокинг. – Изд-во Питер, 2016. – 336 с.
10. Панфилова, Д.Г. Новое измерение ГИС: Contex 3D увеличивает точности эффективность макетирования [Электронный ресурс] / Д. Г. Панфилова // Режим доступа: <http://www.cadmaster.ru/magazin/articles/cm32contex3d> – 06.04.2017.

11. Петрухин, С. Создание 3D модели средствами ГИС Карта 2008 [Электронный ресурс] / С. Петрухин. – Географические информационные системы и дистанционное зондирование GIS LAB // Режим доступа: <http://gislab.info/qa/giskarta-3d.html>. – 12.01.17

12. Дикинсон, К. Оптимизация игр в Unity 5 / К. Дикинсон. – Москва : ДМК Пресс, 2017. – 306 с. – ISBN 978-5-97060-432-8. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/90109> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

13. Тюкачев, Н. А. C#. Основы программирования : учебное пособие / Н. А. Тюкачев, В. Г. Хлебостроев. – 3-е изд., стер. – Санкт-Петербург : Лань, 2018. – 272 с. – ISBN 978-5-8114-2567-9. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/104962> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

14. Ларкович, С. Н. Unity на практике. Создаем 3d-игры и 3d-миры : учебное пособие / С. Н. Ларкович. – Санкт-Петербург : Наука и Техника, 2019. – 272 с. – ISBN 978-5-94387-780-3. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/139150> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

15. Иванов, В. В. 3D-конструирование : учебно-методическое пособие / В. В. Иванов, А. В. Фирсов, А. Н. Новиков. – Москва : РГУ им. А.Н. Косыгина, 2016. – 20 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/128010> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

16. Технология трехмерного моделирования в Blender 3d : учебное пособие / А. А. Кузьменко, А. Д. Гладченков, Л. Б. Филиппова [и др.]. – Москва : ФЛИНТА, 2018. – 79 с. – ISBN 978-5-9765-4015-6. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/113463> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

17. Технология трехмерного моделирования и текстурирования объектов в Blender 3d и 3d Max : учебное пособие / А. А. Кузьменко, А. Д. Гладченков, В. А. Шкаберин [и др.]. – Москва : ФЛИНТА, 2019. – 142 с. – ISBN 978-5-9765-4216-7. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/125515> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

18. Основы работы в Photoshop : учебное пособие. – 2-е изд. – Москва : ИНТУИТ, 2016. – 1393 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/100338> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

19. Молочков, В. П. Adobe Photoshop CS6 : учебное пособие / В. П. Молочков. – 2-е изд. – Москва : ИНТУИТ, 2016. – 388 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/100563> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

20. Джонатан, Л. UV Mapping в Unity / Л. Джонатан ; перевод с английского Р. Н. Рагимов. – Москва : ДМК Пресс, 2016. – 316 с. – ISBN 978-5-97060-234-8. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/93271> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

21. Марчуков, А. В. Работа в Microsoft Visual Studio : учебное пособие / А. В. Марчуков, А. О. Савельев. – 2-е изд. – Москва : ИНТУИТ, 2016. – 384 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/100439> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

22. Данилина, И. И. Программирование на языке C# в среде Microsoft Visual Studio : учебно-методическое пособие / И. И. Данилина. – Екатеринбург : , 2018. – 65 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/121392> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

23. Берг, Д. Б. Модели жизненного цикла : учебное пособие / Д. Б. Берг, Е. А. Ульянова, П. В. Добряк. – Екатеринбург : УрФУ, 2014. – 74 с. – ISBN 978-5-7996-1311-2. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/98755> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

24. Сычёв, С. А. Перспективные технологии строительства и реконструкции зданий : монография / С. А. Сычёв, Г. М. Бадьин. – 2-е изд., испр. и доп. – Санкт-Петербург : Лань, 2019. – 368 с. – ISBN 978-5-8114-4483-0. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/123464> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

25. Аббасов, И. Б. Основы трехмерного моделирования в графической системе 3ds Max 2009 : учебное пособие / И. Б. Аббасов. – Москва : ДМК Пресс, 2009. – 176 с. – ISBN 978-5-94074-570-9. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/1341> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

26. Иванов, В. В. 3D-конструирование : учебно-методическое пособие / В. В. Иванов, А. В. Фирсов, А. Н. Новиков. – Москва : РГУ им. А.Н. Косыгина, 2016. – 20 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/128010> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

27. Гужов, В. И. Методы измерения 3D-профиля объектов. Фазовые методы : учебное пособие / В. И. Гужов. – Новосибирск : НГТУ, 2016. – 83 с. – ISBN 978-5-7782-2727-9. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/118225> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

28. Келлер, Э. Введение в ZBrush® 4 / Э. Келлер. – Москва : ДМК Пресс, 2012. – 768 с. – ISBN 978-5-94074-794-9. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/4154> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

29. Меженин, А. В. Технологии разработки 3D-моделей : учебное пособие / А. В. Меженин. – Санкт-Петербург : НИУ ИТМО, 2018. – 100 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/136470> (дата обращения: 11.06.2020). – Режим доступа: для авториз. пользователей.

30. Бугаев С.Н. Компьютерная историко-топографическая реконструкция Албазинского острога // Молодежь XXI века: шаг в будущее: мат. XX регион. науч.-прак. конф. – Благовещенск: АмГУ, 2019. – Т. 3. – С. 192-193.

31. Еремин И.Е., Трухин В.И., Бугаев С.Н. Трехмерное компьютерное моделирование Албазинского острога периода 1684г // Информатика и системы управления – Благовещенск: АмГУ, 2019. – Т. 4. – С. 10-25

СЕРТИФИКАТ

участника

Бурасев Станислав Николаевич

**XX региональной
научно-практической конференции
«Молодежь XXI века: шаг в будущее»**

Секция Информационные технологии
Модернизация современного общества

**Председатель
совета ректоров
вузов Амурской области**

Тихончук

П.В. Тихончук



**Благовещенск
23 мая 2019 г.**

