

РЕФЕРАТ

Выпускная квалификационная работа: 90 страниц, 3 части, 69 рисунков, 25 источников, 8 таблиц, 1 приложение.

ВИРТУАЛЬНАЯ ЛАБОРАТОРИЯ, КОМПЬЮТЕРНЫЕ СЕТИ, ИГРОВАЯ ПЛАТФОРМА, BLENDER, UNITY, ТРЕХМЕРНОЕ МОДЕЛИРОВАНИЕ, IDEF0, РАЗРАБОТКА, ВИРТУАЛЬНАЯ ЛАБОРАТОРИЯ, UI, ТРЁХМЕРНЫЕ МОДЕЛИ, ПРОГРАММНЫЙ МОДУЛЬ, АЛГОРИТМ.

Цель – разработка виртуальной лаборатории для обучения по дисциплине «Компьютерные сети» (на примере Рубцовского института (филиала) АлтГУ).

Объектом исследования является процесс обучения по дисциплине «Компьютерные сети».

Предметом исследования является геймификация процесса обучения по дисциплине «Компьютерные сети».

Методы решения поставленных задач: системный анализ, разработка программного обеспечения, программирование.

Результат работы – разработанная, в рамках геймификации, виртуальная лаборатория для обучения по дисциплине «Компьютерные сети», позволяющая повысить уровень обучения и степень вовлеченности студентов в образовательном процессе.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Аналитическая часть	6
1.1 Техничко-экономическая характеристика предметной области	6
1.2 Анализ функционирования объекта исследования.....	12
1.3 Определение цели и задач по разработке приложения	17
1.4 Обзор и анализ существующих технологий проектирования	18
1.4.1 Autodesk Maya	18
1.4.2 ZBrush.....	21
1.4.3 Blender	22
1.4.4 Unreal Engine	26
1.4.5 Платформа Unity	27
1.4.6 Godot.....	30
1.5 Выбор и обоснование проектных решений.....	32
1.5.1 Техническое обеспечение	32
1.5.2 Программное обеспечение.....	33
1.5.3 Лингвистическое обеспечение	34
1.5.4 Обоснование выбора «Blender».....	34
1.5.5 Обоснование выбора «Unity»	35
2 Проектная часть	37
2.1 Решение проблем после анализа объекта исследования	37
2.2 Архитектура и принцип функционирования программы.....	39
2.3 Проектирование трехмерных моделей лаборатории и мебели.....	41
2.4 Проектирование трёхмерных моделей для информационного	

обеспечения лаборатории.....	42
2.5 Разработка программного модуля контроля знаний.....	52
2.6 Разработка программного модуля проектирования сетей	56
2.6.1 Программная реализация модуля проектирования сетей.....	61
2.7 Разработка программного модуля отслеживания прогресса	66
2.8 Разработка системы управления и пользовательского интерфейса...	70
2.9 Разработка модуля воспроизведения звукового сопровождения	75
3 Оценка эффективности внедрения программного обеспечения	79
3.1 Общие положения	79
3.2 Показатели эффективности.....	79
3.3 Расчет экономических затрат.....	80
3.4 Оценка педагогической эффективности.....	83
3.5 Оценка социальной эффективности.....	84
ЗАКЛЮЧЕНИЕ	86
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	87
ПРИЛОЖЕНИЕ А	91

ВВЕДЕНИЕ

На текущем этапе развития цифровых технологий, устоявшиеся методики обучения в образовательных учреждениях, заключающиеся в изучении лекционных материалов, выполнение перечня практических и лабораторных работ зачастую оказывается не всегда достаточно для поддержания высокой заинтересованности студентов в процессе изучения. Изучение дисциплин по классической модели обучения в малой или большей степени показывает слабую заинтересованность студентов в той или иной изучаемой дисциплине.

Геймификация в образовании представляет собой определенную совокупность правил, элементов и механик, использующихся в обычных и виртуальных играх, которые применяются в образовательной среде с целью повышения эффективности и вовлеченности в образовательный процесс.

Внедрение геймификации в образовательный процесс позволит студентам лучше погрузиться в процесс обучения по дисциплине «Компьютерные сети». Это положительным образом повлияет на показатели успеваемости студентов, что в свою очередь сделает общий образовательный процесс изучения компьютерных сетей более эффективным. Именно по этому ряду причин, интегрирование процесса геймификации в дисциплину «Компьютерные сети» посредством виртуальной лаборатории, является актуальной задачей для института.

Цель исследования заключается в разработке виртуальной лаборатории для обучения по дисциплине «Компьютерные сети» (на примере Рубцовского института (филиала) АлтГУ).

В таком случае, объектом исследования является процесс обучения по дисциплине «Компьютерные сети».

Предметом исследования является геймификация процесса обучения по дисциплине «Компьютерные сети».

Для достижения цели необходимо решить следующие задачи:

- провести анализ предметной области;
- выработать проектные решения по обеспечивающим системам;
- реализовать проектные решения разрабатываемой виртуальной лаборатории «Компьютерные сети»;
- провести расчет экономических затрат на разработку виртуальной лаборатории «Компьютерные сети».

1 Аналитическая часть

1.1 Технико-экономическая характеристика предметной области

В настоящее время, Рубцовский институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Алтайский государственный университет» (далее – Рубцовский институт (филиал) АлтГУ) является некоммерческой, входящее в состав АлтГУ, организацией, осуществляющей предоставление образовательных услуг всем близлежащим районам Алтайского края, в том числе, жителям, проживающим на территории города Рубцовска.

Приказом Государственного комитета РФ по высшему образованию от 22 февраля 1996г. № 324 г. был создан филиал Алтайского государственного университета в г. Рубцовске [1].

Рубцовский институт (филиал) АлтГУ осуществляет образовательную деятельность на некоммерческой и коммерческой основе, предоставляя бюджетные и платные места абитуриентам, желающим получить среднее специальное и высшее образование.

Основанием ведения образовательной деятельности Рубцовского института (филиала) АлтГУ, является предоставленная лицензия на осуществление образовательной деятельности.

Настоящая лицензия предоставлена на основании распоряжения Федеральной службы по надзору в сфере образования и науки от 29 июля 2016, №2296. Лицензия предоставлена институту на бессрочной основе.

Дата создания образовательной организации 22 февраля 2006 года.

Миссия Рубцовского института (филиала) АлтГУ заключается в том, чтобы удовлетворить потребности жителей г. Рубцовска и близлежащих районов Алтайского края в получении современного, качественного,

актуального образования, а также в удовлетворении образовательных потребностей общества путем осуществления подготовки и переподготовки кадров.

Главная цель – подготовка выпускников, способных обеспечить функционирование и развитие профессиональных сфер деятельности на уровне мировых достижений, являющихся достойными гражданами России, конкурентноспособными в России и за рубежом [2].

Первоначальной задачей Института в ближайший период времени является разработка, улучшение и внесение новшеств в систему образования и общий образовательный процесс.

Политика Рубцовского института (филиала) АлтГУ в области качества образования является неотъемлемым элементом стратегии Института и основой планирования его образовательной деятельности [4].

График учебного процесса образовательной организации:

- Пн – Пт: 8:00 – 21:10;
- Сб: 08:00 – 20:50.

График работы специалистов и структурных подразделений:

- Пн – Чт: 08:00 – 17:00 (обед 12:00 – 12:48);
- Пт: 08:00 – 16:00 (обед 12:00 – 12:48).

Рубцовский институт (филиал) АлтГУ формирует квалифицированные кадры по следующим направлениям:

- педагогическое образование;
- юриспруденция;
- государственное и муниципальное управление;
- менеджмент;
- экономика;
- психология;
- прикладная информатика;
- юриспруденция (спо);
- правоохранительная деятельность;

- финансы;
- экономика и бухгалтерский учет (по отраслям) (в промышленности, бюджетных отраслях);
- земельно-имущественные отношения;
- информационные системы и программирование.

Опубликованная информация о поступлениях и расходах финансовых и материальных средств продемонстрированы в таблице 1.1.

Таблица 1.1 – Поступление и расходы финансовых и материальных средств

Год	Поступление финансовых и материальных средств	Расходы финансовых и материальных средств
2022	100 587 614,46	101 564 387,14
2023	106 229 760,63	111 486 653,87
2024	107 055 952,02	113 779 095,78

Рубцовский институт (филиал) АлтГУ подразделен на два образовательных корпуса. Один из них находится по адресу г. Рубцовск, пр-кт Ленина 200б.

Второй корпус Рубцовского института (филиала) АлтГУ расположен по адресу г. Рубцовск пр-кт Ленина 243 [22].

На данный момент, в Рубцовском институте (филиале) АлтГУ в штате находится 96 сотрудников.

Рубцовский институт (филиал) АлтГУ находится под управление директора. Директор осуществляет управление филиалом. Директор находится в подчинении у учёного совета.

В подчинении у директора находятся следующие структурные подразделения:

- заместитель директора по стратегическому развитию;
- заместитель директора по воспитательной работе;
- заместитель директора по научной работе;
- заместитель директора по учебной работе;
- документовед;
- отдел правового и кадрового обеспечения;

- отдел технического и программного обеспечения;
- отдел хозяйственного обеспечения;
- специалист по охране труда;
- финансово экономический отдел.

У заместителя директора по воспитательной работе находятся следующие структурные подразделения:

- сектор по связи с общественностью;
- студенческий центр.

Заместитель директора по учебной работе имеет в подчинении следующие организационные единицы:

- научная библиотека;
- бюро расписания;
- отдел по работе со студентами;
- сектор методического сопровождения и мониторинга качества

образования;

- кафедра экономики;
- кафедра государственного и муниципального управления и права;
- кафедра математики и прикладной информатики;
- кафедра психологии и общественных дисциплин.

Центр сертифицированного обучения находится под управлением Кафедры экономики.

Центр повышения квалификации и развития профессиональных компетенций «Управление, оценка и право» находится в подчинении у Кафедры Государственного и муниципального управления и Права.

Центр информационных технологий управляется Кафедрой математики и прикладной информатики.

Центр дополнительного образования «Логос» находится под управлением Кафедры психологии и общественных дисциплин.

Организационная структура управления Рубцовским институтом (филиалом), актуальная на 2025 год, продемонстрирована на рисунке 1.1.

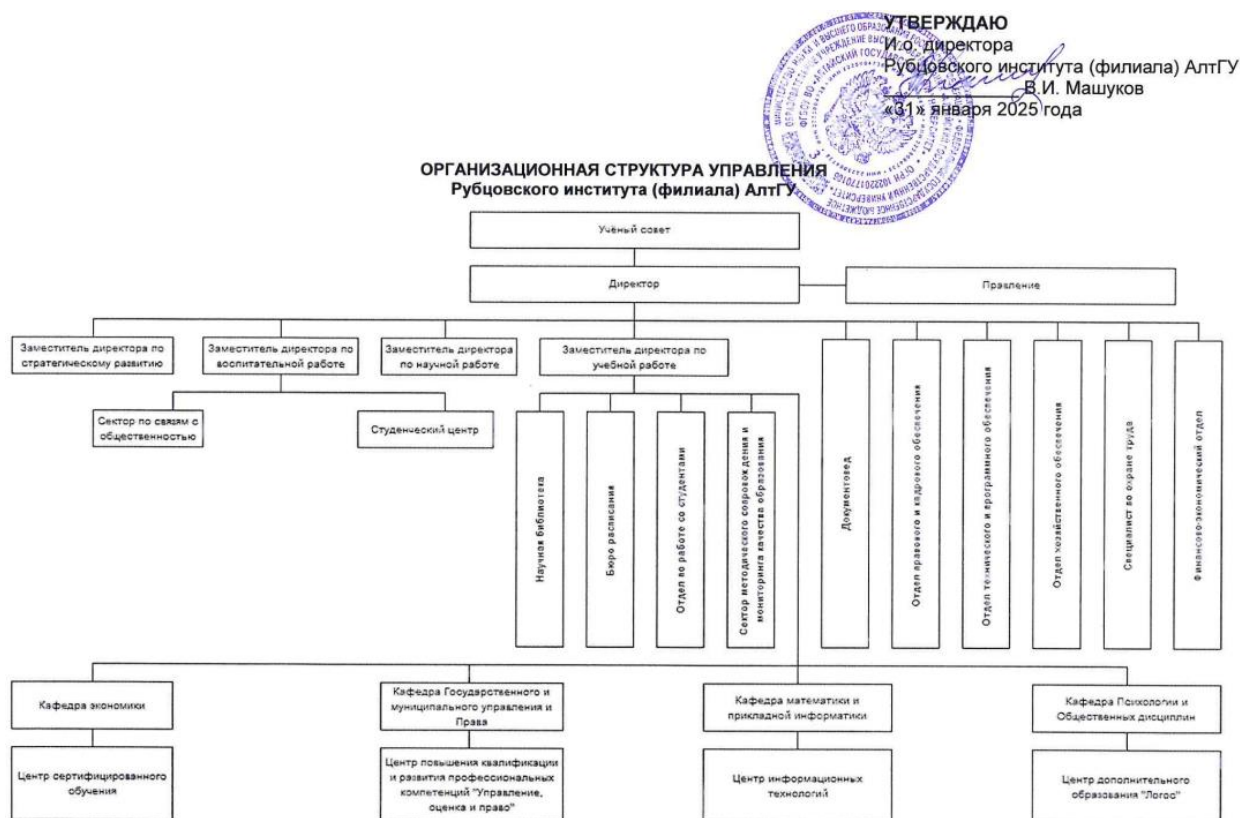


Рисунок 1.1 – Организационная структура управления Рубцовского института (филиала) АлтГУ

В Рубцовском институте (филиала) АлтГУ создана и настроена экосистема, где студенты могут быстро получать знания в удобном для них формате.

Правительство Рубцовского института (филиала) АлтГУ заключило контракты с популярными электронными библиотечными системами. У студентов всех кафедр есть непрерывный доступ к электронным библиотечным системам на весь период обучения.

Рубцовский институт (филиала) АлтГУ сотрудничает со следующими библиотечными системами:

1. ЭБС «Консультант студента».
2. ЭБС Elibrary.ru.
3. ЭБС «Юрайт».
4. ЭБС издательства «Лань».
5. ЭБС «Универсальная библиотека онлайн».

6. ЭБС East View.
7. ЭБС «Система Главбух VIP».

Помимо этого, у Рубцовского института (филиала) АлтГУ находится собственная электронная библиотечная система под названием «ЭК библиотека РИ(ф)АлтГУ».

Сведения об оборудованных учебных кабинетах, аудиториях, лабораториях представлены в таблице 1.2.

Таблица 1.2 – Сведения об оборудованных учебных кабинетах филиала

Адрес	Номер кабинета	Оснащенность
658225, Алтайский край, город Рубцовск, проспект Ленина, дом 200б	108, 111, 205, 204, 215, 305, 307, 311, 304	Учебная мебель, оборудованные места для обучения, место преподавателя, набор оборудования для демонстрирования учебного материала, компьютерная техника с обеспеченным доступом в интернет, мультимедийное оборудование
658225, Алтайский край, город Рубцовск, проспект Ленина, дом 200б	317, 316, 314, 310, 309, 306, 303, 301, 214, 213, 208, 106,	Мультимедийное оборудование, оборудованное место преподавателя, технические средства обучения, специальная аудиторная мебель
658227, Алтайский край, город Рубцовск, проспект Ленина, дом 243	108, 205, 207, 209, 211	Мультимедийное оборудование, оборудованное место преподавателя, технические средства обучения, специальная аудиторная мебель, переносной ноутбук, мультимедийное оборудование
658227, Алтайский край, город Рубцовск, проспект Ленина, дом 243	222 – стрелковый тир	Интерактивный лазерный тир «РУБИН», ИЛТ-110 «Кадет» (лазерная камера, лазерный автомат Калашникова, пневматический лазерный пистолет), специализированная мебель, оборудованное место для преподавателя, проектор

Средствами обучения и воспитания в Рубцовском институте (филиале) АлтГУ выступают: разработанные образовательные программы, учебные планы, нормативные документы, регулирующие образовательную деятельность, современное оборудование, поддерживающее образовательный процесс на высоком уровне, электронно-образовательные ресурсы, воспитательная и педагогическая поддержка обучающихся сотрудниками и

научно-педагогическими работниками Института [3].

1.2 Анализ функционирования объекта исследования

Контекстная диаграмма процесса обучения по дисциплине «Компьютерные сети» представлена на рисунке 1.2.

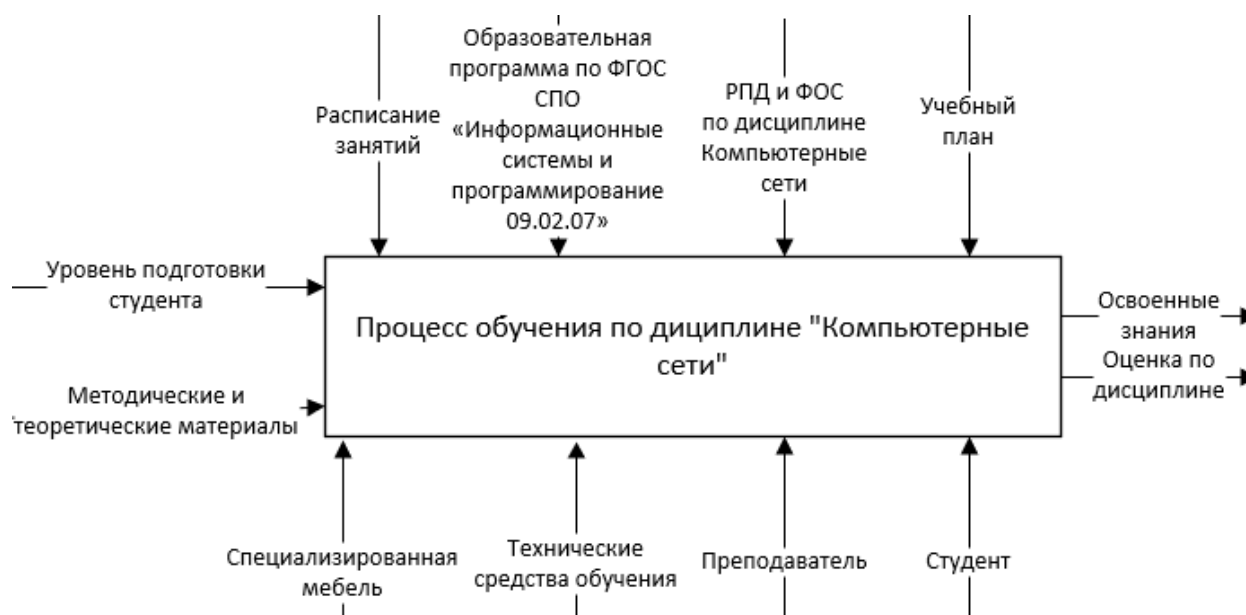


Рисунок 1.2 – Контекстная диаграмма «как есть» процесса обучения по дисциплине «Компьютерные сети»

Входными параметрами является, уровень подготовки студента, методические и теоретические материалы.

Выходными параметрами являются приобретённые студентом знания и оценка по дисциплине.

В качестве механизма выступает:

- преподаватель по дисциплине;
- технические средства обучения (учебные ноутбуки и компьютеры с выходом в сеть Ethernet, периферийные устройства, ноутбук преподавателя, проектор);
- специализированная мебель (стулья, компьютерные столы);

– студент.

В качестве управления выступает:

1. Модуль расписания занятий. В нем выставляются занятия для очного посещения студентами.
2. Образовательная программа по ФГОС среднего специального образования «Информационные системы и программирование 09.02.07».
3. Рабочая программа учебной дисциплины (РПД) и фонд оценочных средств (ФОС).
4. Учебный план.

Образовательный процесс по дисциплине «Компьютерные сети» состоит из нескольких этапов.

Изучение дисциплины «Компьютерные сети» начинается с посещения регулярных лекционных занятий, где преподаватель знакомится с учениками, повествует им теоретический материал, в том числе, излагает учебный материал в удобном формате. Лекционные занятия выставляются в расписании занятий согласно учебному плану.

Повествование учебного материала сопровождается жестикулированием, показом презентационного материала, видеоматериалом и попутным общением со студентами. Во время лекционных занятий, студенты могут задавать вопросы в рамках дисциплины преподавателю и получать развернутые ответы на них. Изучение лекционного материала студентами сопровождается попутным конспектированием. Во время регулярного посещения лекционных занятий, студенты получают и закрепляют структурированную информацию, которую можно будет применить на практических занятиях.

Посещение лабораторных занятий студентами представляет собой регулярный процесс, при котором осуществляется приобретение практических навыков и углубленного изучения в области компьютерных сетей.

Взаимодействие преподавателя с учениками в данном процессе играет

важную роль, поскольку не все люди, умеют одинаково быстро и беспрепятственно усваивать получаемую информацию. Роль преподавателя заключается в том, чтобы направить студента на путь решения, интересующей их проблемы посредством упрощения стиля подачи информации или наглядной демонстрации различных методик/способов решения проблемы.

На лабораторных занятиях студенты знакомятся с учебным оборудованием, знакомятся с программами, которые позволяют администрировать компьютерные сети, изучают способы взаимодействия с сетями, проводят настройку сетей через компьютерные сети и так далее.

Процесс выполнения лабораторной работы заключается в составлении отчетов о выполненной работе, изучении определенного материала и последующей защите одного перед преподавателем.

Преподаватель изучает сформированные отчеты по выполненной работе и задает вопросы студенту, попутно выдавая перечень задач, которые необходимо выполнить до определенного срока. Это может быть день или два в зависимости от сложности выполняемого задания.

Студент выполняет предложенные задания и демонстрирует их преподавателю.

На основании выполненной работы, преподаватель выставляет студенту соответствующую оценку. Оценка определяется качеством выполненной работы и полнотой продемонстрированных знаний.

После выполнения лабораторных работ, студенты своевременно проходят итоговое тестирование по дисциплине в системе дистанционного образования под названием (СДО) «Moodle». Итоговое тестирование демонстрирует преподавателю качество приобретенных знаний студентами. Итоговое тестирование является одним из критериев выставления оценки по дисциплине. График прохождения итогового тестирования назначается преподавателем.

Итоговая аттестация выставляется в расписании занятий согласно учебному плану. На итоговой аттестации студенты выполняют перечень

заданий, предоставленных преподавателем. После выполнения заданий, учитель выставляет итоговую оценку по дисциплине. Этот этап является заключительной частью в изучении дисциплине «Компьютерные сети». В результате прохождения всех этапов обучения, студенты получают приобретенные знания, практический опыт и оценку по дисциплине.

Для описания процесса обучения по дисциплине «Компьютерные сети» в графическом формате, была построена диаграмма декомпозиции «как есть» согласно нотации IDEF0. Данная диаграмма интерпретирует в графическом формате, все этапы обучения, протекающие во время обучения по дисциплине.

Диаграмма декомпозиции процесса обучения по дисциплине «Компьютерные сети» представлена на рисунке 1.3.

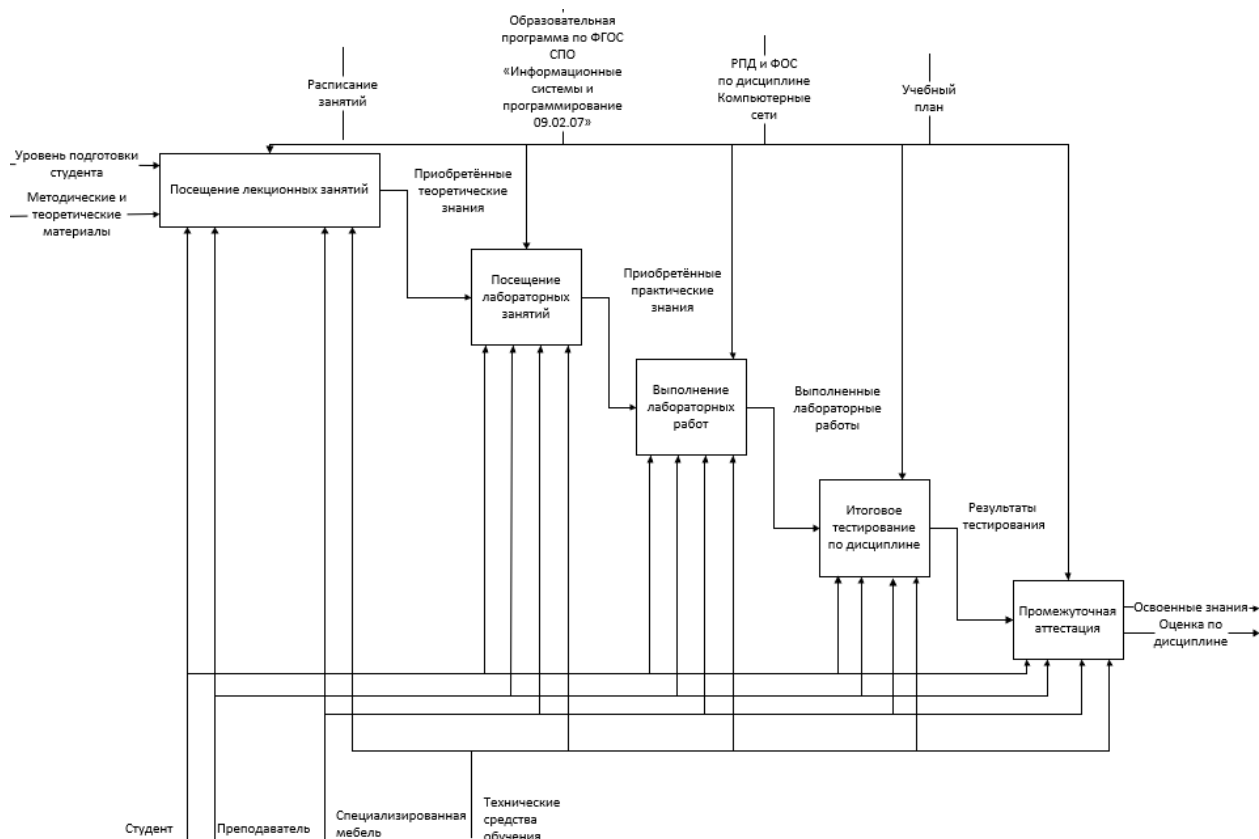


Рисунок 1.3 – Диаграмма декомпозиции «как есть» процесса обучения по дисциплине «Компьютерные сети»

После проведенного анализа объекта исследования были выявлены недостатки в общем образовательном процессе по дисциплине «Компьютерные сети». Эти недостатки связаны с тем, что учебный материал,

предоставляемый преподавателем в устоявшемся, классическом формате, может быть не всегда усвоен учениками в процессе обучения. Это может случиться не только потому, что преподаватель плохо преподносит учебный материал, или объема учебного материала не всегда достаточно для представления полноты картины изучаемой темы.

