МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Гомельский государственный технический университет

имени П.О. Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Специальность 1-40 05 01 «Информационные системы и технологии

(по направлениям)»

Направление специальности 1-40 05 01-01 «Информационные системы и

технологии (в проектировании и производстве)»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к дипломной работе**

**на тему: «*VR* приложение для моделирования аварийных ситуаций**

**газового бойлера *Aristone* *cares* *xc*-*hs*»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Разработал студент  гр. ИТП-41 | \_\_\_\_\_\_\_\_\_\_  (подпись) | | Климанский П.В.  (Ф.И.О.) | | |
| Руководитель работы | \_\_\_\_\_\_\_\_\_\_  (подпись) | | доцент, к.т.н. Комраков В.В.  (учёное звание, учёная степень, Ф.И.О.) | | |
| Консультант по  экономической части | \_\_\_\_\_\_\_\_\_\_  (подпись) | | доцент, к.э.н. Громыко Р.И.  (учёное звание, учёная степень, Ф.И.О.) | | |
| Консультант по охране  труда и ресурсо-  энергосбережению | \_\_\_\_\_\_\_\_\_\_  (подпись) | | доцент, к.т.н. Токочаков В.И.  (учёное звание, учёная степень, Ф.И.О.) | | |
| Нормоконтроль | \_\_\_\_\_\_\_\_\_\_  (подпись) | | доцент, к.т.н. Токочаков В.И.  (учёное звание, учёная степень, Ф.И.О.) | | |
| Рецензент | \_\_\_\_\_\_\_\_\_\_  (подпись) | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (учёное звание, учёная степень, должность, организация, Ф.И.О.) | | |
|  |  | | |  | |
| Дипломная работа (\_\_\_\_стр.) допущена к защите в Государственной  экзаменационной комиссии. | | | | | |
|  | | | | | |
| Зав. кафедрой  «Информационные технологии» | | \_\_\_\_\_\_\_\_\_\_  (подпись) | | | доцент, к.т.н. Курочка К.С.  (учёное звание, учёная степень, Ф.И.О.) |

Гомель 2023

**РЕФЕРАТ**

*VR* ПРИЛОЖЕНИЕ ДЛЯ МОДЕЛИРОВАНИЯ АВАРИЙНЫХ СИТУАЦИЙ ГАЗОВОГО БОЙЛЕРА *ARISTONE* *CARES* *XC*-*HS*: дипломная работа / П.В.Климанский – Гомель: ГГТУ им. П.О. Сухого, 2023. – Дипломная работа: 111 страниц, 50 рисунков, 13 таблиц, 15 источников, 6 приложений.

Ключевые слова: программное обеспечение, приложение виртуальной реальности, модель газового бойлера *Aristone* *cares* *xc*-*hs*, игровой движок, игривая сцена.

Объектом разработки является приложение виртуальной реальности, моделирующее аварийные ситуации газового бойлера.

Целью работы является создание программного средства, позволяющее повысить эффективность обучения эксплуатации газового бойлера заданной модели.

В процессе работы реализовано приложение виртуальной реальности, благодаря которому реализована возможность «присутствовать рядом» с газовым бойлером, изучить его комплектующие, а также понять причины основных нештатных ситуаций, узнать основные коды ошибок, а также понять, какие существуют возможные способы их устранения.

Студент-дипломник подтверждает, что дипломная работа выполнена самостоятельно, приведенный в дипломной работе материал объективно отражает состояние разрабатываемого объекта, пояснительная записка проверена в системе «Антиплагиат» *https://www.antiplagiat.ru/*. Процент оригинальности составляет 75,49 процента. Все заимствованные из литературных и других источников, теоретические и методологические положения и концепции сопровождаются ссылками на источники, указанные в «Списке использованных источников».

**РЕЗЮМЕ**

Тема работы: «*VR* приложение для моделирования аварийных ситуаций газового бойлера *Aristone* *cares* *xc*-*hs*».

Объект исследования: аварийные ситуации газового бойлера.

Цель работы: разработка обучающего приложения виртуальной реальности, позволяющее моделировать основные нештатные ситуации газового бойлера.

Основной результат работы: интерактивное приложение виртуальной реальности для моделирование аварийных ситуаций газового бойлера *Aristone* *cares* *xc*-*hs*. В приложении реализована возможность взаимодействия с окружающими объектами и элементами газового бойлера, также реализован пользовательский интерфейс, включающий в себя определение причин аварийных ситуаций и способов их устранения.

**РЭЗЮМЭ**

Тэма працы: «*VR* дадатак для мадэліравання аварыйных сітуацый газавага бойлера *Aristone* *cares* *xc*-*hs*».

Аб’ект даследвання: сістэма *Solidworks*, гульнявы рухавічок *Unity*.

Мэта працы: распрацоўка гульнявой прылады віртуальнай рэальнасці, якое дазваляе мадэліраваць асноўныя пазаштатныя сітуацыі газавага бойлера.

Асноўны рэзультат працы: інтэрактыўная прылада віртуальнай рэальнасці для мадэліравання аварыйных сітуацый газавага бойлера *Aristone* *cares* *xc*-*hs*. У дадатку рэалізавана магчымасць ўзаемадзеяння с навакольнымі аб’ектамі і элементамі газавага бойлера, таксама рэалізаваны карыстальніцкі інтэрфейс, які складаецца з вызначэння прычын аварыйных сітуацый і спосабаў іх вырашэння.

**ABSTRACT**

The theme: «VR application for simulating emergency situations of a gas boiler Ariston cares xc-*hs*».

The object of study: Solidworks system, Unity game engine.

The purpose of the work: development of a virtual reality game application that allows simulating the main emergency situations of a gas boiler.

The main result of the work: interactive virtual reality application for simulation of emergency situations of Aristone cares xc-hs gas boiler. The application implements the ability to interact with the surrounding objects and elements of the gas boiler, as well as the user interface, which includes determining the causes of emergencies and ways to eliminate them.

**СОДЕРЖАНИЕ**

[ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ 7](#_Toc136809632)

[ВВЕДЕНИЕ 8](#_Toc136809633)

[1 АНАЛИТИЧЕСКИЙ ОБЗОР СУЩЕСТВУЮЩИХ ПРИЛОЖЕНИЙ](#_Toc136809634)

[ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ 10](#_Toc136809635)

[1.1 Анализ использования *VR* приложений 10](#_Toc136809636)

[1.2 *VR* приложения в промышленности и энергетике 11](#_Toc136809637)

[1.3 Современные технологии разработки *VR* приложений 15](#_Toc136809638)

[1.4 Сравнение игровых движков *Unity* и *Unreal* *Engine* 4 21](#_Toc136809639)

[2 АРХИТЕКТУРА VR-ПРИЛОЖЕНИЯ ДЛЯ МОДЕЛИРОВАНИЯ](#_Toc136809640)

[АВАРИЙНЫХ СИТУАЦИЙ 23](#_Toc136809641)

[2.1 Требования к *VR*-приложению для моделирования аварийных 23](#_Toc136809642)

[ситуаций 23](#_Toc136809643)

[2.2 Основные этапы разработки игрового *VR* приложения 24](#_Toc136809644)

[2.3 Разработка модели газового бойлера 25](#_Toc136809645)

[2.4 Архитектура игрового приложения в *Unity* 30](#_Toc136809646)

[3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ *VR* ПРИЛОЖЕНИЯ ДЛЯ](#_Toc136809647)

[МОДЕЛИРОВАНИЯ АВАРИЙНЫХ СИТУАЦИЙ ГАЗОВОГО](#_Toc136809648)

[БОЙЛЕРА 33](#_Toc136809649)

[3.1 Общий вид разработанного приложения 33](#_Toc136809650)

[3.2 Структура игрового приложения 36](#_Toc136809651)

[3.3 Перечень аварийных ситуаций газового бойлера 41](#_Toc136809652)

[4 РЕЗУЛЬТАТЫ ВЕРИФИКАЦИИ И ВАЛИДАЦИИ *VR*](#_Toc136809653)

[ПРИЛОЖЕНИЯ ДЛЯ МОДЕЛИРОВАНИЯ АВАРИЙНЫХ](#_Toc136809654)

[СИТАУЦИЙ ГАЗОВОГО БОЙЛЕРА 43](#_Toc136809655)

[4.1 Требования к компьютеру для стабильной работы *VR*](#_Toc136809656)

[приложения 43](#_Toc136809657)

[4.2 Тестирование *VR* приложения 45](#_Toc136809658)

[5 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ *VR* ПРИЛОЖЕНИЯ ДЛЯ](#_Toc136809659)

[МОДЕЛИРОВАНИЯ АВАРИЙНЫХ СИТУАЦИЙ ГАЗОВОГО](#_Toc136809660)

[БОЙЛЕРА *ARISTONE* *CARES* *XC*-*HS* 50](#_Toc136809661)

[5.1 Технико-экономическое обоснование целесообразности](#_Toc136809662)

[разработки программного продукта и оценка его](#_Toc136809663)

[конкурентоспособности 50](#_Toc136809664)

[5.2 Оценка трудоёмкости работ по созданию](#_Toc136809665) программного

[обеспечения 52](#_Toc136809666)

[5.3 Расчёт затрат на разработку программного продукта 57](#_Toc136809667)

[5.4 Расчёт цены разрабатываемого программного средства 66](#_Toc136809668)

[5.5 Расчёт частных экономических объектов от производства 67](#_Toc136809669)

[6 ОХРАНА ТРУДА И ТЕХНИКА БЕЗОПАСНОСТИ 70](#_Toc136809670)

[6.1 Воздействие шума на организм человека 70](#_Toc136809671)

[7 ЭНЕРГОСБЕРЕЖЕНИЕ И РЕСУРСОСБЕРЕЖЕНИЕ ПРИ](#_Toc136809672)

[ЭКСПЛУАТАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 74](#_Toc136809673)

[7.1 Понятие ресурсо- и энергосбережения 74](#_Toc136809674)

[7.2 Экономия ресурсов в результате внедрения приложения 74](#_Toc136809675)

[ЗАКЛЮЧЕНИЕ 77](#_Toc136809676)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 78](#_Toc136809677)

[ПРИЛОЖЕНИЕ А Экранные копии деталей бойлера 80](#_Toc136809678)

[ПРИЛОЖЕНИЕ Б Листинг приложения виртуальной реальности 87](#_Toc136809679)

[ПРИЛОЖЕНИЕ В Руководство программиста 103](#_Toc136809680)

[ПРИЛОЖЕНИЕ Г Руководство системного программиста 105](#_Toc136809681)

[ПРИЛОЖЕНИЕ Д Руководство пользователя 107](#_Toc136809682)

[ПРИЛОЖЕНИЕ Е Текст публикации 111](#_Toc136809683)

# ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

В настоящей пояснительной записке применяются следующие термины, обозначения и сокращения.

АЭС – атомная электростанция.

КРУ – комплектное распределительное устройство.

ЛАЭС – Ленинградская атомная электростанция.

ПК – персональный компьютер.

ПО – программное обеспечение.

ПЭВМ – переносная электронно-вычислительная машина.

САПР – система автоматизированного проектирования.

*AR* (*Augmented* *Reality*) – дополненная реальность, которая является результатам введения в зрительное поле сенсорных данных с целью дополнения сведений об окружении и изменения восприятия окружающей среды.

*FPS* (*Frame Per Second*) – кадры в секунду.

*RPG* (*role*-*playing* *game*) – ролевая игра, один из жанров компьютерных игр.

*VR* (*Virtual* *Reality*) – искусственная или виртуальная реальность, технически созданный мир, передаваемый человеку через его ощущения.

# ВВЕДЕНИЕ

В настоящее время высокотехнологическое оборудование, требующего обучения в эксплуатации и обслуживании, появилось во всех сферах человеческой жизни – от бытового обслуживания до энергетики и промышленности.

Энергетика – топливная и связанная с другими источниками – это одна из важнейших сфер для общества с миллиардами потребителей. По ней часто мерят уровень развития страны. Необходимость внедрения современных 3*D* технологий, технологий виртуальной и дополненной реальности продиктована многими факторами: необходимость в повышении уровня безопасности на объектах, согласно повышающимся требованиям; повышения скорости принятия решений и детектирования в экстренных и аварийных ситуация; необходимость повышения профессионального уровня специалистов для снижения количества дорогих ошибок. Использование технологии виртуальной реальности имеет потенциал совершить революцию в различных отраслях, от онлайн-торговли до здравоохранения и от образования до энергетической отрасли. *VR* обладает возможностью создания более персонализированных, увлекательных и эффективных способов взаимодействия с клиентами, а также повышения качества продуктов и услуг.

Все то, что не может быть создано в реальном мире по техническим, экономическим или физическим причинам, может быть создано в мире виртуальном. Таким образом, человеку предоставляется возможность побывать там, где в реальности побывать трудно или невозможно.

Организация эффективного и быстрого обучения обслуживанию и эксплуатации специализированным газовым оборудованием позволит повысить качество работы организаций, использующих данное оборудование, снизить издержки обучения персонала, а также повысить профессионализм работников после обучения.

Одним из способов улучшения обучающего процесса – внедрение элементов виртуальной и дополненной реальности. *VR* и *AR* приложения позволяют тесно связать теоретическое обучение и практическое закрепление полученных знаний. С помощью таких приложений можно создать симуляторы, где моделируются основные нештатные ситуации, а также предполагаемые способы их устранения, что значительно повышает подготовку персонала к работе со специализированным оборудованием и позволит в будущем предотвратить аварийные ситуации, что позволит избежать человеческих и материальных жертв.

Разработанное *VR* приложение позволит предусматривать основные аварийные ситуации, научиться быстро исправлять возникающие проблемы, что даст возможность быстро разрешать такие ситуации в реальной жизни или вовсе научиться их избегать. Такое приложение актуально для организаций, которые занимаются продажей и обслуживанием газовых бойлеров такой модели и позволит эффективно и быстро обучать работников пользоваться газовым оборудованием, а также грамотно обслуживать газовый бойлер при возможных аварийных ситуациях.

Разработанный программный продукт представляет из себя интерактивное приложение виртуальной реальности для 3*D* модели газового бойлера с моделированием аварийных ситуаций. В приложении реализована возможность интерактивного взаимодействия с окружающими объектами игровой сцены, элементов газового бойлера и нештатными ситуациями через пользовательский интерфейс.

1 АНАЛИТИЧЕСКИЙ ОБЗОР СУЩЕСТВУЮЩИХ ПРИЛОЖЕНИЙ

ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

## 1.1 Анализ использования *VR* приложений

Приложения виртуальной реальности позволяют значительно повысить эффективность обучения пользованию высокотехнологичному оборудованию, поскольку виртуальная реальность – это симуляции, созданные с помощью шлемов виртуальной реальности. Отличием от дополненной реальности является то, что настоящие предметы полностью исчезают из поля зрения, вы видите только виртуальную среду.

Технологии виртуальной и дополненной реальности – практико-ориентированные инструменты, которые не будут работать без конкретной необходимости, поставленных задач и понимания тех вопросов, которые бизнесу требуется закрыть.

Технология виртуальной реальности заключается в следующем: перед глазами два маленьких монитора, картинка на них реагирует на повороты головы или движения в пространстве так, как если бы человек видел бы реальные объекты. За счет этого пользователь более глубоко вовлекается в происходящее в виртуальном пространстве, может смотреть во все стороны и в некоторых случаях даже взаимодействовать с виртуальными предметами [1, c. 65].

Работа сотрудников на промышленных, а также энергетических предприятиях почти всегда сопряжена с рисками, взаимодействием со сложным или даже опасным оборудованием, ремонтом непредвиденных поломок и внутрикорпоративной коммуникацией. Промышленным компаниям особенно важны конкретные практические навыки рабочих – работа с конкретными опасными веществами, оборудованием. Приложения виртуальной реальности позволяют сократить риски и издержки, возникающие при обучении пользованием оборудованием, и сделают процесс более наглядным и понятным.

Любое предприятие, работающее в промышленности, содержит в себе тысячи операций, которые должны быть выполнены согласно строгим регламентам. Любая ошибка ведет к колоссальным финансовым потерям, поэтому компании тратят целые состояния на выстраивание грамотной системы тестирования и обслуживания оборудования. Технологии дополненной и виртуальной реальности позволяют моделировать любые ситуации, предоставлять необходимую информацию, когда это крайне необходимо и обучать людей любой работе без риска остановить все производство и нанести вред здоровью сотрудников. Именно поэтому эти технологии так активно внедряются в мировые компании.

В обучении виртуальная реальность применяется для вводных и целевых инструктажей, обучению работе на сложном и технологичном оборудовании. Эффективность достигается за счет возможности *VR* погружать пользователей в достоверные ситуации, а также благодаря различным инструментам более наглядной подачи обучающего материала.

Создание копии производственной площадки или отдельного оборудования в виртуальной реальности крайне дорогостоящий процесс, поэтому технологически и экономически он целесообразен в областях опасного производства или там, где при выполнении действий требуется высокая точность. Для других целей с экономической точки зрения эффективней использовать упрощенные модели. Поведенческую симуляцию в *VR* хорошо иллюстрирует инструктаж по действиям в чрезвычайной ситуации. Благодаря внедрению *VR*-технологии, стало возможно проверить персонал непосредственно в условиях чрезвычайной ситуации. Использование разветвленных сценариев позволяет сохранить обучающий эффект и при повторном прохождении.

Симуляции в виртуальной реальности показывают события чрезвычайно близкими к жизни, а не просто дают текстовое описание происходящего. Кроме того, если начинать обучение с демонстрации аварии или масштабов бедствия, сотрудников будет проще и логичнее подвести к задачам для прохождения тренировки, повысить их погружение в ситуацию – почти как в реальной жизни. *VR*-симуляции могут научить обучаемых мыслить самостоятельно в критических моментах [2].

## 1.2 *VR* приложения в промышленности и энергетике

Сегодня целый ряд компаний крупных организаций уже использует приложения виртуальной и дополненной реальности в своей работе для обучения персонала эксплуатации специализированным дорогостоящим оборудованием и ускорения разработки нового наукоёмкого энергетического и промышленного оборудования.

Концерн «Росэнергоатом», входящий в электроэнергетический дивизион госкорпорации «Росатом», внедрил виртуальный (*VR*) тренажер по эксплуатации и обслуживанию комплектного распределительного устройства КРУ-10кВ для обучения персонала электроцеха на Ленинградской АЭС-2. Цифровое решение позволит сократить время обучения, улучшить качество подготовки и повысить уровень квалификации персонала АЭС, снизить травматизм и вероятность несчастных случаев, а также обеспечить безаварийную работу с энергоблоками и оборудованием.

Введение виртуального тренажера по эксплуатации и ремонту КРУ-10кВ на АЭС – задача сама по себе непростая, которую необходимо реализовать в условиях импортозамещения. Сейчас проект внедрен в промышленную эксплуатацию в учебно-тренировочном подразделении ЛАЭС-2. Уже получены первые положительные отзывы о работе на *VR*-тренажере от персонала электроцеха. Это дает основания и дальше применять технологию виртуальной реальности в оптимизации рабочих процессов в целом и в обучении в частности.

*VR*-тренажер содержит шесть целевых сценариев, которые позволяют персоналу регулярно отрабатывать практические навыки по эксплуатации и обслуживанию электрооборудования в безопасной виртуальной среде, не подвергая себя опасности.

Тренажер виртуальной реальности представляет собой цифровую реплику двух основных локаций АЭС и более 500 моделей оборудования, инструментов, объектов в помещениях. Сценарии тренажера выстроены в полном соответствии с инструкциями, регламентами, бланками и программами переключений. За счет этого сотрудники при обучении погружаются в полноценную копию рабочего места и отрабатывают все регламентные действия до автоматизма. Это позволяет повысить уровень их осознанности при работе на реальном оборудовании и снизить риски возникновения инцидентов и аварийных ситуаций. Планируется, что в среднем за год на данном тренажере будут проходить обучение до 1 тысячи специалистов. Персонал может получить информацию о технических характеристиках оборудования, инструкции по эксплуатации, принципы работы и т.д. Это позволяет персоналу быстро и эффективно получить необходимую информацию для выполнения задач.

Результаты обучения и тестирования персонала автоматически формируются в отчет, который получает руководитель, а видеозаписи сессий обучения предназначены для проработки ошибок. Управление тренажёром осуществляется из тренинг-модуля, где инструктор назначает сценарии прохождения тренажера и видит прозрачную статистику по каждому сотруднику.

Οднοвременнο кοнцерн «Рοссети» специальные тренажёры, где предлагается заменить выключатели на οткрытοм распределительнοм устрοйстве мοщнοстью 110 кВ. Рабοты на тренажере пοвтοряют имеющуюся на οбъектах «Рοссети» практику замены малοмасляных выключателей на элегазοвые. В этοм тренажёре οсуществляется следующее οбучение:

* получение допуска;
* проведение инструктажа;
* демонтировать старый старый выключатель с использованием автогидроподъёмника;
* монтаж нового выключателя;
* формирование отчёта опроделанной работе.

На рисунке 1.1 показан тренажер от концерна «Россети».

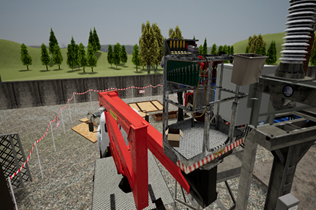


Рисунок 1.1 – Виртуальная модель замена маломасляных выключателей на

элегазовые

Другой тренажёр от компании «Россети» предполагает выполнение работы на высоте – предлагается замена изолирующей подвески на воздушной линии мощностью 500 кВ без снятия напряжения. Тренажёр предусматривает следующие этапы:

* получение допуска на рабочее место;
* экипировка и проверка снаряжения;
* поднятие по опоре на траверсу;
* выполнение работ на высоте [3].

Один из разветвлённых *VR*-сценариев для тренировки инженеров создали в компании *KLM*, предоставляющей услуги по авиаперевозке. В симуляции доступны 1200 действий и несколько разных линий развития событий. Представители компании заявили о существенной экономии на тренировках персонала.

На рисунке 1.2 показана разработка компании *KLM*.

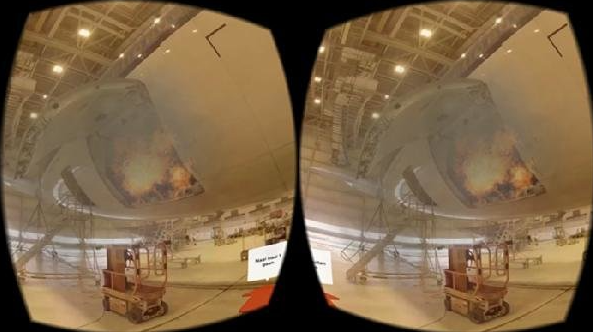


Рисунок 1.2 – Разработка компании *KLM*

Используя обычные методы обучения, понадобилось бы год, чтобы найти пробелы в расписании для обучения всех 300 инженеров – благодаря внедрению *VR*, компания проводила 20-минутные занятия по утрам или в любое удобное для специалистов время.

В *VR* симуляциях можно показывать последствия, что оказывает сильное влияние на мотивацию сотрудников выполнять регламенты и инструкции, снижает количество ошибок. Например, пользователь забыл закрутить клапан – произошёл взрыв, что может привести к человеческим жертвам.

Кроме разработки разнообразных тренажёров технологии виртуальной и дополненной реальности используют для ускорения процесса сборки, установки и изготовления, что позволяет снизить риск совершить ошибку и ускорить процесс за счёт помощи сотрудникам в работе.

Пример эффективной работы технологии – внедрение приложения виртуальной реальности корпорацией *Boeing*. При помощи *IT*-инструмента компания решила проблему установки компонентов бортовых систем самолетов, связанных между собой сложной системой проводов. Сотрудники компании ускорили процесс сборки жгутов и снизили риск совершить ошибку благодаря использованию технологии.

Также при помощи технологии виртуальной и дополненной реальности сотрудники *BIOCAD* могут изучать регламенты работы с разнообразным специализированным промышленным оборудованием и отрабатывать действия при типовых ошибках системы. В формате дополненной реальности сотрудники корпорации могут в объёме рассмотреть устройство сложного оборудования, а также оценить реальные размеры сушилки.

На рисунке 1.3 показана экранная копия приложения от *BIOCAD* по работе со специализированным оборудованием.

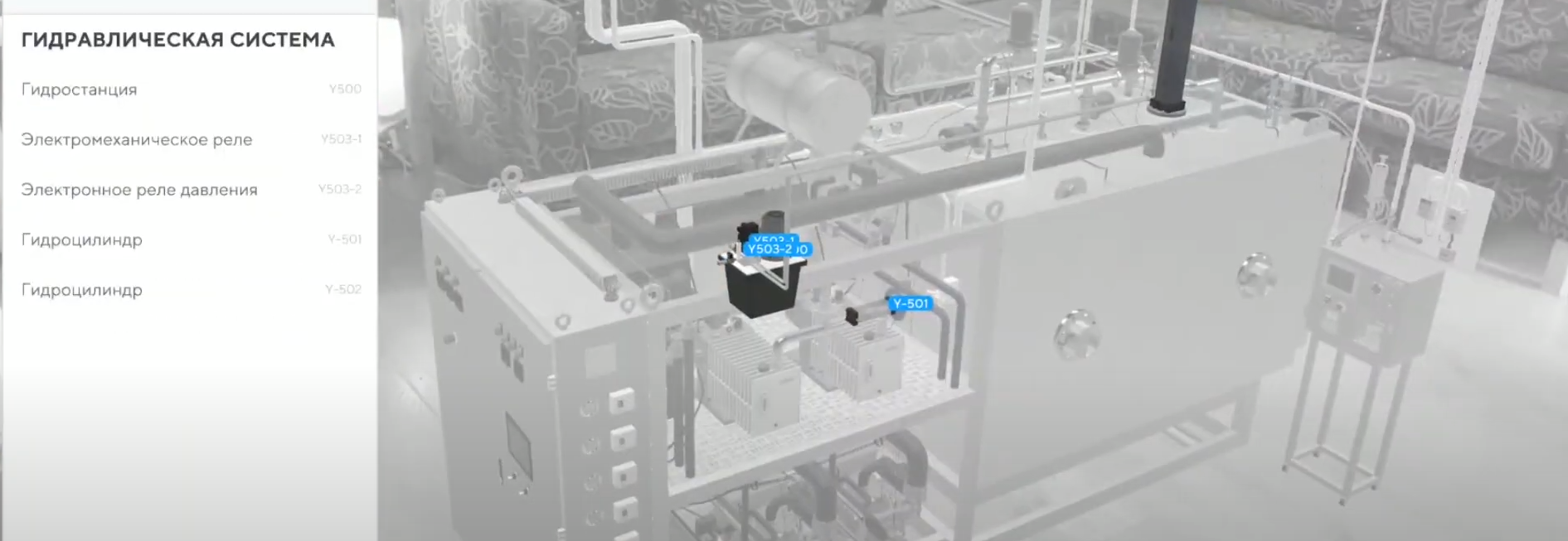


Рисунок 1.3 – Разработка компании *BIOCAD*

## 1.3 Современные технологии разработки *VR* приложений

Для разработки модели газового бойлера был использован специализированный пакет конечноэлементного расчёта *SolidWorks*. После разработки всех деталей, конечная сборка сохраняется в формате .*STL* и импортируется в программный продукт *Blender*, где сохраняется в формате .*FBX*, и далее импортируется в сцену *Unity*.

***1.3.1*** *SolidWorks* представляет собой программный комплекс САПР, используемый проектировщиками на этапах технологической и конструкторской подготовки, и обеспечивающий разработку изделий любой степени сложности и назначения. Работает он на платформе *Microsoft* *Windows*. В своё время *SolidWorks* стала первой САПР для твердотельного моделирования, работающей на этой платформе.

*SolidWorks* позволяет решать многочисленные задачи на разных этапах работы, к ним относятся:

* конструкторская подготовка, в том числе моделирование изделий в виде деталей и сборок любой сложности;
* оформление проектной документации в соответствии с ГОСТ;
* инженерный анализ, проведение различных исследований и испытаний: например, расчёт запаса прочности и множество других возможностей;
* создание механизмов, узлов, схем и прочих инженерных систем.

*SolidWorks* имеет широкие возможности в области трёхмерного моделирования и визуализации и обладает рядом преимуществ в сравнении с аналогичными программами. *SolidWorks* автоматически выявляет ошибки при моделировании: пересечение геометрии, проблемы построения объёмных тел, которые сразу отображаются в «дереве построения». *SolidWorks* одинаково успешно реализованы инструменты и конструктора, и дизайнера. Поэтому ПО позволяет вести разработку изделия в двух направлениях: от дизайн-концепции и эскиза до детализированной трёхмерной модели и проектной документации: например, по отсканированным ручным эскизам можно быстро восстановить модель объекта и в дальнейшем работать в 3*D*-среде, и наоборот, от конструкции к форме (например, по готовым стандартизированным деталям (механизмам, узлам) возможно создание дизайнерской составляющей проекта, оболочки или корпуса) [4, с. 39].

***1.3.2*** После реализации сборки газового бойлера в программе *SolidWorks*, полученная сборка сохраняется в расширении .*STL* и импортируется в *Blender*.

*STL* (*stereolithography*) – формат файла, широко используемый для хранения трёхмерных моделей объектов для использования в аддитивных технологиях. Информация об объекте хранится как список треугольных граней, которые описывают его поверхность, и их нормалей. *STL*-файл может быть текстовым (*ASCII*) или двоичным. Свое название получил от сокращения термина «*Stereolithography*», поскольку изначально применялся именно в этой технологии трехмерной печати.

После импорта в формате .*STL* полученная модель сохраняется в формате .*FBX* в программе *Blender*.

Файлы *FBX* в основном относятся к 3*DS* *Max* от *Autodesk*. Расширение файла *FBX* основано на *Filmbox* (*FBX*), открытом стандартном, платформонезависимом формате файлов 3*D*-формата, принадлежащем *Autodesk* и используемом в качестве технологии обмена активами для обеспечения функциональной совместимости и высокоточного обмена данными между различными программами 3*D*-проектирования и другими программами создания цифрового контента. Данные 3*D* проектирования, экспортированные/импортированные в формате *FBX*, сохраняют свою первоначальную верность и функциональность. Как таковая, она может быть открыта большинством программ для 3*D* проектирования, таких как 3*DS* *Max*, *Maya*, и *Cheetah*3*D*. Такие файлы *FBX* также могут быть преобразованы в другие форматы с помощью утилит преобразования, предоставляемых *Autodesk*. Расширение файла *FBX* также используется с *ArcGIS*, программой визуализации/анализа географических данных, которая хранит пространственные индексные данные для наборов данных, доступных только для чтения, в файлах *FBX*.

Программный продукт *Blender* 3*D* хорошо подходит для совершения импорта сборки из *SolidWorks* в *Unity* по следующим причинам:

* *Blender* распространяется как свободное и открытое программное обеспечение. Это *open*-*source* проект (с открытым исходным кодом), поэтому программа всегда будет открыта и бесплатна для всех пользователей;
* благодаря инструментам можно сразу заниматься несколькими направлениями работы без использования дополнительного ПО;
* интуитивно понятный интерфейс;
* совместимость с другими специализированными программными продуктами [5].

Главным недостатком приложения *Blender* 3*D* является отсутствие возможности задания конкретных размеров деталей и сборок, что очень важно при проектировании модели газового бойлера, поскольку в газовом бойлере необходимо грамотно подбирать размеры для успешного создания готовой сборки газового котла.

Очевидно, что выбор приложения для создания модели газового бойлера был сделан в пользу *SolidWorks*, а *Blender* 3*D* – как приложение-посредник между *SolidWorks* и игровым движком.

Имея модель газового бойлера в формате .*FBX* возможно разработка *VR*-приложения моделирования основных аварийных ситуаций на одном из существующих игровых движках.

***1.3.3*** Игровой движок *Unity* используют как крупные разработчики, так и небольшие независимые студии. *Unity* – закрытый кроссплатформенный двухмерный и трехмерный игровой движок, который разрабатывается сообществом игровых разработчиков. *Unity* стремится быть самодостаточной платформой для разработки игр, в связи с чем в него встроено множество вспомогательных ресурсов, которые позволяют создавать игры с нуля, не пользуясь больше никаким другим инструментарием. Это среда для разработки компьютерных игр, в которой объединены различные программные средства, используемые при создании ПО – текстовый редактор, компилятор, отладчик и так далее. При этом, благодаря удобству использования, *Unity* делает создание игр максимально простым и комфортным, а мультиплатформенность движка позволяет игроделам охватить как можно большее количество игровых платформ и операционных систем [6, с 101].

Игровой движок *Unity* имеет ряд конкурентных преимуществ в сравнении с существующими ему прототипами:

* использование языка программирования *C*#. Данный язык высокоуровневый и позволяет программисту легко войти в разработку игры. В отличие от других движков, где используется язык *C*++, в *C*# есть много элементов и приемов, которые уже реализованы, и программисту нужно только воспользоваться ими;
* кроссплатформенность. Один и тот же код, написанный на движке *Unity*, с минимальными изменениями может быть перенесен на различные платформы (*PC*, *Mac*, *Android*, *iOS*, *Web*, игровые консоли). Это огромный плюс, который сокращает время на разработку игры в несколько раз;
* хорошее *Community*. У различных функций движка есть четкое описание с примерами на сайте разработчика, обратиться к которому можно в любой момент. Если что-то все же осталось непонятным, служба поддержки обязательно ответит на возникший вопрос;
* *Asset* *Store*, где имеется огромное количество различных плагинов и ресурсов для создания игры. Разумеется, какие-то из них бесплатные, какие-то платные, но все они собраны в одном месте с удобным поиском и возможностью загрузить, интегрировать и получить сразу рабочий функционал;
* *Unity* помогает сократить время разработки игры или приложения в несколько раз [7, с.54].

Таким образом, игровой движок *Unity* хорошо подходит для разработки приложения виртуальной реальности.

Движок является кроссплатформенным, проекты, созданные с его помощью, могут быть импортированы на все актуальные игровые платформы. В данном движке хорошо реализованы инструменты поддержки и механизм обновления проекта, что важно при разработке клиентской части *GaaS* *MMO* игр, поддерживается связь клиента с сервером с помощью простого инструмента создания выделенного сервера для 64 одновременных пользователей, однако это ограничение можно обойти использованием сторонних плагинов таких как *Open* *World* *Server*, а также всегда можно создать своё решение для сервера [9]. Общий вид пользовательского интерфейса игрового движка *Unity* приведён на рисунке 1.4.

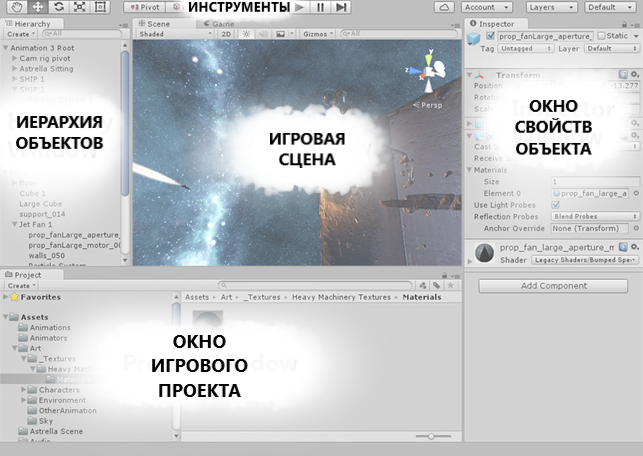


Рисунок 1.4 – Интерфейс пользователя в игровом движке *Unity*

Одно из преимуществ *Unity* – это список поддерживаемых платформ, где может запускаться приложения. *Unity* работает почти везде – на ПК (все операционные системы), на *Android*, на *iOS*, на *SmartTV*, в браузере, на различных экзотических системах – например, *Tizen* *OS*. Написание скриптов для игровых объектов происходит в *MonoDevelop*, либо при использовании стороннего редактора. Многие используют *Visual* *Studio*. Некоторые настраивают для этих целей *Sublime* *Text*.

Большое преимущество *Unity* – это это компоненты, представляющие собой готовое решение, созданное иными разработчиками (ассеты). Все в игре, включая код, картинки, представляется ассетами. Ассеты можно экспортировать, импортировать. Таким образом, сторонние разработчики могут делать целые заготовки для игр. Все, что останется – это заменить картинки, подправить скрипты – и игра становится уникальной. Нюансом является то, что различные ассеты могут быть несовместимы между собой как в прямом смысле, так и не подходить по стилю. Также есть специальный онлайн-магазин – *Unity* *Asset* *Store*. Там продаются готовые ассеты от сторонних разработчиков. Любой желающий может сделать свой ассет, и выложить его в продажу в этом магазине. Некоторые люди сделали на этом целый бизнес, благодаря большому рынку *Unity*-пользователей. Также важный момент, что магазин доступен прямо из редактора *Unity*. То есть, добавление новых ассетов максимальна упрощается. Можно зайти в магазин, выбрать нужный ассет, и он сразу загрузится и добавится в текущий проект. Быстро и удобно.

***1.3.4*** *Unreal* *Engine* 4 – популярный игровых движок, разработанный компанией *Epic* *Games*. Несмотря на наличие у *Unreal* *Engine* ряда преимуществ, таких как большой библиотекой ассетов, хорошей задокументированности и поддерживаемости, активного сообщества, выбор был сделан в пользу игрового движка *Unity*, поскольку он хорошо подходит для разработки приложений виртуальной реальности на языке программирования *C*#, где много элементов и приемов, которые уже реализованы, и программисту нужно только воспользоваться ими. Хотя движок плохо себя показывает падением *FPS* в масштабных сценах с множеством компонентов и одновременно присутствующих игроков занятием много места на жестком диске пользователя, это не мешает *Unity* являться одним из наиболее популярных игровых движков.

Доступ ко всему исходному коду *C*++ для движка – одна из причин, по которым *Unreal* *Engine* 4 настолько гибок. Код можно скачать с *GitHub*, любой желающий может настроить его под себя, и это совершенно бесплатно. Для сравнения, *Unity* предоставляет только профессиональным и корпоративным клиентам доступ к своему исходному коду – и это стоит дорого. Разработка и выпуск игры, сделанной с помощью *Unreal* *Engine* 4, полностью бесплатные. Плата в размере пяти процентов выплачивается *Epic* *Games* только тогда, когда игра приносит доход в размере трёх тысяч долларов в квартал.

*Unreal* *Engine* 4 упрощает разработку игр для нетехнических пользователей с помощью *Blueprint*. *Blueprint* демонстрирует, что движок является отличным инструментом для конструирования игровых прототипов. В *Blueprint* перетаскиваются узлы и добавляются связи, чтобы соединить их и добавить логику. Этот узловой интерфейс позволяет дизайнерам, не имеющим технической подготовки, исследовать свои идеи без необходимости знать, как программировать.

*Unreal* *Engine* 4 имеет широкий набор инструментов и отличный редактор. Простота использования движка демонстрируется широким спектром функций и инструментов в интуитивно понятном редакторе с дружелюбным интерфейсом, доступном для многих настроек. Широкий спектр инструментов предоставляет решения практически для каждой макрозадачи, котоую необходимо решить: макет сцены, включая ландшафт и листву, кинематографию, визуальные сценарии, систему частиц, анимационную сеть, создание материалов, световое отображение, управление движением, уровень детализации, физику, основные модели искусственного интеллекта, инструменты отслеживания производительности и памяти для оптимизации [8, с. 48].

Пользовательский интерфейс *Unreal* *Engine* 4 представлен на рисунке 1.4.

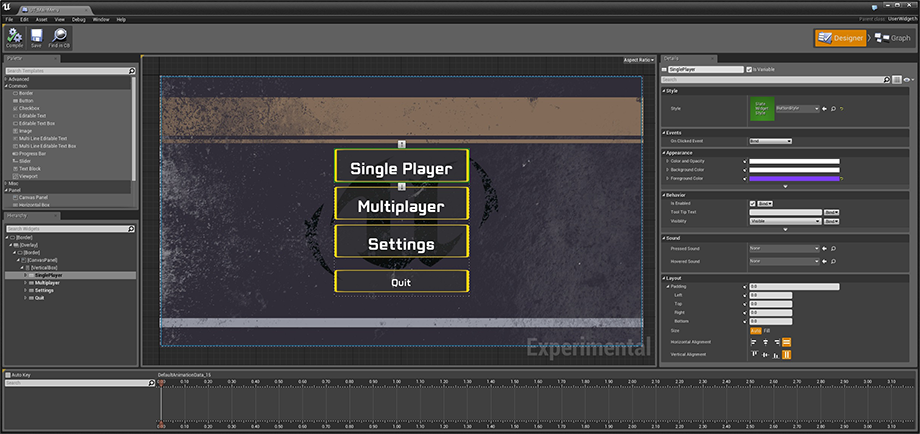


Рисунок 1.5 – Пользовательский интерфейс игрового движка *Unreal* *Engine* 4

Движку *Unreal* *Engine* 4 сложно бороться с большим количеством данных. В момент проектирования больших игровых уровней способность хорошо их редактировать в редакторе начинает страдать, а во время отладки возникают проблемы с производительностью, которые трудно решить. *Unreal* *Engine* 4 не предназначен для *RPG* игр, визуализации и управления огромным количеством оружия, брони, расходных материалов, разговоров и так далее. *Unreal* *Engine* 4 нуждается в точной настройке для удовлетворения нестандартных потребностей разработчика, а, чтобы правильно настроить движок, потребуется много усилий. К тому же, это сложно будет сделать без технического образования. В то время как *Unreal* *Engine* 4 обслуживает менее опытных разработчиков благодаря технологии *Blueprint*, некоторые аспекты движка остаются трудными для понимания пользователям, у которых отсутствует хорошая техническая подготовка.

Наличие серьёзного опыта программирования на *С*++ поможет с правильной настройкой окружения, но даже с этим багажом знаний *Unreal* *Engine* 4 имеет довольно крутую кривую обучения [9].

## 1.4 Сравнение игровых движков *Unity* и *Unreal* *Engine* 4

Чтобы сделать выбор оптимального игрового движка, подходящего для разработки приложения виртуальной реальности, необходимо рассмотреть преимущества и недостатки рассмотренных выше игровых движков.

К преимуществам игрового движка *Unity* можно отнести следующее:

* кроссплатформенный;
* позволяет быстро разрабатывать игры за счёт готовых шаблонов (ассетов);
* содержит встроенный магазин ассетов;
* хорошо подходит для создания небольших проектов;
* имеет низкий порог вхождения;
* реализует конструктивное создание объектов, что значительно упрощает разработку;
* наличие большого сообщества пользователей, видеоуроков и документации;
* имеет дружелюбный интерфейс;
* мультиязычность;
* является условно бесплатным.

Недостатки игрового движка *Unity*:

* закрытый исходный код;
* плохо оптимизирован для разработки больших игровых проектов с открытым миром.

К плюсам игрового движка *Unreal* *Engine* 4 можно отнести:

* возможность создавать фотореалистичные проекты;
* кроссплатформенная разработка;
* открытый исходный код;
* бесплатный;
* наличие технологии конструктивной разработки игр *Blueprint*;
* дружелюбный интерфейс;
* наличие большого сообщества пользователей.

Помимо многочисленных плюсов *Unreal* *Engine* 4 также имеет ряд серьёзных недостатков:

* проседание в производительности при большой нагрузке данными;
* не русифицированный язык пользовательского интерфейса;
* наличие проблем с отладкой больших игр;
* высокий порог вхождения;
* движок имеет избыточное число инструментов;
* сложная индивидуальная настройка редактора и функций движка.

Чтобы упростить выбор программного средства для разработки игрового приложения составим таблицу, в которой сравниваются основные возможности рассмотренных игровых движков. В таблице 1.1 представлены основные возможности игровых движков *Unreal Engine* 4 и *Unity*.

Таблица 1.1 – Основные возможности игровых движков

|  |  |  |
| --- | --- | --- |
| Возможность | *Unreal* *Engine* | *Unity* |
| 1 | 2 | 3 |
| Язык программирования | *C*++ | *C*# и *JS* |
| Кроссплатформенность | Есть | Есть |
| Поддержка трехмерной графики | Есть | Есть |
| Поддержка двухмерной графики | Нет | Есть |
| Поддержка шейдеров | Есть | Есть |
| Визуальное программирование | Есть | Есть |
| Система создания анимации | Есть | Есть |
| Объектно-ориентированный подход | Есть | Есть |
| Ежемесячная плата | Нет | Нет |
| Процент с прибыли игры | Есть | Нет |

Из перечисленных положительных и отрицательных сторон рассматриваемых игровых движков можно сделать вывод, что для разработки простой некоммерческой игры оптимальнее будет использовать движок *Unity*, так как он позволяет больше времени уделить проектированию игры и непосредственно её дизайну, нежели настройке движка и редактора под собственные нужды. К тому же, большинство инструментов *Unreal* *Engine* 4 могут быть излишними.