Причиной этих проблем может являться то, что некоторые студенты, зачастую, плохо воспринимают информацию через призму классического формата обучения. Монотонное чтение теоретического материала не всегда может аккумулироваться в полезную, осознанную информацию для студента.

По этой причине было предпринято внедрить процесс геймификации в процесс обучения компьютерных сетей путем разработки виртуальной лаборатории для изучения по дисциплине «Компьютерные сети». Геймификация позволит вызвать интерес у студентов, посредством изучения виртуальной лаборатории по дисциплине «Компьютерные сети». Компьютерная лаборатория будет иметь упрощенный стиль подачи информации, полное звуковое сопровождение, и различные трёхмерные модели оборудования. Всё это в купе повлияет, исключительно, положительным образом на усвоение и закрепление информации студентами. Помимо обычного изучения лаборатории у студентов будет возможность на любом этапе изучения лаборатории пройти различные тестирования внутри виртуальной лаборатории и оценить свой уровень знаний.

Методы подачи теоретического и практического материала через геймификацию процесса обучения поспособствует лучше запоминать и усваивать материал студентам. У студентов будут более углубленные знания в области технических аспектов оборудования компьютерных сетей, теоретических аспектов и так далее.

При использовании геймификации используются активные формы обучения, поэтому восприятие и запоминание информации при подаче материала с элементами геймификации приравнивается к усвоению информации, применяемой на практике или даже путем использования в

реальной жизни, и дает от 75 до 90% эффективности усвоения учебного материала [21, с. 62-63].

1.3 Определение цели и задач по разработке приложения

Цель исследования заключается в разработке виртуальной лаборатории для обучения по дисциплине «Компьютерные сети».

Виртуальная лаборатория для обучения по дисциплине «Компьютерные сети» позволит студентам повысить их заинтересованность, вовлеченность в образовательный процесс, углубить теоретические знания по дисциплине путем изучения образовательного материала через визуальное и слуховое восприятие, и применение этих знаний при построении компьютерных сетей в этой лаборатории.

Виртуальная лаборатория для обучения по дисциплине «Компьютерные сети» должна выполнять ряд следующих задач:

- обеспечивать интерактивное взаимодействие пользователя с лабораторией;
- обеспечивать получение пользователем теоретического материала через визуальное представление;
- обеспечить получение пользователем теоретического материала через слуховое восприятие;
- обеспечить интерактивное обучение пользователя, путем взаимодействия его с трёхмерными объектами лаборатории;
- предоставить возможность пользователям осуществлять проектирование сетей различных типов.

При полной реализации виртуальной лаборатории, образовательный процесс по дисциплине «Компьютерные сети» станет более образовательным, повысится вовлеченность студентов в образовательный процесс.

1.4 Обзор и анализ существующих технологий проектирования

После того как была сформулирована и поставлена цель исследования необходимо провести обзор программного обеспечения, в котором будет осуществляться трёхмерное моделирование требуемых моделей и непосредственно разработка виртуальной лаборатории.

Необходимо рассмотреть наиболее востребованные варианты программного обеспечения, проанализировать их преимущества и недостатки, провести сравнительный анализ и далее сделать выбор в пользу того или иного программного обеспечения с аргументированием выбора.

1.4.1 Autodesk Maya

Autodesk Maya является мощной программной средой для создания 3D-графики, визуальных специальных эффектов применяемых в видеоиграх, фильмах, всё различных приложениях. Maya содержит в себе полный набор инструментов для трёхмерного моделирования, создания анимации, рендеринга, скрининга и так далее.

Autodesk Maya является универсальной программной средой. Это означает, что проекты, которые были разработаны в этой программной среде, могут быть спокойно перенести на другую программу для трёхмерного моделирования, например – Blender, и продолжить разработку проектов.

Еще одним достоинством Autodesk Maya является тот факт, что данная программная среда, содержит полный набор инструментов, которые позволяют моделировать 3D-объекты различными способами, например, осуществлять различные манипуляции с вершинами и полигонами. Данный процесс моделирования называется «полигональным моделированием».

В простонародье, этот процесс называется «экструдированием вершин».

Также, в Maya, можно создавать трёхмерные объекты с использованием техники «скульптинга» и NURBS-моделирования.

Режим «скульптинг» – это один из методов трехмерного моделирования, который основан на взаимодействии с единой сеткой модели, а не с отдельными её элементами. Для создания модели в режиме «скульптинга», обычно, используется определенный набор инструментов, который позволяет осуществлять различные манипуляции с сеткой объекта (далее – мэша), например, выдавливать, скручивать, сжимать, выравнивать и так далее. В различных программных средах, ориентированных на трехмерное моделирование, наименования инструментов и методы взаимодействия с моделью в режиме «скульптинга» могут несколько отличаться, однако, процесс «скульптинга» в большинстве своём протекает похожим образом.

NURBS-моделирование – это процесс моделирования трёхмерных объектов, с использованием кривых поверхностей. Сокращение «NURBS» означает неоднородную рациональную поверхность В-сплайнов. Это может показаться сложным, но это просто означает, что все поверхности-NURBS это – кривые. Кривые поверхности строятся на основе математических формул, это означает, что редактирование таких кривых можно осуществлять бесконечное количество раз, и они могут иметь бесконечную длину и различную, плавно перетекающую форму.

Ближайшей аналогией NURBS-моделирования может выступать создание изображений с использованием векторной графики, основа которой, составляет математические формулы.

Maya идеально подойдет для всех, кому предстоит создавать отрисовывать сюжеты с максимальной реалистичностью, то есть представителям кино, мультипликации и игр [5, с. 184].

К недостаткам Autodesk Maya можно отнести достаточно высокую стоимость лицензии для одиночного пользователя, поэтому, данную программную среду нельзя отнести к доступным для большинства пользователей. Однако, Autodesk предоставляет бесплатный доступ для

студентов и преподавателей. Лицензионный пакет данного программного обеспечения будет вполне выгодно оплачивать крупным компаниям, специализирующимся на разработке проектов, либо тем пользователям, которые имеют высокий доход на регулярной основе.

Еще одним недостатком Maya является то, что порог вхождения для новичков может быть несколько затруднен из-за сложности освоения интерфейса программы. Связано это с тем, что большинство обучающего материала по данной программе отсутствует на русском языке.

В интернете можно найти множество обучающих видеороликов, но большинство их них не переведены на русский язык [6].

Ко всему прочему, погружение в трёхмерное моделирование, требует дополнительных знаний в области 3D-моделирования, например, что такое полигоны, виды, топология моделей, принципы создания моделей, знать общую логику построения моделей и так далее. Однако, этот недостаток можно отнести ко всему программному обеспечению, которое специализируется на создании трёхмерных объектов.

Интерфейсная часть программы «Autodesk Maya» продемонстрирован на рисунке 1.4.

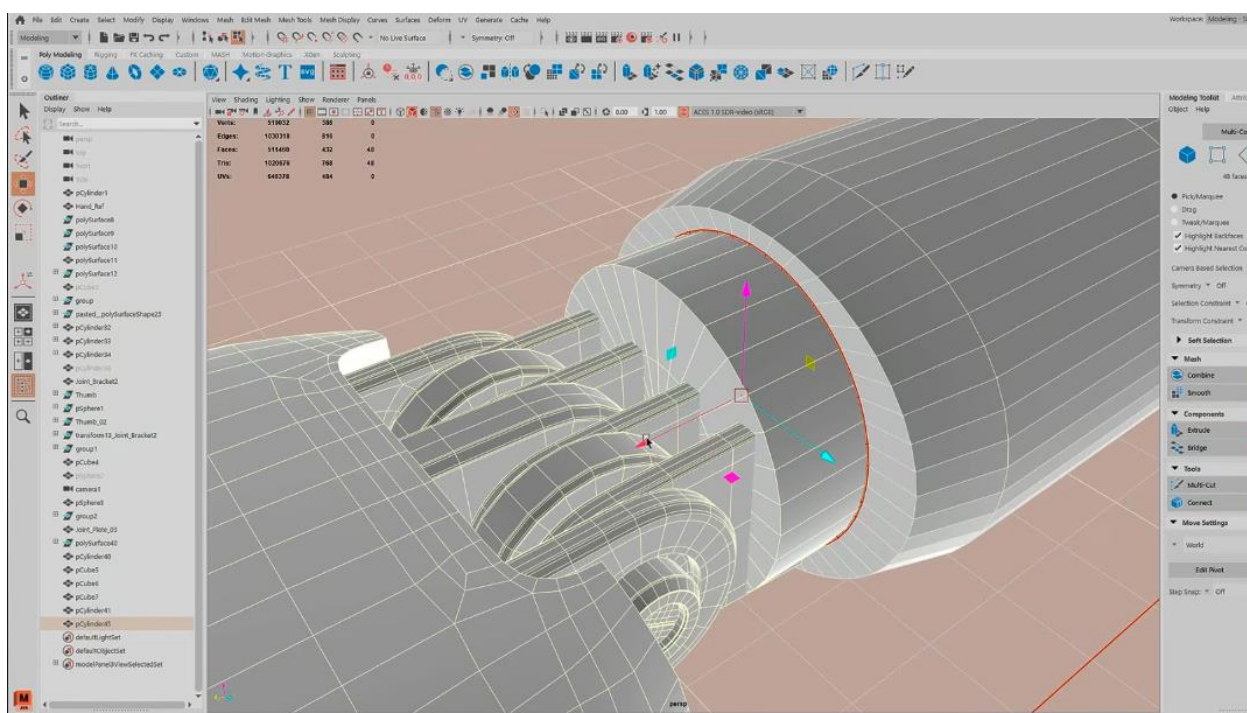


Рисунок 1.4 – Интерфейс «Autodesk Maya»

1.4.2 ZBrush

ZBrush – это крайне популярная, прогрессивная программная среда предназначенная для 3D моделирования. Главным преимуществом ZBrush является продуманный, комфортный процесс формирования различных органических трёхмерных объектов, иными словами, режим «скульптинга» или «лепки» (об этом режиме упоминалось в предыдущей главе). Если 3D-художник пребывает в поиске программы, которая предоставляет мощную среду для «скульптинга», то лучше, чем «ZBrush», программы нет, на данный момент.

Самым популярным и мощным 3D-редактором, в котором «лепят», является Zbrush. Поскольку Zbrush разработан для создания отдельных объектов, его обычно используют вместе с классическим редактором (например 3dsmax), где и собирают всю сцену вместе [14, с. 81-82].

Еще одной уникальной особенностью ZBrush, до, относительно, недавнего времени, считалась система симуляции ткани наделения физическими свойствами тот или иной объект. Существует набор кистей, которые наделены уникальными свойствами и могут по-разному взаимодействовать с мэшами. Например, существует кисть, которая при контакте с мэшем, наделяет его физическими свойствами, например утяжеляет объект, или наоборот, создает эффект ряби на плоскости. Так же существует кисть, которая придает объекту физические свойства, присущие определенному типу ткани (волнистость, шероховатость, скомканность). Стоит уточнить, что влияние кисти распределяется не на весь объект, а на определенный радиус воздействия кисти. Радиус воздействия можно регулировать.

При нынешних тенденциях быстрого развития компьютерной графики, ZBrush и по сей день остается крайне востребованной программной средой среди опытных и продвинутых пользователей, графических художников в индустрии кино, в сфере CG-анимации, игровой индустрии и так далее.

Рабочая среда и пользовательский интерфейс программной среды ZBrush продемонстрирован на рисунке 1.5.

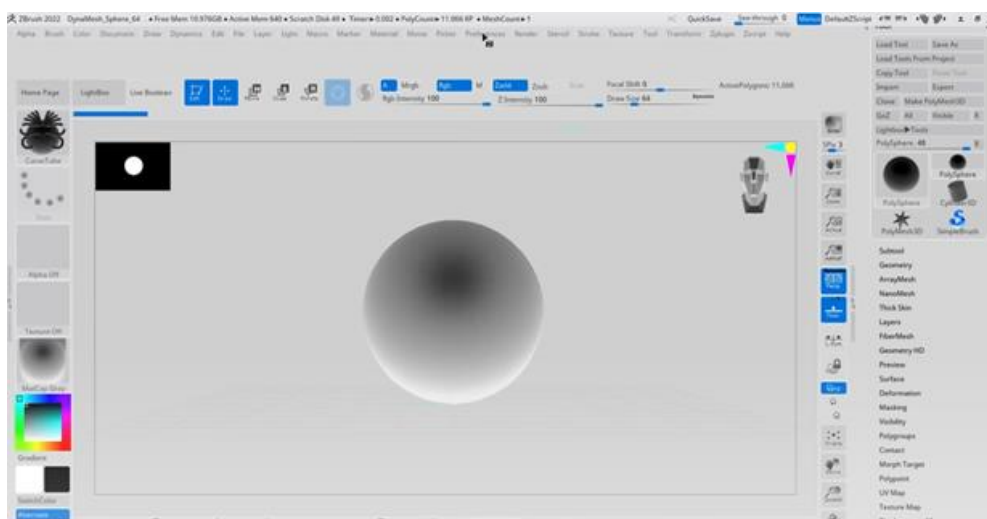


Рисунок 1.5 – Интерфейс программной среды «ZBrush»

Существует и ZBrush Core Mini – бесплатная версия, которая подходит для тех, кто только начинает свой путь в 3D мире и хочет познакомиться с возможностями Pixologic. Такое разнообразие вариантов лицензирования дает пользователям возможность выбрать оптимальный вариант, наиболее подходящий для их потребностей и бюджета [20].

К недостаткам ZBrush можно отнести то, что данное программное обеспечение распространяется исключительно на платной основе. К этой же причине можно добавить, относительно, высокую стоимость приобретения лицензии для инструмента, ключевой особенностью которого, заключается именно режим «скульптинга». Если пользователю нужен режим «скульптинга» из «ZBrush» только единожды, в определенный период времени, то есть не используя это приложение на постоянной основе, то, возможно, стоит обратить внимание на программное обеспечение, предоставляющее похожие возможности.

1.4.3 Blender

Blender является бесплатным программным обеспечением с открытым

исходным кодом, которое объединяет в себе широкий спектр возможностей, начиная от создания 2D-анимации, заканчивая продвинутым трёхмерным моделированием, созданием UV-разверток, текстурированием, созданием геометрических нод и всеми вышеописанными возможностями предыдущих программных продуктов.

На данный момент, Blender является полностью бесплатным программным обеспечением (за исключением некоторого количества расширений, призванных облегчить работу), которое активно поддерживается сообществом «Blender Foundation», в том числе и самой компанией.

Миссию Blender'а можно сформулировать так: «Передать лучшую в мире 3D-технологию в руки художников в виде открытого исходного кода и создавать с её помощью потрясающие вещи» [7].

На первый взгляд, трудно поверить, что Blender является полностью бесплатным программным обеспечением, учитывая тот факт, что Blender объединяет в себе огромное количество функций, однако, факт остается фактом.

Blender объединяет в себе следующие возможности:

- создание анимации и риггинг, маппинг;
- рендеринг с использованием трассировки лучей;
- трёхмерное моделирование;
- двухмерное моделирование;
- симуляция ткани, жидкостей, разрушения;
- физика объектов;
- видеомонтаж;
- написание скриптов на языке программирования «Python»;
- рисование, создание текстур;
- создание UV-карт моделей;
- запекание текстур;
- построение геометрических нод (geometry nodes);
- продвинутый режим «скульптинга»;

- полная интеграция проектов в игровые движки.

Помимо всех вышеописанных возможностей данной программы, Blender постоянно пополняется новыми расширениями и плагинами, которые позволяют ускорить большинство рутинных задач, например, сделать быструю ретопологию модели после «лепки» (для этого используется знаменитый плагин «Quad Remesher», но существуют и другие плагины) вместо того, чтобы вручную делать ретопологию, на которую может уйти не одни сутки. Другой пример: вместо того, чтобы создавать UV-развертку вручную, можно использовать плагины, например «UVPackMaster 3» или «Zen UV». В этом, как раз-таки, и заключается одна из ключевых возможностей Blender – любая рутинная задача, которая выполняется в Blender, может быть ускорена (упрощена) в несколько раз при использовании различных плагинов. Естественно, использование плагинов не является обязательным требованием к использованию программы, однако, если 3D-художнику необходимо сэкономить несколько часов своей жизни, он может прибегнуть к использованию специальных инструментов.

Частой практикой Blender является то, что сторонние плагины, которые были созданы энтузиастами, зачастую становятся частью программы. Например, в ранних версиях Blender (до 2.5), чтобы сделать булеву операцию между двумя объектами (вычесть один объект из другого или наоборот получить разницу объектов, объединить их в единую непрерывную сетку), необходимо установить сторонний плагин под названием «Bool Tool».

Данный плагин позволял быстро и без ручного труда мягко «сшивать» сетку (полигоны) двух разных объектов, при использовании нескольких сочетаний клавиш. Однако, спустя некоторое время, разработчики «Blender» внедрили эту функцию под видом одного из модификаторов, который впредь стал называться модификатором «Boolean». Таким образом, сторонний плагин стал частью данной среды разработки трёхмерных моделей.

Blender предоставляет возможности моделирования трёхмерных объектов путем различных методов, таких как: полигональное моделирование,

«лепка» моделей (на уровне ZBrush), NURBS-моделирование.

Функциональные возможности блендер позволяют реализовывать различного уровня симуляции ткани, физику объектов и так далее.

Главным достоинством Blender, является отсутствие требования в приобретении лицензии для этого программного обеспечения. Сам Blender выпускается под лицензией GNU General Public License [8].

Еще одним решающим фактором при выборе Blender является большая, структурированная и актуальная документация, которая постоянно обновляется и совершенствуется. С каждым выходом новой версии, появляется новая документация, разработанная специально для этой версии.

В отличие от Autodesk Maya, где основное инфо-поле сосредоточено вокруг иностранного комьюнити, Blender имеет просто огромное количество русскоязычных видеороликов обучающего характера, большое количество различных бесплатных курсов по созданию персонажей, сцен, в том числе и другого обучающего контента, любой сложности и характера на русскоязычных просторах интернета.

Стандартный интерфейс распространяемого бесплатного программного обеспечения «Blender» продемонстрирован на рисунке 1.6.

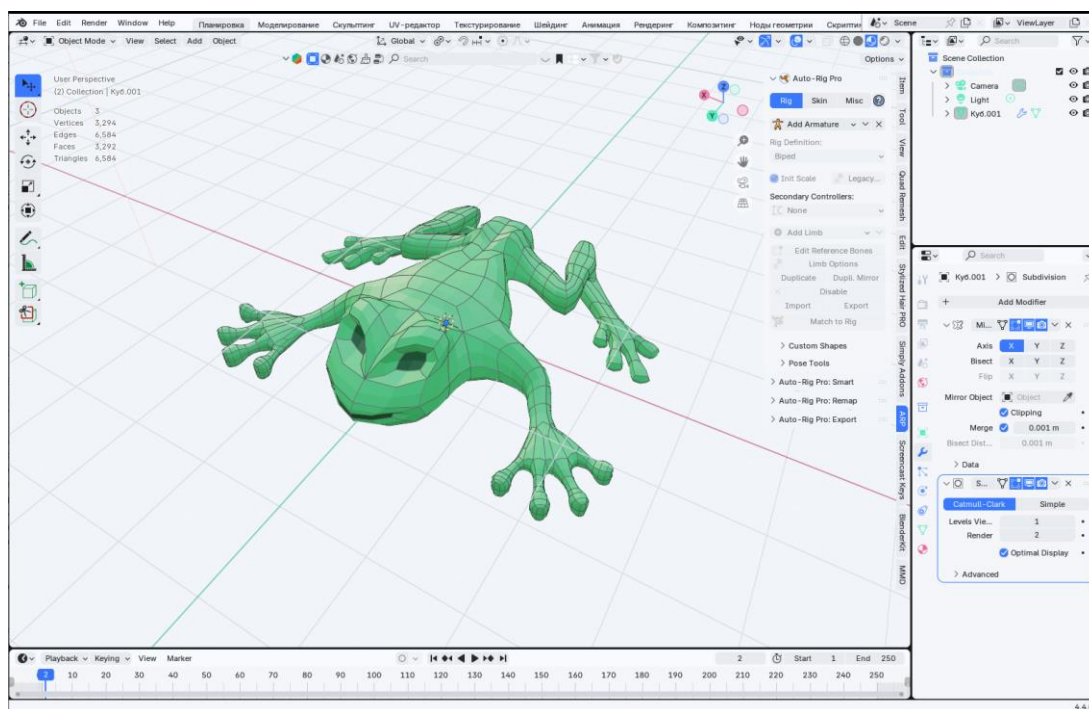


Рисунок 1.6 – Интерфейс «Blender»

1.4.4 Unreal Engine

Unreal Engine 4 (UE4) – игровой движок и редактор, разработанный Epic Games для создания игр и приложений, начиная от высокобюджетного консольного AAA-класса и заканчивая нишевой инди-разработкой для мобильных устройств [9, с. 13].

Данный игровой движок пользуется большой популярностью среди больших игровых студий, и компаний.

Unreal Engine является универсальным игровым движком, который содержит в себе большое количество функций, необходимых для создания проектов любого уровня сложности.

Unreal Engine обладает возможностью кроссплатформенной разработки, это означает что можно разработать проект который будет функционировать на нескольких операционных системах, таких как Linux, Android, Playstation и так далее.

Unreal Engine является универсальным игровым и доступным игровым движком. Данный движок имеет большое количество ассетов, функций, плагином и т.д. Данным игровым движком можно пользоваться бесплатно новичкам и профессионалам, до того момента, пока проект не принесёт прибыли от проекта до одного миллиона долларов. После превышения данных доходов, разработчикам придется выплачивать до 5% от общей прибыли.

Дополнительным преимуществом игрового движка «Unreal Engine» является поддержка визуального программирования. Инструмент позволяющий

заниматься визуальным программированием называется – Blueprint. Blueprint позволяет быстро реализовывать игровую логику без особых знаний программирования, например, логику управления персонажем (управления камерой с помощью захвата мыши, перемещением персонажа и так далее).

Эта технология позволит познакомиться с программированием людям, которые далеки от программирования без изучения синтаксиса языка

программирования, но это не обозначает, что разработчику не нужно знать теоретическую базу по программированию.

Если разработчик не планирует заниматься визуальным программированием, а хочет использовать классические языки программирования для скриптов, то придется изучить язык программирования C++.

Еще одним преимуществом данного игрового движка можно отнести работу с визуальной обработкой, шейдера, постобработкой и так далее. Если перед разработчиками стоит цель разработать проект с максимально красивой графикой, без особых материальных вложений в проект, то лучше проекта не найти, среди бесплатных аналогов.

К высокой доступности можно отнести огромное количество обучающего материала, которым делится объединенное сообщество пользователей.

Интерфейс Unreal Engine представлен на рисунке 1.7.



Рисунок 1.7 – Интерфейс игрового движка «Unreal Engine»

1.4.5 Платформа Unity

Unity – это мощная платформа для разработки 2D и 3D приложений и игр [10, с. 10].

Unity объединяет в себе большое количество программных компонентов, которое активно используется при разработке игр, такие как:

- систему создания пользовательских интерфейсов (ui);
- мощные графический рендереры;
- движок позволяющий реализовывать физику объектов и не только;
- системы быстрой интеграции скриптов на языке C#;
- системы быстрой интеграции ассетов в проект;
- систему частиц (particle system, vfx);
- модули позволяющие создавать анимации и интегрировать уже существующие из других смежных программ;
- редактор сцены;
- компонентный подход к разработке игр.

Unity поддерживает установку сторонних расширений, которые называются – ассетами. Для этого была создана отдельная онлайн-библиотека ассетов и плагинов, в которой можно приобретать различные, полезные плагины, ассеты, модели, текстуры, либо находить аналоги платным решениям. Всё это нужно для того, чтобы ускорить процесс разработки приложений и обеспечивать дополнительный комфорт разработчикам во время процесса разработки.

Еще одной сильной стороной Unity считается поддержка большого количества сторонних технологий, в том числе и API. Возможности Unity позволяют интегрировать приложения на различные устройства и операционные системы. Например, игру, разработанную на операционной системе «Windows» при желании, без особого труда можно интегрировать на операционную систему «Android», «Linux», и так далее. Иными словами, Unity обладает широкими возможностями в области построения кроссплатформенных приложений.

Еще одной важной особенностью Unity является поддержка VR-

технологий. Это будет важным аспектом для разработчиков, которые захотят импортировать свой проект на платформу VR.