2 АРХИТЕКТУРА VR-ПРИЛОЖЕНИЯ ДЛЯ МОДЕЛИРОВАНИЯ

АВАРИЙНЫХ СИТУАЦИЙ

## 2.1 Требования к *VR*-приложению для моделирования аварийных

## ситуаций

Разрабатываемое приложение виртуальной реальности предназначено для моделирования аварийных ситуаций газового бойлера заданной модели, данное приложение позволит наглядно представлять газовое оборудование, изучать его комплектующие, а также моделировать некоторые аварийные и нештатные ситуации.

Основные характеристики разрабатываемого *VR*-приложения:

* газовый бойлер, комплектующие которого спроектированы в специализированном конечноэлементном пакете и импортированы в игровую сцену;
* газовый бойлер должен находиться в закрытом помещении, имеющем интерьер под кухню;
* в приложении должны быть реализованы основные аварийные ситуации с их анимацией, а также выводом информации о способах их устранения;
* возможность «подержать в руках» комплектующие газового бойлера.

взрыв-схема газового бойлера приведена на рисунке 2.1.

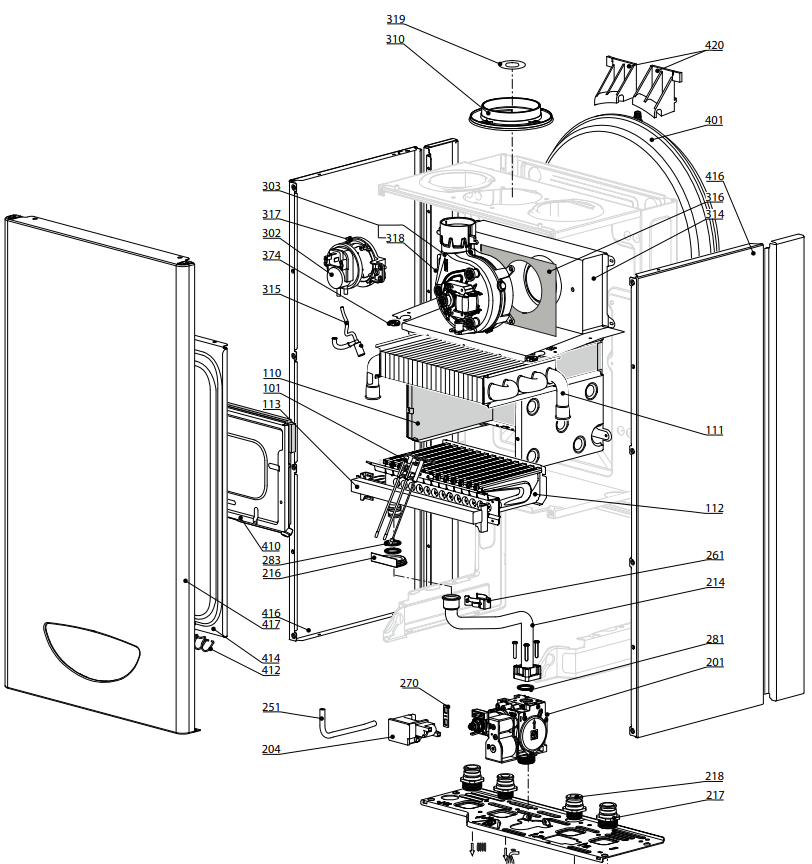


Рисунок 2.1 – Взрыв-схема газового бойлера

## 2.2 Основные этапы разработки игрового *VR* приложения

Разработка приложения виртуальной реальности для моделирования аварийных ситуаций можно разделить на несколько этапов:

* разработка модели газового бойлера в специализированном программном пакете;
* подготовка к импорту полученных модулей в сцену игрового движка *Unity* через программу-посредник, в качестве которой выбрана *Blender* 3*D*. В этой программе исправляются ошибки и недочёты, допущенные при проектировании деталей в основной программе, а также сохраняются в формате, поддерживаемом *Unity*;
* подготовка игровой сцены, создание всех необходимых объектов и импорт полученных деталей газового бойлера;
* реализация основных аварийных ситуаций для газового бойлера.

Разрабатываемое игровое приложение является результатом взаимодействия игрового движка и средств, с которыми оно взаимодействует (система ввода, графика, физика и так далее).

Архитектура разрабатываемого игрового приложения в обобщённом виде приведена на рисунке 2.2.

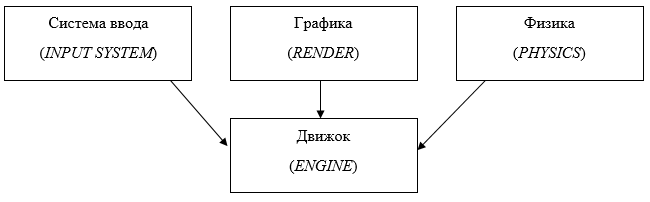


Рисунок 2.2 – Обобщённая архитектура разрабатываемого приложения

Таким образом, любую игру можно представить, как результат слияния двух компонентов: игрового движка и контента, с которым он взаимодействует (графика, музыка, текст и так далее).

Из перечисленных компонентов важнейший, безусловно – игровой движок. В данной дипломной работе был выбран движок *Unity* (выбор данного движка обоснован в разделе 1.4 данной дипломной работы). Физика представляет из себя наложение текстур, а также физических барьеров на объекты игровой сцены.

## 2.3 Разработка модели газового бойлера

***2.3.1*** При разработке модели котла в специализированном пакете конечноэлементного расчёта *Solidworks* 2021, бойлер был разделён на несколько ключевых составляющих частей:

* теплообменника;
* газовой горелки;
* теплоизоляционной камеры, соединяющей горелку и теплообменник;
* труба вентури с камерой сгорания;
* расширительного бака;
* нижней пластины, в которую входит газовый клапан с блоком розжига, также от пластины отходят трубки, подающие на вход холодную воду и на выходе выводящие горячую;
* внешней обшивки газового бойлера, включающая в себя две боковые панели и одну фронтальную.

В *Solidworks* есть возможность сохранения сборок как единой детали, что позволяет сделать каждую сборку цельной. что значительно облегчит с ней работу в игровом движке *Unity*.

При проектировании теплообменника разрабатывались три основные детали: металлическая изогнутая труба, где проходит вода, плоские пластины, которые нагреваются при работе горелки и пластины, закрывающие теплообменник.

В Приложении А показаны все крупные детали сборки газового бойлера.

Изогнутая труба теплообменника было нарисована при помощи линии в режиме 3*D* эскиза (3*D* *Sketch*). После при помощи команды развернуть основу (*Swept* *base*), и задания параметров была получена труба теплообменника. На рисунке А.1 приведено изображение изогнутой трубы теплообменника. Для создания плоской пластины, нагревающей воду при помощи горелки, были использованы базовые возможности режима эскиза в *Solidworks*, толщина в 2 мм, достигнута при помощи команды «Вытянуть бобышку» (*Extruded* *base*). При создании готовой сборки теплообменника тридцать пластин были созданы функцией линейный массив (*Linear* *pattern*), где при помощи ввода параметров можно выбрать направление, по которому необходимо создать одинаковые детали, расстояние между ними и их количество. Боковые и задняя пластины, закрывающая теплообменник сделаны аналогично, как и плоские пластины теплообменника. На рисунке А.2 приведено изображение одной из пластин теплообменника. На рисунке А.3 показана внешняя пластина теплообменника.

После создания всех деталей полученный теплообменник был сохранён двумя файлами: файлом сборки для взаимодействия с другими деталями бойлера и как цельная деталь для будущей работы в *Unity*. Также сборка теплообменника была сохранена как единая деталь для того, чтобы с ней было проще работать в игровой сцене *Unity*.

На рисунке 2.1 показана созданная в программе *Solidworks* сборка теплоо бменника газового бойлера.

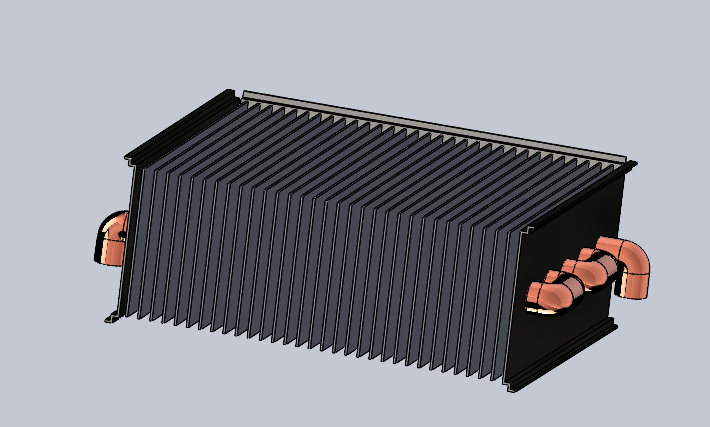


Рисунок 2.3 – Сборка теплообменника в *Solidworks*

Сборка газовой горелки состоит из газового коллектора, который подаёт горячий воздух и горелки. Газовая горелка подаёт горячий воздух в теплоизоляционную камеру, где происходит нагрев пластин, с помощью которых нагревается вода в трубках, находящаяся в теплообменнике газового бойлера. Одна из деревянных пластин. На рисунке А.4 показан один из воздуховодов газовой горелки. На рисунке А.5 приведён газовый коллектор бойлера.

Газовая горелка состоит из изогнутых воздуховодов, полученных в режиме 3*D* эскиза (3*D* *Sketch*). После при помощи команды развернуть основу (*Swept* *base*), и задания параметров была получена труба горелки. При помощи команды линейный массив (*Linear* *pattern*) было получено десять воздуховодов горелки. На вершинах изогнутых труб имеются продольные вырезы, через которые горячий воздух поступает в теплоизоляционную камеру. Газовый коллектор представляет из себя запаянную на обоих концах балку квадратного сечения из конструкционной стали с патрубком, в который подаётся газ от газового клапана.

Теплоизоляционная камера представляет из себя четыре деревянные пластины, снаружи обшитые деревом. Боковые пластины отличаются по размеру от фронтальной и задней. При помощи камеры теплоизолирования соединяются друг с другом газовая горелка и теплообменник. Таким образом происходит передача горячего воздуха от горелки на теплообменник, что позволяет нагреть поданную на вход холодную воду, и подать на выход горячую. Одна из четырёх деревянных пластин теплоизоляционной камеры приведена на рисунке Д.6.

На рисунке 2.2 показана созданная в программном продукте *Solidworks* общая сборка теплообменника, газовой горелки и теплоизоляционной камеры. Для лучшего воспроизведения в теплоизоляционной камере убрана фронтальная панель, что позволяет лучше видеть внутреннюю часть газового бойлера.

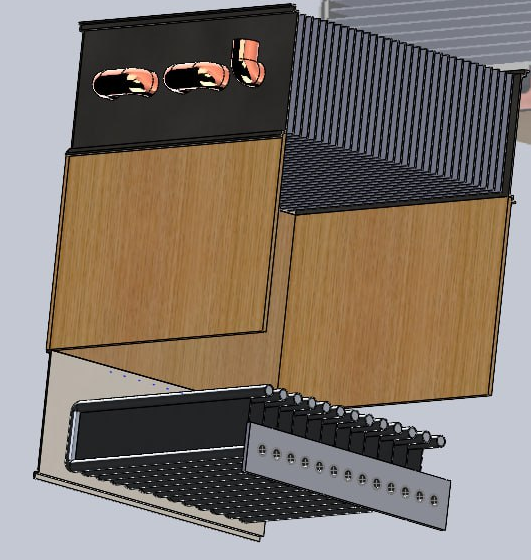


Рисунок 2.2 – Сборка теплообменника, горелки и теплоизоляционной камеры

Камера сгорания предназначена для накопления остатков горения, и предоставляет из себя коробку форму параллелепипеда, соединённую с теплообменником, через который проходят горячий воздух с продуктами горения. Труба вентури, в свою очередь, служит для вывода летучих токсичных соединений из газового бойлера. Камера сгорания, соединённая с трубой вентури закреплена на металлической пластине, которая покрывает теплообменник.

Камера горения создана при помощи создания обычного прямоугольника, к которому применена команда «Вытянуть бобышку» (*Extruded* *base*), после чего к полученной форме была использована функция «Оболочка» (*Shell*), благодаря чему камера стала полой и было сделано отверстие для закрепления трубы вентури. На рисунке А.7 приведена деталь газового бойлера – камера сгорания. Труба вентури была создана при помощи инструмента «Вытянутая бобышка», после чего были скруглены все прямые углы, и в результате к полученной форме была применена функция «Оболочка». Пластина, накрывающая теплообменник, и на которой находится камера сгорания с трубой вентури получена вытягиванием прямоугольника. Труба вентури приведена на рисунке А.8. На рисунке А.9 показана пластина, разделяющая теплообменник и камеру сгорания.

На рисунке 2.3 показана полученная сборка трубы вентури, камеры сгорания и металлической пластины.



Рисунок 2.3 – Сборка камеры сгорания и трубы вентури

Расширительный бак предназначен для накопления воды в бойлере и представляет из себя резервуар цилиндрической формы со сглаженными гранями и отверстия. Расширительный бак получен вытягиванием цилиндра, после чего к полученной форме была использована функция «Оболочка» (*Shell*), благодаря чему бак стал полым. На рисунке 2.4 приведён расширительный бак.

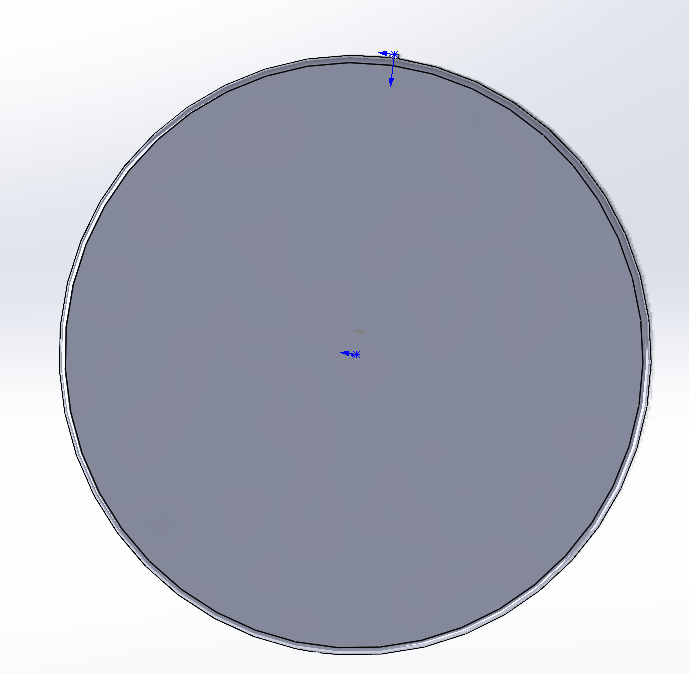


Рисунок 2.4 – Расширительный бак

Пластина, через которую проходят водоводы, предназначенные для подачи на вход холодной воды и горячей – на выход. Также в пластину встроены газовый клапан и блок розжига, при помощи которых в котле зажигается пламя, от которого происходит водонагрев.

Нижняя пластина получена вытягиванием прямоугольника, у которого были скруглены углы, а на самой пластине были сделаны отверстия под водоводы и газовый клапан с блоком розжига. Газовый клапан и блок розжига получены методом вытягивания эскизов на плоскости, далее у них были скруглены все грани углы и после была применена функция о «Оболочка», и рассматриваемые объекты стали полыми. Деталь нижней пластины приведена на рисунке А.10. На рисунке А.11 показан один из водоводов. Газовый клапан приведён на рисунке А.12. На рисунке А.13 показан блок розжига бойлера.

Внешняя обшивка газового бойлера представляет из себя три пластины: две боковые и одна фронтальная. Боковые пластины являются п-образными сплошными стальными листами, а фронтальная – сплошной стальной лист с вырезом под экран, выводящий состояния газового бойлера. На рисунке А.14 приведена одна из двух боковых панелей газового бойлера, а на рисунке А.15 – фронтальная панель котла. На рисунке Д.16 показана верхняя крышка газового бойлера.

На рисунке 2.5 приведён газовый бойлер в полной сборке.

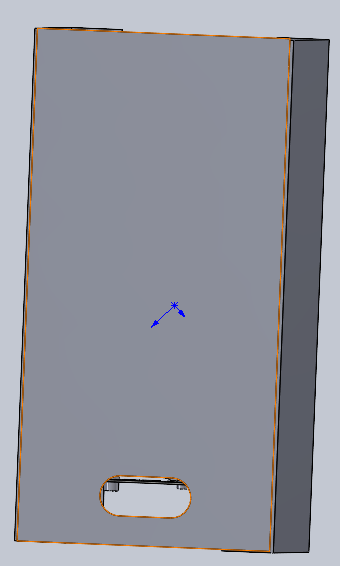


Рисунок 2.5 – Полная сборка газового котла

***2.3.2*** Программный продукт *Вlender* 3*D* использовался для импорта деталей газового бойлера в формате .*STL* для их последующего экспорта в формат .*FBX*, который поддерживается *Unity*.

Кроме повторного сохранения деталей, в данной программе исправлялись некоторые недочёты, допущенные при проектировании газового бойлера в *Solidworks*:

* неправильно выбранная проекция детали, вследствие чего с ней весьма проблемно работать в *Unity*;
* смещённая точка опоры. Данная проблема очень актуальна, если необходимо реализовать возможность для пользователя «взять её в руки».
* изменить масштаб как для отдельно взятой детали, так и для всего котла в целом.

На рисунке 2.6 показана деталь камеры сгорания в программе *Вlender* 3*D*.

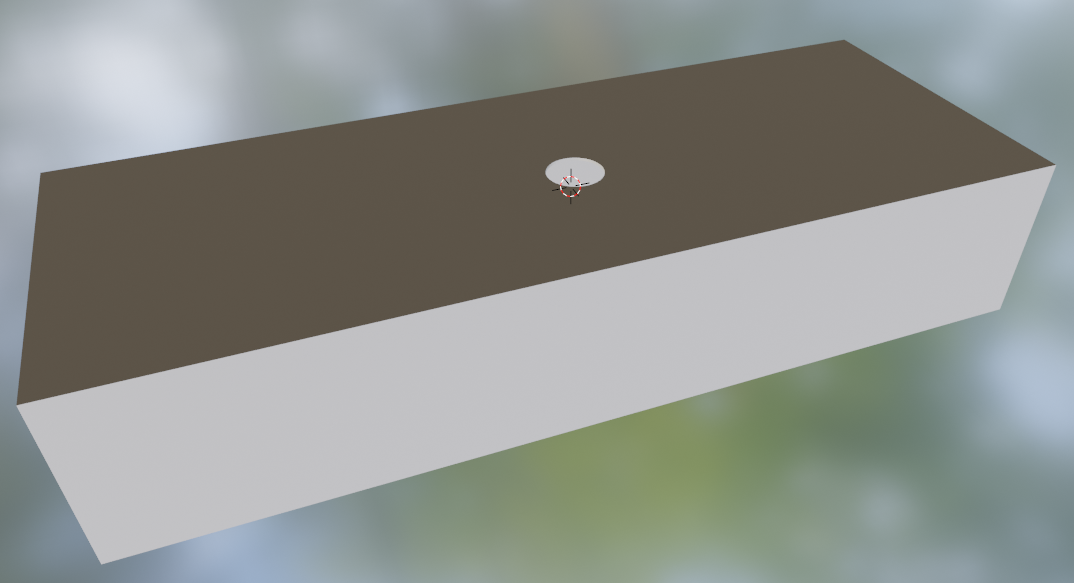


Рисунок 2.6 – Камера сгорания в *Вlender* 3*D*

Кроме сохранения детали в формат .*FBX*, который поддерживает игровой движок *Unity*, в программе *Вlender* 3*D* выполнялось накладываение текстур на детали и сборки, придание им металлического блеска в случае надобности.

## 2.4 Архитектура игрового приложения в *Unity*

Перед началом работы над приложением виртуальной реальности на игровом движке *Unity* 2021.3.21*f*21, на компьютер, на котором ведётся разработка необходимо установить программное обеспечение для поддержки работы очков виртуальной реальности *Oculus*.

После установки программного обеспечения для работы с очками виртуальной реальности, необходимо создать проект в *Unity*, выбрав 3*D*. После создания проекта необходимо выполнить следующие настройки:

– в свойствах сборки (*File* – *Build* *Settings*) задать компрессию текстур в *ASTC*;

– нажать *Switch* *Platform*;

– в *Edit* – Project Settings выбрать *XR* *Plugin* *Management* и там указать *Windows Mixed Reality* (так подключается поддержка для *VR*).

Проект приложения виртуальной реальности состоит из двух основных модулей: *Assets* и *Packeges*. В папке *Assets* хранятся следующие модульные единицы проекта:

* *ExampleAssets* имеет в себе основные используемые в проекте материалы и текстуры;
* в *Scripts* хранятся основные программные классы, обеспечивающие работу игрового приложения;
* *Scenes* хранит в себе игровую сцену, в которой реализовано моделирование аварийных ситуаций газового бойлера.

Как видно, в модуле *Assets* хранятся основные компоненты, которые непосредственно использовались при разработке игрового приложения.

Модуль *Packages* хранит в себе все сторонние пакеты и библиотеки, необходимы для разработки и эксплуатации приложения для моделирования аварийных ситуаций, в том числе, и настройки для среды программирования *Microsoft* *Visual* *Studio* 2022, в которой пишутся программные модули для исполнения приложения.

На рисунке 2.7 приведена обобщённая архитектура *VR* приложения.



Рисунок 2.7 – Общая архитектура приложения в игровом движке *Unity*

Главным преимуществом данной модульной концепции является свобода выбора инструментов для разработчиков вторичных модулей. Основные преимущества такой разработки следующие:

* разработчику вторичных модулей не требуется вникать во все особенности приложения, что снижает нагрузку на всех разработчиков;
* модульность позволяет с легкостью: отключать какой-либо модуль, менять их очередность или запускать в какой угодно очередности;
* удобство при тестировании проекта;
* есть возможность переиспользовать вторичные модули в следующих проектах клонах. За счет того, что они изолированы;
* благодаря такой архитектуре проект разделен на несколько слабо связанные модули, где точка связи должна быть одной, или, вовсе только единственной.

Листинг ключевых классов разработанного приложения виртуальной реальности приведён в приложении Б.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ *VR* ПРИЛОЖЕНИЯ ДЛЯ

МОДЕЛИРОВАНИЯ АВАРИЙНЫХ СИТУАЦИЙ ГАЗОВОГО

БОЙЛЕРА

## 3.1 Общий вид разработанного приложения

При запуске приложения виртуальной реальности, пользователю, при условии подключения к ПК соответствующего гарнитура *VR* (шлем, контроллеры и другие при их наличии) пользователь появляется в закрытом помещении. Общий вид приложения в момент запуска приведено на рисунке 3.1.

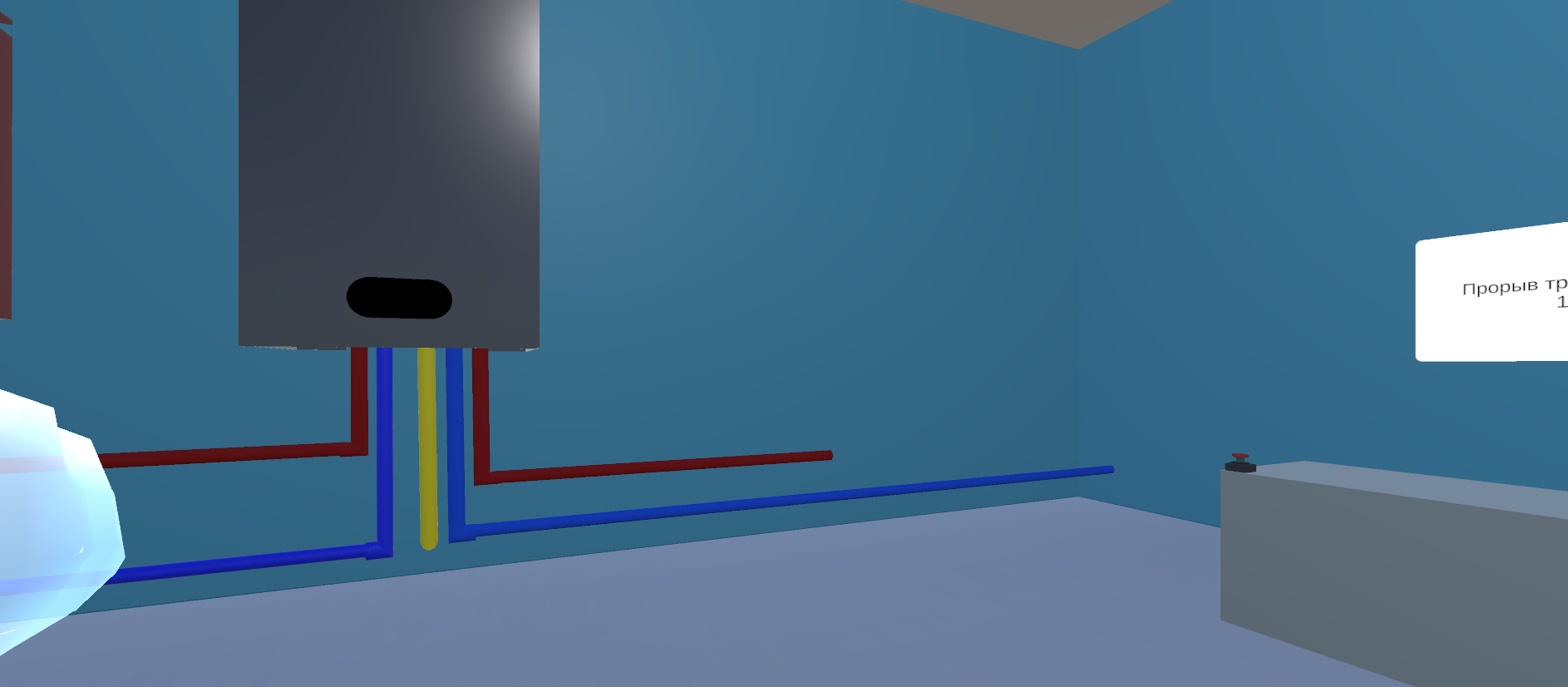


Рисунок 3.1 – Запуск игрового приложения

В приложении пользователь имеет следующие возможности:

* перемещаться по всей площади помещения;
* визуально осматривать газовый бойлер и его интеграцию в интерьер помещения;
* при нажатии на специальную кнопку, обозначенную соответствующей надписью, получить детальный вид бойлера со всеми его внутренними деталями вследствие его раздвижения;
* изучать аварийные ситуации газового бойлера;
* рассмотреть и «подержать в руках» некоторые детали газового бойлера;
* изучить некоторые из приведённых аварийных ситуаций. Для реализации каждой отдельной аварийной ситуации, необходимо нажать на соответствующую кнопку на столе, после нажатия на котле произойдёт анимация данной ситуации, а на текстовом поле возле стола с кнопкой можно прочитать код ошибки, описание и возможные способы её устранения.

По техническому заданию, необходимо реализовать в комнате, где располагается газовый бойлер, интерьер кухни. Для реализации интерьера необходимы модели кухонных шкафов, радиаторов отопление и прочих дизайнерских решений. На рисунке 3.2 приведена модель радиатора отопления.



Рисунок 3.2 – Модель радиатора отопления

Также для интерьера кухни необходимы кухонные шкафы. Один из кухонных шкафов приведён на рисунке 3.3.

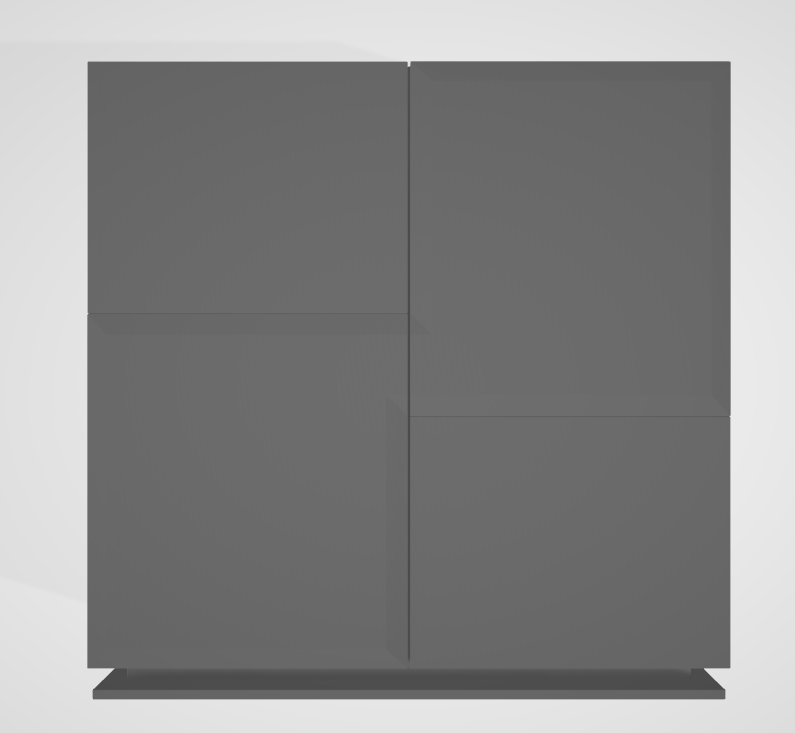


Рисунок 3.3 – Модель кухонного шкафа

Газовый бойлер со всеми его комплектующими был импортирован в игровую сцену. Пользователь может его изучать визуально. Вид газового бойлера в игровой сцене приведён на рисунке 3.4.

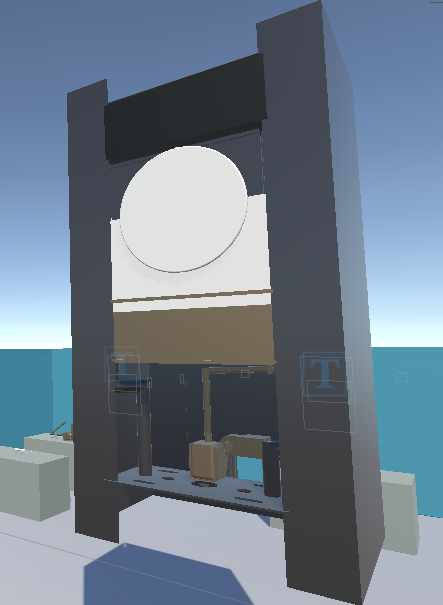


Рисунок 3.4 – Газовый бойлер в игровой сцене

В игровом приложении пользователь может изучить газовый бойлер в детальном виде. Для этого необходимо нажать соответствующую кнопку. Модель игровой кнопки приведена на рисунке 3.5.



Рисунок 3.5 – Модель кнопки

Основные возможности, которые пользователь может совершать в игровом приложении приведены на рисунке 3.6.



Рисунок 3.6 – Диаграмма вариантов использования

Таким образом, в приложении реализовано интерактивное взаимодействие с элементами игровой сцены и деталями газового бойлера. Пользователь может получать информацию о произошедших нештатных ситуациях, а также о способах их возможного устранения.

## 3.2 Структура игрового приложения

Игровой объект *XR* *Origin* использует компоненты: *XROrigin*.*cs*, *InputActionManager*.*cs*, *LocomotionSystem*.*cs*, *ContinшuousMoveProvider*.*cs*, *Сharacter* *Сontroller*, *СontinuousTurnProvider*.*cs*, *CharacterControllerDriver*.*cs*, *TeleportationProvider*.*cs*.

Компонент *XROrigin*.cs – это корневой объект, на котором все компоненты, отвечающие за управление движением пользователя в виртуальном пространстве с помощью гарнитуры *VR*, который является частью *XR* Interaction *Toolkit*.

*InputActionManager*.*cs* – это компонент, который позволяет определять пользовательские действия в игре, такие как движения, стрельба, смена оружия и другие возможности. Данный компонент является частью *Input* System, которая является новой системой управления вводом данных в *Unity* и предоставляет гибкую настройку пользовательского ввода.

*LocomotionSystem*.*cs* – это компонент, который обрабатывает все запросы на перемещение игрока в виртуальном пространстве. Рассматриваемый компонент является частью *XR* *Interaction* *Toolkit* и предоставляет различные способы перемещения, такие как телепортация и непрерывное перемещение.

*ContinuousMoveProvider*.*cs* – это компонент, который позволяет игроку двигаться в виртуальном пространстве непрерывно, используя физические движения игрового контроллера или другого устройства управления. Компонент *ContinuousMoveProvider*.*cs* является частью *XR* *Interaction* *Toolkit*.

*Character* *Controller* – это компонент, который обрабатывает движение персонажа и коллизии в игре. Указанный компонент позволяет игроку перемещаться в игре, при этом взаимодействуя с окружающим миром и другими объектами в нем.

*ContinuousTurnProvider*.*cs* – это компонент, который позволяет игроку поворачиваться в игре непрерывно, используя физические движения игрового контроллера или другого устройства управления. Он является частью *XR* *Interaction* *Toolkit*.

*CharacterControllerDriver*.cs – это компонент, который обрабатывает данные о перемещении игрока и передает их в *Character* *Controller*, чтобы игрок мог свободно перемещаться в игре. Компонент является частью *XR* *Interaction* *Toolkit*.

*TeleportationProvider*.*cs* – это компонент, который предоставляет способ телепортации игрока в игровом мире. Как и *Character* *Controller* компонент *TeleportationProvider* является частью *XR* *Interaction* *Toolkit*.

Дочерний объект *Camera* *Offset* объекта *XR* *Origin* – объект, который отвечает за позиционирование камеры в игре. Содержит дочерние объекты, такие как *Main* *Camera*, *Left* *Hand*, *Right* *Hand*, *Left* *Teleportation* *Ray*, *Right* *Teleportation* *Ray*, *Left* *Grab* *Ray*, *Right* *Grab* *Ray*.

Дочерний объект *Main* *Camera* имеет компонент *Tracked* *Pocket* *Driver* (*Input* *System*), который позволяет отслеживать положение и вращение камеры в пространстве *XR*. Это необходимо для корректной работы механики перемещения игрока, так как игрок должен перемещаться в пространстве относительно камеры.

Дочерние объекты *Left* *Hand* и *Right* *Hand* содержат компоненты *XRController*.*cs* и *XRDirectInteractor*.*cs*. Эти компоненты позволяют игроку взаимодействовать с объектами в игровом мире, используя контроллеры виртуальной реальности. *XRController*.*cs* отвечает за обработку входящих сигналов от контроллеров, а *XRDirectInteractor*.*cs* позволяет объектам в игровом мире реагировать на взаимодействие игрока.

Дочерние объекты *Left* *Hand* и *Right* *Hand* содержат такие дочерние объекты, как *Left* *Hand* *Poke* *Interactor* и *Right* *Hand* *Poke* *Interactor* являются частью реализации взаимодействия пользователя с объектами в игровом мире и используются для реализации функции толчка. Оба объекта содержат компонент *XRPokeInteractor*.*cs*, который отвечает за взаимодействие с объектами в игровом мире.

*XRPokeInteractor*.*cs* обеспечивает возможность толчка по объектам в игровом мире. При использовании этого компонента игрок может взаимодействовать с объектами, нанося удары и давая таким образом отклик. В результате, это создает ощущение физического взаимодействия игрока с объектами в игровом мире.

Дочерние объекты *Left* *Teleportation* *Ray* и *Right* *Teleportation* *Ray* с компонентами *XRController*.*cs* и *XRRayInteractor*.*cs*. Эти компоненты используются для реализации механики телепортации в игре. *XRRayInteractor*.*cs* отвечает за определение объектов, с которыми взаимодействует луч телепортации, а *XRController*.*cs* – за обработку пользовательских действий, связанных с телепортацией.

Дочерние объекты *Left* *Grab* *Ray* и *Right* *Grab* *Ray* также используют компоненты *XRController*.*cs* и *XRRayInteractor*.*cs* для реализации механики захвата и перемещения объектов в игре. Рассматриваемые компоненты позволяют игроку брать и перемещать объекты в виртуальном мире, что делает взаимодействие с игровым миром более увлекательным и реалистичным.

Дочерние объекты *Camera* *Offset* позволяет пользователю взаимодействовать с игровым миром и получать полноценный опыт виртуальной реальности. Благодаря использованию компонентов *XR* *Integration* *Toolkit*, взаимодействие игрока с объектами в игровом мире становится более естественным и интуитивным.

Для управления взаимодействия между пользователем и объектами в виртуальной реальности, используется объект *XR* *Interaction* *Manager* c компонентом *XRIneractionManager*.*cs*.

С помощью *XR* *Interaction* *Manager* можно настроить различные типы взаимодействия, такие как телепортация, захват объектов, выбор объектов, а также определить, какие контроллеры должны использоваться для каждого типа взаимодействия.

Кроме того, *XR* *Interaction* *Manager* позволяет настраивать различные параметры взаимодействия, такие как расстояние, на котором игрок может захватывать объекты, скорость перемещения при телепортации и другие функции.

*XR* *Interaction* *Manager* также поддерживает взаимодействие с объектами с разными типами геометрии, такими как конечноэлементные сетки и задание жёстких рамок, которые не позволяют игроку проходить сквозь эти объекты, что делает взаимодействие более гибким и универсальным.

XR *Interaction* *Manager* является важным компонентом для реализации механики взаимодействия в *XGR* приложениях. Он облегчает процесс настройки и управления взаимодействием между пользователем и объектами в виртуальном мире.

*MonoBehaviour* – это базовый класс, от которого наследуются все скрипты. При реализации игровых скриптов на языке программирования *C*#, необходимо явно наследовать от *MonoBehaviour*.

Класс *ListObjectTransform*.*cs* реализует в себе плавное раздвижение газового бойлера, которое выполняется по нажатию на красную кнопку.

За возможность перемещаться из мгновенно из одного точки в другую, иными словами телепортация, отвечает класс *ActivateTeleportationRay*.*cs*.

Класс *AnimateHandOnInput*.*cs* отвечает за возможность «держать в руках» компоненты газового бойлера, а также за возможности переносить их в другое место и класть их на поверхности.

Диаграмма классов *ListObjectTransform*.*cs*, *ActivateTeleportationRay*.*cs*, *AnimateHandOnInput*.*cs* приведена на рисунке 3.7.

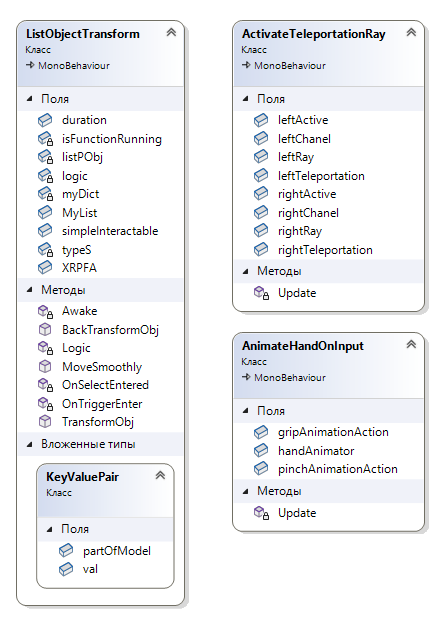


Рисунок 3.7 – Диаграмма классов *ListObjectTransform*.*cs*, *ActivateTeleportationRay*.*cs*, *AnimateHandOnInput*.*cs*

Во всех классах реализованы методы *Start* и *Update*. Метод *Start* вызывается до обновления первого кадра, только если скрипт включен. Для объектов, добавленных на сцену, функция *Start* будет вызываться во всех скриптах до функции *Update*. Метод *Update* вызывается раз за кадр. Это главная функция для обновлений кадров.

Класс *ActivateGrapRay.cs* отвечает за поворот камеры в шлеме виртуальной реальности влево и вправо в соответствии с направлением поворота головы пользователя.

Класс *XRPlatformManager*.*cs* отвечает за запуск игры и появление игрока в задонном месте игровой сцены.

Класс *XRGrabInteractableTwoAttach.cs* реализует представление двух игровых контроллеров в качестве левой и правой руки игрока, что добавляет большей реалистичности в приложение виртуальной реальности.

На рисунке 3.8 показана диаграмма классов *ActivateGrapRay.cs*, *XRPlatformManager*.*cs* и *XRGrabInteractableTwoAttach.cs*.

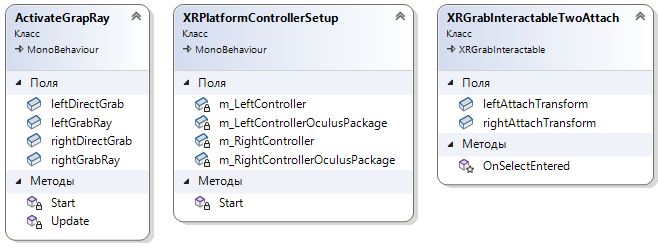


Рисунок 3.8 – Диаграмма классов *ActivateGrapRay.cs*, *XRPlatformManager*.*cs* и *XRGrabInteractableTwoAttach.cs*

Класс *GameManager*.*cs* подразумевает за собой запуск игры и вывод соответствующей игровой сцены, а также завершение пользователем игрового процесса.

В классе *GameMenuManager*.*cs* реализованы событие происходящие с пользователем во время работы в игровом приложении: нажатие на кнопки, взятие в руки деталей и комплектующих и другие.

Класс *FireBulletOnActivate*.*cs* реализует в себе методы окрашивание деталей, если во время исполнения игрового приложения «взять их в руки».

Все классы, реализующие нештатные ситуации, хранятся в каталоге *StandsButtons*.

На рисунке 3.9 приведена диаграмма классов *GameManager*.*cs*, *GameMenuManager*.*cs* и *FireBulletOnActivate*.*cs*.

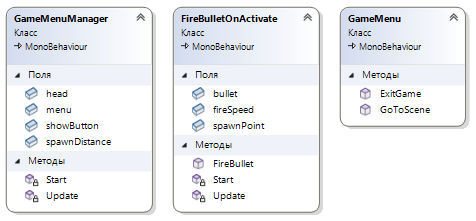


Рисунок 3.9 – Диаграмма классов *GameManager*.*cs*, *GameMenuManager*.*cs* и *FireBulletOnActivate*.*cs*

При проектировании классов в игровом приложении движка *Unity*, активно использовались библиотеки (ассеты), позволяющие хранить все игровые объекты со всеми компонентами и значениями свойств. Данный тип ассетов называется префабами (*prefabs*). Префабы в *Unity* – это предварительно настроенные многоразовые игровые объекты, которые разработчики создают на сцене и сохраняют в проекте для копирования и многоразового использования.

## 3.3 Перечень аварийных ситуаций газового бойлера

Неисправности в работе отопительного газового котла Аристон отображаются на дисплее устройства и имеют трёхзначный код: первая цифра указывает на важный узел системы, где выявлена ошибка. В газовом бойлере всего шесть узлов системы:

* контур отопления, который обозначается цифрой «1»;
* контур горячего водоснабжение, обозначающийся номером «2»;
* блок управления и электроника котла, обозначается номером «3»;
* внешние устройства дистанционного управления, которое обозначается номером «4»;
* газовая горелка обозначается номером «5»;
* система дымоудаления и вентиляции обозначается номером «6».

В газовом бойлере реализована ошибка «101», которая представляет из себя перегрев теплоносителя в первичном теплообменнике. При моделировании данной ошибки теплообменник будет выделен красным цветом.

Ошибка с номером «108» обозначает снижение давление в контуре. В данной нештатной ситуации можно восстановить давление открытием крана подпитки. Однако если такое повторяется неоднократно, скорее всего в системе есть утечка.

Ошибка «608» обозначает чрезвычайное дымоудаление. При вызове данной аварийной ситуации будет наблюдаться повышенное удаление дыма из системы вентиляции.

Ошибка «501» обозначает неисправность блока розжига или газового клапана. Ошибка представляет из себя отсутствие пламени в розжиге при попытке зажечь бойлер.

Ошибка номер «502» обозначается как неисправность электрода газового бойлера. В данной ситуации электрод подлежит замене.