Как и большинство современных игровых движков, по типу «Unreal Engine 5», Unity поддерживает современные технологии OpenGL и DirectX 12, в том числе и современную технологию трассировки лучей (ray tracing) и так далее.

Платформа Unity отличной подойдет начинающим разработчикам, студентам, преподавателям и небольшим студиям, которые разрабатывают небольшие проекты, поскольку Unity является бесплатной программной средой при соблюдении определенных условий. Приобретение лицензии необходимо только большим студиям и студиям доход которых превышает сотню тысяч USD.

Интерфейс платформы Unity представлен на рисунке 1.8.

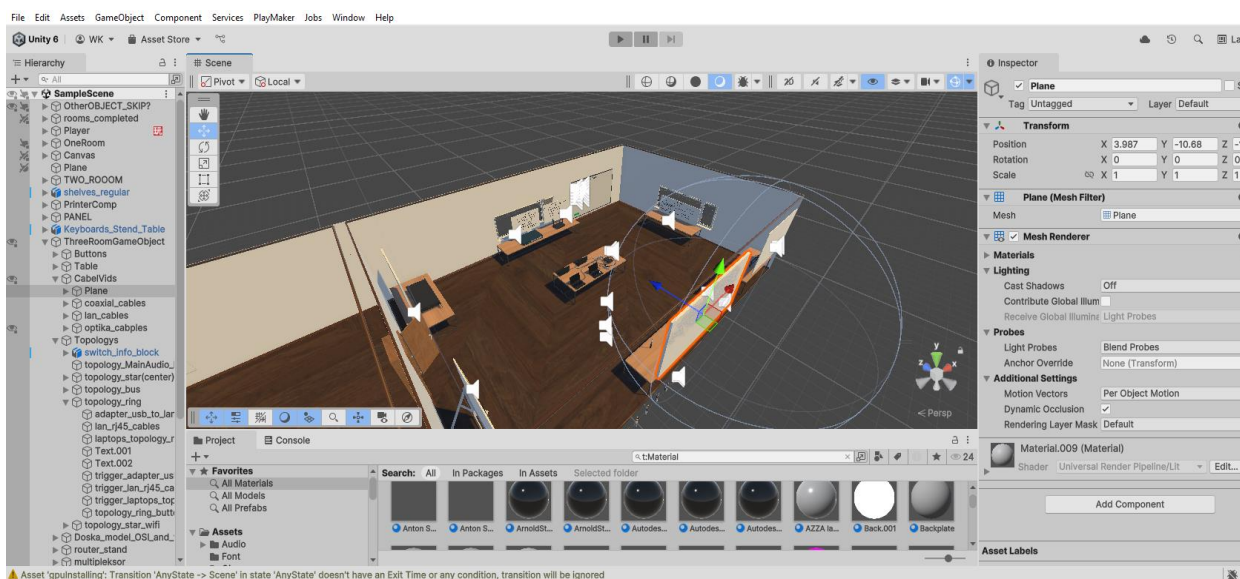


Рисунок 1.8 – Интерфейс платформы «Unity»

Кроме этого, Unity обладает дополнительным приложением под названием «Unity Hub», благодаря которому, можно без труда управлять проектами Unity, получать дополнительные новости об обновлениях, и осуществлять обновление или переустановку компонентов игрового движка. Unity Hub – это отдельное приложение, которое упрощает способ навигации, загрузки и управления проектами и установками Unity [19].

Помимо всего прочего, обучающий материал и официальная документация по платформе Unity непрерывно обновляется, поэтому, не составляет больших усилий найти тот или иной обучающий материал.

1.4.6 Godot

Godot – это бесплатный и открытый игровой движок, разрабатываемый сообществом. Он ориентирован на создание 2D- и 3D-игр и пользуется популярностью среди независимых разработчиков [11, с. 339].

По началу, «Godot» считался закрытым движком и был ориентирован на нужды частных компаний, иными словами, движок был проприетарным. Но спустя некоторое время, разработка и поддержка Godot перешла в руки сообщества «Godot Engine Community». Поддержка данного движка продолжается и по сей день.

По сравнению со своими конкурентами, движок считается очень компактным игровым движком, размер его составляет около сотни мегабайт, в то время как «Unity» вести около десяти гигабайт, и это без учёта веса проектов (базовый «скелет» проекта весит около 2гб).

Godot поставляется с полноценным игровым редактором со встроенными инструментами для решения самых распространенных задач. Он включает редактор кода, редактор анимации, редактор тайлмапов, редактор шейдеров, отладчик, профайлер и многое другое [18].

Для некоторых пользователей ручная установка может показаться слишком долгим процессом, однако, комьюнити обо всём позаботилось, чтобы пользователю было удобно загружать все необходимые плагины, аддоны, ассеты и так далее.

Доступ к репозиторию движка и его плагинам доступен в окне при запуске движка. Одним из плюсов для новичков в «Godot» является то, что документация по этому игровому движку доступна не только в главном репозитории в режиме онлайн, но и напрямую в самом движке. Таким образом,

можно сделать вывод о том, что данный игровой движок является достаточно лояльным по отношению к новичкам. В свою очередь, такая доступность к документации обеспечивает непрерывную разработку приложений, без необходимости искать информацию на внешних источниках, что может быть очень удобно для большинства пользователей.

Для разработки скриптов и реализации игровой механики в данной программной среде используется собственный язык программирования, который называется «GDScript». Данный язык программирования является высокоуровневым, скриптовым языком программирования с динамической типизацией.

Однако, если разработчику необходимо использовать другой язык программирования, без временных затрат на изучение «GDScript», Godot имеет возможность проводить интеграцию других популярных языков программирования, таких как: язык программирования Rust^б, C#, C++.

Достоинством данного языка программирования является то, что все игровые ресурсы, такие как, текстуры, скрипты, шейдеры, конфиги, дополнения, хранятся в движке как обычный набор файлов в системе проводника. Таким образом, данный подход позволяет команде разработчиков обеспечивать быстрый доступ к файлам проекта, используя штатные инструменты операционной системы, не пользуясь движком напрямую. Всё это благоприятно сказывается на контроле версии продукта и его «облегченности».

Из недостатков данного игрового движка можно отметить его низкую популярность на русскоязычных просторах интернета.

Еще одним незначительным недостатком может быть то, что данный игровой движок имеет слабые инструменты для поддержки AR/VR технологий.

Незначительным недостатком в этом движке для некоторых разработчиков может быть то, что выделить дополнительное время на некоторую подготовку перед разработкой, иными словами, установить все

необходимые плагины и расширения, перед началом разработки.

В общем и целом, из недостатков данного движка можно отметить отсутствие некоторых узконаправленных компонентов в базовой сборке, например, инструменты для создания UI, невозможность настройки визуальных эффектов, инструменты для настройки физических свойств двухмерных частиц и так далее. Необходимо дополнительное время на подготовку движка.

Если не учитывать все вышеописанные недостатки, то игровой движок «Godot» является отличным решением, как для начинающих разработчиков, которые хотят попробовать себя в разработке игр, так и профессионалов своего дела.

Интерфейс игрового, бесплатного движка «Godot» продемонстрирован на рисунке 1.9.

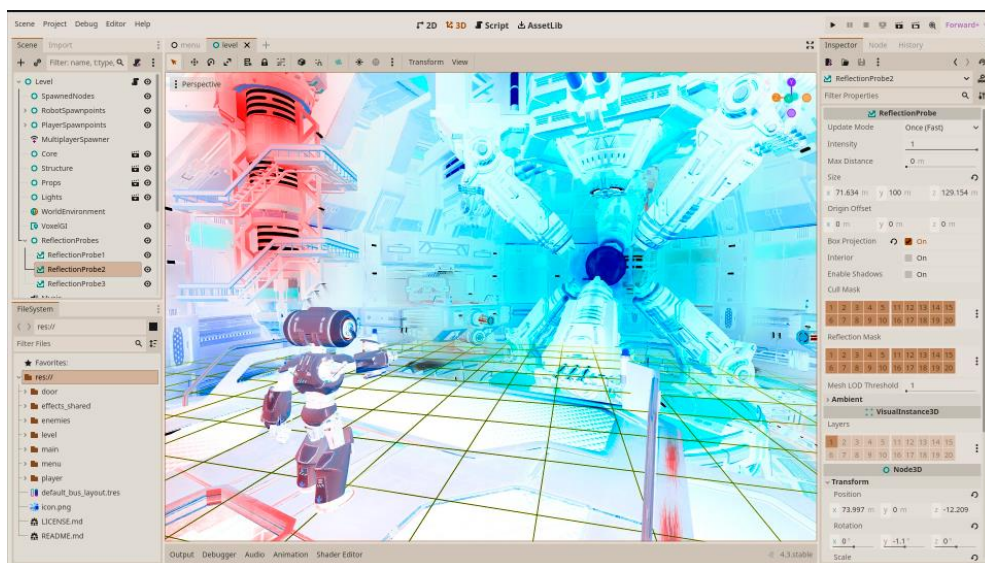


Рисунок 1.9 – Интерфейс игрового движка «Godot»

1.5 Выбор и обоснование проектных решений

1.5.1 Техническое обеспечение

Техническое обеспечение Рубцовского института (филиала) АлтГУ (то есть, рабочие станции) рекомендуется привести к соответствующим

минимальным системным требованиям программного обеспечения, на базе которого будет функционировать разработанная виртуальная лаборатория.

Системные требования представляют из себя набор стандартов, требований, которые выдвигаются программным обеспечением, для нормального, стабильного функционирования приложения.

Для стабильного функционирования виртуальной лаборатории для обучения по дисциплине «Компьютерные сети», необходимо соответствовать следующим минимальным системным требованиям, выдвинутым для рабочих станций.

Минимальные системные требования для стабильного функционирования представлены в таблице 1.3.

Таблица 1.3 – Минимальные системные требования для работы для стабильного функционирования виртуальной лаборатории.

Характеристика	Минимальные системные требования
Операционная система	Windows 10, Windows 11
Процессор	Intel core i5 2400 или AMD FX-6300
Оперативная память, гб.	8
Наличие твердотельного накопителя	Да
Объем ПЗУ, гб.	256
Видеокарта	GTX 1050ti или RX480

1.5.2 Программное обеспечение

Системное программное обеспечение (далее – ПО) – это набор программных продуктов, предназначенных для работы компьютера или сети, который может включать в себя операционные системы (далее – ОС), драйверы устройств, программы для управления файлами и базами данных, а также многое другое [12].

После проведения обзора на программное обеспечение предоставляющие возможности для трёхмерного моделирования и не дополнительных возможностей связанным с этим, был сделан выбор в сторону

«Blender».

В качестве платформы предоставляющей методы реализации игровой механики (игрового движка) была сделан выбор в сторону Unity.

1.5.3 Лингвистическое обеспечение

Лингвистическое обеспечение (ЛО) включает совокупность научно-технических термином и других языковых средств, используемых в информационных системах, а также правил формализации естественного языка, включающих методы сжатия и раскрытия текстовой информации с целью повышения эффективности автоматизированной обработки информации и облегчающих общение человека с ИС [15, с. 37].

Взаимодействие пользователя с виртуальной лабораторией будет происходить посредством изучения текстовой и звуковой информации. Звуковая и текстовая информация ориентирована на русскоговорящих пользователей. Элементы интерфейса, такие как, экранные подсказки, элементы пользовательского интерфейса (UI), информационные стенды содержат исключительно русский язык.

При дальнейшей разработке, не исключено внедрение дополнительной локализации в проект.

1.5.4 Обоснование выбора «Blender»

Выбор данного программного обеспечения обусловлен тем, что «Blender» является бесплатным программным обеспечением с открытым исходным кодом, таким образом, у пользователя отсутствует необходимость в приобретении лицензии, либо подписки, как это может произойти при выборе таких программ, как «ZBrush» или «Autodesk Maya». Пользователь может без лишних материальных затрат, создавать проекты в данной программе и монетизировать их по своему желанию.

Следующим решающим фактором, является то, что «Blender» содержит большое количество обучающего контента на русском языке в сети Ethernet, таким образом, будет минимизировано потраченное время на поиск и дополнительное изучение образовательного материала.

1.5.5 Обоснование выбора «Unity»

Сравнительная таблица программного обеспечения по функциональным возможностям и прочим критериям, предназначенного для трёхмерного моделирования продемонстрирована в таблице 1.4.

Таблица 1.4 – Сравнение программного обеспечения для трехмерного моделирования по функциональным возможностям.

Критерий	Autodesk Maya	ZBrush	Blender
Полигональное моделирование	+++	+	+++
Создание анимации	++	-	+++
Универсальность	++	+	+++
Риггинг	+++	-	++
Текстурирование	++	+++	++
Необходимость в покупке лицензии	+	+	-
Открытый исходный код	-	-	+
Поддержка	++	++	+++

Сравнительная таблица для игровых платформ разработки приложений по функциональным возможностям продемонстрирована в таблице 1.5.

Таблица 1.5 – Сравнение платформ по функциональным возможностям.

Критерий	Unreal Engine	Unity	Godot
Универсальность	++	+++	++
Сложность в обучении	+	+++	++
Реализация в VR	++	+++	+
Стоимость	++	+++	+++
Графика	+++	++	+

В качестве игрового движка, на которой будет функционировать виртуальная лаборатория, была выбрана платформа Unity. Выбор Unity обусловлен тем, что данный игровой движок с последним выходом обновлений является одним из самых стабильных среды своих предыдущих

представителей (предыдущие версии Unity). С каждым выходом обновлений, Unity совершенствуется, исправляются технические неполадки и т.д. Еще одним решающим фактором стало то, что Unity является бесплатной платформой для разработки приложений. Этот вариант отлично подойдет для энтузиастов, новичков, индивидуальных разработчиков, студентов и небольших инди-студий, доходы которых не превышают определенный порог.

Следующим фактором выбора Unity является то, что основным языком программирования в Unity является объектно-ориентированный язык программирования C#.

Данный язык программирования наделен, относительно, простым и логически понятным синтаксисом, современными решениями в области программирования, и в целом, зарекомендовал себя как «хороший» язык программирования, на котором написано огромное количество программного обеспечения.

На основании представленных таблиц 1.4 и 1.5 можно сделать вывод о том, что для выполнения поставленных задач наилучшим образом подходят именно Blender и Unity для проектирования и разработки виртуальной лаборатории для обучения по дисциплине «Компьютерные сети».

2 Проектная часть

2.1 Решение проблем после анализа объекта исследования

После проведенного анализа объекта исследования и контекстной диаграммы «как есть» процесса обучения по дисциплине «Компьютерные сети» и проведения декомпозиции диаграммы была построена контекстная диаграмма IDEF0 «как будет» для графического интерпретирования.

Контекстная диаграмма «как будет» процесса обучения по дисциплине «Компьютерные сети» продемонстрирована на рисунке 2.1.

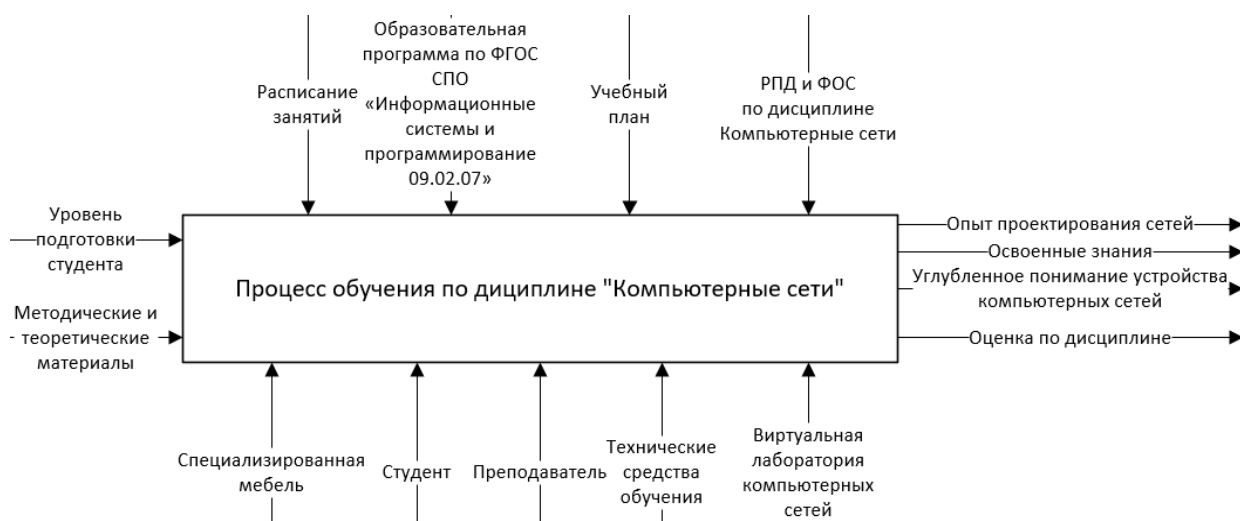


Рисунок 2.1 – Контекстная диаграмма «ТО-ВЕ» процесса обучения по дисциплине «Компьютерные сети»

На данной диаграмме представлен общий процесс обучения по дисциплине «Компьютерные сети» после внедрения процесса геймификации в образовательный процесс.

Можно заметить, что появился новый механизм «Виртуальная лаборатория компьютерных сетей». Данный механизм выступает как средство интерактивного взаимодействия пользователя с цифровым сетевым оборудованием и средством, позволяющим пользователю приобретать

практические навыки в построении сетей различных типов. Помимо интерактивного взаимодействия с оборудованием, пользователь получит возможность приобретать знания в альтернативном формате (звуковом формате) в области компьютерного оборудования, устройства сетей.

Декомпозиция контекстной диаграммы «как будет» процесса обучения по дисциплине «Компьютерные сети» представлена в приложении А (рисунок А1).

В диаграмме декомпозиции контекстной диаграммы «как будет» появились новые процессы:

- знакомство с виртуальной лабораторией;
- приобретение знаний в виртуальной лаборатории;
- построение сетей в виртуальной лаборатории;
- контроль и проверка знаний в виртуальной лаборатории.

Под процессом «Знакомство с виртуальной лабораторией» понимается знакомство с устройством лаборатории, изучение планировки лаборатории, изучение управления внутриигровым персонажем, знакомство с внутриигровым интерфейсом и сопроводительными подсказками. После знакомства с виртуальной лабораторией, пользователь будет понимать, как управлять персонажем, как взаимодействовать с лабораторией.

Процесс «Приобретение знаний в виртуальной лаборатории» подразумевает изучение пользователем следующей информационной составляющей в рамках виртуальной лаборатории:

- изучение информационных стендов с оборудованием, изучение устройства оборудования, изучение инструментов, которое применяется при построении сетей, изучение различных топологий сетей;
- изучения информационных стендов с оборудованием посредством звукового сопровождения;
- изучение моделей «OSI» и «TCP/IP»;
- изучение глоссария по компьютерным сетям.

Процесс «Построение сетей в виртуальной лаборатории» представляет из себя процесс, в ходе которого пользователь приобретает практические навыки в построении сетей по различным топологиям. После процесса построения сетей в виртуальной лаборатории у пользователя появится понимание, каким образом осуществляется построение сетей различных типов, и какое оборудование используется при построении.

Процесс «Контроль и проверка знаний в виртуальной лаборатории» описывает процесс прохождения всех этапов тестирования в рамках виртуальной лаборатории. Прохождения тестов по теоретическому материалу (тест по моделям OSI и TCP/IP), тестирование по глоссарию, получение оценок за проектирование сетей являются промежуточными этапами тестирования в рамках виртуальной лаборатории. После процесса контроля и проверки знаний, пользователь проверит свои знания, которые он приобрел при изучении виртуальной лаборатории.

На выходе появилось «Углубленное понимание устройства компьютерных сетей». Это означает, что пользователь получает углубленные знания от виртуальной лаборатории, которые возможно мог не усвоить при классическом формате обучения.

Помимо этого, пользователь ближе познакомится с типовым оборудованием (получит знания, об устройстве оборудования, из каких компонентов состоит оборудование и т.д.) применяющимся при построении компьютерных сетей.

2.2 Архитектура и принцип функционирования программы

Виртуальная лаборатория будет функционировать в пределах разработанной концепции, которая детально описана далее.

При инициации запуска приложения на экране отображается логотип

программы. После того, как приложение полностью загрузилось, управляемый персонаж появляется в лаборатории в центральном помещении. Центральное помещение представляет собой хаб (hub), который является нейтральной комнатой, связующим звеном между двумя комнатами (комнатой с теоретическим материалом и комнатой, предназначенной для проектирования сетей). В центральном помещении будет содержаться следующая информация:

1. Информация о прохождении этапов тестирования. Список тестов, их названия и количество полученных баллов за каждый этап контроля.
2. Карта-ориентир. На карте представлены все комнаты и указано, что находится в каждой комнате.

При появлении управляемого персонажа на карте, в интерфейсе пользователя будут отображаться следующие подсказки по управлению:

1. Перемещение с помощью клавиш «W», «A», «S», «D».
2. Бег при нажатии на клавишу «SHIFT».
3. Приседание при нажатии на клавишу «CTRL»
4. Подобрать предмет при нажатии на клавишу «E».
5. Выбросить предмет при нажатии на клавишу «Q».
6. Установить предмет при нажатии на клавишу «F».
7. Вернуться к справочной при нажатии на клавишу «T».

Перед началом изучения виртуальной лаборатории, настоятельно рекомендуется изучить карту-ориентир, которая расположена перед камерой у управляемого персонажа при появлении.

После изучения карты-ориентир можно будет приступить к изучению лаборатории по сетям.

Лаборатория компьютерных сетей разделена на две комнаты:

1. Первая комната содержит в большей мере информационный контент (информационные стенды, глоссарий, интерактивная доска по моделям «OSI/TCP/IP» анимационный стенд, оборудование для компьютерных сетей, применяемые при построении компьютерных сетей

кабели, роутеры, коммутаторы, повторители и так далее).

2. Вторая комната представляет из себя полигон для закрепления навыков в области проектирования компьютерных сетей (построение сетей по топологии «звезда» и менее распространенной топологии «общая шина»).

Каждая комната содержит системы контроля знаний (тесты по теоретическому материалу). За прохождение каждого теста, в системе фиксируется оценка (балл), которые можно посмотреть после прохождения оных. Тесты можно проходить неограниченное количество раз. По прохождении всех тестов вычисляется средняя оценка за все тесты.

2.3 Проектирование трехмерных моделей лаборатории и мебели

После того как была построена общая концепция приложения, можно приступить к разработке трёхмерной модели лаборатории. Для этого использовалась среда трёхмерного моделирования «Blender».

При разработке трёхмерной модели комнаты, применялось полигональное моделирование с применением простых геометрических форм.

Трёхмерная модель лаборатории с изображена на рисунке 2.2.

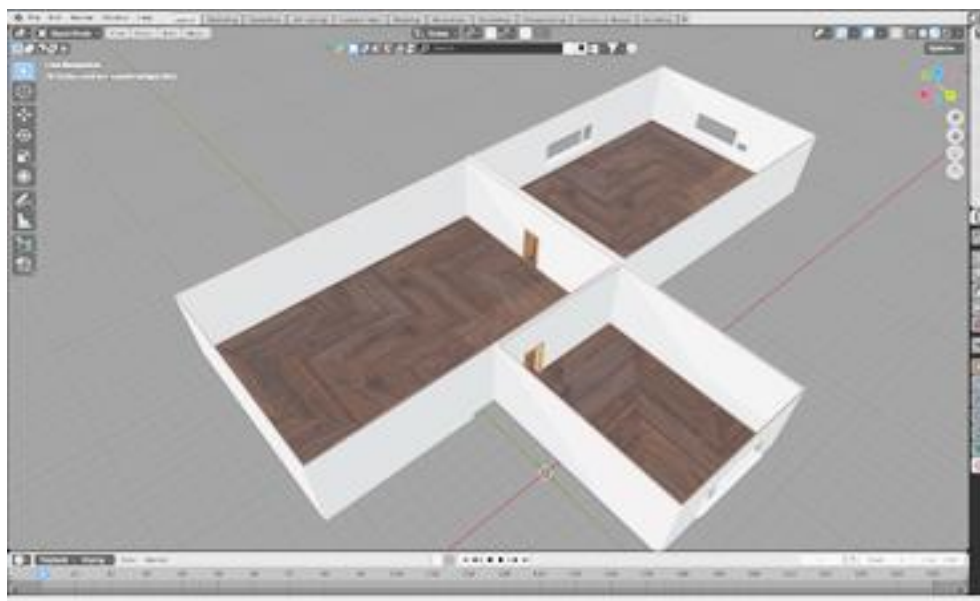


Рисунок 2.2 – Трёхмерная модель лаборатории компьютерных сетей

Чтобы лаборатория не выглядела слишком монотонно и была приятно воспринимаема человеческими глазами, были добавлены текстуры древесины для пола, добавлены плинтусы, двери, окна. Всё это влияет на иммерсивность, которая в свою очередь будет положительным образом влиять на общее восприятие пространства лаборатории.

После того как была создана модель трёхмерной лаборатории, можно приступить к созданию мебели для информационных стендов лаборатории.

При создании мебели использовались референсы мебели в стиле «Лофт», средних, малых и больших размеров. Данный стиль мебели отлично сочетается со стилем лаборатории и не перегружает общий дизайн комнат.

Спроектированные трёхмерные модели мебели в среде Blender представлены на рисунке 2.3.