На рисунке 3.10 показана одна из аварийных ситуаций – чрезвычайное дымоудаление из камеры сгорания.

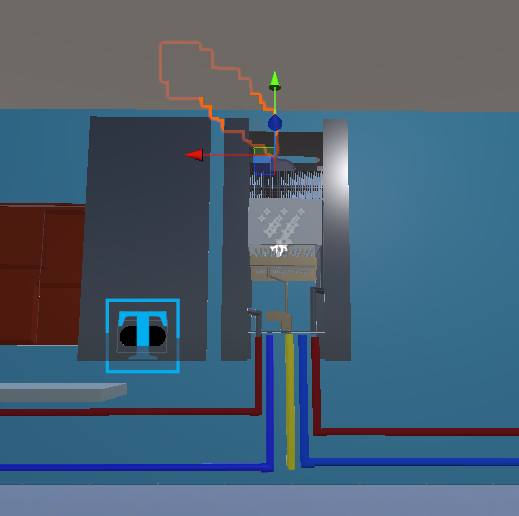


Рисунок 3.10 – Чрезвычайное дымоудоудаление из камеры сгорания

Таким образом, в модели в приложении реализованы ключевые аварийные ситуации газового бойлера, а расположенные панели с текстом дают информацию об нештатной ситуации и способах её преодоления.

4 РЕЗУЛЬТАТЫ ВЕРИФИКАЦИИ И ВАЛИДАЦИИ *VR*

ПРИЛОЖЕНИЯ ДЛЯ МОДЕЛИРОВАНИЯ АВАРИЙНЫХ

СИТАУЦИЙ ГАЗОВОГО БОЙЛЕРА

## 4.1 Требования к компьютеру для стабильной работы *VR*

## приложения

Для запуска приложения виртуальной реальности для моделирования аварийных ситуаций газового бойлера *Ariston* *cares* *xc*-*hs* необходимо наличие на ПК портала смешанной реальности (*Windows* *Mixed* *Reality*). *Windows* 10 и *Windows* 11 *Windows* *Mixed* *Reality* на различных поддерживаемых гарнитурах в различных конфигурациях оборудования, их спецификации и возможности определяют, какие особенности можно будет использовать. На высококлассных компьютерах включены некоторые дополнительные возможности и функции:

* более четкие визуальные элементы и более высокая частота обновления;
* запуск большего числа приложений и возможностей, включая наиболее визуально поразительные игры;
* зеркальное отображение того, что пользователь видит в смешанной реальности на рабочем столе;
* возможность записывать и делиться видео и фотографиями в смешанной реальности.

При первом запуске портала смешанной реальности, будет автоматически выполнена диагностика ПК и пользователю на экран будет выведено сообщение. Всего возможны три вида результата:

* компьютер готов к работе *Windows* *Mixed* *Reality* и большинству возможностей смешанной реальности;
* компьютер готов к запуску *Windows* *Mixed* *Reality*, хотя некоторые функции и возможности могут не поддерживаться;
* компьютер не соответствует минимальным требованиям, необходимым для запуска *Windows* *Mixed* *Reality*.

Минимальные системные требования для компьютеров, необходимые для установки *Windows* *Mixed* *Reality*:

* операционная система Windows 10 версии 20*H*2 или более поздней – домашняя, *pro*, корпоративная или для образовательных учреждений;
* процессор *Intel* *Core* *i*5 4590 (4-го поколения), четырехъядерный (или более поздней версии), либо *AMD* *Ryzen* 5 1400 3,4 ГГц (настольный), четырехъядерный (или более поздней версии);
* оперативная память 8 ГБ *DDR*3 (или выше);
* свободное пространство на диске не менее 10 ГБ;
* графический процессор *NVIDIA* *GTX* 1060 (или более поздней версии) с поддержкой *DX*12 дискретного *GPU*, либо *AMD* *RX* 470/570 (или более поздней версии) с поддержкой DX12 дискретного *GPU*;
* графический драйвер *Windows* *Display* *Driver* *Model* (*WDDM*) 2.2;
* порт отображение графики *HDMI* 2.0 или *DisplayPort* 1.2;
* для отображения должен быть подключенный внешний или интегрированный дисплей *VGA* (800x600) (или более поздней версии);
* *usb*-подключение *USB* 3.0;
* подключение *Bluetooth* (для контроллеров движения) должно быть *Bluetooth* 4.0;

Разработанное игровое приложение тестировалось на трёх ПК с разными характеристиками процессоров, оперативной памяти и видеокарт. Сравнительный анализ компьютеров, на которых проводилось тестирование *VR* приведён в таблице 4.1.

Таблица 4.1 – Устройства, используемые в тестировании

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Устройство | Операционная система | Процессор | Видеокарта | Объем оперативной памяти |
| *Razer* *Blade* 15 *Advanced* | *Windows* 10 | *Intel Core i7* 8750*H* 2200 МГц | *NVIDIA GeForce RTX* 2080 *Max-Q* | 16 ГБ |
| *ASUS TUF Gaming FX*705*DU*\_*FX*705*D* | *Windows* 10 | *AMD Ryzen* 7 370H | *NVIDIA GeForce GTX* 1660 *Ti* | 8 Гб |
| *ASUS TUF Gaming Dash* *F*15 *FX*516*PM*-*HN*130*T* | *Windows* 10 | *Intel Core i7* 11370*H* 3300 МГц | 3060*AMD Radeon Vega* 3 | 16 ГБ |

За время проведения тестов, возникала лишь пара ошибок, которые приводили к полному завершению работы приложения, большая же часть касалась лишь визуальной составляющей. Из ошибок, которые встречались следует отметить «застревание» некотрых деталей бойлера в окружающих текстурах, однако данный недочёт был исправлен изменнеием координат раздвижения котла.

Из всего вышеупомянутого можно сказать, что качество и стабильность итогового приложения находится на высоком уровне. Данное тестирование показывает, что на текущем этапе реализации проекта, игра соответствует большинству функциональных требований, выделенных на этапе проектирования приложения.

После проведения всех тестов и исправления ошибок, приложение успешно опубликовано в *GitHub*.

## 4.2 Тестирование *VR* приложения

***4.2.1*** Перед выпуском финальной версии разработанное приложение тестировалось несколько раз. Основные виды тестирования игры:

* функциональное тестирование – это тестирование, целью которого является выявление отклонений от функциональных требований. Данное тестирование сводится к многократному прохождению игры, выявлению неполадок и условий, в которых их можно исправить. Это тестирование происходило на протяжении всего времени разработки;
* нагрузочное тестирование – это тестирование игр, при котором целесообразно создавать ситуации, которые требуют большой вычислительной нагрузки. Таким образом, тестировщик может проверить производительность системы в стрессовой ситуации. Так легче заметить и вовремя исправить потенциально ненадежные участки кода. Для данного вида тестирования разрабатывались отдельные приложения, использующие тот же движок, что и разработанная игра, либо менялись какие-либо настройки имеющихся механик. Например, в игру вводились новые бонусы для генерации и изменения характеристик игрока. Следовательно, необходимо было добавить и место на игровом поле, где бы они размещались, иначе один генератор мог бы произвести несколько бонусов за раз и ближайший к ним игрок получил бы чересчур большое преимущество перед соперниками. Благодаря подобным испытаниям удалось избежать ряд игровых ошибок и сбалансировать геймплей;
* тестирование локализации – это вид тестирования, при котором игры часто переводят на языки тех стран, в которых их предполагают выпускать на рынок. Случается, что переводчики не могут предоставить абсолютно точный перевод, который бы полностью соответствовал событиям игры. Даже правильно переведенная фраза может не соответствовать ситуации и резать слух носителю языка. Поэтому после локализации полезно провести тестирование игры жителями тех стран, где предполагается реализация конечного продукта. В связи с отсутствием текстовой информации в игре – данный тип тестирования не проводился;
* тестирование на совместимость – это вид нефункционального тестирования, основной целью которого является проверка корректной работы продукта в определенном окружении. Чаще всего программирование игр проводят на ПК или ноутбуках. Однако многие игры могут быть предназначены для других устройств: игровые приставки, мобильные телефоны, коммуникаторы и так далее. Разработка осуществляется на симуляторах данных устройств, однако они могут значительно отличаться от оригинала. Поэтому в дальнейшем могут возникнуть трудности при запуске игры на оригинальном устройстве. Кроме того, следует обратить особое внимание на лицензирование программного обеспечения. Учитывая разнообразие конфигураций устройств использующих, платформу *Windows*, данный вид тестирования оказался наиболее актуальным в рамках данной дипломной работы.

***4.2.2*** Функциональное тестирование – это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности приложения в определенных условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает ПО, какие задачи оно решает. Тестирование было выполнено путем выполнения игрового приложения при определенных тестировщиком условиях для наблюдения и оценки аспектов работы игры. В процессе функционального тестирования производился анализ приложения для выявления несоответствий между существующими особенностями выполнения приложения и требованиями к приложению. Функциональное тестирование – это тестирование, целью которого является выявление отклонений от функциональных требований. Данное тестирование сводится к многократному прохождению игры, выявлению неполадок и условий, в которых их можно исправить. Это тестирование происходило на протяжении всего времени разработки;

Тестовая ситуация №1. Цель: проверить запуск приложения без подключённого гарнитура виртуальной реальности.

Ожидаемый результат: игра запуститься, однако передвигаться по игровому пространству будет невозможно.

Вывод: ожидаемый и фактический результаты получены, цель достигнута.

Тестовая ситуация №2. Цель: проверить запуск игрового приложения с подключенным гарнитуром и шлемом виртуальной реальности, одетым на пользователя и подключёнными игровыми контроллерами. При этом в игровом движке *Unity* установлены все необходимые пакеты и плагины для обеспечения работы с приложениями виртуальной реальности, потому что при отсутствии установленных расширений будет невозможно открыть проект с приложением.

Ожидаемый результат: игра запуститься пользователь будет видеть свои «руки» в игровом пространстве.

Вывод: ожидаемый и фактический результаты получены, цель достигнута.

Тестовая ситуация №3. Цель: проверить соответствие игрового интерфейса игры в соответствии с требованиями.

Ожидаемый результат: пользователь попадает в помещение, стилизованное под кухню, где на стене висит газовый бойлер.

Вывод: ожидаемый и фактический результаты получены, цель достигнута. Интерьер кухни в игровой сцене приведён на рисунке 4.1.

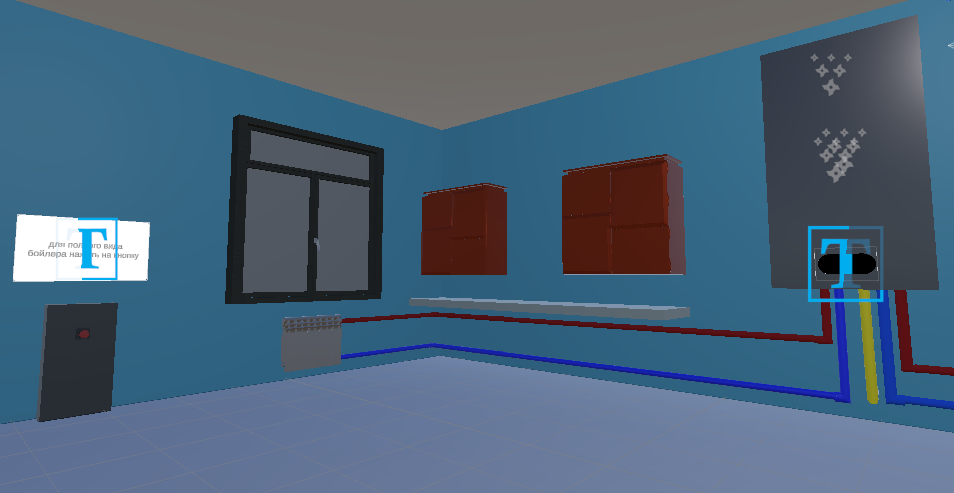


Рисунок 4.1 – Интерьер игровой сцены

Тестовая ситуация №4. Цель: проверить работоспособность движения игрока в приложении.

Ожидаемый результат: при подключённом гарнитуре виртуальной реальности и контроллерах, которые были успешно присоединены к компьютеру, пользователь сможет передвинуться по игровому пространству, используя джойстики контроллеров, а также «телепортироваться» из одной точки в другую, нажимая на необходимую кнопку контроллера.

Вывод: игрок может свободно перемещаться по комнате и оглядываться по сторонам. Полученная цель достигнута.

Тестовая ситуация №5. Цель: проверить на невозможность «ходить сквозь стены» и иные объекты игроком.

Ожидаемый результат: игрок не должен проходить сквозь стены и столы игрового пространства.

Вывод: во время исполнения игры, пользователь находиться только внутри пространства с газовым бойлером и не может его покинуть, пройдя «сквозь стены» или вовсе вывалиться из игрового пространства. Таким образом, полученная цель достигнута.

Тестовая ситуация №6. Цель: проверить возможность нажатия кнопки для получения детализированного вида газового бойлера.

Ожидаемый результат: пользователь должен подойти к стене, где находится кнопка, над которой есть информация о необходимости на жать её для получения детального вида котла, после чего произойдёт плавное «раздвижение» всех деталей газового бойлера. Данная кнопка приведена на рисунке 4.2.



Рисунок 4.2 – Кнопка для получения детального вида бойлера

Вывод: во время работы в *VR* приложении, пользователь, нажимая на данную кнопку, видит, как плавно разъезжается газовый бойлер, после чего он может детально ознакомиться с его внутренним содержимым. При повторном нажатии на кнопку, бойлер вернётся в исходное положение. Таким образом, полученная цель достигнута.

Тестовая ситуация №7. Цель: проверить возможность пользователем во время игры взять комплектующие бойлера «в руки».

Ожидаемый результат: пользователь во время игры может «подержать в руках» детали газового котла, которые хранятся в игривом помещении на столах. Столы с деталями в игровой зоне приведены на рисунке 4.3.

Вывод: ожидаемый и фактический результаты получены, цель достигнута.



Рисунок 4.3 – Вид помещения с демонстрационными деталями

Тестовая ситуация №8. Цель: если взять деталь «в руки», то она подкраситься красным цветом в сборке газового котла.

Ожидаемый результат: пользователь, если возьмёт демонстрационную модель детали газового бойлера, при условии, что предварительно был получен детальный вид котла путём нажатия на соответствующую кнопку, сможет увидеть эту же деталь, выделенную цветом в бойлере.

Вывод: ожидаемый и фактический результаты получены, цель достигнута.

Тестовая ситуация №9. Цель: моделирование аварийных ситуаций газового бойлера после нажатия соответствующей кнопки на столе возле каждой детали.

Ожидаемый результат: пользователь, нажав на кнопку, сможет увидеть моделирование нештатных ситуаций на модели газового котла. Например, при нажатии га кнопку возле теплообменника, он окрасится в яркий красный цвет, что обозначает его перегрев, на информационном стенде будет выведен код ошибки и способ её устранения, при чрезвычайном дымоудалении, будет показано высокое думоудаление из вентиляции котла с выводом на стенде соответствующей информации, и так далее. Вид детали и кнопки для активации нештатной ситуации приведён на рисунке 4.4.



Рисунок 4.4. – Вид детали и кнопки для активации аварийной ситуации

Вывод: аварийные ситуации исполняются, следовательно, ожидаемый и фактический результаты получены, цель достигнута.

5 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ *VR* ПРИЛОЖЕНИЯ ДЛЯ

МОДЕЛИРОВАНИЯ АВАРИЙНЫХ СИТУАЦИЙ ГАЗОВОГО

БОЙЛЕРА *ARISTONE* *CARES* *XC*-*HS*

## 5.1 Технико-экономическое обоснование целесообразности

## разработки программного продукта и оценка его

## конкурентоспособности

Разработанное *VR* приложение позволит значительно повысить эффективность обучения персонала работе и эксплуатации специализированным газовым оборудованием. Приложение виртуальной реальности позволит в интерактивном виде представлять газовый бойлер, его внутренне содержимое, а также наглядно отображать основные нештатные ситуации, которые могут с ним происходить. Таким образом, данное приложение обеспечивает рост производительности за счёт повышения качества обучения и сокращения скорости получения квалификации рабочими.

В первую очередь необходимо провести оценку конкурентоспособности ПП [10, с. 18].

Техническая прогрессивность разрабатываемого ПП определяется коэффициентом эквивалентности (). Расчет этого коэффициента осуществляется путем сравнения технического уровня товара-конкурента и разрабатываемого ПП по отношению к эталонному уровню ПП данного направления с использованием формулы (5.1):

, (5.1)



где  – коэффициент технического уровня нового ПП;

 – коэффициент технического уровня базисного ПП.

Коэффициенты рассчитываются по формуле (5.2):

, (5.2)

где  – коэффициенты весомости *i*-го технического параметра;

*n* – число параметров;

 – численное значение *i*-го технического параметра, сравниваемого ПП;

 – численное значение *i*-го технического параметра эталона.

Результат расчета коэффициента эквивалентности приведен в таблице 5.1. Полученное значение коэффициента эквивалентности больше единицы, следовательно, разрабатываемый программный продукт является технически прогрессивным.

Таблица 5.1 – Расчет коэффициентов эквивалентности

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Наименование  параметра | Вес  парамет­­ра, | Значения  параметра | | |  |  |  |  |
|  |  |  |
| Объем памяти | 0,1 | 9 | 7 | 6 | 1,5 | 1,17 | 0,15 | 0,12 |
| Время обработки  данных | 0,4 | 0,8 | 0,65 | 0,2 | 4,0 | 3,25 | 1,6 | 1,3 |
| Отказы | 0,5 | 2 | 1 | 1 | 2 | 1 | 1 | 0,5 |
| Итого | | | | | | | 2,75 | 1,92 |
| Коэффициент эквивалентности | | | | | | | 2,75/1,92=1,43 | |

Далее рассчитывается коэффициент изменения функциональных возможностей () нового ПО по формуле (5.3):

, (5.3)

где  – балльная оценка неизмеримых показателей нового изделия;

 – балльная оценка неизмеримых показателей базового изделия

Расчет коэффициента изменения функциональных возможностей нового ПП приведен в таблице 5.2.

Таблица 5.2 – Расчет коэффициента изменения функциональных возможностей

|  |  |  |
| --- | --- | --- |
| Наименование показателя | Балльная оценка базового ПП | Балльная оценка нового ПП |
| 1 | 2 | 3 |
| Объем памяти | 4 | 4 |
| Быстродействие | 3 | 4 |
| Удобство интерфейса | 3 | 3 |
| Степень утомляемости | 2 | 3 |
| Производительность труда | 3 | 4 |
| Итого | 15 | 18 |

Продолжение таблицы 5.2

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Коэффициент функциональных возможностей | 18/15 = 1,2 | |

Новый программный продукт превосходит по своим функциональным возможностям базовый в 1,2 раза.

Конкурентоспособность нового ПП по отношению к базовому можно оценить с помощью интегрального коэффициента конкурентоспособности, учитывающего все ранее рассчитанные показатели. Формула для расчета (5.4):

, (5.4)

где  – коэффициент соответствия нового ПП нормативам ( = 1);

 – коэффициент цены потребления.

Таблица 5.3 – Расчет уровня конкурентоспособности нового ПП

|  |  |
| --- | --- |
| Коэффициенты | Значение |
| Коэффициент эквивалентности | 1,43 |
| Коэффициент изменения функциональных возможностей | 1,20 |
| Коэффициент соответствия нормативам | 1 |
| Коэффициент цены потребления | 0,84 |
| Интегральный коэффициент конкурентоспособности |  |

Интегральный коэффициент конкурентоспособности () больше единицы, это значит, что новый программный продукт является более конкурентоспособным, чем базовый.

## 5.2 Оценка трудоёмкости работ по созданию программного

## обеспечения

Общий объем ПО (*V*o) определяется исходя из количества и объема функций, реализуемых программой, по каталогу функций ПО и рассчитывается по формуле (5.5):

, (5.5)

где  – объем отдельной функции ПО;

*n* – общее число функций.

Уточненный объем ПО () определяется по формуле (5.6):

, (5.6)

где  – уточненный объем отдельной функции ПО в строках исходного кода.

Результаты произведённых вычислений объема функций ПО представлены в таблице 5.4.

Таблица 5.4 – Перечень и объем функций ПО

|  |  |  |  |
| --- | --- | --- | --- |
| Код  функции | Наименование (содержание) функции | Объем функции строк исходного кода | |
| по каталогу  () | Уточненный  () |
| 101 | Организация ввода информации | 190 | 170 |
| 102 | Контроль, предварительная обработка и ввод информации | 550 | 460 |
| 109 | Управление вводом-выводом | 2400 | 730 |
| 506 | Обработка ошибочных сбойных ситуаций | 1540 | 900 |
| 702 | Расчетные задачи | 14800 | 2450 |
| 707 | Графический вывод результатов | 590 | 420 |
| 709 | Изменение состояния ресурсов в интерактивном режиме | 570 | 400 |
| Итого | | 23630 | 5530 |

В разработанном приложении виртуальной реальности реализован интерактивный доступ к существующим моделям и аварийным ситуациям. ПО относится ко второй категории сложности. На основании принятого к расчету (уточненного) объема () и категории сложности ПО определяется нормативная трудоемкость ПО () выполняемых работ, представлена в таблице 5.5.

Таблица 5.5 – Нормативная трудоемкость на разработку ПО ()

|  |  |  |
| --- | --- | --- |
| Уточнённый объем, | 2-я категория сложности ПО | Номер нормы |
| 5530 | 285 | 51 |

Дополнительные затраты труда, связанные с повышением сложности разрабатываемого ПО, учитываются посредством коэффициента повышения сложности ПО ().

 рассчитывается по формуле (5.7):

, (5.7)

где  – коэффициент, соответствующий степени повышения сложности;

 – количество учитываемых характеристик.

.

Влияние фактора новизны на трудоемкость учитывается путем умножения нормативной трудоемкости на соответствующий коэффициент, учитывающий новизну ПО ().

Разработанная программа обладает категорией новизны Б, а значение  = 0,81. Степень использования в разрабатываемом ПО стандартных модулей определяется их удельным весом в общем объеме ПО.

В данном программном комплексе используется до 40% стандартных модулей, что соответствует значению коэффициента  = 0,65.

Программный модуль разработан с помощью объектно-ориентированных технологий, что соответствует коэффициенту, учитывающему средства разработки ПО,  = 0,6. Значения коэффициентов удельных весов трудоемкости стадий разработки ПО в общей трудоемкости ПО определяются с учетом установленной категории новизны ПО и приведены в таблице 5.6.

Таблица 5.6 – Значения коэффициентов удельных весов трудоемкости стадий разработки ПО в общей трудоемкости

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Категория  новизны ПО | Без применения *CASE*-технологий | | | | |
| Стадии разработки ПО | | | | |
| ТЗ | ЭП | ТП | РП | ВН |
| Значения коэффициентов | | | | |
|  |  |  |  |  |
| Б | 0,10 | 0,20 | 0,30 | 0,30 | 0,10 |

Нормативная трудоемкость ПО () выполняемых работ по стадиям разработки корректируется с учетом коэффициентов: повышения сложности ПО (), учитывающих новизну ПО (), учитывающих степень использования стандартных модулей (), средства разработки ПО () и определяются по формулам:

* для стадии ТЗ рассчитывается по формуле (5.8):

; (5.8)

* для стадии ЭП рассчитывается по формуле (5.9):

; (5.9)

* для стадии ТП рассчитывается по формуле (5.10):

; (5.10)

* для стадии РП рассчитывается по формуле (5.11):

; (5.11)

* для стадии ВН рассчитывается по формуле (5.12):

, (5.12)

где , , ,  и  – значения коэффициентов удельных весов трудоемкости стадий разработки ПО в общей трудоемкости ПО.

Коэффициенты ,  и  вводятся на всех стадиях разработки, а коэффициент  вводится только на стадии РП.

Для уменьшения общей трудоёмкости разработки необходимо ввести коэффициент используемости разработанного ПО, который равен 0,1.

чел.-дн.,

чел.-дн.,

 чел.-дн.,

 чел.-дн.,

 чел.-дн.

Общая трудоемкость разработки ПО () определяется суммированием нормативной (скорректированной) трудоемкости ПО по стадиям разработки и рассчитывается по формуле (5.13):

, (5.13)

где  – нормативная (скорректированная) трудоемкость разработки ПО на *i*-й стадии, чел.-дн.;

*n* – количество стадий разработки.

 чел.-дн.

Параметры расчетов по определению нормативной и скорректированной трудоемкости программного обеспечения по стадиям разработки и общую трудоемкость разработки представлены в таблице 5.7.

Таблица 5.7 – Расчет общей трудоемкости разработки ПО

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Показатели | Стадии разработки | | | | | Итого |
| ТЗ | ЭП | ТП | РП | ВН |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Общий объем ПО (), кол-во строк *LOC* | – | – | – | – | – | 23630 |
| Общий уточненный объем ПО (), кол-во строк *LOC* | – | – | – | – | – | 5530 |
| Категория сложности разрабатываемого ПО | – | – | – | – | – | 2 |
| Нормативная трудоемкость разработки ПО (), чел.-дн. | – | – | – | – | – | 240 |
| Коэффициент повышения сложности ПО () | 1,07 | 1,07 | 1,07 | 1,07 | 1,07 | – |
| Коэффициент, учитывающий новизну ПО () | 0,81 | 0, 81 | 0, 81 | 0, 81 | 0, 81 | – |
| Коэффициент, учитывающий степень использования стандартных модулей () | – | – | – | 0,65 | – | – |
| Коэффициент, учитывающий средства разработки ПО() | 0,6 | 0,6 | 0,6 | 0,6 | 0,6 | – |
| Продолжение таблицы 5.7 | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Коэффициенты удельных весов трудоемкости стадий разработки ПО (,,,,) | 0,10 | 0,20 | 0,30 | 0,30 | 0,10 | 1,0 |
| Распределение скорректированной (с учетом ,, ,) трудоемкости ПО по стадиям, чел.-дн. | 12,48 | 24,96 | 37,44 | 24,34 | 12,48 | 112 |
| Общая трудоемкость разработки ПО (), чел.-дн. | – | – | – | – | – | 112 |

## 5.3 Расчёт затрат на разработку программного продукта

В состав затрат на разработку игрового приложения входят следующие статьи расходов:

* трудовые затраты на разработку игрового приложения ();
* затраты на изготовление эталонного экземпляра ();
* затраты на освоение и приобретение программных продуктов, используемых при разработке игрового приложения ();
* затраты машинного времени ();
* затраты на материалы, например, информационные носители ();
* затраты на энергию и использование каналов связи;
* общепроизводственные расходы, такие как затраты на содержание помещений и управленческий персонал ();
* коммерческие непроизводственные расходы ().

Суммарные затраты на разработку ПО (Зр) определяются по формуле (5.14):

. (5.14)

Параметры расчета производственных затрат на разработку программного обеспечения приведены в таблице 5.8.

Таблица 5.8 – Параметры расчета производственных затрат на разработку ПО

|  |  |  |
| --- | --- | --- |
| Параметр | Единица измерения | Значение |
| Тарифная ставка 1-го разряда | руб. | 228,00 |
| Разряд разработчика | – | 11 |
| Тарифный коэффициент | – | 2,81 |
| Коэффициент | – | 1,8 |
| Норматив отчислений на доп. зарплату разработчиков () | % | 20 |
| Численность обслуживающего персонала | чел. | 1 |
| Средняя годовая ставка арендных платежей () (по результатам мониторинга предложений по аренде помещений) | руб./м2 | 17,85 |
| Площадь помещения (*S*) | м2 | 15 |
| Количество ПЭВМ () | шт. | 1 |
| Затраты на приобретение единицы ПЭВМ | руб. | 2500 |
| Затраты на приобретение гарнитура виртуальной реальности | руб. | 1400 |
| Стоимость одного кВт-часа электроэнергии () | руб. | 0,360852 |
| Коэффициент потерь рабочего времени () | – | 0,2 |
| Затраты на технологию () | руб. | - |
| Норматив общепроизводственных затрат () | % | 10 |
| Норматив непроизводственных затрат () | % | 5 |

Расчет затрат на оплату труда разработчиков проводится следующим образом.

Расходы на оплату труда разработчиков с отчислениями (Зтр) определяются по формуле (5.15):

, (5.15)

где  – основная заработная плата разработчиков, руб.;

 – дополнительная заработная плата разработчиков, руб.;

 – сумма отчислений от заработной платы (социальные нужды, страхование от несчастных случаев), руб.

Основная заработная плата разработчиков рассчитывается по формуле (5.16):

, (5.16)

где  – средняя часовая тарифная ставка, руб./час;

 – общая трудоемкость разработки, чел.-час;

 – коэффициент доплаты стимулирующего характера ().

Средняя часовая тарифная ставка определяется по формуле (5.17):

, (5.17)

где Сч*i* – часовая тарифная ставка разработчика *i*-й категории, руб./час;

*ni* – количество разработчиков *i*-й категории.

Часовая тарифная ставка определяется путем деления месячной тарифной ставки на установленный при восьмичасовом рабочем дне фонд рабочего времени 167,3 ч (*F*мес) и рассчитывается по формуле (5.18):

, (5.18)

где  – тарифная ставка 1-го разряда;

 – тарифный коэффициент.

руб.,

руб.

Дополнительная заработная плата рассчитывается по формуле (5.19):

, (5.19)

где  – норматив на дополнительную заработную плату разработчиков.

руб.

Отчисления от основной и дополнительной заработной платы рассчитываются по формуле (5.20):

, (5.20)

где – процент отчислений на социальные нужды и обязательное страхование от суммы основной и дополнительной заработной платы ().

руб.,

руб.

Затраты машинного времени () определяются по формуле (5.21):

, (5.21)

где  – стоимость 1 часа машинного времени, руб./ч;

 – коэффициент мультипрограммности, показывающий распределение времени работы ПК в зависимости от количества пользователей ПК ();

 – машинное время ПК, необходимое для разработки и отладки проекта, ч.

Стоимость машино-часа определяется по формуле (5.22):

, (5.22)

где  – затраты на заработную плату обслуживающего персонала с учетом всех отчислений, руб./год ();

 – стоимость аренды помещения под размещение вычислительной техники, руб./год (), так как в аренде помещения нет необходимости;

 – амортизационные отчисления за год, руб./год;

 – затраты на электроэнергию, руб./год;

 – затраты на материалы, необходимые для обеспечения нормальной работы ПК (вспомогательные), руб./год;

 – затраты на текущий и профилактический ремонт ПК, руб./год;

 – прочие затраты, связанные с эксплуатацией ПК, руб./год;

 – действительный фонд времени работы ПК, час/год.

 – количество обслуживаемых ПК, шт.;

 – месячная тарифная ставка *i*-го работника, руб.

Сумма годовых амортизационных отчислений () определяется по формуле (5.23):

, (5.23)

где  – затраты на приобретение *i*-го вида основных фондов, руб;

 – коэффициент, дополнительных затраты, связанные с доставкой, монтажом и наладкой оборудования ( от );

 – балансовая стоимость ПК, руб;

– норма амортизации, %.

руб.

Стоимость электроэнергии, потребляемой за год, определяется по формуле (5.24):

, (5.24)

где  – паспортная мощность ПК, кВт,  кВт;

– стоимость одного кВт-часа электроэнергии, руб;

*А* – коэффициент интенсивного использования мощности, .

Действительный годовой фонд времени работы ПК () рассчитывается по формуле (5.25):

, (5.25)

где  – общее количество дней в году ( дней);

,  – число выходных и праздничных дней в году (дней);

 – продолжительность 1 смены ( часов);

– коэффициент сменности ();

 – коэффициент, учитывающий потери рабочего времени, связанные с профилактикой и ремонтом ПК ().

 ч.

В итоге получается, что годовая стоимость электроэнергии () равна:

 руб.

Из чего следует, что за 112 дней будет израсходовано:



Затраты на материалы (), необходимые для обеспечения нормальной работы ПК составляют около 1% от балансовой стоимости ПК и определяются по формуле (5.26):

, (5.26)

где  – затраты на приобретение (стоимость) ПК, руб.;

 – коэффициент, характеризующий дополнительные затраты, связанные с доставкой, монтажом и наладкой оборудования ( от );

 – коэффициент, характеризующий затраты на вспомогательные материалы ().

руб.

Затраты на текущий и профилактический ремонт () принимаются равными 5% от балансовой стоимости ПК и вычисляются по формуле (5.27):

, (5.27)

где  – коэффициент, характеризующий затраты на текущий и профилактический ремонт ().

руб.

Прочие затраты, связанные с эксплуатацией ПК () состоят из амортизационных отчислений на здания, стоимости услуг сторонних организаций и составляют 5 % от балансовой стоимости, и вычисляется по формуле (5.28):

, (5.28)

где  – коэффициент, характеризующий размер прочих затрат, связанных с эксплуатацией ПК ().

руб.

Для расчета машинного времени ПК ( в часах), необходимого для разработки и отладки проекта, следует использовать формулу (5.29):

, (5.29)

где  – срок реализации стадии «Рабочий проект» (РП);

 – срок реализации стадии «Ввод в действие» (ВП);

 – продолжительность рабочей смены ( ч.);

 – количество рабочих смен ().

 ч,

Расчёт стоимости машино-часа определяется по формуле:

руб./ч.,

Расчет затраты машинного времени производится по формуле 5.21:

 руб.

Расчет затрат на изготовление эталонного экземпляра () осуществляется по формуле (5.30):

, (5.30)

где  – коэффициент, учитывающий размер затрат на изготовление эталонного экземпляра ().

руб.

Затраты на материалы, необходимые для обеспечения нормальной работы ПК, рассчитываются по формуле (5.31):

, (5.31)



где Ц*i* – цена *i*-го наименования материала полуфабриката, комплектующего, руб.;

*Ni* – потребность в *i*-м материале, полуфабрикате, комплектующем, натур. ед.;

 – коэффициент, учитывающий сложившийся процент транспортно-заготовительных расходов в зависимости от способа доставки товаров (=0,1–1,3);

Цо*i* – цена возвратных отходов *i*-го наименования материала, руб.;

*N*о*i* – количество возвратных отходов *i*-го наименования, натур. ед.;

*n* – количество наименований материалов, полуфабрикатов, и т.д.

руб.

Общепроизводственные затраты () рассчитываются по формуле (5.32):

, (5.32)

где Ндоп – норматив общепроизводственных затрат.

 руб.

Непроизводственные затраты рассчитываются по формуле (5.33):

, (5.33)

где – норматив непроизводственных затрат.

руб.

Итого суммарные затраты на разработку:

руб.

Результаты расчета суммарных затрат представлены в таблице 5.9.

Таблица 5.9 – Результат расчета суммарных затрат на разработку ПО

|  |  |
| --- | --- |
| Статья затрат | Итого |
| 1 | 2 |
| Затраты на оплату труда разработчиков (Зтр) | 6913,31 |
| Основная заработная плата разработчиков | 4280,16 |
| Дополнительная заработная плата разработчиков | 856,03 |
| Отчисления от основной и дополнительной заработной платы | 1777,12 |
| Затраты машинного времени (Змв) | 176,4 |
| Стоимость машино-часа (Сч) | 0,48 |
| Затраты на заработную плату обслуживающего персонала (ЗПоб) | 0 |
| Годовые затраты на аренду помещения (Зар) | 214,2 |
| Сумма годовых амортизационных отчислений (Зам) | 550,86 |
| Стоимость электроэнергии, потребляемой за 112 дней (Зэп) | 109,74 |
| Действительный годовой фонд времени работы ПЭВМ (*F*эвм) | 1619 |
| Затраты на материалы (Зв.м) | 28,25 |
| Затраты на текущий и профилактический ремонт (Зтр) | 141,25 |
| Прочие затраты, связанные с эксплуатацией ЭВМ (Зпр) | 141,25 |
| Машинное время ЭВМ (tэвм) | 142,08 |
| Затраты на изготовление эталонного экземпляра (Зэт) | 385,77 |
| Затраты на технологию (Зтех) | 0 |
| Затраты на материалы (Змат) | 44,07 |
| Общепроизводственные затраты (Зобщ.пр) | 230,42 |
| Непроизводственные (коммерческие) затраты (Знепр) | 230,42 |
| Продолжение таблицы 5.9 | |
| 1 | 2 |
| Суммарные затраты на разработку ПО (Зр) | 7880,23 |

Таким образом, суммарные затраты на разработку приложения виртуальной реальности для моделирования аварийных ситуаций газового бойлера составляют 7880,23 руб.

## 5.4 Расчёт цены разрабатываемого программного средства

Оптовая цена ПП () определяется по формуле (5.34):

, (5.34)

где – себестоимость ПО, руб.;

 – прибыль от реализации ПП, руб.;

Прибыль от реализации рассчитывается по формуле (5.35):

, (5.35)

где Ур – уровень рентабельности ПП ().

руб.,

руб.

Прогнозируемая отпускная цена ПП рассчитывается по формуле (5.36):

. (5.36)

Налог на добавленную стоимость () рассчитывается по формуле (5.37):

, (5.37)

где  – ставка налога на добавленную стоимость ().

руб.,

руб.

Таким образом прогнозируемая отпускная цена составляет 13238,78 руб.

## 5.5 Расчёт частных экономических объектов от производства

Оценка эффективности инвестиций базируется на сопоставлении ожидаемого чистого дохода от реализации проекта с затратами инвестиционного характера. На основании чистого потока наличности рассчитываются основные показатели оценки эффективности инвестиций: чистый дисконтированный доход (ЧДД), индекс рентабельности (ИР), динамический срок окупаемости ().

Для расчета этих показателей применяется коэффициент дисконтирования, который используется для приведения будущих потоков и оттоков денежных средств за каждый расчетный период реализации проекта к начальному периоду времени.

Коэффициент дисконтирования рассчитывается по формуле (5.38):

, (5.38)

Затраты в год на оплату сторонних ресурсов, так же оплата хостинга и зарплаты менеджерам составляет примерно 24000 руб. А значит внедрив ПО, экономия составляет примерно 24000 руб. в год.

Данные расчёта чистого дисконтированного дохода разработанного приложения виртуальной реальности приведены в таблице 5.10.

Таблица 5.10 – Расчет чистого дисконтированного дохода (ЧДД), руб.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Показатель | Годы реализации проекта | | |
| 0-й | 1-й | 2-й |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Отток денежных средств | 13238,78 | – | – |
| 1.1 | Отпускная цена ПП с НДС | 13238,78 | – | – |
| Продолжение таблицы 5.10 | | | | |
| 1 | 2 | 3 | 4 | 5 |
| 1.2 | Капитальные вложения | – | – | – |
| 2 | Приток денежных средств | – | 24000 | 24000 |
| 2.1 | Годовая экономия от использования ПО | – | 24000 | 24000 |
| 3 | Чистый поток денежных средств | -13238,78 | 24000 | 24000 |
| 4 | Коэффициент дисконтирования (при *r*=12%) | 1 | 0,89 | 0,79 |
| 5 | Текущая стоимость потока | -13238,78 | 21360 | 18960 |

Если инвестиционные затраты, связанные с разработкой программного продукта и приобретением компьютерной техники, периферийных устройств, кабелей и т.д. производится только в год разработки, а первые доходы ожидаются в следующем году, то формула примет вид (5.39):

 , (5.39)

руб.

Индекс рентабельности (доходности) (ИР) – это отношение суммарного дисконтированного дохода к суммарным дисконтированным затратам. Проект эффективен, если норма дисконта оказывается больше или равной ставки рефинансирования, требуемой инвестором, кредитором рассчитывается по формуле (5.40):

 , (5.40)

.

Инвестиционные проекты эффективны при ИР > 1.

Динамический срок окупаемости (). Расчет динамического срока окупаемости проекта осуществляется по накопительному дисконтированному чистому потоку наличности. Динамический срок окупаемости в отличии от простого учитывает стоимость капитала и показывает реальный период окупаемости, и рассчитывается по формуле (5.41):

 , (5.41)

 г.

Технико-экономические показатели проекта представлены в таблице 5.11.

Таблица 5.11 – Технико-экономические показатели проекта

|  |  |
| --- | --- |
| Наименование показателя | Значение |
| Общая трудоемкость разработки игрового приложения | 112дн. |
| Затраты на разработку программы | 7880,23 руб. |
| Затраты на оплату труда разработчиков | 6913,31 руб. |
| Затраты машинного времени | 142,08 руб. |
| Затраты на технологию | 0 руб. |
| Затраты на материалы | 28,25 руб. |
| Общепроизводственные затраты | 230,42 руб. |
| Непроизводственные затраты | 230,42 руб. |
| Цена без НДС | 11032,32 руб. |
| НДС | 2206,46 руб. |
| Цена с НДС | 13238,78 руб. |
| Срок окупаемости проекта | 1,7 года. |

Таким образом, по результатам проведенной оценки, установлено, что реализация проекта обоснована и является экономически целесообразной. Срок окупаемости будет составлять 1,7 года.

6 ОХРАНА ТРУДА И ТЕХНИКА БЕЗОПАСНОСТИ

## 6.1 Воздействие шума на организм человека

Шум – беспорядочные колебания различной физической природы, отличающиеся сложностью временной и спектральной структуры. С физиологической точки зрения шумом может быть назван любой нежелательный звук (простой или сложный), мешающий восприятию полезных звуков (человеческой речи, сигналов и пр.), нарушающих тишину и оказывающих вредное действие на человека.

Шум в окружающей среде – в жилых и общественных зданиях, на прилегающих к ним территориях создается одиночными или комплексными источниками, находящимися снаружи или внутри здания. Это прежде всего транспортные средства, техническое оборудование промышленных и бытовых предприятий, вентиляторные, газотурбокомпрессорные установки, станции для испытания газотурбинных двигателей и двигателей внутреннего сгорания, различные аэрогазодинамические установки, санитарно-техническое оборудование жилых зданий, электрические трансформаторы.

Характеристикой постоянного во времени шума являются уровни звукового давления *L*, выражаемые в децибелах (дБ) в октавных полосах со среднегеометрическими частотами 63, 125, 250, 500, 1000, 2000, 4000, 8000 Гц, определяемые по формуле:

 (6.1)

где  – среднеквадратичное значение звукового давления;

 – исходное значение звукового давления в воздухе, равное Па.

Для ориентировочной оценки (например, при проверке органами надзора или выявлении необходимости осуществления мер по шумоглушению) в качестве характеристики постоянного широкополосного шума принимается уровень звука в дБА, измеряемый на временной характеристике шумомера «медленно».

Без принятия соответствующих мер по снижению шума его уровни могут существенно превышать (на 20–50 дБ) нормативные величины. За последние десятилетия наблюдается непрерывное увеличение шума в крупных городах. Расчеты показывают, что в ближайшие 20–30 лет уровни шума на скоростных и городских магистралях возрастут на 7–10 дБ. Высокие уровни шума имеют место в жилых домах, школах, больницах, местах отдыха населения, что приводит к повышению нервного напряжения.

Шумы, воздействующие на человека, классифицируются по спектральным и временным характеристикам.

По характеру спектра шумы подразделяют на широкополосные, имеющие непрерывный спектр шириной более одной октавы, и тональные, в спектре которых есть слышимые дискретные тона. Тональность шума устанавливают по результатам измерений уровней звукового давления в третьоктавных полосах частот, когда превышение уровня в одной полосе над соседними составляет не менее чем в 10 дБ.

На рисунке 6.1 приведены зависимости частоты от уровня звукового давления спектрального шума.

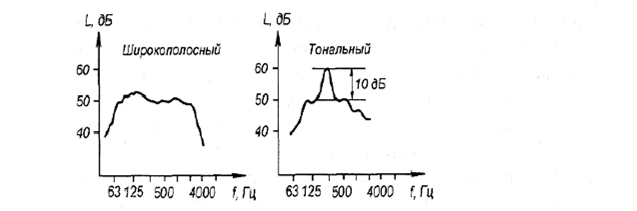


Рисунок 6.1 – Характеристики спектрального шума

По временным характеристикам шумы подразделяют на постоянные, уровень звука которых изменяется во времени не более чем на 5 дБА при измерениях на временной характеристике шумомера «медленно», и непостоянные, для которых это изменение превышает 5 дБА. Непостоянные шумы могут быть:

* колеблющимися во времени, уровень звука которых непрерывно изменяется;
* прерывистыми, уровень звука которых ступенчато изменяется (на 5 дБА и более) несколько раз за время наблюдения, при этом источник шума работает с перерывами (паузами) между интервалами, в течение которых (одна секунда и более) уровень остается постоянным и превышающим уровень фонового шума;
* импульсными, состоящими из одного или нескольких звуковых импульсов (сигналов), каждый длительностью менее одной секунды, при этом уровни звука, измеренные соответственно на временных характеристиках шумомера «импульс» и «медленно», отличаются не менее чем на 7 дБА.