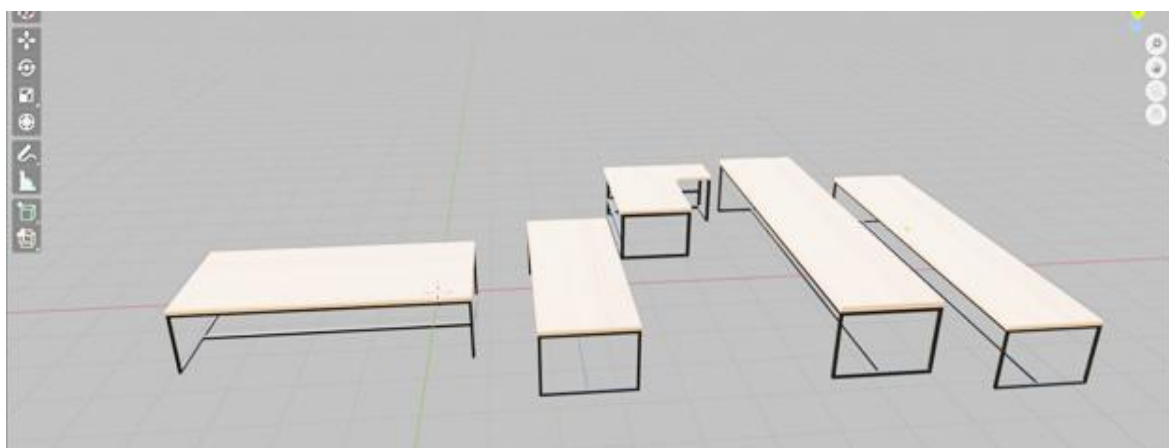


Рисунок 2.3 – Спроектированная мебель для лаборатории

2.4 Проектирование трёхмерных моделей для информационного обеспечения лаборатории

Информационное обеспечение представляет собой информационные стенды с оборудованием и соответствующей информацией к ним, которую будет изучать пользователь в процессе исследования лаборатории.

При создании информационного обеспечения использовались реальные модели оборудования компьютерных сетей, которые на регулярной основе

используются при построении локальных и глобальных сетей различных типов. Для информационного обеспечения лаборатории были разработаны следующие трехмерные модели оборудования:

- маршрутизатор (роутер);
- коммутатор (switch);
- мультиплексор;
- повторитель;
- коаксиальные кабели;
- кабели типа «витая пара»;
- оптоволоконные кабели;
- анимационный стенд с демонстрацией процесса обжима кабеля типа «витая пара» по стандарту «t568b»;
- топологии различных сетей (общая шина, звезда, кольцо, беспроводная сеть типа «звезда»);
- интерактивный глоссарий;
- интерактивная доска с теоретическим материалом на тему «модель osi» и «модель tcp/ip»;
- карта-ориентир;

Каждый информационный стенд с оборудованием содержит информационный блок, в котором содержится описание оборудования (общая информация про оборудование, его прямое назначение), кнопка «Воспроизвести», и кнопка запуска анимации (квадратная кнопка с закругленными краями, в центре которой находится стрелка, символизирующая воспроизведение анимации), на стендах где применяются анимации.

Каждый информационный стенд содержит в себе не только блок с информацией, но и дополнительные элементы взаимодействия с моделью, называемые – триггерами.

Триггер представляет собой простейший, объемный объект, при наведении на который срабатывает определенное действие на сцене –

отображение дополнительной информации на пользовательском интерфейсе.

Данный триггер является переиспользуемым компонентом программы и поэтому он используется для всех интерактивных стендов с оборудованием, где необходимо донести до пользователя определенную информацию. Триггер представлен на рисунке 2.4.

При наведении курсора на триггер, срабатывает OnTriggerEnter и края объекта триггера подсвечиваются. Это означает, что пользователь действительно навёл курсор на этот элемент и подсвеченный триггер является активным элементом в текущий момент.

OnTriggerEnter вызывается, когда Collider other входит в триггер [24].

Такой подход к взаимодействию пользователя с объектами положительным образом влияет на пользовательский опыт.



Рисунок 2.4 – Триггер для взаимодействия с информационными стендами

Помимо блока с информацией, каждый стенд сопровождается звуковой информацией. Чтобы воспроизвести звук и получить информацию об оборудовании, необходимо нажать на соответствующую кнопку «Воспроизвести» которая находится возле каждого блока с текстовой информацией. Кнопка представляет собой относительно плоский объект квадратной формы, оранжевого цвета с закругленными краями, с иконкой динамика в центре объекта.

Динамик символизирует процесс начала воспроизведения звука. При

наведении курсора на кнопку, срабатывает хOVER-область и края кнопки подсвечиваются. Это означает, что элемент активен. Это указывает на то, что пользователь может воспроизвести звук и прослушать описание оборудования в ином формате изложения. Текст и звуки с информацией отличаются по содержательности, также, как, например, речь выступающего и его презентация, которая лишь подкрепляет сказанную информацию выступающим. Кнопка воспроизведения звуковой информации так же является переиспользуемым компонентом программы, и поэтому используется везде, где необходимо передать информацию в звуковом формате.

Кнопка «Воспроизвести» продемонстрирована на рисунке 2.5.



Рисунок 2.5 – Кнопка «Воспроизвести»

На рисунке 2.6 представлен стенд с маршрутизатором, разработанным в среде Blender.

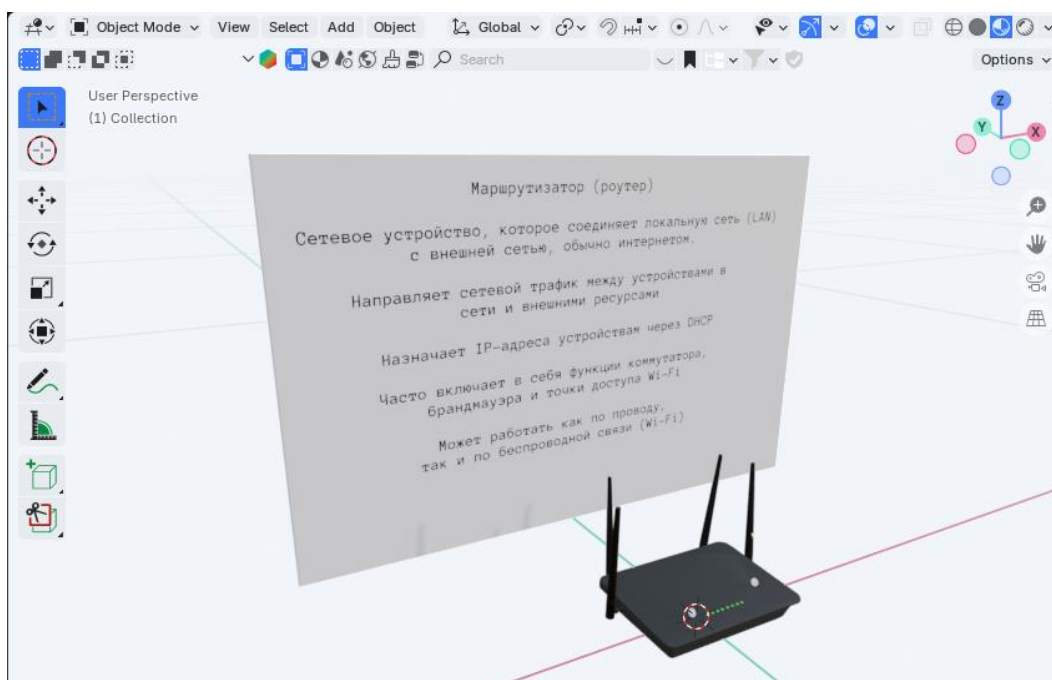


Рисунок 2.6 – Информационный стенд «Маршрутизатор (роутер)»

На рисунке 2.7 изображена трёхмерная модель мультиплексора, разработанная в среде Blender.

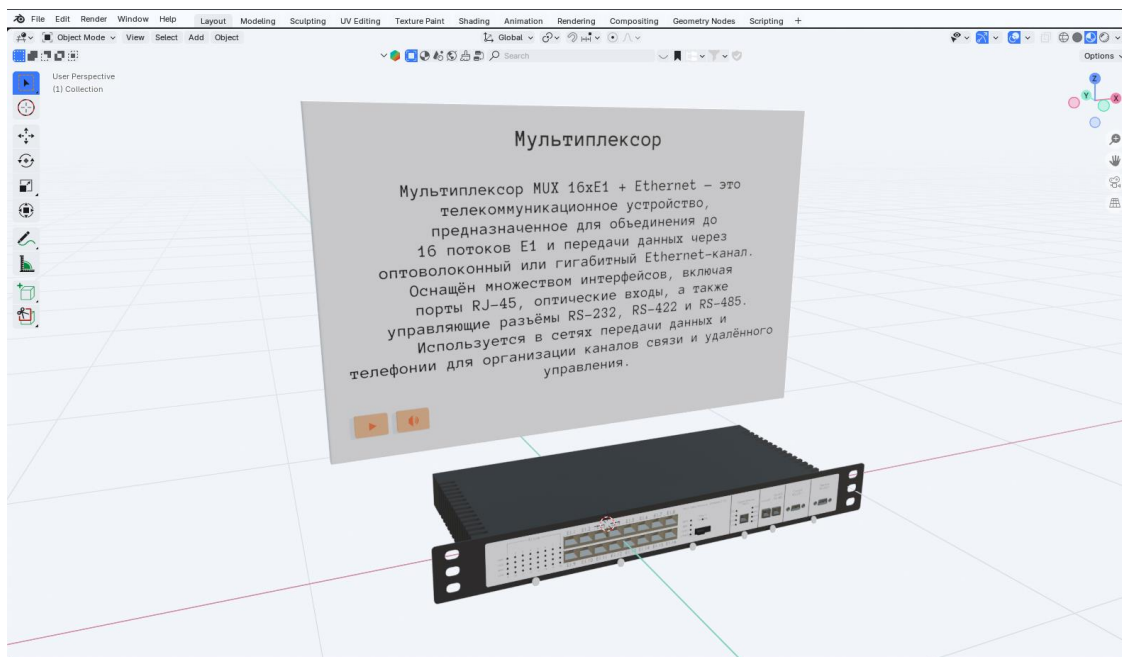


Рисунок 2.7 – Информационный стенд «Мультиплексор»

На рисунке 2.8 изображена трёхмерные модели повторителей, разработанные в среде Blender.

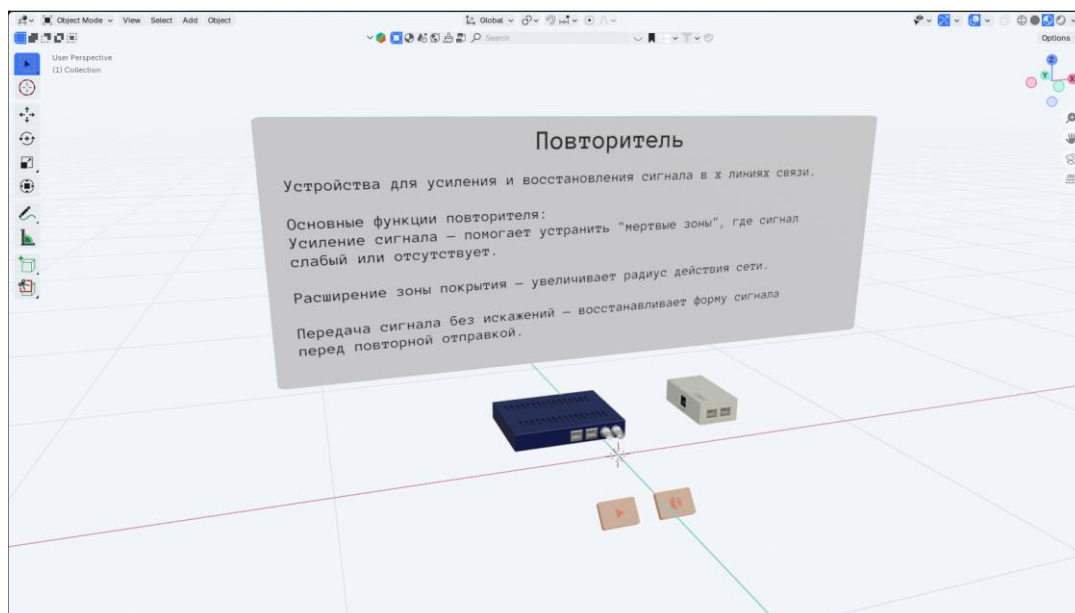


Рисунок 2.8 – Информационный стенд «Повторитель»

На рисунке 2.9 изображены трёхмерные модели коаксиальных кабелей, разработанных в среде Blender.

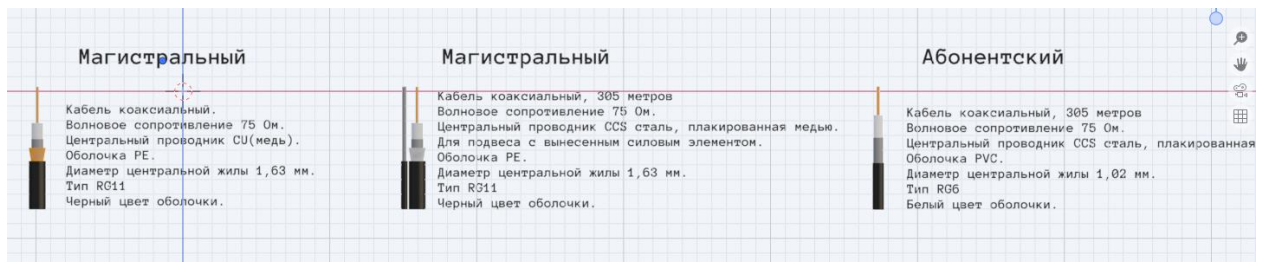


Рисунок 2.9 – Информационный стенд с коаксиальным кабелем

На рисунке 2.10 изображены трёхмерная модели кабелей типа «витая пара», разработанные в среде Blender.

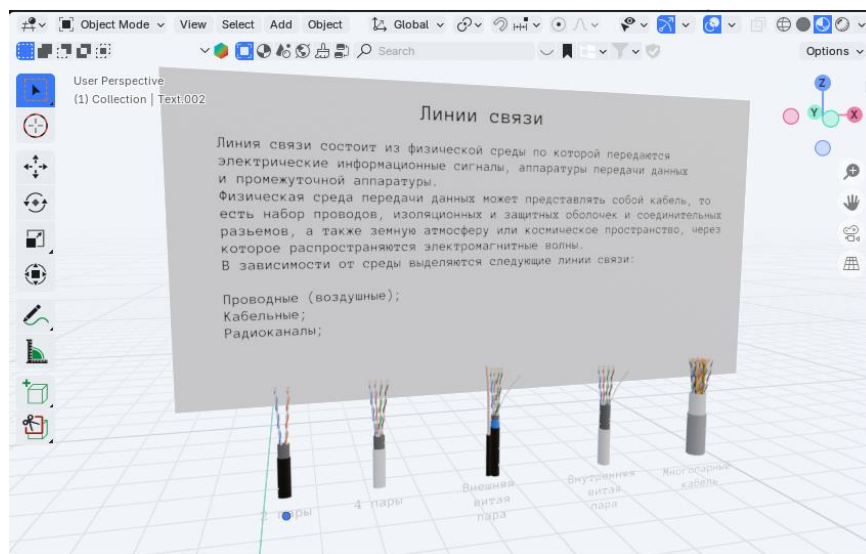


Рисунок 2.10 – Информационный стенд кабеля типа «витая пара»

На рисунке 2.11 изображены трёхмерные модели оптоволоконных кабелей, разработанных в среде Blender.

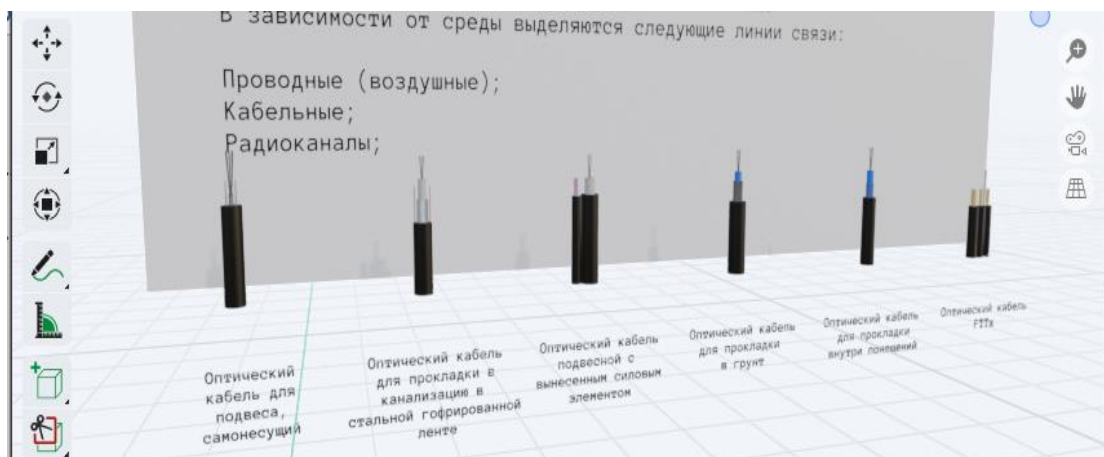


Рисунок 2.11 – Информационный стенд топологии «Оптоволоконный кабель»

Анимационный стенд с процессом обжима кабеля типа «витая пара» по стандарту «Т568В» продемонстрирован на рисунке 2.12.

Данный информационный стенд проектировался с применением анимации, которая будет проигрываться при нажатии на кнопку «Воспроизвести».

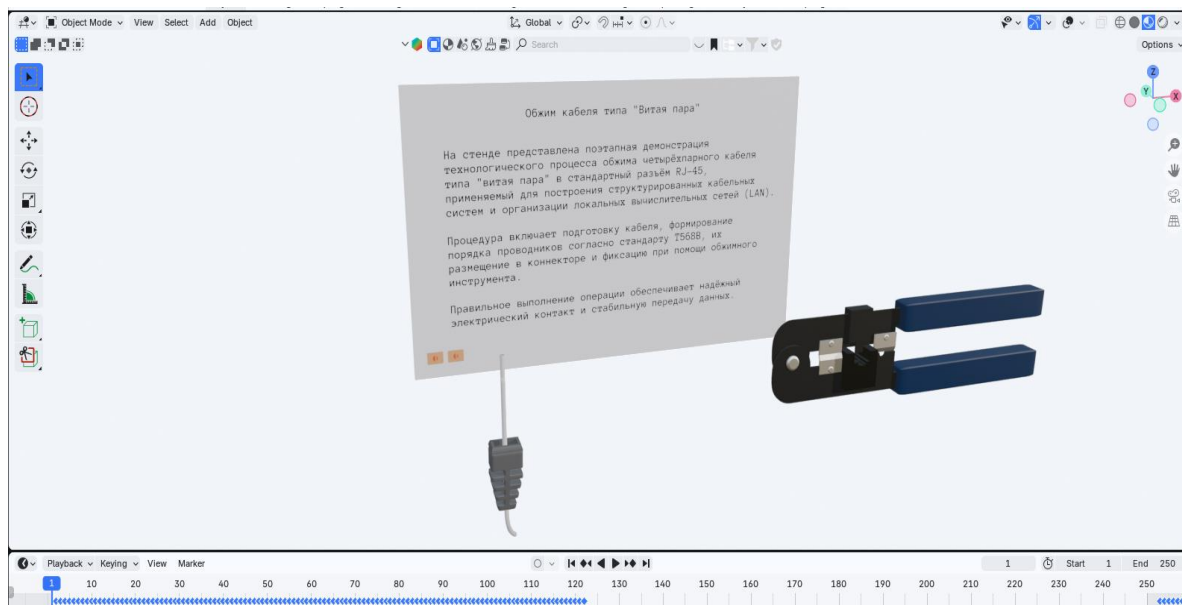


Рисунок 2.12 – Информационный стенд обжима кабеля типа «витая пара»

На рисунках 2.13-2.16 продемонстрированы трёхмерные модели топологий различных сетей («общая шина», «звезда», «кольцо», беспроводная сеть типа «звезда»).

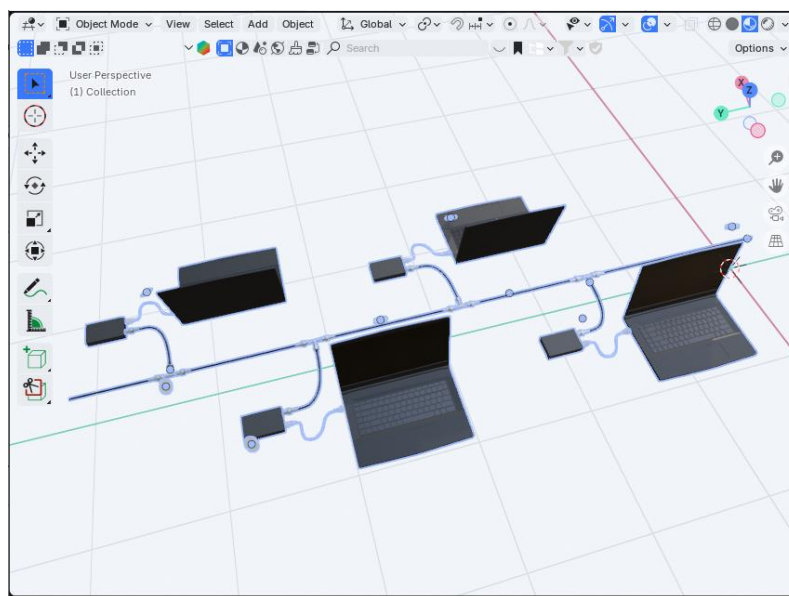


Рисунок 2.13 – Информационный стенд топологии «Общая шина»

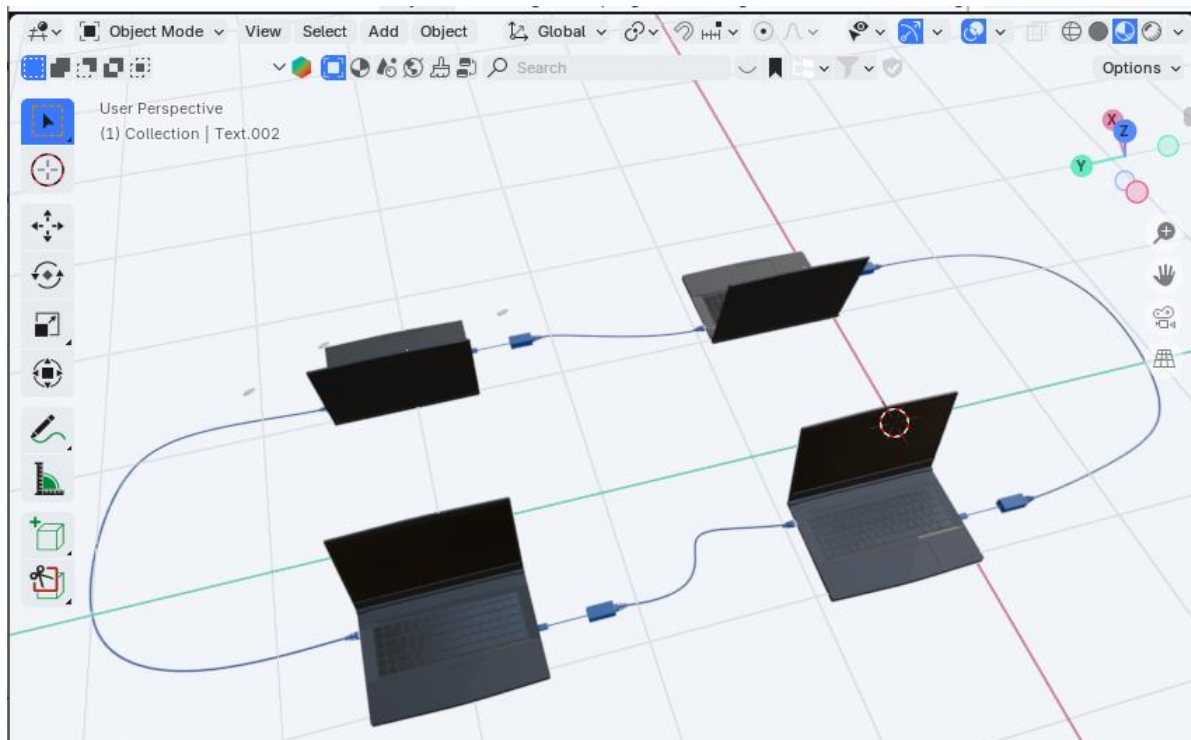


Рисунок 2.14 – Информационный стенд топологии «Кольцо»