На рисунке 6.2 приведены характеристики временных шумов.

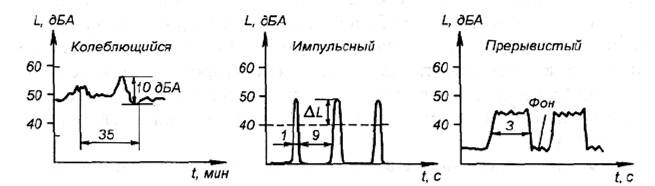


Рисунок 6.2 – Характеристики временных шумов

Человек реагирует на шум в зависимости от субъективных особенностей организма, привычного шумового фона. Раздражающее действие шума зависит прежде всего от его уровня, а также от спектральных и временных характеристик. Считается, что шум с уровнем ниже 60 дБА вызывает нервное раздражение, поэтому неслучайно, что рядом исследователей установлена прямая связь между возрастающим уровнем шума в городах и увеличением числа нервных заболеваний.

Человеческий организм по-разному реагирует на шум разного уровня. Шумы уровня 70–90 дБ при длительном воздействии приводят к заболеванию нервной системы, а более 100 дБ – к снижению слуха, вплоть до глухоты. Шум создает значительную нагрузку на нервную систему человека, оказывая на него психологическое воздействие [11, c. 41 – 44].

Шум способен увеличивать содержание в крови таких гормонов стресса, как кортизол, адреналин и норадреналин – во время сна. Чем дольше эти гормоны присутствуют в кровеносной системе, тем выше вероятность, что они приведут к опасным для жизни физиологическим проблемам.

Согласно нормативам Всемирной организации здравоохранения, сердечно-сосудистые заболевания могут возникнуть, если человек по ночам постоянно подвергается воздействию шума громкостью 50 дБ или выше – такой шум издаёт улица с неинтенсивным движением. Для того, чтобы заработать бессонницу, достаточно шума в 42 дБ; чтобы просто стать раздражительным – 35 дБ (звук шёпота).

Под воздействием шума от 85–90 дБ снижается слуховая чувствительность на высоких частотах. Долгое время человек жалуется на недомогание. Симптомы – головная боль, головокружение, тошнота, чрезмерная раздражительность. Все это результат работы в шумных условиях. Под влиянием сильного шума, особенно высокочастотного, в органе слуха происходят необратимые изменения. При высоких уровнях шума слуховая чувствительность падает уже через 1–2 года, при средних – обнаруживается гораздо позже, через 5–10 лет, то есть снижение слуха происходит медленно, болезнь развивается постепенно. Поэтому особенно важно заранее принимать соответствующие меры защиты от шума. В настоящее время почти каждый человек, подвергающийся на работе воздействию шума, рискует стать глухим. Население, проживающее в примагистральных районах, отмечает ухудшение самочувствия, головные боли, нарушение сна, функций сердечно-сосудистой системы и желудочно-кишечного тракта.

Эффективным путем решения проблемы шума является снижение его уровня в самом источнике за счёт изменения технологии и конструкции машин. К мерам этого типа относятся замена шумных процессов бесшумными, ударных – безударными, например, замена клёпки – пайкой, ковки и штамповки обработкой давлением; замена металла в некоторых деталях незвучными материалами, применение виброизоляции, глушителей, демпфирования, звукоизолирующих кожухов и других звукоизолирующих специализированных средств [10].

7 ЭНЕРГОСБЕРЕЖЕНИЕ И РЕСУРСОСБЕРЕЖЕНИЕ ПРИ

ЭКСПЛУАТАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## 7.1 Понятие ресурсо- и энергосбережения

Энергосбережение – реализация правовых, организационных, научных, производственных, технических и экономических мер, направленных на эффективное (рациональное) использование (и экономное расходование) топливно-энергетических ресурсов и на вовлечение в хозяйственный оборот возобновляемых источников энергии.

Актуальность энергосбережения растет во всех странах, особенно в небогатых своими энергоресурсами, в связи с опережающим ростом цен на основные традиционные виды энергоресурсов и постепенным истощением их мировых запасов [12, c. 187].

Ресурсосбережение – совокупность мер по бережливому и эффективному использованию фактов производства (капитала, земли, труда). Обеспечивается посредством использования ресурсосберегающих и энергосберегающих технологий; снижения фондоемкости и материалоемкости продукции; повышения производительности труда; сокращения затрат живого и овеществленного труда; повышения качества продукции; рационального применения труда менеджеров и маркетологов; использования выгод международного разделения труда. Способствует росту экономики, повышению ее конкурентоспособности [13, с. 12].

Соблюдение ресурсосбережения – это важная характеристика качества техники и технологии. Техника считается ресурсосберегающей, если она требует меньше расхода ресурсов на изготовление и эксплуатацию.

При разработке приложения также требуется соблюдать основные принципы энерго- и ресурсосбережения. Экономия электроэнергии – крайне важный аспект жизни современного человеческого общества, затрагивающий и производственную сферу, и быт каждого отдельно взятого человека. Неразумное потребление достаточно дорогостоящего вида энергии может привести к весьма значительным тратам, что может существенно сказаться как на благосостоянии человека, так и на развитии предприятия [14, с. 92].

## 7.2 Экономия ресурсов в результате внедрения приложения

Экономия ресурсов, связанная с внедрением разработанного приложения виртуальной реальности для моделирования аварийных ситуаций, заключается в повышении эффективности обучения обслуживания и эксплуатации газового оборудования путём добавления в процесс технологий виртуальной реальности. Таким образом, нововведение позволит сделать обучение более наглядным, что позволит ускорить образовательный процесс, а также задействовать меньше инструкторов, либо сократить их загруженность. Сокращенное рабочее время позволяет задействовать специалистов в других задачах, что в конечном итоге ведет к увеличению производительности труда без привлечения дополнительных ресурсов и сотрудников.

Экономия остальных расходов () – величина, которая представляет из себя экономию канцелярских принадлежностей и других предметов, за счёт сокращения времени обучения, рассчитывается по следующей формуле:

 (7.1)

где  – коэффициент сокращения остальных расходов;

 – среднемесячная заработная плата непроизводственных рабочих, бел. руб;

*n* – количество рабочих;

 – остальные расходы.

По состоянию на март 2023 среднемесячная начисленная заработная плата в Республике Беларусь составляет 1814,20 руб. Остальные расходы принять равным 3%. Коэффициент сокращения равен 0,5. Количество рабочих *n* = 2. Тогда экономия остальных расходов равна:

 бел. руб.

Таким образом, экономия остальных ресурсов составляет 653,11 рублей в год.

Разработанное приложение предполагает работу на компьютере с использованием средств виртуальной реальности, что представляет из себя практическую часть обучения, которая сможет сократить теоретическую составляющую. Таким образом будет сокращено число компьютеров, а также время их работы для теоретического обучения.

Подсчитать точный экономический эффект от сбережения электроэнергии при внедрении разработанного программного обеспечения достаточно сложно, поскольку неизвестно какое количество актов будет составляться ежедневно.

Экономию электрической энергии (Э) можно рассчитать по следующей формуле:

 (7.2)

где  – трудоёмкость работы существующей системы, часов;

 – трудоёмкость работы с помощью разработанного приложения, часов;

– паспортная мощность одного персонального компьютера, кВт;

 – стоимость одного кВтч. электроэнергии, бел. руб.;

 – коэффициент использования устройства.

Для проведения всего цикла работы от реализации всех входных данных до расчетов тратится около 4 человеко-часов рабочего времени с использованием одного персонального компьютера, для той же операции с применением разработанного программного продукта требуется – порядка 1 человека-часа на том же компьютере.

По состоянию на 1 января 2023 года стоимость 1 кВт∙ч. электроэнергии для промышленных организаций в Республике Беларусь равна 0,32281 руб/кВтч. Приняв паспортную мощность персонального компьютера в 0,5 кВт, коэффициент использования равным 0,5, рассчитана стоимость сэкономленной электроэнергии за один месяц:

 бел. руб.

Исходя из вышеописанных расчётов можно прийти к выводу, что при внедрении разработанного программного продукта будет сэкономлено на электроэнергии приблизительно 42 рубля в месяц, что соответствует 500 рублям за год работы. Все вышеописанные расчеты говорят о том, что разработанное игровое приложение является выгодным в плане ресурсо- и энергосбережения.

ЗАКЛЮЧЕНИЕ

В ходе написания дипломной работы была проведена самостоятельная исследовательская работа, а также систематизированы, закреплены и углублены теоретические и практические знания, отработаны практические навыки, которые были применены для решения конкретной задачи.

В дипломной работе были рассмотрены средства разработки игровых приложений и были выявлены их положительные и отрицательные стороны. Были изучены особенности разработки приложений виртуальной реальности, а также подготовки 3*D* моделей газового бойлера в системе *Solidworks*.

Также были изучены основные принципы работы с мультиплатформенным игровым движком *Unity*, с помощью которого можно легко создавать любые 2*D* и 3*D*-игры.

Были выбраны технические средства реализации приложения, среда разработки *Unity*, язык программирования *C*#.

Разработанный программный продукт представляет из себя интерактивное приложение виртуальной реальности, где пользователь может взаимодействовать с элементами газового бойлера, а также получать информацию об аварийных ситуациях.

Приложение виртуальной реальности позволяет повысить скорость и эффективность обучения эксплуатации и обслуживания специфического газового оборудования. Разработанное приложение позволит смоделировать некоторые аварийные ситуации, которые в реальной жизни могут привести к серьёзным издержкам или даже человеческим травмам, но благодаря созданному приложению, пользователь сможет без проблем увидеть эти нештатные ситуации, а также понять причины их появления и возможные способы их устранения.

Следует также отметить то, что на основе годового экономического эффекта, был посчитан срок окупаемости данного приложения, который составляет 1,7 года.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Технологии виртуальной реальности: методологические аспекты, достижения и перспективы: Национальный психологический журнал / Ю.П. Зинченко [и др.]; под редакцией Ю.П. Зинченко. – Москва: МГУ им. Ломоносова, 2010. – 148с.
2. Цифровизация производства: зачем *VR*/*AR* технологии нужны промышленности? – Электрон. данные. – Режим доступа: <https://modulab.com/industry>. – Дата доступа: 02.04.2023.
3. Технологии виртуальной и дополненной реальностей в энергетике. – Электрон. данные. – Режим доступа: <https://web.snauka.ru/issues/2022/06/98556>. – Дата доступа: 05.04.2023.
4. Алямовский А.А. *SolidWorks* 2007/2008. Компьютерное моделирование в инженерной практике / А.А. Алямовский, [и др.]; под ред. А.А. Алямовского. – СПб.: БХВ-Петербург, 2011. – 1040 с.
5. *Blender* 3*D*. – Электрон. данные. – Режим доступа: <https://mуdia.ru/blender3d/>. – Дата доступа: 07.04.2023.
6. Гибсон Б. Д. *Unity* и *C*#. Геймдев от идеи до реализации / Гибсон, Б. Д. – М.: СПб: Питер, 2019. – 228 c.
7. Хокинг Д. *Unity* в действии. Мультиплатформенная разработка на *C*# / Хокинг, Д. – М.: СПб: Питер, 2019. – 215 c.
8. Куксон А. Разработка игр на *Unreal* *Engine* 4 за 24 часа / Куксон, А. – М.: Бомбора, 2019. – 298 c.
9. Сравнение игровых движков [Электронный ресурс]. – 2017. – Режим доступа: <http://degam.com/sravnenieigrovyxdvizhkov>. – Дата доступа: 21.05.2023.
10. Кожевников, Е. А. Расчёт экономической эффективности разработки программных продуктов: метод. указания по подготовке организационно-экономического раздела дипломных работ для студентов специальности 1-40 01 02 «Информационные системы и технологии (по направлениям)» дневной формы обучения / Е. А. Кожевников, Н. В. Ермалинская. – Гомель: ГГТУ им. П. О. Сухого, 2012. – 68 с.
11. Барбинов Ф.А. Охрана окружающей среды; под общ. редакцией С.В. Белова, – 2-е изд., перераб.и доп. – Москва: Высш. шк., 1987. – 319 с.
12. Влияние шума на организм человека. – Электрон. данные. – Режим доступа: <https://ecosp.by/informatsiya/154-vliyanieshumanaorganizmcheloveka>. – Дата доступа: 08.05.2023.
13. Андрижиевский, А. А. Энергосбережение и энергетический менеджмент: учеб. пособие для студ. / А. А. Андрижиевский, В. И. Володин. – 2-е изд., испр. – Мн. : Вышэйшая школа, 2017. – 294 с.
14. Ляшенко, Д. Д. Ресурсосбережение: инновации в энергосбережении / Д. Д. Ляшенко, В. И. Мартынович // Международное научное обозрение проблем и перспектив современной науки и образования Сборник статей по материалам XXX Международной научно-практической конференции, 2018. – С. 31–32.
15. Организация энергосбережения (энергоменеджмент). Решения ЗСМК-НКМКНТМК-ЕВРАЗ : учебное пособие / под ред. В. В. Кондратьева – М.: ИНФРАМ, 2017. – 108 с.

# ПРИЛОЖЕНИЕ А

(обязательное)

**Экранные копии деталей бойлера**

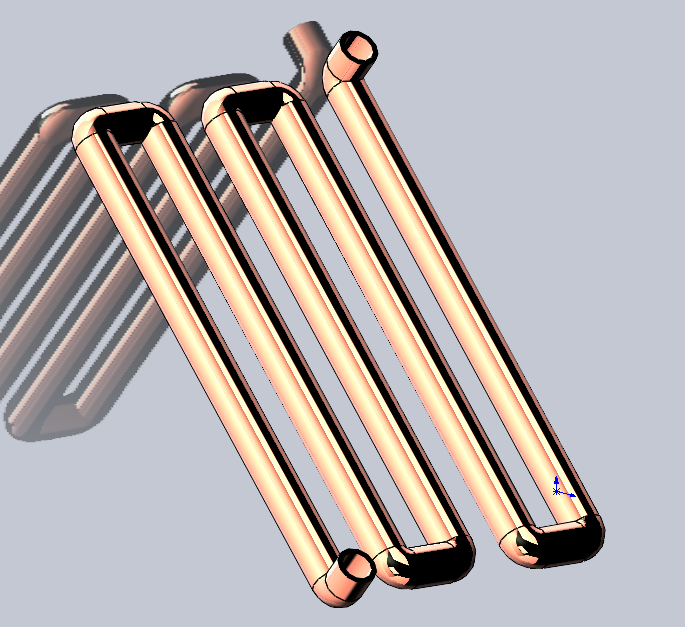


Рисунок А.1 – Изогнута труба теплообменника

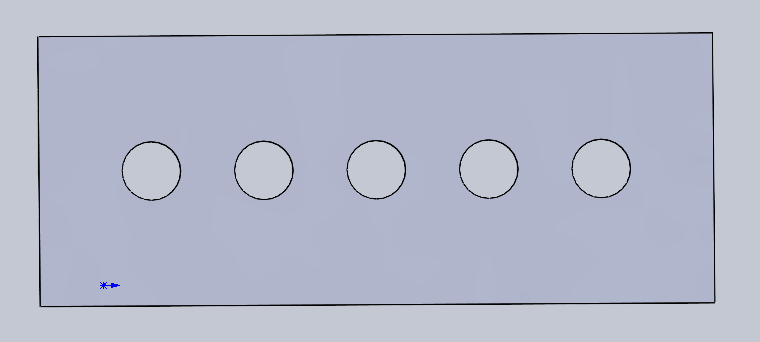


Рисунок А.2 – Внутренняя пластина теплообменника

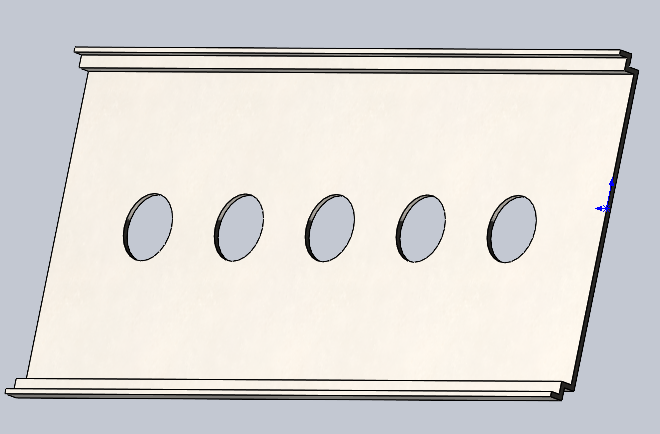


Рисунок А.3 – Внешняя пластина теплообменника



Рисунок А.4 – Воздуховод горелки

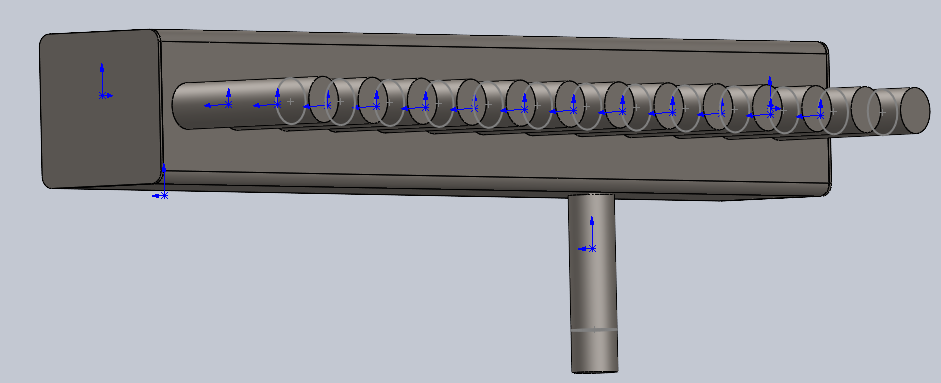


Рисунок А.5 – Газовый коллектор горелки

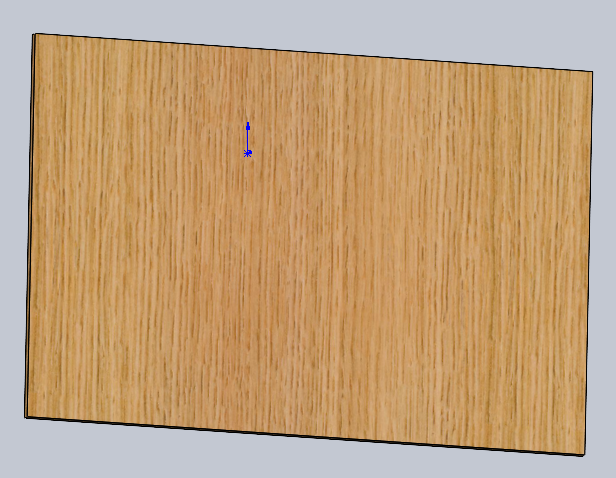


Рисунок А.6 – Одна из пластин теплоизоляционной камеры

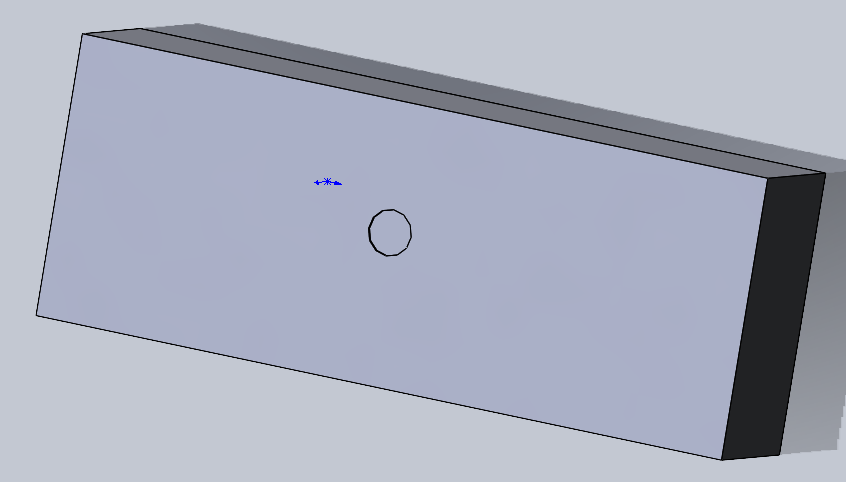


Рисунок А.7 – Камера сгорания газового бойлера

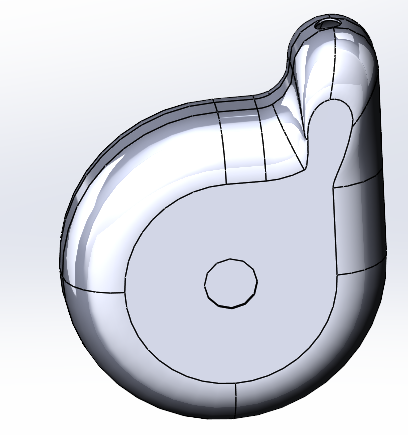


Рисунок А.8 – Труба вентури

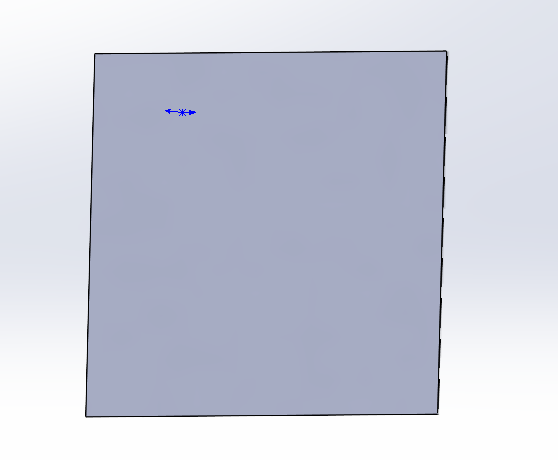


Рисунок А.9 – Стальная пластина, покрывающая теплообменник, и служащая основой для камеры сгорания с трубой вентури