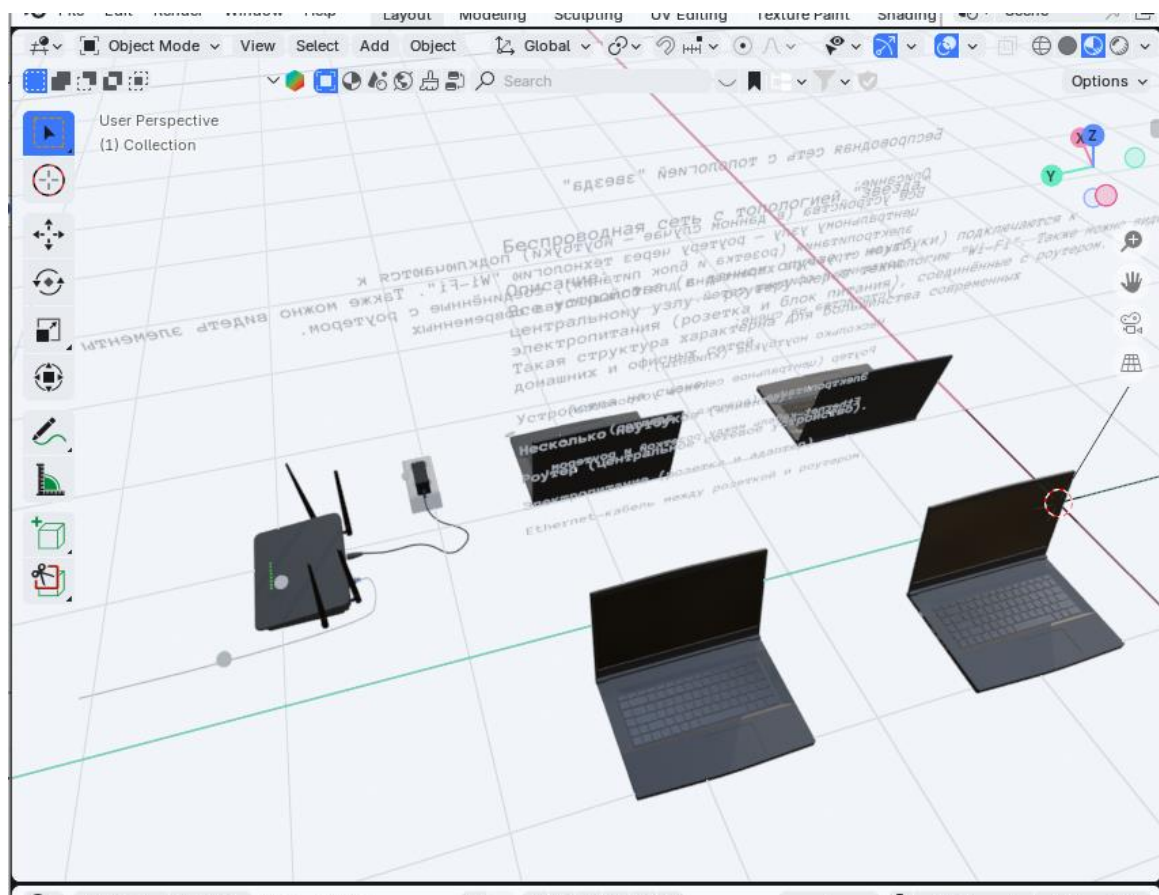


Рисунок 2.15 – Информационный стенд топологии беспроводной сети типа «Звезда»

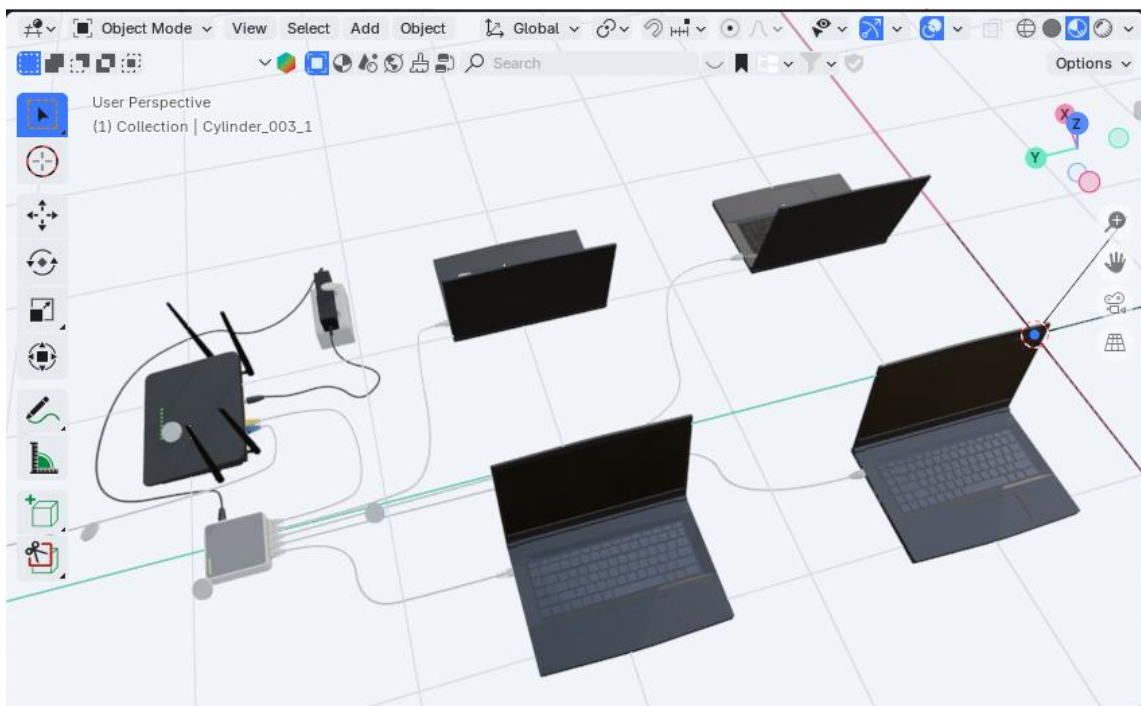


Рисунок 2.16 – Информационный стенд топологии «Звезда»

На рисунке 2.17 представлен спроектированный интерактивный глоссарий.

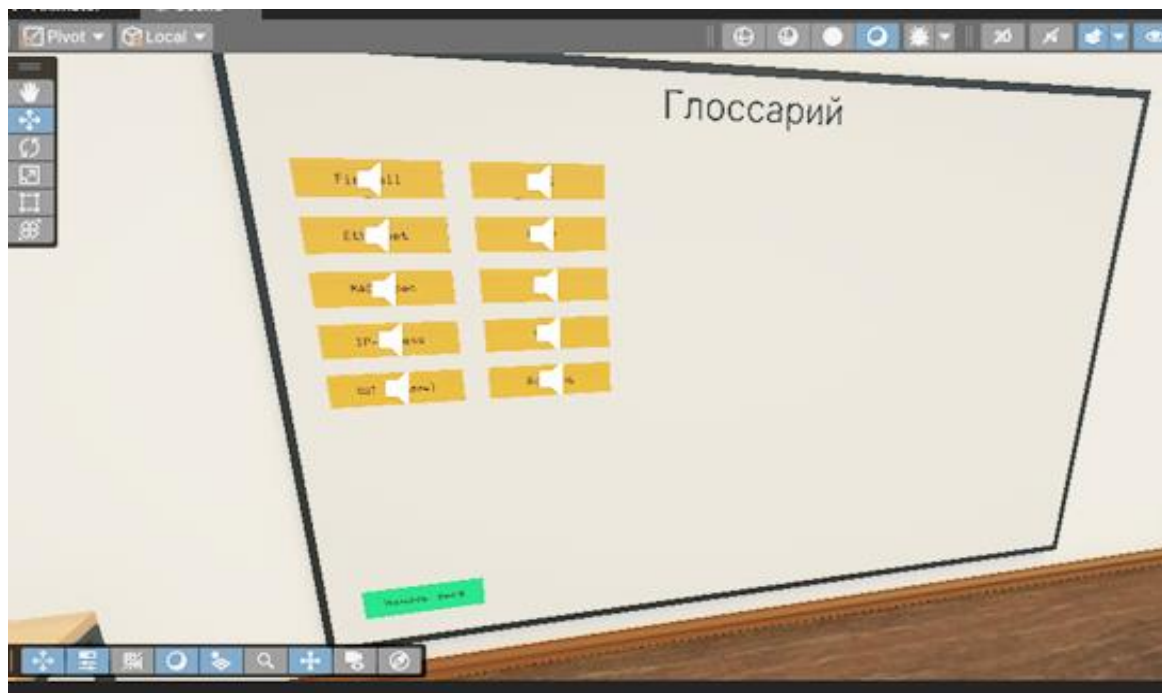


Рисунок 2.17 – Информационный стенд «Глоссарий»

На рисунке 2.18 продемонстрирована интерактивная доска с теоретическим материалом на тему «модель OSI» и «модель TCP/IP»;

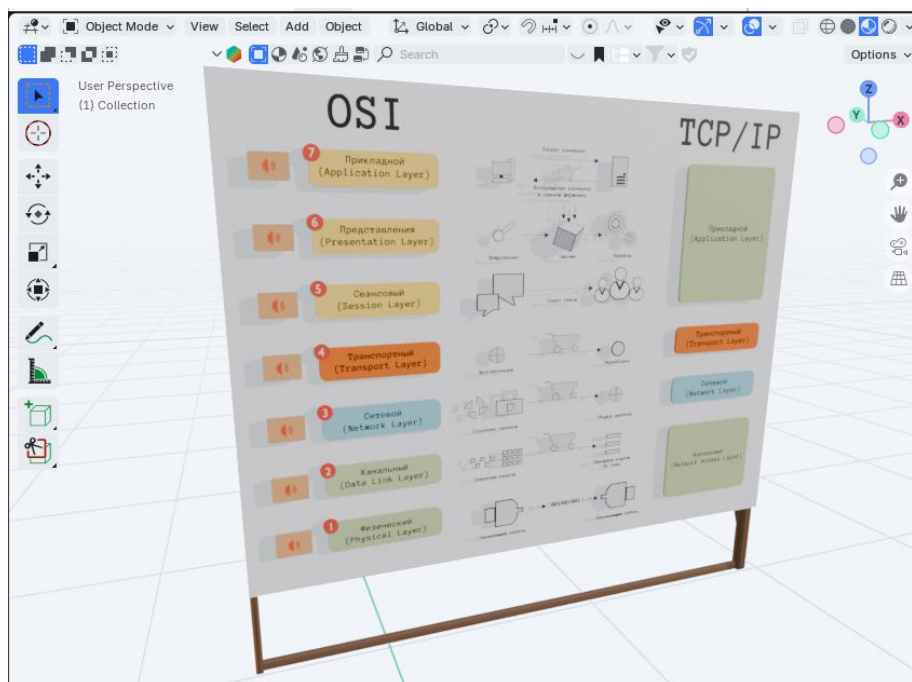


Рисунок 2.18 – Информационный стенд «модель OSI» и «модель TCP/IP»

На рисунке 2.19 продемонстрированная карта-ориентир для удобной навигации по лаборатории.



Рисунок 2.19 – Информационный стенд «Карта-ориентир»

На рисунке 2.20 представлен стенд с коммутатором, разработанным в среде Blender.

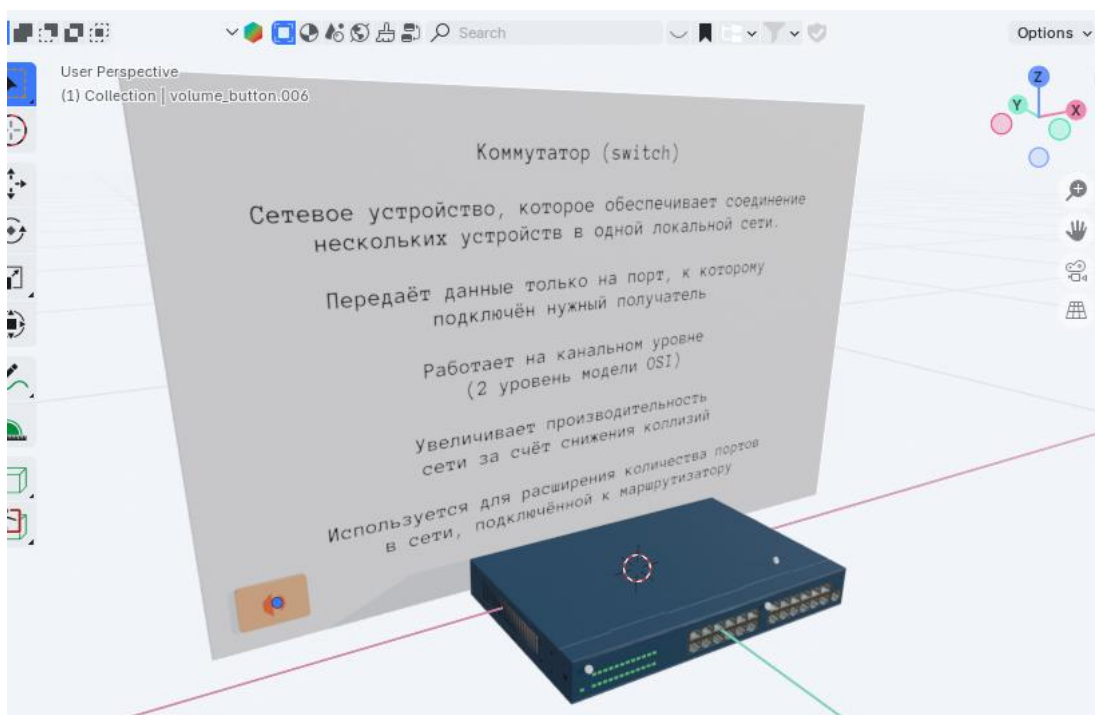


Рисунок 2.20 – Информационный стенд «Коммутатор»

2.5 Разработка программного модуля контроля знаний

Программный модуль контроля знаний представляет из себя алгоритм, набор правил, которые высчитывают оценку по количеству правильных ответов, предоставленных пользователем. Другими словами, это набор вопросов, с предложенными вариантами ответа, на которые, пользователю необходимо ответить в процессе тестирования. Далее программный модуль высчитывает оценку за его прохождение на основании количества правильных ответов, по отношению к общему количеству вопросов.

Программный модуль контроля знаний реализован с использованием временного массива строк данных, который хранит строго фиксированное количество вопросов и ответов на них. Каждый вопрос представляет собой объект, который хранит в себе следующий набор свойств:

- индекс вопроса;
- текст вопроса;

- массив строк с вариантами ответов;
- индекс правильного ответа.

Такой подход позволит быстро добавлять нужное количество вопросов без больших изменений в коде.

Запуск этого программного модуля осуществляется при нажатии на соответствующую кнопку «Начать тест».

Во время прохождения тестирования, управление становится временно недоступным. Такое решение обусловлено тем, что при движении персонажа во время тестирования могут возникать возможные ошибки, и чтобы этого не происходило, было принято решение временно отключить элементы управления. После того, как тестирование завершится, управление персонажем вновь станет доступным.

Выбор вариантов ответа осуществляется при помощи клавиш 1-4 на клавиатуре. Подтверждение выбора осуществляется через клавишу «ENTER».

Процесс тестирования представлен на рисунке 2.21.

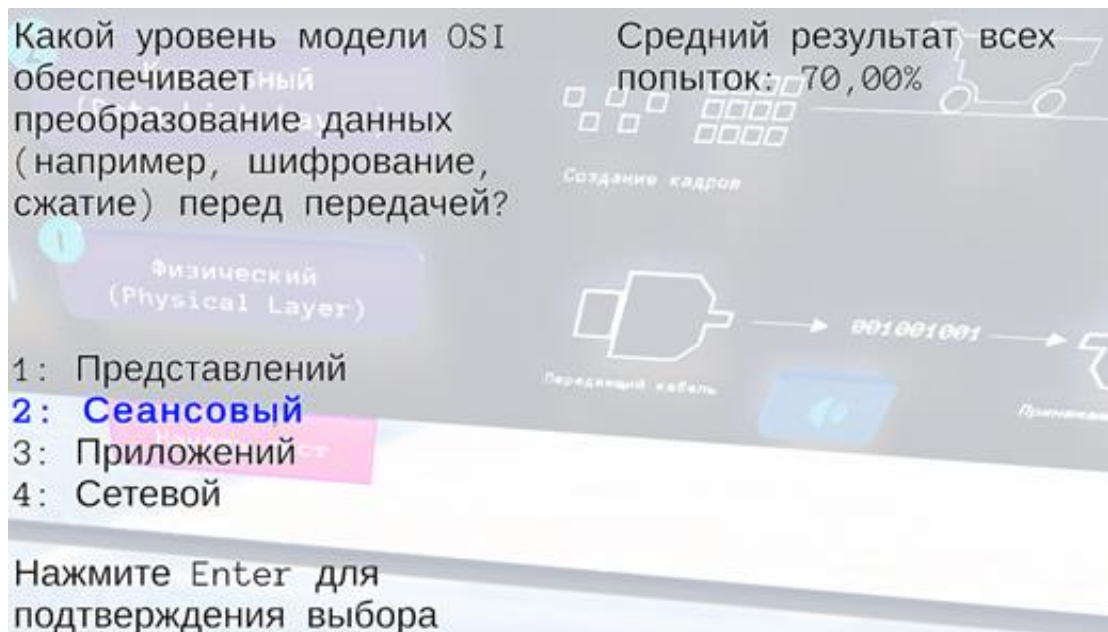


Рисунок 2.21 – Процесс тестирования

После того как пользователь отвечает на все вопросы теста, вычитаются результаты тестирования текущей попытки.

Результаты тестирования представлены на рисунке 2.22.

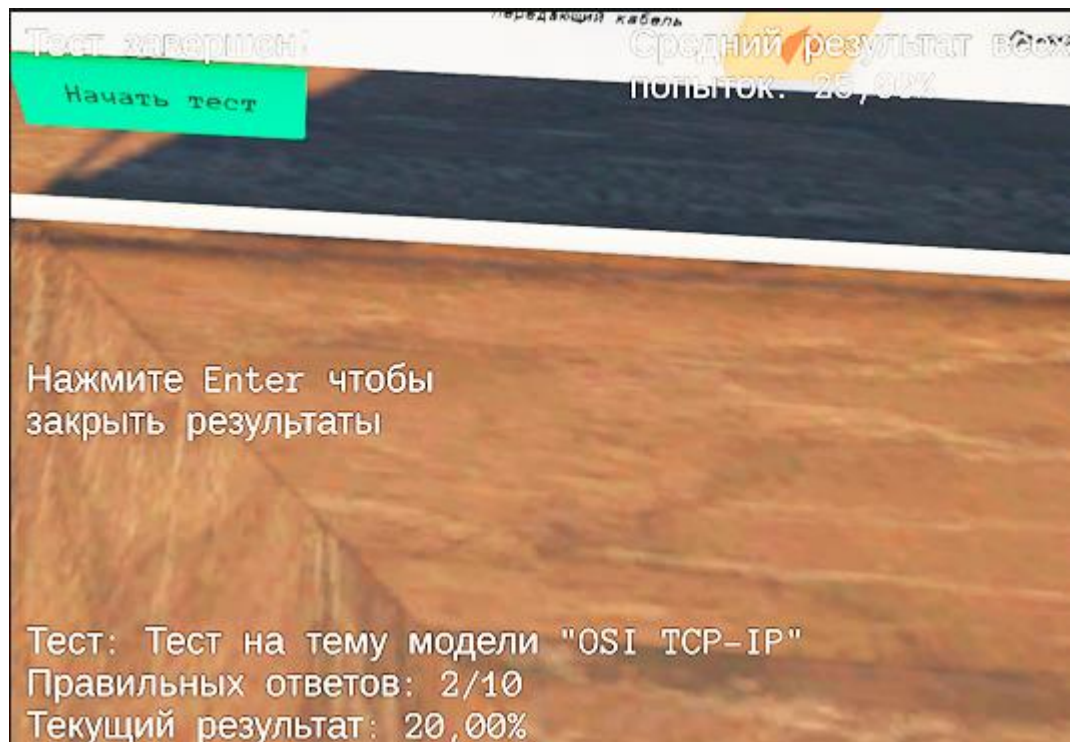


Рисунок 2.22 – Результаты тестирования

На рисунке 2.23 продемонстрирован алгоритм вывода результатов тестирования для завершенного теста.

Ссылка 1

```
void CompleteTestRun()
{
    testActive = false;
    float currentScore = (float)currentTestCorrect / currentTestQuestions.Count;
    currentTestScores.Add(currentScore);
    allTestScores.Add(currentScore);
    ShowAverageScore();
    if (currentTestQuestions == test1Questions)
    {
        UpdateBestScore(currentScore, bestTest2Score, bestTest3Score);
    }
    else if (currentTestQuestions == test2Questions)
    {
        UpdateBestScore(bestTest1Score, currentScore, bestTest3Score);
    }
    else if (currentTestQuestions == test3Questions)
    {
        UpdateBestScore(bestTest1Score, bestTest2Score, currentScore);
    }
    uiResultsText.text = $"Тест: {currentTestName}\n" +
        $"Правильных ответов: {currentTestCorrect}/{currentTestQuestions.Count}\n" +
        $"Текущий результат: {(currentScore * 100f):F2}%\n";
    uiQuestionText.text = "Тест завершен.";
    uiOptionsText.text = "Нажмите Enter чтобы закрыть результаты";
    testCompleted = true;
    testFinished = true;
    imageFon.SetActive(false);
    FSM.EnableFSMs();
}
```

Рисунок 2.23 – Алгоритм отображения результатов тестирования

Алгоритм отображения варианта вопроса во время тестирования продемонстрирован на рисунке 2.24.

```
Ссылка 3
void DisplayQuestion(Question q)
{
    uiQuestionText.text = q.questionText;
    string optionsDisplay = "";
    for (int i = 0; i < q.answerOptions.Length; i++)
    {
        if (selectedAnswer == i + 1)
            optionsDisplay += $"<b><color=yellow>{i + 1}: {q.answerOptions[i]}</color></b>\n";
        else
            optionsDisplay += $" {i + 1}: {q.answerOptions[i]}\n";
    }
    if (awaitingConfirmation)
    {
        optionsDisplay += "\nНажмите Enter для подтверждения выбора";
    }
    uiOptionsText.text = optionsDisplay;
}
```

Рисунок 2.24 – Алгоритм отображения варианта вопроса

Результаты прохождения всех тестов можно отследить не только после прохождения теста, но и в главной комнате. Результаты тестов сохраняются в глобальные переменные, и их значения отображаются в соответствующих полях вывода. Отображение результатов тестирования для каждого теста представлено на рисунке 2.25.

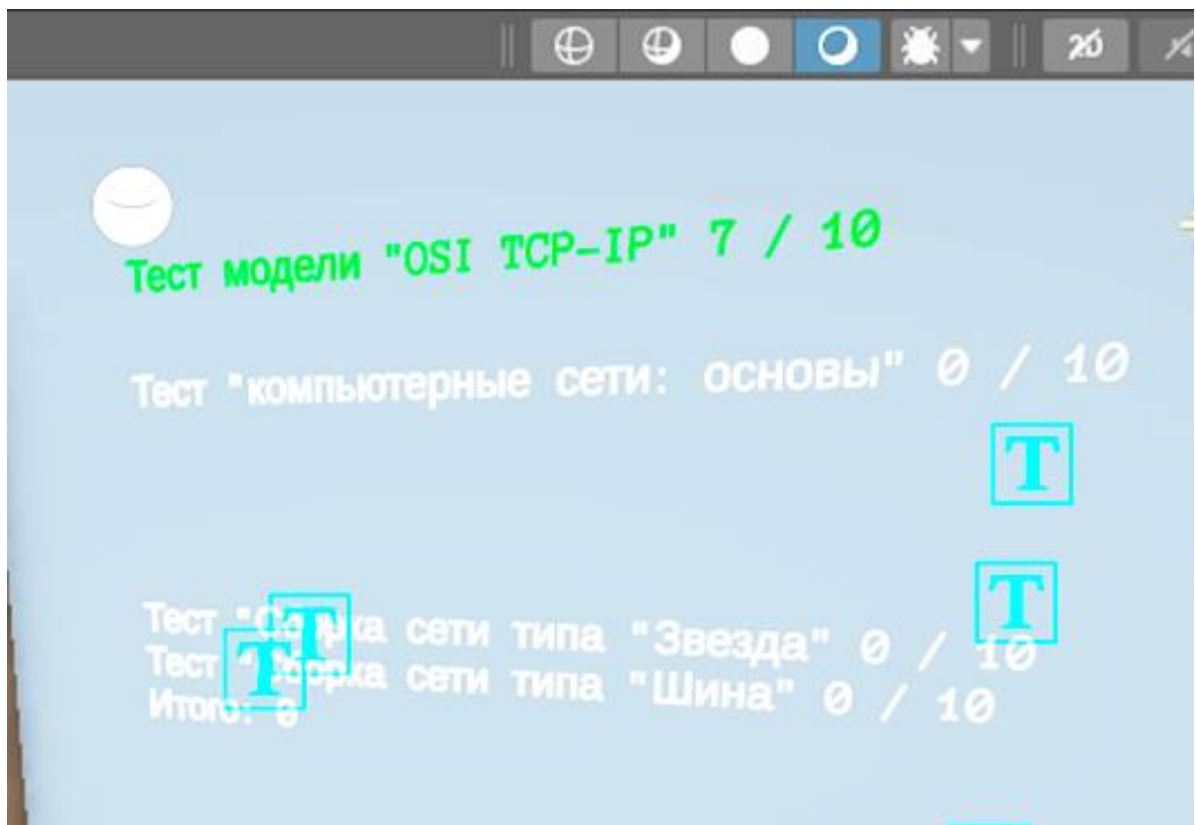


Рисунок 2.25 – Место отслеживания результатов тестирования для всех тестов

2.6 Разработка программного модуля проектирования сетей

Программный модуль проектирования сетей является одним из основных модулей виртуальной лаборатории по компьютерным сетям, поскольку данный модуль оценивает навыки пользователя в сборке компьютерных сетей на физическом уровне модели OSI (первый уровень модели), проверяет усвоенные пользователем знания после изучения основных компонентов, которые напрямую участвуют при сборке различных сетей. Программный модуль проектирования сетей учит пользователя самостоятельно строить различные сети на основе полученных знаний после изучения лаборатории.

Важно отметить, что доскональное изучение лаборатории не является первостепенной задачей. Пользователь может начать проектировать сети в случайном порядке.

Далее будет детально описан функционал программного модуля.

Данный программный модуль функционирует исключительно в рамках полигона проектирования сетей. Это значит, что пользователь может осуществлять сборку только в рамках полигона (комнаты в которой происходит сборка сетей).

Когда пользователь заходит кабинет проектирования сетей, перед ним предстает четыре стенда с оборудованием:

1. Оборудование для проектирования сетей по топологии «Общая шина». На столе находятся три ноутбука, различные виды коаксиальных кабелей, т-образные коннекторы, терминатор, конвертеры, и LAN-кабели различной длины.

2. Оборудование для проектирования сетей по топологии «Звезда». На столе находятся два ноутбука, роутер, коммутатор, блок питания для роутера, коммутатора, три вида LAN-кабеля с коннектором типа «RJ-45».

3. Стенд с моделью-индикатором, для сети с топологией «Общая шина» и точно такая же модель, только для топологии «Звезда». Данные стенды с сетями полностью окрашены в белый цвет. Это означает, что с ними нельзя взаимодействовать, и что она является всего лишь ориентиром при сборке сети.

Стоит отметить, что оборудование, с помощью которого осуществляется проектирование сетей тоже является недоступным до запуска режима проектирования.

Кроме стендов с оборудованием, на стенах закреплены информационные блоки для каждого стола с оборудованием, которые объясняют суть и назначение этой сцены.

Стенды с оборудованием, предназначенным для проектирования сетей представлены на рисунке 2.26.

Запуск режима проектирования сети осуществляется через соответствующую кнопку запуска, которая находится в нижнем углу информационного блока. После того как пользователь нажмёт на соответствующую кнопку, запустится режим проектирования, и пользователь сможет взаимодействовать с предметами, которые необходимо использовать при построении сети.

Если во время тестирования пользователь покинет комнату, то режим тестирования завершится досрочно, и пользователю придется заново запустить режим проектирования.



Рисунок 2.26 – Стенды с оборудованием для проектирования сетей

Если пользователь передумал проектировать сеть, случайно нажал на кнопку запуска режима проектирования, необязательно покидать комнату. Режим проектирования можно досрочно завершить при нажатии на клавишу «ESC».

Построение сети осуществляется в три этапа. Каждый этап представляет собой сборку сети с последующим повышением сложности сборки:

1. При первом запуске сборке сети, пользователю необходимо следовать подсказкам программы. Исполняемый алгоритм подсвечивает компоненты сети, которые нужно установить на указанные места. Пользователь захватывает необходимый предмет (например, роутер или коммутатор), перемещается вместе с ним и устанавливает его в необходимое место. После того как пользователь таким образом установит всё оборудование, программа завершится и пользователю будет предложено перезапустить режим проектирования, чтобы активировать второй этап сборки. Запуск первого этапа представлен на рисунке 2.27.

2. Второй этап сборки сети подразумевает сборку сети без подсказок от исполняющего алгоритма. Исполняющий алгоритм подсвечивает, какой элемент (оборудование сети) нужно установить на указанное место, но не подсказывает пользователю, что нужно взять в руки. Пользователь должен сам выбрать указанное оборудование и затем осуществить установку. После сборки сети, пользователю будет предложено перезапустить режим сборки сети, чтобы стал доступен третий этап сборки сети. Запуск второго этапа представлен на рисунке 2.28.

3. Третий этап сборки сетей отличается повышенной сложностью, по отношению к первому и второму этапу. На третьем этапе пользователь не получает сопроводительные подсказки от программы, и ему необходимо самому собрать сеть в той последовательности, которая была заложена в предыдущих этапах. На этом этапе пользователю стоит быть максимально осторожным, чтобы не перепутать оборудование для сборки и не потерять баллы за неправильный выбор оборудования запуск третьего этапа

продемонстрирован на рисунке 2.29.

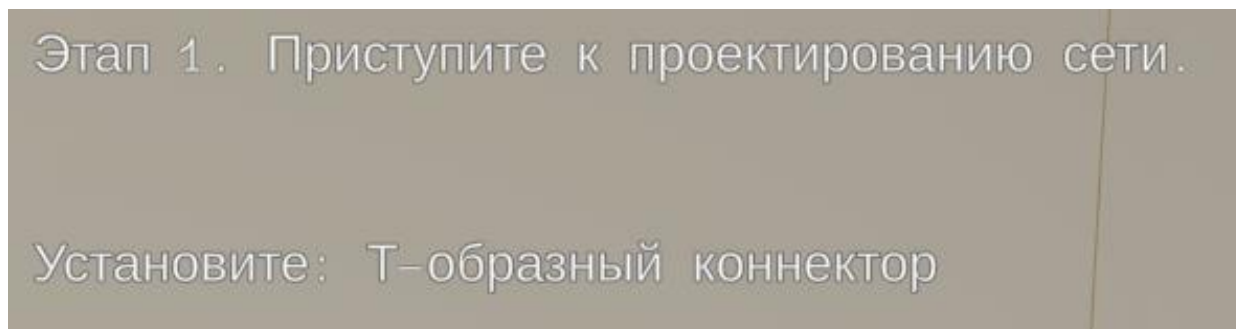


Рисунок 2.27 – Первый этап сборки сети

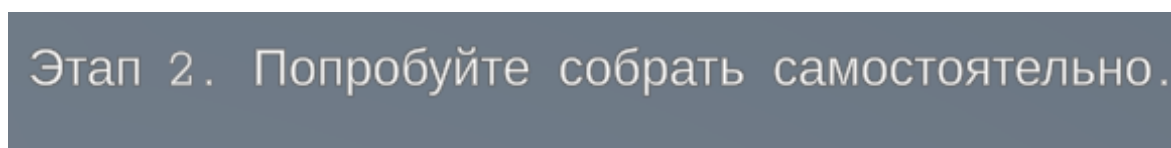


Рисунок 2.28 – Второй этап сборки сети

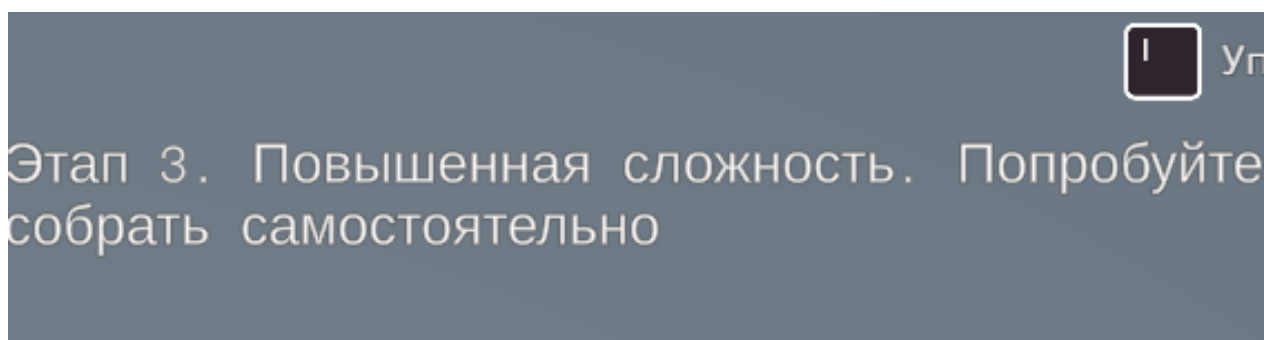


Рисунок 2.29 – Третий этап сборки сети с повышенным уровнем сложности

После того как пользователь прошел третий этап сборки сети, алгоритм вновь возвращается к первому этапу сборки сети. Таким образом сеть можно собирать неограниченное количество раз и тем самым тренировать навыки в сборке сети.

При попытке установить оборудование в неправильное место, программа начнет вычитать по одному баллу за каждую неправильную попытку. Аналогичным образом, при попытке взять неправильный предмет, программа также будет вычитать один балл (рисунок 2.30). После перезапуска сети, общее количество баллов обновляется до исходного количества, то есть, пользователь при каждой итерации сборки сети на счету имеет 10 баллов.

При запуске программы у пользователя на счету находится общее количество баллов – 10. При необходимости можно искусственно повысить сложность, путем уменьшения общего количества баллов у пользователя.

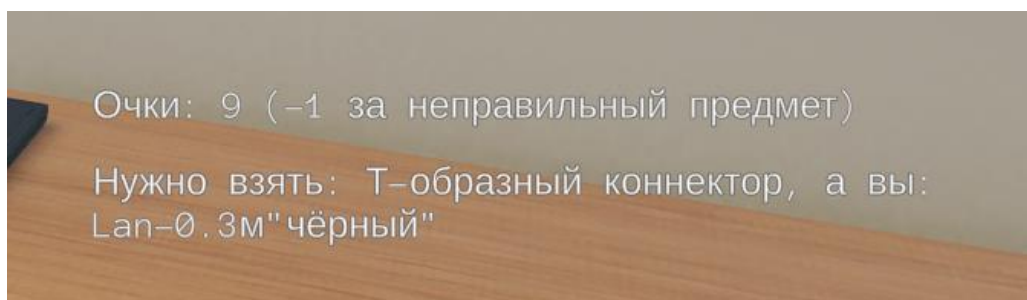


Рисунок 2.30 – Демонстрация системы вычитания баллов

Если пользователь израсходовал все балл, которые у него были, программа завершается с сообщением о том, всё баллы израсходованы, и сборка сети прекращена (рисунок 2.31). В таком случае, пользователю нужно снова запустить режим проектирования и повторить этап сборки.

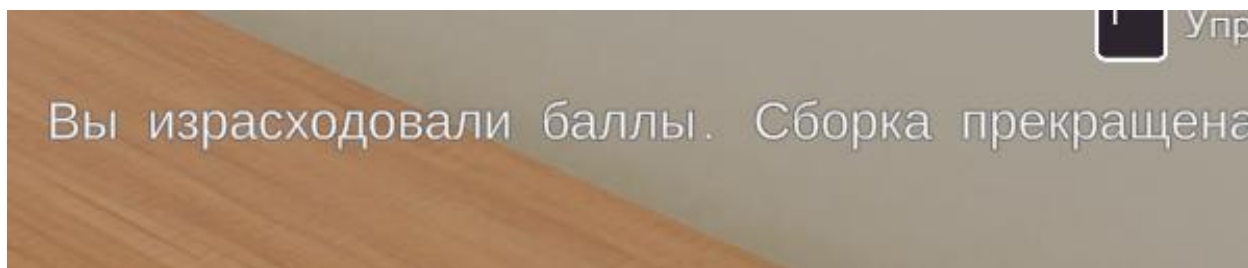


Рисунок 2.31 – Сообщение в израсходовании баллов

После того как пользователь завершил сборку сети, в интерфейсе отобразится итоговый результат сборки. Программа посчитает итоговую оценку на основании оставшихся баллов у пользователя. Если пользователь потратил все баллы, все ранее установленные предметы возвращаются на исходные места, программа завершается и на экране появляется сообщение о том, что тестирование завершено. Оценка вычисляется по методу наивысшей оценки (рисунок 2.32).

Стоит также отметить, что осуществлять сборку двух сетей одновременно не представляется возможным, поскольку при одновременной сборке двух сетей сразу могут возникнуть ошибки при проектировании.

Поэтому было принято решение ограничить процесс взаимодействия с оборудованием для одной сети, при активном построении другой сети. Например, пользователь решил запустить режим проектирования сети по топологии «Звезда». При активной сборке сети по топологии «Звезда», сборка сети по топологии «Общая шина» блокируется. Аналогичным образом, при сборке сети по топологии «Общая шина», сборка сети по топологии «Звезда» становится недоступной.

Таким образом, проектирование сетей по различным топологиям вырабатывает навыки построения сетей, у пользователя тренируется внимательность и общие навыки установки компонентов сети, сборка становится увлекательной и интересной.

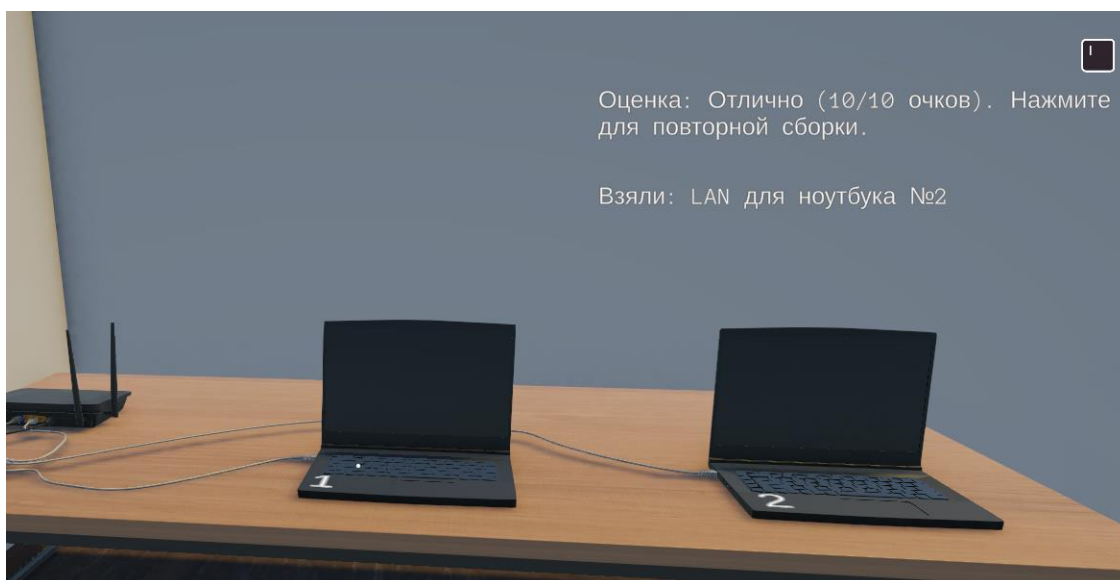


Рисунок 2.32 – Результат построения сети по топологии «Звезда»

2.6.1 Программная реализация модуля проектирования сетей

Программный модуль и исполняющие алгоритмы программы реализованы таким образом, что их можно быстро адаптировать по построение или сборку других типов сети и не только сети.

Общий принцип программного модуля устроен таким образом, что порядок установки работает с использованием массива и индекса шагов (этапов сборки.)

Порядок сборки сети зависит от порядка установки элементов в массиве.

Когда пользователь пытается захватить предмет через метод «tryPickUp», метод сравнивает захваченный предмет с предметом, который указан в качестве его копии (рисунок 2.33). Если предметы совпадают, то пользователь успешно захватывает предмет через метод «PickUpItem» и далее может установить его на указанное место (рисунок 2.34).

```
void TryPickUp()
{
    if (!Input.GetKeyDown(pickUpKey)) return;

    var ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    if (!Physics.Raycast(ray, out RaycastHit hit)) return;

    var itm = hit.collider.GetComponent<NetW1>();
    if (itm == null) return;

    string currentExpectedItem = items[currentStep].name;
    bool isGroupItem = false;
    string[] currentGroup = null;

    foreach (var group in itemGroups)
    {
        if (group.Value.Contains(currentExpectedItem))
        {
            isGroupItem = true;
            currentGroup = group.Value;
            break;
        }
    }

    if (isGroupItem && currentGroup.Contains(itm.name))
    {
        PickUpItem(itm);
    }
    else if (itm.name == currentExpectedItem)
    {
        PickUpItem(itm);
    }
    else
    {
        string russianNameCurrentStep = translations.ContainsKey(currentExpectedItem)
            ? translations[currentExpectedItem]
            : currentExpectedItem;

        string russianNameItm = translations.ContainsKey(itm.name)
            ? translations[itm.name]
            : itm.name;

        if (restartNew == 0 || restartNew == 1)
        {
            text.text = $"Нужно взять: {russianNameCurrentStep}, а вы: {russianNameItm}";
        }
        else if (restartNew == 2)
        {
            text.text = "Неправильный предмет установки.";
        }

        AdjustScore(-1, "Неправильный предмет");
    }
}
```

Рисунок 2.33 – Метод реализующий сравнивающий предметы

```

Ссылка: 2
void PickupItem(NetW1 itm)
{
    heldItem = itm;
    var rb = itm.GetComponent<Rigidbody>();
    rb.isKinematic = true;

    itm.transform.parent = handAnchor;
    itm.transform.localPosition = Vector3.zero;
    itm.transform.localRotation = Quaternion.identity;

    text.text = "Вы захватили: " + (translations.ContainsKey(itm.name)
    ? translations[itm.name]
    : itm.name);
}

```

Рисунок 2.34 – Метод захвата предмета

Метод установки предмета, который был захвачен игроком представлен на рисунке 2.35.

```

Ссылка: 1
void TryPlace()
{
    var snap = placeholders[currentStep].snapPoint;
    float distToSnap = Vector3.Distance(handAnchor.position, snap.position);

    if (distToSnap <= magnetRadius)
        placeholders[currentStep].DisableVisual();
    else
        placeholders[currentStep].EnableVisual();

    if (distToSnap <= magnetRadius)
    {
        if (heldItem.transform.parent == handAnchor)
            heldItem.transform.parent = null;

        heldItem.transform.position = Vector3.Lerp(
            heldItem.transform.position,
            snap.position,
            Time.deltaTime * magnetSpeed);
        heldItem.transform.rotation = Quaternion.Slerp(
            heldItem.transform.rotation,
            snap.rotation,
            Time.deltaTime * magnetSpeed);
    }
    else
    {
        if (heldItem.transform.parent != handAnchor)
        {
            heldItem.transform.parent = handAnchor;
            heldItem.transform.localPosition = Vector3.zero;
            heldItem.transform.localRotation = Quaternion.identity;
        }
    }

    if (Input.GetKeyDown(placeKey))

```

Рисунок 2.35 – Метод установки предмета в указанное место

Метод установки предмета в указанное место отвечает за вычисление расстояния от захваченного предмета до точки установки предмета. Если предмет находится очень близко к месту установки, то он притягивается к точке установки, занимая позицию предмета представляющего копию этого

же предмета. Если пользователь отходит вместе с предметом на большее расстояние от места установки, то предмет возвращается обратно к пользователю и вновь двигается вместе с ним.

Так же этот метод отвечает за выключение предмета, который указывает как он должен быть установке.

Если все условия соблюдаются, предмет устанавливается на место фиксации при помощи метода «PlaceComplete» (рисунок 2.36). Также этот метод реализует завершение сборки сети.

```
Ссылка: 1
void PlaceComplete()
{
    placeholders[currentStep].DisableVisual();
    heldItem = null;
    currentStep++;

    if (currentStep >= placeholders.Length)
    {
        finish.text = "Сеть собрана, режим проектирования завершен.";
        EvaluateResults();
        testStarted = false;
        assemblyComplete = true;
        ChildOBJ = true;
    }
    else
    {
        string nextItemName = items[currentStep].name;

        bool isNextGroupItem = false;
        foreach (var group in itemGroups)
        {
            if (group.Value.Contains(nextItemName))
            {
                isNextGroupItem = true;
                break;
            }
        }

        if (restartNew == 0)
        {
            text.text = "Теперь установите: " + (translations.ContainsKey(nextItemName)
                ? translations[nextItemName]
                : nextItemName);
        }
        else if (restartNew == 1 || restartNew == 2)
        {
            text.text = " ";
            number.text = " ";
            finish.text = " ";
        }

        RefreshHighlights(true);
    }
}
```

Рисунок 2.36 – Метод установки предмета на место фиксации

Метод вычитания баллов и прекращения сборки представлен на рисунке 2.37.

```

Ссылка: 1
void AdjustScore(int delta, string reason)
{
    score += delta;

    number.text = $"Очки: {score} ({(delta < 0 ? "-" : "+" ){Mathf.Abs(delta)} за {reason})";

    if (score <= 0)
    {
        finish.text = "Вы израсходовали баллы. Сборка прекращена";
        Invoke(nameof(timeReset), 3f);
        Invoke(nameof(ClearText), 3f);
        ResetAllItems();
        number.text = "";
        text.text = "";
    }
}

```

Рисунок 2.37 – Метод вычитания баллов прекращения сборки сети

Метод, реализующий принудительное прекращение сборки сети продемонстрирован на рисунке 2.38.

```

Ссылка: 3
public void StopTest()
{
    startFlags = false;
    testStarted = false;

    if (heldItem != null)
    {
        var rb = heldItem.GetComponent<Rigidbody>();
        rb.isKinematic = false;
        heldItem.transform.parent = null;
        heldItem = null;
    }

    ResetAllItems();
    RefreshHighlights(false);
    finish.text = " ";
    finish.text = "Сборка остановлена, и все предметы возвращены.";
    Invoke(nameof(ClearText), 3f);
}

```

Рисунок 2.38 – Метод принудительного прекращения сборки сети

Метод реализующий запуск режима проектирования сети представлен на рисунке 2.39.

```

public void StartTest()
{
    startFlags = true;
    if (testStarted) return;
    testStarted = true;
    currentStep = 0;
    score = maxScore;

    finish.text = "Этап 1. Приступите к проектированию сети.";
    Invoke(nameof(ClearText1), 3f);
    RefreshHighlights(true);

    if (restartNew == 0)
    {
        text.text = "Установите: " + (translations.ContainsKey(items[currentStep].name)
        ? translations[items[currentStep].name]
        : items[currentStep].name);
    }
    else if (restartNew == 1)
    {
        text.text = " ";
        finish.text = "Этап 2. Попробуйте собрать самостоятельно.";
        number.text = " ";
    }
    else if (restartNew == 2)
    {
        text.text = " ";
        finish.text = "Этап 3. Повышенная сложность. " +
        "Попробуйте собрать самостоятельно";
        number.text = " ";
    }
}

```

Рисунок 2.39 – Метод запуска режима проектирования

На рисунке 2.40 представлены поля для установки элементов оборудования в определенном порядке.

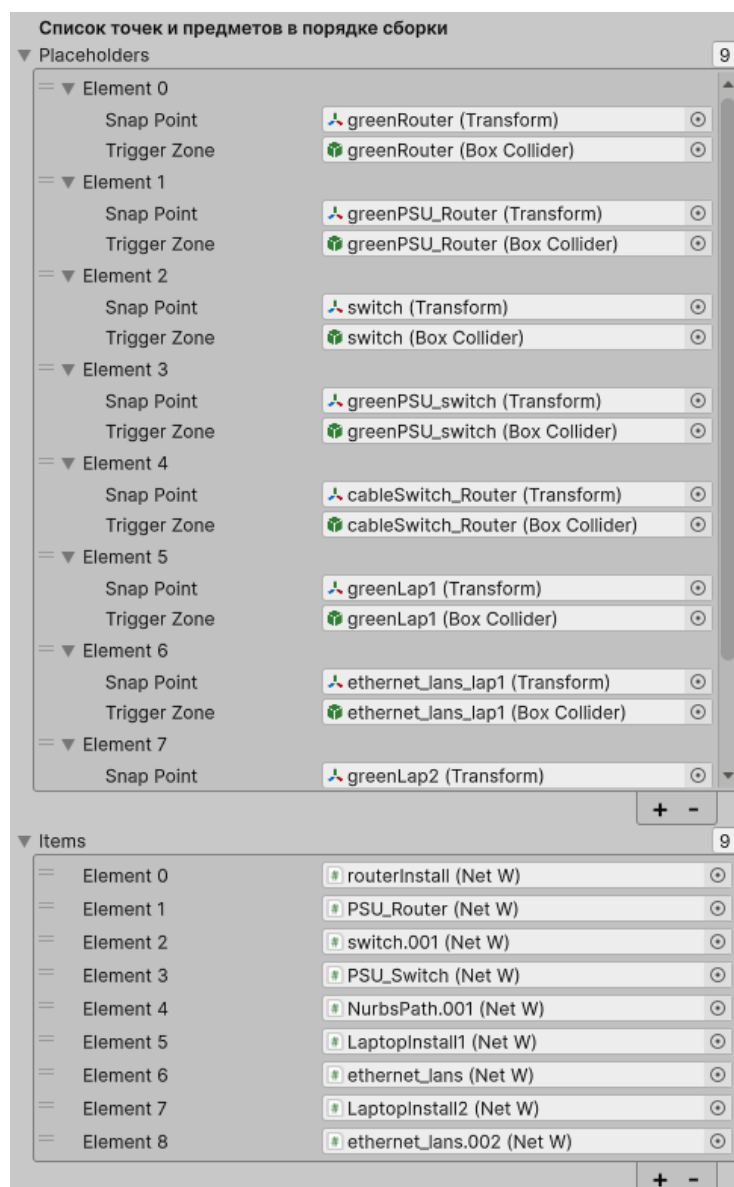


Рисунок 2.40 – Массивы элементов сборки сетей

2.7 Разработка программного модуля отслеживания прогресса

Программный модуль механизма отслеживания прогресса представляет из себя набор определенных действий, которые должен совершить пользователь, находясь в конкретной сцене.

Примером так такого прогресса может выступать: сборка сети без

ошибок (сохранить максимальное количество баллов), взять в руки предмет, завершить сборку сети, выбросить предмет из рук, изучить информационный стенд и так далее.

Этот модуль необходим для того, чтобы пользователь понимал, когда исследование лаборатории близится к завершению. Помимо вышеописанного, данный модуль поможет пользователю отслеживать прогресс действий, которые выполнил или должен пользователь, находясь на определенной сцене.

Программный модуль механизма отслеживания отображается исключительно в интерфейсной части программы.

Программный модуль механизма отслеживания прогресса построен из следующих двухмерных компонентов (объектов):

- тёмный фон (тёмное изображение с небольшим уровнем прозрачности);
- текст, описывающий тот или иной прогресс;
- изображение чекбокса (флажковое поле);
- галочка (чек-поинт, иконка, символизирующая подтверждение действия);
- программный модуль, выполняющий отображение галочки, чекбокса и изображений.

Программный модуль работает при помощи флагов в программном коде, которые срабатывают при выполнении определенных действий. Например, пользователь взял в руки предмет. В этот момент срабатывает флаг, который приводит в действие алгоритм, и тот отображает, галочки в чекбоксах.

Метод «OnTriggerExit», отслеживает момент, когда пользователь покидает комнату.

Если пользователь покинул комнату, метод «OnTriggerExit» вызывает другой метод, отвечающий за запуск таймера – «RoomExitGo». Он начинает отчитывать время до скрытия текста и галочек, находящихся в поле прогресса (рисунок 2.41). Это необходимо для того, чтобы в каждой комнате появлялись и скрывались пункты с соответствующим прогрессом.

```

Сообщение Unity | Ссылка: 0
private void OnTriggerExit(Collider other)
{
    if (other.CompareTag("Player") && other.gameObject == GT_RoomSeti_Exit)
    {
        RoomExitGo(ref isPlayerInside_One_ROOM, GT_RoomSeti_Exit, GT_RoomSeti, text_setiText);
        // Запускаем таймер скрытия текста и галочки
        timerText = timeTextTime;
    }
    else if (other.CompareTag("Player") && other.gameObject == GT_RoomSetiPractika_Exit)
    {
        stopTestSetiStar = FindObjectOfType<NetworkAssemblyManager>();
        stopTestSetiBus = FindObjectOfType<NetworkAssemblyManager1>();

        RoomExitGo(ref isPlayerInside_Two_ROOM, GT_RoomSetiPractika_Exit, GT_RoomSetiPractika, text_setiTextPractika);
        // Запускаем таймер скрытия текста и галочки
        timerText = timeTextTime;
        stopTestSetiStar.StopTest();
        stopTestSetiBus.StopTest();
    }
}

```

Рисунок 2.41 – Программный модуль скрытия окна прогресса

Появление объектов в интерфейсной части программы обеспечивает метод «OnTriggerEnter».

На входе в комнату стоит коллайдер который постоянно отслеживает местоположение персонажа, и если персонаж и коллайдер начнут пересекаться, то сработает метод «OnTriggerEnter». Важно учесть, чтобы сработал триггер, персонаж должен иметь компонент «Rigidbody». Этот компонент придает физические свойства объекту (вес, гравитацию, объем и т.д.). Метод, отвечающий за появление объектов в интерфейсе представлен на рисунке 2.42.

```

Сообщение Unity | Ссылка: 0
private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Player") && other.gameObject == GT_RoomSeti)
    {
        RoomEnterGo(ref isPlayerInside_One_ROOM, GT_RoomSeti, GT_RoomSeti_Exit, text_setiText);
    }
    else if (other.CompareTag("Player") && other.gameObject == GT_RoomSetiPractika)
    {
        RoomEnterGo(ref isPlayerInside_Two_ROOM, GT_RoomSetiPractika, GT_RoomSetiPractika_Exit, text_setiTextPractika);
    }

    triggerColliderRoms(other, TriggersSetiPractik[0], ref flagOne);
    triggerColliderRoms(other, TriggersSetiPractik[1], ref flagTwo);

    triggerColliderRoms(other, TriggersSetiPractik[2], ref PC_triggerRom1);
    triggerColliderRoms(other, TriggersSetiPractik[3], ref PC_triggerRom2);
    triggerColliderRoms(other, TriggersSetiPractik[4], ref PC_triggerRom3);
    triggerColliderRoms(other, TriggersSetiPractik[5], ref PC_triggerRom4);
}

```

Рисунок 2.42 - Метод отображения объектов интерфейса

После того как пользователь нажмет на заданные в программе кнопки, совершит какие-либо интерактивные действия с информационными стендами, сработает метод «OneInputOneCondition». Данный модуль отображает галочки в нужный момент.

Программный модуль появления галочек представлен на рисунке 2.43.

```
Ссылка: 10
private (GameObject, bool, int) OneInputOneCondition(GameObject galaChik,
    GameObject objInput, ref bool why, int indexImageChil = 0)
{
    if (Input.GetMouseButtonDown(0) && hit.collider.gameObject == objInput)
    {
        why = true; if (why == true) galaChik.transform.GetChild(indexImageChil).gameObject.SetActive(true);
    }
    return (objInput, why, indexImageChil);
}
```

Рисунок 2.43 – Программный модуль появления галочек

Для действий осмотра и кликов по объектам (кнопкам) срабатывает «HitRayCam» (рисунок 2.44).

```
void HitRayCam()
{
    trigger_rj45ports = GameObject.Find("trigger_rj45ports");
    S_trigger_case = GameObject.Find("S_trigger_case");
    swap_button_topologi_threeRoom = GameObject.Find("swap_button_topologi_threeRoom");
    button_CrimperAnim = GameObject.Find("button_CrimperAnim");
    swap_button_CableVids = GameObject.Find("swap_button_CableVids");
    Test_TSPIP = GameObject.Find("Test_TSPIP");
    Test_seti = GameObject.Find("Test_seti");
    trigger_fiber1 = GameObject.Find("trigger_fiber1");
    w_trigger_led = GameObject.Find("W_trigger_led");

    if (Physics.Raycast(Camera.main.transform.position, Camera.main.transform.forward, out hit, distanceRa)
    {
        if (hit.collider.gameObject)
        {
            // сети
            if (isPlayerInside_One_ROOM)
            {
                OneInputOneCondition(galchiGalchi_SetiRoom, button_CrimperAnim, ref chilTwo, 3);
                OneInputOneCondition(galchiGalchi_SetiRoom, swap_button_CableVids, ref cableBol, 4);
                OneInputOneCondition(galchiGalchi_SetiRoom, swap_button_topologi_threeRoom, ref topologBol, 2);
                TwoInputTwoCondition(galchiGalchi_SetiRoom, S_trigger_case, w_trigger_led, ref chilOne, ref chilZero, 1);
                TwoInputTwoCondition(galchiGalchi_SetiRoom, trigger_rj45ports, trigger_fiber1, ref chilZero, ref chilOne, 0);
                TwoInputTwoCondition(galchiGalchi_SetiRoom, Test_TSPIP, Test_seti, ref testOne, ref testTwo, 5);
            }
        }
    }
}
```

Рисунок 2.44 – Метод «HitRayCam»

Метод отображения чек-боксов представлен на рисунке 2.45.

```
private void RoomEnterGo(ref bool roomFlag, GameObject entrance, GameObject exit, string textTask)
{
    roomFlag = true;
    if (entrance != null && entrance.activeSelf)
    {
        entrance.SetActive(false);
        if (exit != null) exit.SetActive(true);
    }
    if (textHelps != null)
    {
        textHelps.text = textTask;
        textHelps.enabled = true;
    }
    if (squareRoom != null)
    {
        squareRoom.SetActive(true);
    }
    if (isPlayerInside_One_ROOM == true)
    {
        AllGalchi[0].SetActive(true);
        Seti(galchiGalchi_SetiRoom, chilZero, 0);
        Seti(galchiGalchi_SetiRoom, chilOne, 1);
        Seti(galchiGalchi_SetiRoom, chilTwo, 3);
        Seti(galchiGalchi_SetiRoom, cableBol, 4);
        Seti(galchiGalchi_SetiRoom, topologBol, 2);
        Seti(galchiGalchi_SetiRoom, testOne, 5);
    }
    else if (isPlayerInside_Two_ROOM == true)
    {
        AllGalchi[1].SetActive(true);
    }
}
```

Рисунок 2.45 – Метод отображения чек-боксов «RoomEnterGo»

На рисунке 2.46 представлено окно прогресса в интерфейсе пользователя при первом входе в комнату.

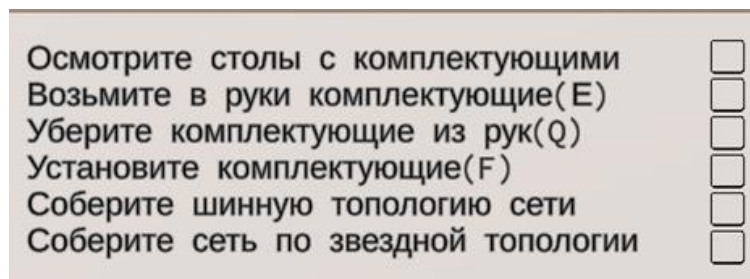


Рисунок 2.46 – Окно прогресса при первом входе в комнату

На рисунке 2.47 представлено окно прогресса после выполнения всех действий и заданий в текущей комнате.

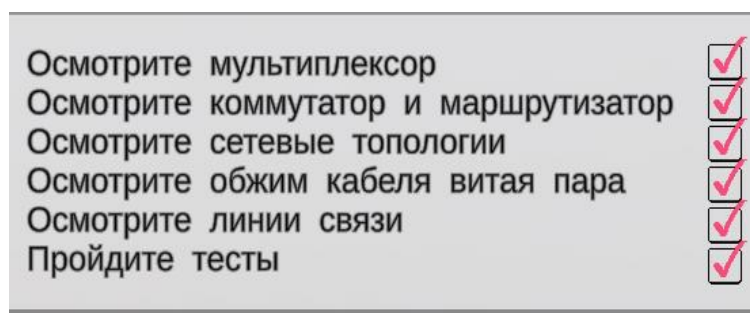


Рисунок 2.47 – Окно с выполненным прогрессом

Стоит отметить, что после того, как пользователь выполнит все действия в комнате, окно прогресса впредь не будет появляться, поскольку все действия являются выполненными.

2.8 Разработка системы управления и пользовательского интерфейса

Элементы управления персонажем представляют собой программные модули, которые отвечают для выполнение конкретных действий при нажатии на определенный набор клавиш, сочетания клавиш. Помимо считывания откликов с клавиатуры, программный модуль отвечает за захват и блокировку курсора мыши в пределах выполняемого окна программы.

Для реализации управления персонажем (ходьба, без, приседание, управление камеры с помощью мыши) было задействованы средства визуального программирования – Playmaker [23]. Выбор визуального программирования при создании управления персонажем обусловлен тем, что этот метод проявил себя наилучшим образом (управление персонажем происходит корректно, без подергиваний, плавно и непринужденно). Еще одним фактором выбора в пользу визуального программирования стало то, что логика реализации не требует углубленных знаний при её создании. Используемый ассет, в данном случае, обладает готовыми решениями, которые позволяют быстро реализовать управление персонажем без особых временных вложений.

Алгоритм управления персонажем представлен на рисунке 2.48.

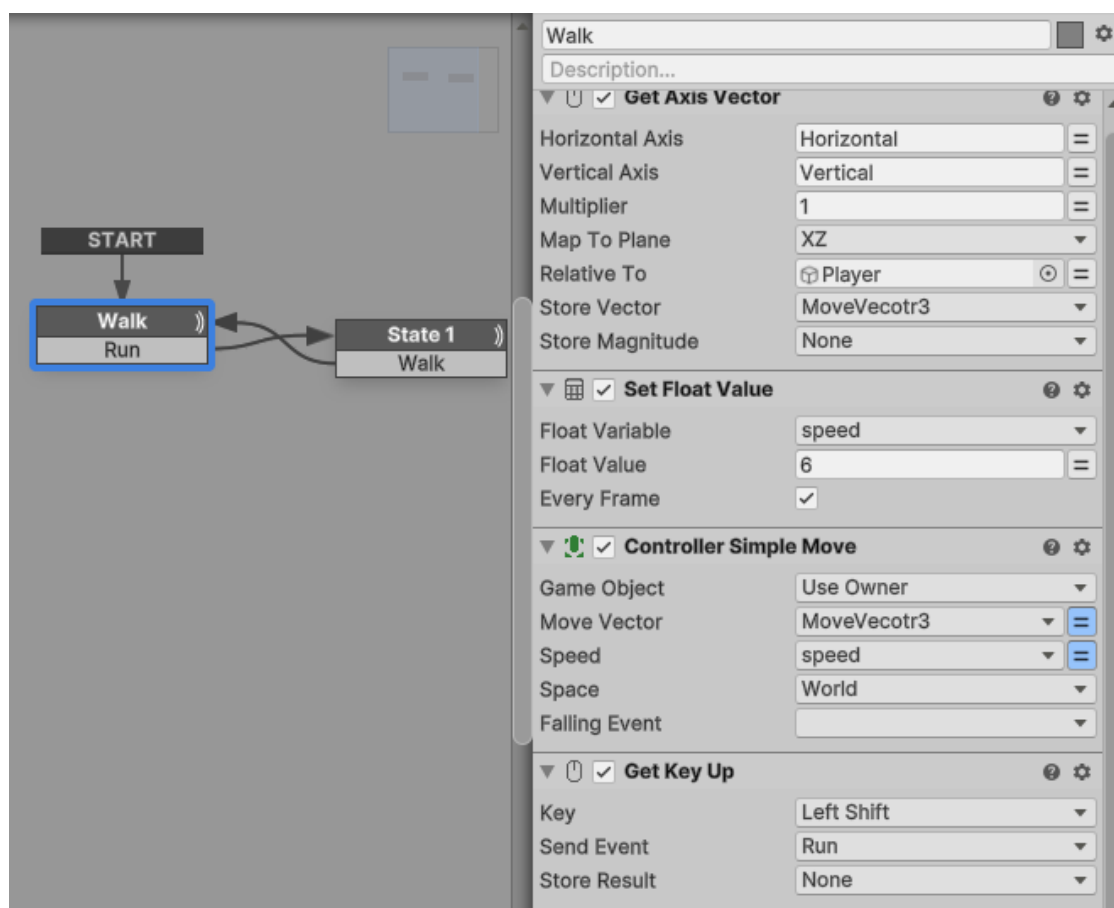


Рисунок 2.48 – Алгоритм управления персонажем

Алгоритм управления персонажем делится на два состояния, которые срабатывают при нажатии клавиш (состояние ходьбы и состояние бега).

Состояние ходьбы представляет из себя плавное и равномерное перемещение камеры со средней скоростью, приближенной к скорости ходьбы среднестатистического человека.

Состояние бега – это та же ходьба, только с увеличенной скоростью перемещения камеры. Добавление скорости бега обусловлено тем, что для некоторых пользователей обычная ходьба может быть слишком медленной в рамках игры, поэтому была реализована механика бега, чтобы пользователь мог быстро и беспрепятственно перемещаться по лаборатории.

Работа алгоритма осуществляется через срабатывание двух этих состояний при нажатии на клавиши. Когда пользователь нажимает клавиши «WASD», срабатывает состояние ходьбы, которое приводит в движение камеру. Стоит пользователю единожды нажать клавишу «SHIFT», и персонаж начнет двигаться с большей скоростью. При повторном нажатии на эту же клавишу, сработает переключение на состояние ходьбы.

Алгоритм управления камерой с помощью мыши представлен на рисунке 2.49.

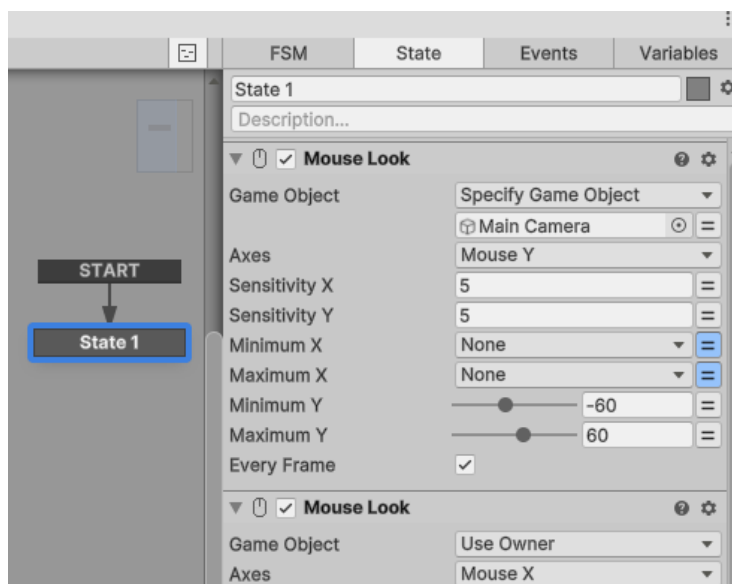


Рисунок 2.49 – Алгоритм управления камерой с помощью мыши

Элементы пользовательского интерфейса – это набор экранных подсказок, которые помогут пользователю сориентироваться при первоначальном исследовании программы.

Для некоторых пользователей, которые мало знакомы с игровыми механиками, может быть неочевидным, как нужно правильно управлять персонажем.

При создании элементов пользовательского интерфейса использовались изображения клавиш в формате «PNG» и набор текстовых подсказок. Изображения клавиш необходимы для того, чтобы показать пользователю, какую клавишу ему необходимо нажать чтобы выполнить определенное действие.

Реализованные элементы пользовательского интерфейса продемонстрированы на рисунке 2.50.

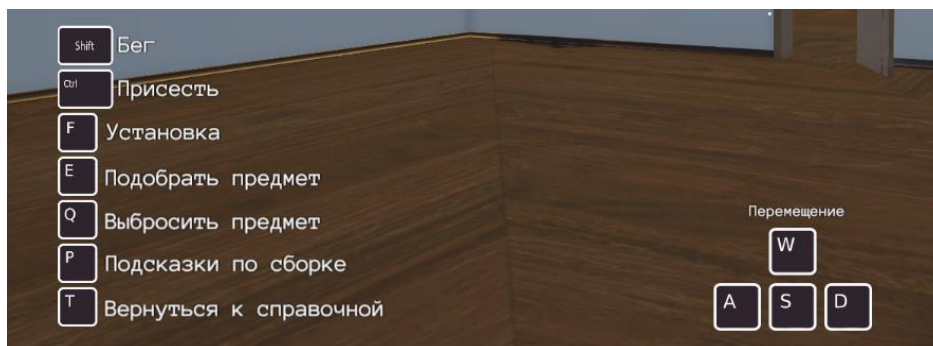


Рисунок 2.50 – Элементы пользовательского интерфейса

Программный модуль реализующий отображение элементов пользовательского интерфейса представлен на рисунке 2.51.

```
Скрипт Unity (1 ссылка на ресурсы) | Ссылка: 0
public class StartUI : MonoBehaviour
{
    [SerializeField] private Text textElement;
    [SerializeField] private float displayDuration = 5f;
    private float timer;
    private bool isDisplaying = true;

    public GameObject key_W;
    public GameObject key_A;
    public GameObject key_S;
    public GameObject key_D;

    Сообщение Unity | Ссылка: 0
    private void Start()
    {
        ShowText("Перемещение");
    }

    Сообщение Unity | Ссылка: 0
    void Update()
    {
        if (isDisplaying)
        {
            timer -= Time.deltaTime;
            if (timer <= 0f)
            {
                HideText();
            }
        }
    }
}
```

Рисунок 2.51 – Программный модуль отображения элементов интерфейса

Загруженные изображения в исполняющий программный модуль представлены на рисунке 2.52.

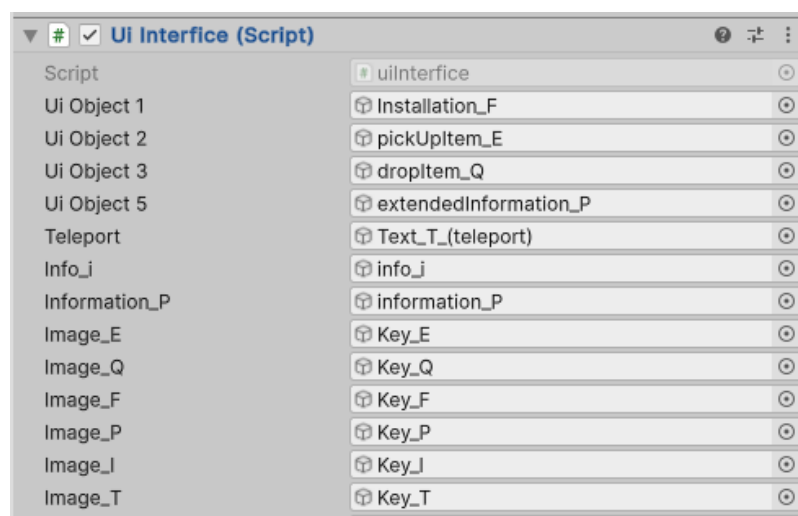


Рисунок 2.52 – Загруженные изображения клавиш для подсказок

На рисунке 2.53 представлены методы отображение подсказок отображающиеся при нажатие на клавиши подсказки на кнопку «I».



Рисунок 2.53 – Мотоды отображения клавиш при вызове подсказки

2.9 Разработка модуля воспроизведения звукового сопровождения

Модуль воспроизведения звукового сопровождения представляет из себя единый модуль (алгоритм), по действию которого осуществляется запуск звукового файла в формате «mp3» при нажатии кнопки запуска (кнопки «Воспроизвести») находящийся в каждом информационном стенде (в информационном блоке) с оборудованием (рисунок 2.54).

Данный модуль необходим для того, чтобы пользователь мог извлекать для себя информацию в формате звука, тем самым получая дополнительную информацию, преподнесенную в альтернативном повествовании. Такой подход к подаче информации повышает доступность к самой информации и помогает лучше понять, усвоить и закрепить материал.

Модуль воспроизведения звукового сообщения состоит из следующих компонентов:

- набор записанных файлов в формате mp3 с описанием для каждой модели оборудования;
- кнопки-префабы для каждого оборудования (шаблоны кнопок, отличающиеся по имени, исходной моделью которых является единая кнопка);
- программный модуль, осуществляющий воспроизведение и приостановку звука.

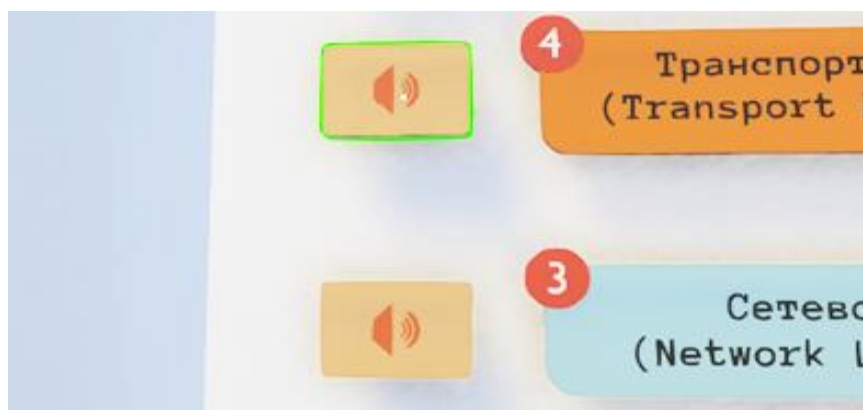


Рисунок 2.54 – Кнопки воспроизведения звукового сопровождения

Программный модуль работает следующим образом:

1. При старте программы, программный модуль находит объекты кнопок и иерархии программы, находит и сохраняет ссылки на файлы звукового сопровождения после обновления «Update()» в виде переменных (doska_OneAudio, doska_TwoAudio и так далее). Поиск файлов и сохранение ссылок на них представлен на рисунке 2.55.

2. При нажатии на кнопку запуска звукового сопровождения, программный модуль запускает метод, который принимает ссылку на файл и воспроизводит его. Пример поиска аудиофайла представлен на рисунке 2.56.

3. Метод воспроизведения звуков ставит на паузу другие звуки, даже если они не были воспроизведены.

```
doska_OneAudio = GameObject.Find("doska_OneAudio");
doska_TwoAudio = GameObject.Find("doska_TwoAudio");
doska_ThreeAudio = GameObject.Find("doska_ThreeAudio");
doska_FourAudio = GameObject.Find("doska_FourAudio");
doska_FiveAudio = GameObject.Find("doska_FiveAudio");
doska_SixAudio = GameObject.Find("doska_SixAudio");
doska_SevenAudio = GameObject.Find("doska_SevenAudio");
doska_TSIPAudio = GameObject.Find("doska_TSIPAudio");
doska_MainAudio = GameObject.Find("doska_MainAudio");

multiplesor_Audio_button = GameObject.Find("multiplesor_Audio_button");
router_Audio_button = GameObject.Find("router_Audio_button");
powtorit_Audio_button = GameObject.Find("powtorit_Audio_button");
crimperMain_Audio_button = GameObject.Find("crimperMain_Audio_button");
crimper_Audio_button = GameObject.Find("crimper_Audio_button");
```

Рисунок 2.55 – Пример сохранения файлов кнопок и ссылок на аудио-файлы

Ссылка: 65

```
public void AudioPlayAndStop(AudioSource audio, ref bool audioFlag, GameObject obj)
{
    if (IsClickMouseAnColliderWhy(Rom.hit.collider, obj))
    {
        if (audio != null)
        {
            if (!audioFlag)
            {
                // Останавливаем все другие аудио
                StopAllAudio();

                // Включаем текущее аудио
                audio.enabled = true;
                if (!audio.isPlaying) audio.Play();
                audioFlag = true;
            }
            else
            {
                audio.Pause();
                audioFlag = false;
            }
        }
    }
}
```

Рисунок 2.56 – Пример метода воспроизведения и остановки аудио

При запуске аудиофайла происходит приостановка всех файлов, которые были воспроизведены. Остановку всех воспроизведенных файлов осуществляет

метод «StopAllAudio». Метод приостановки всех аудиофайлов представлен на рисунке 2.57.

Ссылка 1

```
private void StopAllAudio()
{
    commutator_audio.Pause(); is_ca = false;
    Topology_ring.Pause(); is_tr = false;
    Sas_disk.Pause(); is_sd = false;
    audio_ohlad_main.Pause(); is_ohM = false;
    audio_Vodynka.Pause(); is_voD = false;
    audio_StandartOhlad.Pause(); is_stA = false;
    audio_passivOhlad.Pause(); is_paS = false;
    audio_termoint.Pause(); is_terM = false;

    cabelsMainAudio.Pause(); mainAudio = false;
    cableLanAudio.Pause(); audioCableLan = false;
    cableOptikaAudio.Pause(); audioCableOptika = false;
    cableCoaxialAudio.Pause(); audioCableCoaxial = false;

    sorse_One_Audio.Pause(); why_One_Doska = false;
    sorse_Two_Audio.Pause(); why_Two_Doska = false;
    sorse_Three_Audio.Pause(); why_Three_Doska = false;
    sorse_Four_Audio.Pause(); why_Four_Doska = false;
    sorse_Five_Audio.Pause(); why_Five_Doska = false;
    sorse_Six_Audio.Pause(); why_Six_Doska = false;
    sorse_Seven_Audio.Pause(); why_Seven_Doska = false;
    sorse_TSPIP_Audio.Pause(); why_TSPIP_Doska = false;
    sorse_Main_Audio.Pause(); why_Main_Doska = false;

    sorse_multiplecсор_Audio.Pause(); BOL_multiplecсор = false;
    sourse_router_Audio.Pause(); BOL_router = false;
    sourse_powtorit_Audio.Pause(); BOL_powtorit = false;
    sourse_crimperMain_Audio.Pause(); BOL_crimperMain = false;
    sourse_crimper_Audio.Pause(); BOL_crimper = false;
}
```

Рисунок 2.57 – Пример метода приостановки всех звуков

На рисунке 2.58 представлен метод взаимодействия программного модуля с объектами кнопок

Ссылка 1

```
public bool ClickButtonAudio(bool audio = false)
{
    if (IsClickMouseAnColliderWhy(Rom.hit.collider, buttonAUDIOMainCables))
    {
        if (AreAllAudioFlagsFalse())
        {
            audioCableLan = false;
            audioCableCoaxial = false;
            audioCableOptika = false;
            Debug.Log("Main");
        }

        AudioPlayAndStop(cabelsMainAudio, ref mainAudio, buttonAUDIOMainCables);
    }
    else if (IsClickMouseAnColliderWhy(Rom.hit.collider, buttonAUDIOCableLan))
    {
        if (AreAllAudioFlagsFalseExcept(buttonAUDIOCableLan))
        {
            mainAudio = false;
            audioCableCoaxial = false;
            audioCableOptika = false;
            Debug.Log("LAN Cable");
        }

        AudioPlayAndStop(cableLanAudio, ref audioCableLan, buttonAUDIOCableLan);
    }
}
```

Рисунок 2.58 – Пример метода взаимодействующего с кнопками

Метод «IsClickMouseAnColliderWhy» проверяет – кликнул ли пользователь на объект кнопки (рисунок 2.59).

```
Ссылка 66
public bool IsClickMouseAnColliderWhy(Collider hitCollider, GameObject targetObj)
{
    if (Input.GetMouseButtonDown(0))
    {
        if (hitCollider != null && hitCollider.gameObject == targetObj)
        {
            return true;
        }
    }
    return false;
}
```

Рисунок 2.59 – Метод проверки нажатия на кнопку

Чтобы не возникал конфликт между воспроизводимыми файлами, был написан метод который проверяет наличие активных файлов. Метод используется для того чтобы понять можно ли начинать проигрывать новый аудиофайл без возникновения конфликтов с другими файлами.

Audio Source (источник звука) воспроизводит Audio Clip в сцене. Если Audio Clip является 3D клипом, источник проигрывается в заданном положении в пространстве и будет приглушаться в зависимости от расстояния [25].

Директории с аудиофайлами представлена на рисунке 2.60.

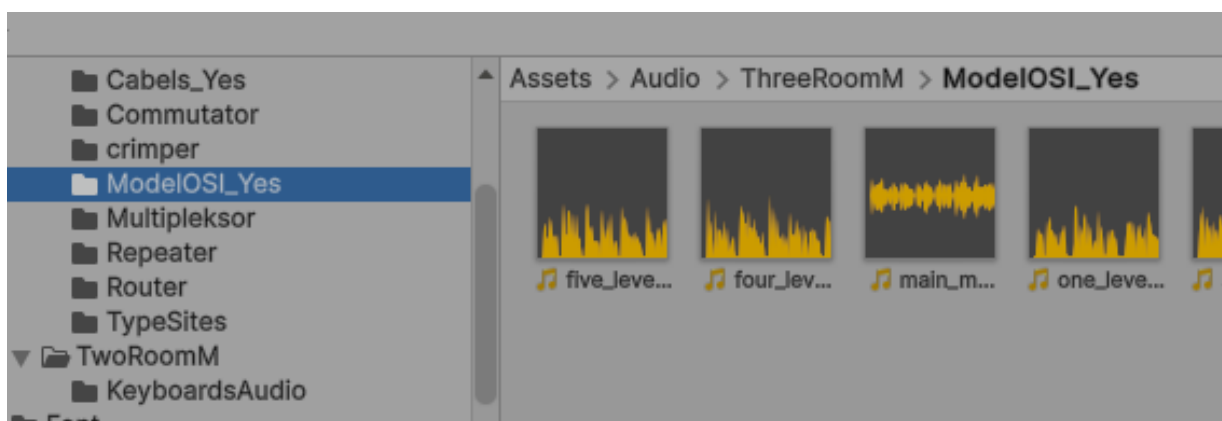


Рисунок 2.60 – Директории с аудиофайлами

3 Оценка эффективности внедрения программного обеспечения

3.1 Общие положения

При разработке программного обеспечения с последующим его внедрением, основное внимание уделяется не только реализации с технической и программной точки зрения, но и с учетом потенциальной эффективности применения разрабатываемого программного обеспечения.

Внедрение ИТ позволяет снижать расходы на учебные материалы, оптимизировать использование физического пространства, создавать персонализированные учебные планы, расширять доступ к образованию и сокращать затраты на дополнительные занятия [16, с. 199].

Эффективность информационной системы – это совокупность показателей информационной системы характеризующие уровень качества выполнения определенных задач в рамках поставленных условий.

Данными показателями являются:

- удобство использования системы;
- безопасность информационной системы;
- отказоустойчивость системы;
- масштабируемость системы;
- экономическая целесообразность разработки.

3.2 Показатели эффективности

Обязательным требованием к внедрению программного обеспечения является оценка эффективности внедрения, однако, это не представляется возможным осуществить без использования показателей эффективности.

В рамках разрабатываемого приложения виртуальной лаборатории для обучения по дисциплине «Компьютерные сети», показателями эффективности выступают:

- экономическая эффективность;
- социальная эффективность;
- педагогическая эффективность.

Экономическая эффективность представляет собой показатель эффективности который отражает экономическую целесообразность разработки приложения в соотношении экономических затрат на разработку, внедрения, сопровождения с последующим получаемым эффектом – снижение трудозатрат, сокращение расходов, ускорение процессов обработки информации и т.д.

Социальная эффективность – это показатель эффективности отражающий влияние разрабатываемого приложения на социальные аспекты образовательного процесса по отношению к обучаемой группе людей.

Педагогическая эффективность отражает уровень качества предоставляемого образовательного контента, степень усвоения материала студентами, а также влияние программного обеспечения на повышение процента успеваемости.

3.3 Расчет экономических затрат

Виртуальная лаборатория для обучения по дисциплине «Компьютерные сети» разрабатывается группой людей состоящей из двух человек: научного руководителя по выпускной квалификационной работы и техника по информационным системам.

Расчет месячного оклада работника высчитывается по формуле 3.3.1:

$$O_m = O_{min} \cdot K_t \quad (3.3.1)$$

Где, O_m – месячный оклад;

O_{min} – минимальная заработная плата на предприятии;

K_T – тарифный коэффициент.

Расчет месячного оклада для руководителя: $O_m = 35800 * 1,46 = 52268$ руб.

Расчет месячного оклада для техника по ИС: $O_m = 21200 * 1,15 = 24380$ руб.

Состав группы работников представлен в таблице 3.1.

Таблица 3.1 – Состав группы работников

Наименование	Количество	Тарифный коэффициент	Месячный оклад O_m , руб.
Научный руководитель	1	1,46	52268
Техник по ИС	1	1,15	24380
Всего	2		76648

Время участия при создания программного обеспечения было определено на основе перечня всех работ и трудоемкости их выполнения.

В таблице 3.2 представлен расчет трудоемкости выполнения перечня работ.

Таблица 3.2 – Расчет трудоемкости выполнения перечня работ

Наименование работ	T_{min}	T_{max}	T_p	Научный руководитель	Техник по ИС
Изучение задания	1	2	1,6	0,8	0,8
Анализ предмета исследования	2	4	3,2	0	3,2
Анализ объекта исследования	1	3	2,2	0	2,2
Определение цели и задач по разработке ПО	1	2	1,6	0	1,6
Обзор и анализ существующих технологий	3	5	4,2	0	4,2
Подбор и изучение необходимой литературы	2	4	3,2	0	3,2
Выбор и обоснование проектных решений	2	4	3,2	1	2,2
Разработка архитектуры и принципа функционирования	6	10	8,4	2	6,4
Проектирование трёхмерных моделей	4	8	6,4	0	6,4
Разработка программного модуля контроля знания	8	12	10,4	1	9,4

Продолжение таблицы 3.2

Разработка программных модулей взаимодействия с информационными стендами	6	10	8,4	0	8,4
Разработка программного модуля проектирования сетей	6	10	8,4	3	5,4
Разработка программного модуля механизма отслеживания прогресса	8	12	10,4	0	10,4
Разработка элементов управления и пользовательского интерфейса	8	12	10,4	0	10,4
Разработка модуля воспроизведения звукового сопровождения	4	6	5,2	0	5,2
Отладка, тестирование, корректировка программы, устранение выявленных ошибок	10	14	12,4	5	7,4
Всего	72	118	99,6	12,8	86,8

Трудоемкость выполнения работ рассчитывалась на основе экспертных оценок и вычислялась она по формуле 3.2:

$$T_p \frac{3 * t_{min} + 2 * t_{max}}{5} \quad (3.2)$$

где: T_p – расчетная трудоемкость выполнения работы;

t_{min} – минимальное время, необходимое для выполнения работы;

t_{max} – максимальное время, необходимое для выполнения работы;

Расчетная трудоемкость выполнения работы составляет 90,4 ($(3 * 72 + 2 * 118) / 5 = 90,4$). Сумма затрат на разработку ПО представлена в таблице 3.2.3.

Таблица 3.3 – Сумма затрат на разработку ПО

№	Наименование статьи затрат	Буквенное обозначение	Формула	Сумма, руб.
1	Зарплата техника по ИС	ЗПтех	$T_p * \text{Ом} / (21 * 8)$	3982,32
2	Зарплата научного руководителя	ЗПрук	$T_p * \text{Ом} / (21 * 8)$	12596,33
3	Итого выплаты	ЗПпо	$\sum \text{всех з/п}$	16578,65
4	Премия	П	$ЗПпо * 0,22$	3647,3

Продолжение таблицы 3.3

5	Выплаты по районному коэффициенту	Врк	$(3\text{Ппо}+\text{П})\cdot 0,15$	3033,89
6	Общий фонд оплаты труда работников	ФОТоб	$3\text{Ппо}+\text{П}+\text{Врк}$	23259,84
7	Единый социальный налог	ЕСН	$\text{ФОТоб}\cdot 0,356$	8280,5
8	Накладные расходы	НР	$3\text{Ппо}\cdot 1,2$	19894,38
9	Итого затрат	Зпо	$\text{ФОТоб}+\text{ЕСН}+\text{НР}$	51434,73
10	Затраты, связанные с работой компьютера при разработке ПО	Зком	$\text{Тр}\cdot \text{оклад}/(8\cdot 21)$	12596,33
11	Прочие затраты связанные с разработкой ПО	Зпр	$3\text{по}\cdot 0,35$	6963,03
12	Итого затрат на разработку	Зрп	$3\text{по}+\text{Зком}+\text{Зпр}$	70994,09
13	Налоги, включаемые в затраты	Нсп	$\text{ФОТоб}\cdot 0,1$	2325,98
14	Затраты на оформление программного обеспечения	Зоф	$3\text{пр}\cdot 0,15$	1044,45
15	Всего затрат на создание ПО	Зсп	$\text{Зрп}+\text{Нсп}+\text{Зоф}$	74364,53

Расчет суммы затрат на разработку программного обеспечения в таблице 3.3 показывает, что суммарные затраты на разработку виртуальной лаборатории для обучения по дисциплине «Компьютерные сети» составляют 74364 руб.

Данные затраты на разработку программного обеспечения не являются экономически выгодным решением, однако, данные затраты направлены на то, чтобы повысить социальную и педагогическую эффективность в образовательном процессе по дисциплине «Компьютерные сети».

На основе анализа литературных источников, предполагается, что применение разработанного программного обеспечения позволит повысить среднюю успеваемость студентов на 5-10% а также снизить педагогические затраты на повторное объяснение материала.

3.4 Оценка педагогической эффективности

Предполагается, что внедрение разрабатываемой виртуальной лаборатории для обучения по дисциплине «Компьютерные сети» сможет обеспечить педагогическую эффективность. Преподаватель сможет лучше

контролировать усвоение методического материала студентами, за счет модулей контроля знаний встроенных в приложение. Модули контроля знаний будут проверят усвоенный материал студентами через систему тестирования, которая в свою очередь будет выставлять промежуточные баллы за прохождения определенных этапов (прохождение тестов, проектирование сетей с оцениванием).

Помимо системы контроля знаний, ожидается, что сократится затрачиваемое время преподавателя на объяснение дополнительного материала.

Ожидаются следующие педагогические эффекты:

- повышенные результаты выполнения лабораторных работ;
- сокращение времени преподавателя повторное объяснение теоретического и практического материала;
- повышение интереса к изучению дисциплины;
- рост вовлеченности за счет применения механик геймификации.

3.5 Оценка социальной эффективности

Предполагается, что внедрение разрабатываемого приложения виртуальной лаборатории для обучения по дисциплине «Компьютерные сети» позволит повысить уровень заинтересованности и вовлеченности студентов в образовательном процессе.

Внедрение элементов геймификации в образовательный процесс влияет исключительно положительным образом на качество обучения. Человек обучающийся с использованием элементов геймификации лучше запоминает и усваивает информацию извлекаемую из игровой среды, поскольку такая информация преподносится в более простом и дружелюбном, игровом формате.

Применение цифровых образовательных ресурсов на игровых

платформах обладает высоким потенциалом в отношении мотивации, вовлеченности в процесс обучения и повышения качества обучения [13, с. 53].

Ярким примером успешного процесса геймификации в образовательной среде является браузерное приложение под названием «Flexbox Froggy». Данное браузерное приложение использует игровые элементы (кувшинки и лягушки) для обучения свойствам «flexbox» в каскадных таблицах стилей (CSS). Это необходимо для понимания, как нужно правильно позиционировать контент на странице внутри тегов-контейнеров.

Аналогичный эффект ожидается и от внедрения виртуальной лаборатории для обучения по дисциплине «Компьютерные сети».

ЗАКЛЮЧЕНИЕ

Цель выпускной квалификационной (бакалаврской) работы заключалась в разработке виртуальной лаборатории для обучения по дисциплине «Компьютерные сети» (на примере Рубцовского института (филиала) АлтГУ)

Для достижения цели были решены следующие задачи:

- проведен анализ предметной области;
- выработаны проектные решения по обеспечивающим системам;
- реализованы проектные решения разрабатываемой виртуальной лаборатории «Компьютерные сети»;
- проведен расчет экономических затрат на разработку виртуальной лаборатории «Компьютерные сети».

Следовательно, можно сделать о том, что цель выпускной квалификационной (бакалаврской) работы была достигнута.

В дальнейшей разработке виртуальной лаборатории для обучения по дисциплине «Компьютерные сети» возможна разработка системы авторизации пользователей для того, чтобы можно было сохранять результаты тестирования и отправлять их преподавателю. Это позволит преподавателю более тщательно отслеживать и контролировать процесс обучения учеников.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Сведения об образовательной организации, Основные сведения [Электронный ресурс]: учебно-методический портал Рубцовского института (филиала) АлтГУ – Режим доступа: <https://rb.asu.ru/sveden/common>. – Загл. с экрана.
2. Миссия и политика Рубцовского института [Электронный ресурс]: учебно-методический портал Рубцовского института (филиала) АлтГУ – Режим доступа: <https://rb.asu.ru/content/article/2270/>. – Загл. с экрана.
3. Сведения о средствах обучения и воспитания [Электронный ресурс]: учебно-методический портал Рубцовского института (филиала) АлтГУ – Режим доступа: <https://rb.asu.ru/sveden/objects>. – Загл. с экрана.
4. Политика института в области качества [Электронный ресурс]: учебно-методический портал Рубцовского института (филиала) АлтГУ – Режим доступа: <https://rb.asu.ru/content/13501>. – Загл. с экрана.
5. Семиглазов, В. А. 3D Технологии: учебное пособие / В. А. Семиглазов. – Москва: ТУСУР, 2023. – 192 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/394100>. – Режим доступа: для авториз. пользователей.
6. Maya от Autodesk – Что это за программа для создания 3D-анимации [Электронный ресурс]: – Режим доступа: <https://media.contented.ru/glossary/maya>. – Загл. с экрана.
7. История Blender’a [Электронный ресурс]: – Режим доступа: https://docs.blender.org/manual/ru/latest/getting_started/about/history.html. – Загл. с экрана.
8. Лицензия – Blender Manual [Электронный ресурс]: – Режим доступа: <https://docs.blender.org/manual/ru/2.82/about/license.html>. – Загл. с экрана.
9. Основы проектирования предприятий отрасли с использованием

систем автоматизированного проектирования. Практикум: учебное пособие / Р. В. Крюк, Р. М. Котов, М. Г. Курбанова [и др.]. – Кемерово: КемГУ, 2024. – 121 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/420755>. – Режим доступа: для авториз. пользователей.

10. Преображенская, Е. В. Создание трёхмерного проекта в Unity: учебно-методическое пособие / Е. В. Преображенская, А. А. Лим. – Москва: РТУ МИРЭА, 2024. – 76 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/432683>. – Режим доступа: для авториз. пользователей.

11. Баланов, А. Н. Комплексное руководство по разработке: от мобильных приложений до веб-технологий: учебное пособие для вузов / А. Н. Баланов. – Санкт-Петербург: Лань, 2024. – 412 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/394577>. – Режим доступа: для авториз. пользователей.

12. Девицын, И. Н. Системное программное обеспечение: учебное пособие / И. Н. Девицын, Т. В. Гавриленко. – Сургут: СурГУ, 2024. – 44 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/422306>. – Режим доступа: для авториз. пользователей.

13. Везиров, Т. Г. Курс лекций по дисциплине «Визуализация и геймификация в образовании»: учебное пособие / Т. Г. Везиров. – Махачкала: ДГПУ, 2024. – 72 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/442670>. – Режим доступа: для авториз. пользователей.

14. Иванько, А. Ф. Информационные системы в издательском деле: учебное пособие / А. Ф. Иванько, М. А. Иванько. – Санкт-Петербург: Лань, 2022. – 148 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/206885>. – Режим доступа: для авториз. пользователей.

15. Довгучиц, С. И. Разработка информационно-аналитических систем: учебное пособие: в 2 частях / С. И. Довгучиц, К. В. Балдин. – Москва: РТУ МИРЭА, 2023 – Часть 1 – 2023. – 80 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/382697>. – Режим доступа: для авториз. пользователей.
16. Баланов, А. Н. Цифровизация в образовательной сфере: учебное пособие для вузов / А. Н. Баланов. – Санкт-Петербург: Лань, 2024. – 400 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/417767>. – Режим доступа: для авториз. пользователей.
17. Структура и органы управления образовательной организацией [Электронный ресурс]: учебно-методический портал Рубцовского института (филиала) АлтГУ – Режим доступа: <https://rb.asu.ru/sveden/struct>. – Загл. с экрана.
18. Введение в Godot [Электронный ресурс]: Документация Godot Engine 4.4 на русском языке – Режим доступа: https://docs.godotengine.org/ru/4.x/getting_started/introduction/introduction_to_godot.html. – Загл. с экрана.
19. Начните с Unity Hub [Электронный ресурс]: – Режим доступа: <https://docs.unity3d.com/hub/manual/index.html>. – Загл. с экрана.
20. Что лучше для скульптинга ZBrush или Blender [Электронный ресурс]: – Режим доступа: <https://jcenters-courses.com/blog/chto-luchshe-dlya-skulptinga-zbrush-ili-blender/>. – Загл. с экрана.
21. Титова, С. В. Цифровая методика обучения иностранным языкам : учебник для вузов / С. В. Титова. – Москва : Издательство Юрайт, 2025. – 248 с. – (Высшее образование). – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/568551>. – Загл. с экрана.
22. Информация о местах осуществления образовательной деятельности, которые включаются в соответствующую запись в реестре

лицензий на осуществление образовательной деятельности [Электронный ресурс]: учебно-методический портал Рубцовского института (филиала) АлтГУ – Режим доступа: <https://rb.asu.ru/sveden/common>. – Загл. с экрана.

23. Playmaker – Visual Scripting for Unity [Электронный ресурс]: – Режим доступа: <https://hutonggames.com/showcase.html>. – Загл. с экрана.

24. API Скриптов: Collider.OnTriggerEnter(Collider) [Электронный ресурс]: Unity Documentation – Режим доступа: <https://docs.unity3d.com/ru/530/ScriptReference/Collider.OnTriggerEnter.html>. – Загл. с экрана.

25. Audio Source – Unity Manual [Электронный ресурс]: Unity Documentation – Режим доступа: <https://docs.unity3d.com/ru/2019.4/Manual/class-AudioSource.html>. – Загл. с экрана.

ПРИЛОЖЕНИЕ А

Исходные данные

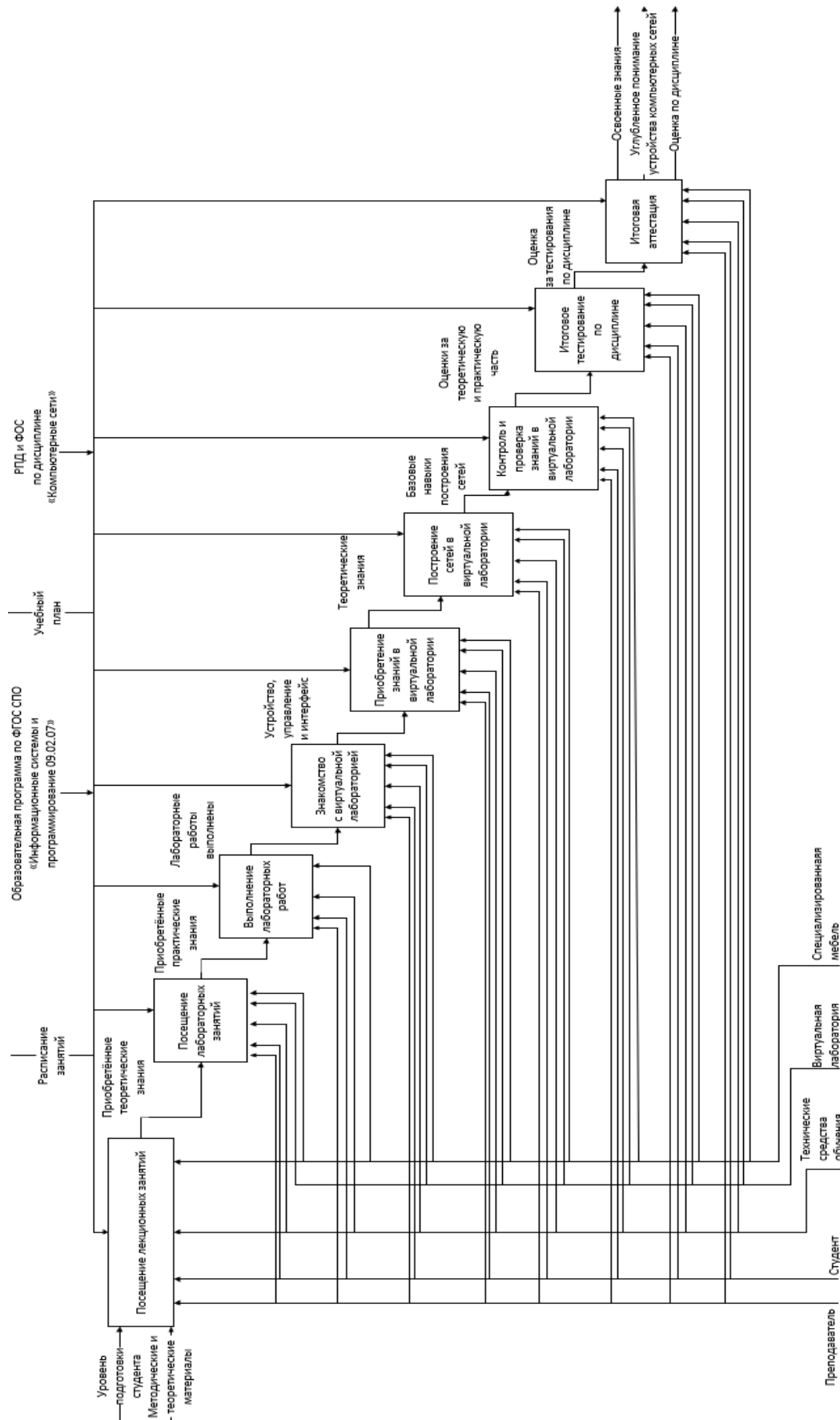


Рисунок А1 – Декомпозиция контекстной диаграммы «как будет» процесса обучения по дисциплине

«Компьютерные сети»