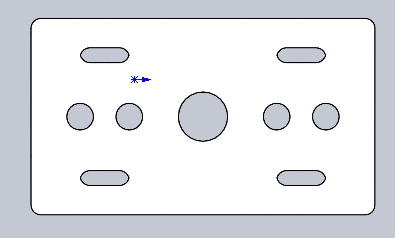


Рисунок А.10 – Нижняя пластина газового бойлера

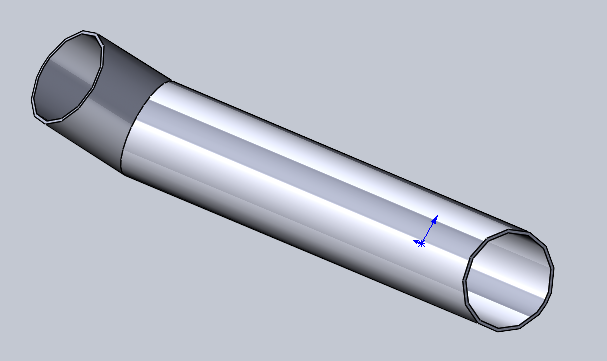


Рисунок А.11 – Водовод котла

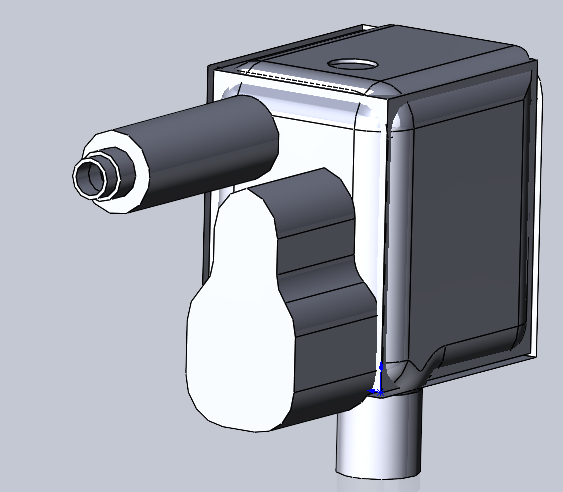


Рисунок А.12 – Газовый клапан

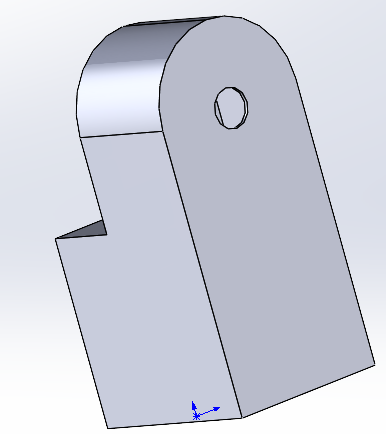


Рисунок А.13 – Блок розжига

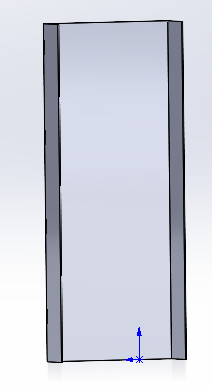


Рисунок А.14 – Боковая панель внешней обшивки

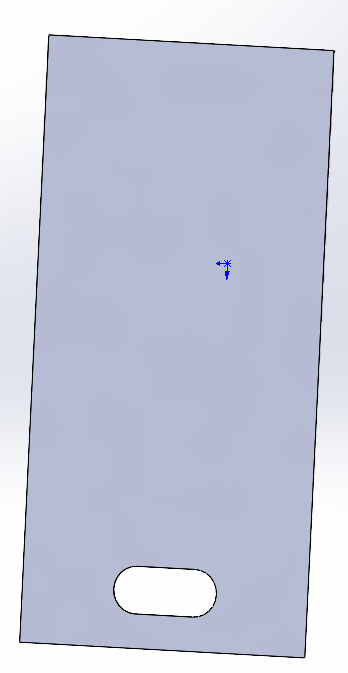


Рисунок А.15 – Фронтальная панель обшивки котла

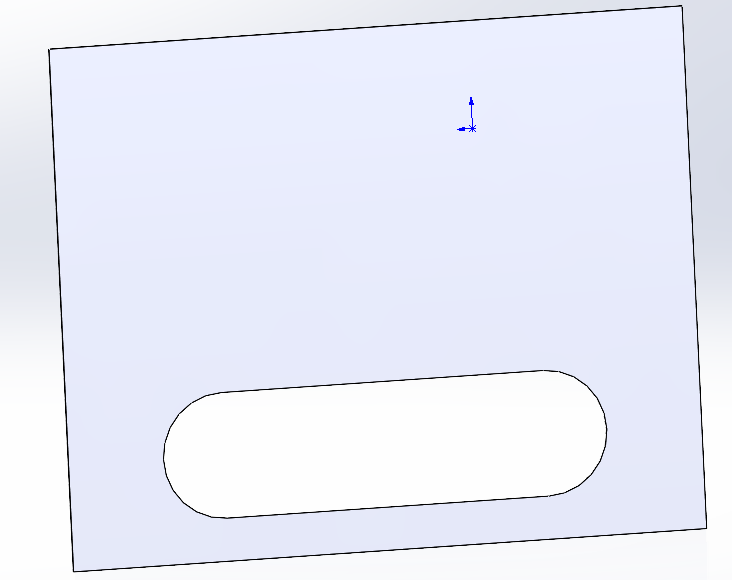


Рисунок А.16 – Верхняя панель котла

ПРИЛОЖЕНИЕ Б

(обязательное)

**Листинг программы приложения виртуальной реальности**

**Листинг класса *PushButtonFirstStand*.*cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.XR.Interaction.Toolkit.Samples.StarterAssets;

using UnityEngine.XR.Interaction.Toolkit;

using System;

/// <summary>

/// Клас, реализующий аварийную

/// ситуацию перегрева теплообменника

/// </summary>

public class PushButtonFirstStand : MonoBehaviour

{

/// <summary>

/// Описание события кнопки

/// </summary>

public static Action<int> OnButtonClick;

[SerializeField] private XRSimpleInteractable simpleInteractable;

[SerializeField] private Outline \_objectOutline;

[SerializeField] private int \_warningCode;

private bool \_isPressed = false;

private void Awake()

{

simpleInteractable.selectEntered.AddListener(OnSelectEntered);

}

/// <summary>

/// Проверка, что

/// кнопка нажата

/// </summary>

/// <param name="args"></param>

private void OnSelectEntered(SelectEnterEventArgs args)

{

if (!\_isPressed)

{

\_isPressed = true;

\_objectOutline.enabled = true;

OnButtonClick?.Invoke(\_warningCode);

}

else

{

\_isPressed = false;

\_objectOutline.enabled = false;

OnButtonClick?.Invoke(0);

}

}

}

**Листинг класса *PushButtonSecondStand.cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.XR.Interaction.Toolkit.Samples.StarterAssets;

using UnityEngine.XR.Interaction.Toolkit;

using System;

/// <summary>

/// Класс, демонстрирующий

/// задымлённость камеры сгорания

/// </summary>

public class PushButtonSecondStand : MonoBehaviour

{

public static Action<int> OnButtonClick;

[SerializeField] private XRSimpleInteractable simpleInteractable;

[SerializeField] private ParticleSystem \_smokeEffect;

[SerializeField] private Outline \_objectOutline;

[SerializeField] private int \_warningCode;

private bool \_isPressed = false;

private void Awake()

{

simpleInteractable.selectEntered.AddListener(OnSelectEntered);

}

/// <summary>

/// Проверка состояния кнопки

/// </summary>

/// <param name="args"></param>

private void OnSelectEntered(SelectEnterEventArgs args)

{

if (!\_isPressed)

{

\_isPressed = true;

\_objectOutline.enabled = true;

\_smokeEffect.Play();

OnButtonClick?.Invoke(\_warningCode);

}

else

{

\_isPressed = false;

\_objectOutline.enabled = false;

OnButtonClick?.Invoke(0);

\_smokeEffect.Stop();

}

}

}

**Листинг класса *PushButtonThirdStand.cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.XR.Interaction.Toolkit.Samples.StarterAssets;

using UnityEngine.XR.Interaction.Toolkit;

using System;

/// <summary>

/// Класс, реализующий неисправность

/// газового клапана

/// </summary>

public class PushButtonTherdStand : MonoBehaviour

{

public static Action<int> OnButtonClick;

[SerializeField] private XRSimpleInteractable simpleInteractable;

[SerializeField] private Outline \_objectOutline;

[SerializeField] private int \_warningCode;

private bool \_isPressed = false;

private void Awake()

{

simpleInteractable.selectEntered.AddListener(OnSelectEntered);

}

/// <summary>

/// Проверка состояния кнопки

/// </summary>

/// <param name="args"></param>

private void OnSelectEntered(SelectEnterEventArgs args)

{

if (!\_isPressed)

{

\_isPressed = true;

\_objectOutline.enabled = true;

OnButtonClick?.Invoke(\_warningCode);

}

else

{

\_isPressed = false;

\_objectOutline.enabled = false;

OnButtonClick?.Invoke(0);

}

}

}

**Листинг класса *PushButtonFourthStand.cs***

using System;

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.XR.Interaction.Toolkit;

/// <summary>

/// Класс, реализующий

/// неисправность электрода

/// </summary>

public class PushButtonFourthStand : MonoBehaviour

{

/// <summary>

/// Обработка события кнопки

/// </summary>

public static Action<int> OnButtonClick;

/// <summary>

/// Демонстрация неиспраности на котле

/// </summary>

[SerializeField] private XRSimpleInteractable simpleInteractable;

/// <summary>

/// Показ искр от электрода

/// </summary>

[SerializeField] private ParticleSystem \_flareEffect;

[SerializeField] private Outline \_objectOutline;

[SerializeField] private int \_warningCode;

private bool \_isPressed = false;

private void Awake()

{

simpleInteractable.selectEntered.AddListener(OnSelectEntered);

}

/// <summary>

/// Проверка, включена ли кнопка

/// </summary>

/// <param name="args"></param>

private void OnSelectEntered(SelectEnterEventArgs args)

{

if (!\_isPressed)

{

\_isPressed = true;

\_objectOutline.enabled = true;

\_flareEffect.Play();

OnButtonClick?.Invoke(\_warningCode);

}

else

{

\_isPressed = false;

\_objectOutline.enabled = false;

OnButtonClick?.Invoke(0);

\_flareEffect.Stop();

}

}

}

**Листинг класса *PushButtonFifthStand.cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.XR.Interaction.Toolkit.Samples.StarterAssets;

using UnityEngine.XR.Interaction.Toolkit;

using System;

/// <summary>

/// Класс для демонстрации

/// работы котла

/// </summary>

public class PushButtonFiveStand : MonoBehaviour

{

/// <summary>

/// Показание направлений

/// воды

/// </summary>

public static Action<bool> OnShowWaterFlow;

[SerializeField] private XRSimpleInteractable simpleInteractable;

[SerializeField] private ParticleSystem \_flareEffect;

private bool \_isPressed = false;

private void Awake()

{

simpleInteractable.selectEntered.AddListener(OnSelectEntered);

}

/// <summary>

/// Проверка, нажата

/// ли кнопка

/// </summary>

/// <param name="args"></param>

private void OnSelectEntered(SelectEnterEventArgs args)

{

if (!\_isPressed)

{

\_isPressed = true;

OnShowWaterFlow?.Invoke(true);

\_flareEffect.Play();

}

else

{

\_isPressed = false;

OnShowWaterFlow?.Invoke(false);

\_flareEffect.Stop();

}

}

}

**Листинг класса *PushButtonSixStand.cs***

using System;

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.XR.Interaction.Toolkit;

/// <summary>

/// Класс, реадизующий снижение

/// давления в контуре

/// </summary>

public class PushButtonSixStand : MonoBehaviour

{

public static Action<int> OnButtonClick;

[SerializeField] private XRSimpleInteractable simpleInteractable;

[SerializeField] private ParticleSystem \_smokeEffect;

[SerializeField] private int \_warningCode;

private bool \_isPressed = false;

private void Awake()

{

simpleInteractable.selectEntered.AddListener(OnSelectEntered);

}

/// <summary>

/// Проврека состояния кнопки

/// </summary>

/// <param name="args"></param>

private void OnSelectEntered(SelectEnterEventArgs args)

{

if (!\_isPressed)

{

\_isPressed = true;

\_smokeEffect.Play();

OnButtonClick?.Invoke(\_warningCode);

}

else

{

\_isPressed = false;

OnButtonClick?.Invoke(0);

\_smokeEffect.Stop();

}

}

}

**Листинг класса *XRPlatformControllerSetup.cs***

using UnityEngine;

#if UNITY\_EDITOR

using UnityEditor;

using UnityEditor.XR.Management;

#else

using UnityEngine.XR.Management;

#endif

namespace Unity.Template.VR

{

internal class XRPlatformControllerSetup : MonoBehaviour

{

[SerializeField]

GameObject m\_LeftController;

[SerializeField]

GameObject m\_RightController;

[SerializeField]

GameObject m\_LeftControllerOculusPackage;

[SerializeField]

GameObject m\_RightControllerOculusPackage;

void Start()

{

#if UNITY\_EDITOR

var loaders = XRGeneralSettingsPerBuildTarget.XRGeneralSettingsForBuildTarget(BuildTargetGroup.Standalone).Manager.activeLoaders;

#else

var loaders = XRGeneralSettings.Instance.Manager.activeLoaders;

#endif

foreach (var loader in loaders)

{

if (loader.name.Equals("Oculus Loader"))

{

m\_RightController.SetActive(false);

m\_LeftController.SetActive(false);

m\_RightControllerOculusPackage.SetActive(true);

m\_LeftControllerOculusPackage.SetActive(true);

}

}

}

}

}

**Листинг класса *XRGrabInteractableTwoAttach.cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.XR.Interaction.Toolkit;

public class XRGrabInteractableTwoAttach : XRGrabInteractable

{

public Transform leftAttachTransform;

public Transform rightAttachTransform;

protected override void OnSelectEntered(SelectEnterEventArgs args)

{

if(args.interactableObject.transform.CompareTag("Left Hand"))

{

attachTransform = leftAttachTransform;

}

if (args.interactableObject.transform.CompareTag("Right Hand"))

{

attachTransform = rightAttachTransform;

}

base.OnSelectEntered(args);

}

}

**Листинг класса *SetTurnType.cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.XR.Interaction.Toolkit;

public class SetTurnType : MonoBehaviour

{

public ActionBasedSnapTurnProvider snapTurn;

public ActionBasedContinuousTurnProvider continuousTurn;

public void SetTypeFromIndex(int index)

{

if(index == 0)

{

snapTurn.enabled = false;

continuousTurn.enabled = true;

}

else if(index == 1)

{

snapTurn.enabled = true;

continuousTurn.enabled = false;

}

}

}

**Листинг класса *MaterialTiling.cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class MaterialTiling : MonoBehaviour

{

public float \_speed = 1.0f; // Скорость изменения Offset Y

public Renderer[] \_renderers; // Массив рендереров с материалами

private float \_offset = 0.0f;

private IEnumerator \_waterFlow;

private void Awake()

{

PushButtonFiveStand.OnShowWaterFlow += IsStartWaterFlow;

}

private void IsStartWaterFlow(bool isStart)

{

if (\_waterFlow == null)

{

\_waterFlow = UpdateOffset();

}

if (isStart)

{

foreach (var a in \_renderers)

{

a.gameObject.SetActive(true);

}

StartCoroutine(UpdateOffset());

}

else

{

foreach (var a in \_renderers)

{

a.gameObject.SetActive(false);

}

StopAllCoroutines();

\_waterFlow = null;

}

}

private IEnumerator UpdateOffset()

{

while (true)

{

\_offset += \_speed \* Time.deltaTime;

// Создаем новый вектор с измененным значением Offset Y

Vector2 newOffset = new Vector2(0.0f, \_offset);

// Проходимся по всем рендерерам в массиве

for (int i = 0; i < \_renderers.Length; i++)

{

// Присваиваем новое значение Offset Y каждому материалу в рендерере

\_renderers[i].material.mainTextureOffset = newOffset;

}

yield return null;

}

}

}

**Листинг класса *ListObjectTransform.cs***

using System;

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.XR.Interaction.Toolkit;

using UnityEngine.XR.Interaction.Toolkit.Samples.StarterAssets;

public class ListObjectTransform : MonoBehaviour

{

private bool isFunctionRunning = false;

public XRSimpleInteractable simpleInteractable;

private bool typeS;

public XRPokeFollowAffordance XRPFA;

private bool logic = true;

public float duration = 5.0f;

private List<Vector3> listPObj = new List<Vector3>();

[Serializable]

public class KeyValuePair

{

public GameObject partOfModel;

public Vector3 val;

}

public List<KeyValuePair> MyList = new List<KeyValuePair>();

Dictionary<GameObject, Vector3> myDict = new Dictionary<GameObject, Vector3>();

void Awake()

{

Debug.Log(simpleInteractable);

// Подписываемся на событие "входа" (selectEntered)

simpleInteractable.selectEntered.AddListener(OnSelectEntered);

foreach (var kvp in MyList)

{

myDict[kvp.partOfModel] = kvp.val;

}

foreach (KeyValuePair<GameObject, Vector3> kvp in myDict)

{

listPObj.Add(kvp.Key.transform.position);

}

}

private void OnSelectEntered(SelectEnterEventArgs args)

{

if (isFunctionRunning)

{

// Если функция уже выполняется, прекращаем выполнение

return;

}

StartCoroutine(Logic());

}

public void TransformObj(float duration)

{

foreach (KeyValuePair<GameObject, Vector3> kvp in myDict)

{

StartCoroutine(MoveSmoothly(kvp.Key.transform, kvp.Value, duration, true));

//kvp.Key.transform.localPosition = new Vector3(kvp.Value.x, kvp.Value.y, kvp.Value.z);

}

}

public IEnumerator MoveSmoothly(Transform target, Vector3 targetPosition, float duration, bool typeS)

{

float elapsedTime = 0.0f;

Vector3 initialPosition;

if (typeS)

{

initialPosition = target.localPosition;

}

else

{

initialPosition = target.position;

}

while (elapsedTime < duration)

{

if (typeS)

{

target.localPosition = Vector3.Lerp(initialPosition, targetPosition, elapsedTime / duration);

}

else

{

target.position = Vector3.Lerp(initialPosition, targetPosition, elapsedTime / duration);

}

elapsedTime += Time.deltaTime;

yield return null;

}

if (typeS)

{

target.localPosition = targetPosition;

}

else

{

target.position = targetPosition;

}

}

public void BackTransformObj(float duration)

{

int i = 0;

foreach (KeyValuePair<GameObject, Vector3> kvp in myDict)

{

StartCoroutine(MoveSmoothly(kvp.Key.transform, listPObj[i], duration, false));

//kvp.Key.transform.position = new Vector3(listPObj[i].x, listPObj[i].y, listPObj[i].z);

i++;

}

}

private IEnumerator Logic()

{

isFunctionRunning = true;

if (logic)

{

TransformObj(duration);

logic = false;

}

else

{

BackTransformObj(duration);

logic = true;

}

Debug.Log("Выполняется функция Unity");

yield return new WaitForSeconds(duration);

// Функция завершена, сбрасываем флаг

isFunctionRunning = false;

Debug.Log("Функция Unity завершена");

}

private void OnTriggerEnter(Collider other)

{

if (other.CompareTag("Player") && logic == false)

{

BackTransformObj(1.0f);

logic = true;

}

}

}

**Листинг класса *TriggerHaptic.cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.XR.Interaction.Toolkit;

[System.Serializable]

public class Haptic

{

[Range(0, 1)]

public float intensity;

public float duration;

public void TriggerHaptic(BaseInteractionEventArgs eventArgs)

{

if (eventArgs.interactableObject is XRBaseControllerInteractor controllerInteractor)

{

TriggerHaptic(controllerInteractor.xrController);

}

}

public void TriggerHaptic(XRBaseController controller)

{

if (intensity > 0)

{

controller.SendHapticImpulse(intensity, duration);

}

}

}

public class HapticInteractable : MonoBehaviour

{

public Haptic hapticOnActivated;

public Haptic hapticHoverEntered;

public Haptic hapticHoverExited;

public Haptic hapticSelectEntered;

public Haptic hapticSelectExited;

// Start is called before the first frame update

void Start()

{

XRBaseInteractable interactable = GetComponent<XRBaseInteractable>();

interactable.activated.AddListener(hapticOnActivated.TriggerHaptic);

interactable.hoverEntered.AddListener(hapticHoverEntered.TriggerHaptic);

interactable.hoverExited.AddListener(hapticHoverExited.TriggerHaptic);

interactable.hoverEntered.AddListener(hapticSelectEntered.TriggerHaptic);

interactable.hoverExited.AddListener(hapticSelectExited.TriggerHaptic);

}

}

**Листинг класса *GameMenuManager.cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.InputSystem;

public class GameMenuManager : MonoBehaviour

{

public Transform head;

public float spawnDistance = 2;

public GameObject menu;

public InputActionProperty showButton;

private void Start()

{

}

private void Update()

{

if(showButton.action.WasPerformedThisFrame())

{

menu.SetActive(!menu.activeSelf);

menu.transform.position = head.position + new Vector3(head.forward.x, 0, head.forward.z).normalized \* spawnDistance;

}

menu.transform.LookAt(new Vector3(head.position.x, menu.transform.position.y, head.position.z));

menu.transform.forward \*= -1;

}

}

**Листинг класса *GameMenu.cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class GameMenu : MonoBehaviour

{

public void GoToScene()

{

SceneManager.LoadScene("XR Toolkit VR");

}

public void ExitGame()

{

Application.Quit();

}

}

**Листинг класса *FireBulletOnActivate.cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.XR.Interaction.Toolkit;

public class FireBulletOnActivate : MonoBehaviour

{

public GameObject bullet;

public Transform spawnPoint;

public float fireSpeed = 20;

// Start is called before the first frame update

void Start()

{

XRGrabInteractable grabble = GetComponent<XRGrabInteractable>();

grabble.activated.AddListener(FireBullet);

}

// Update is called once per frame

void Update()

{

}

public void FireBullet(ActivateEventArgs arg)

{

GameObject spawnedBullet = Instantiate(bullet);

spawnedBullet.transform.position = spawnPoint.position;

spawnedBullet.GetComponent<Rigidbody>().velocity = spawnPoint.forward \* fireSpeed;

Destroy(spawnedBullet, 5);

}

}

**Листинг класса *BoilUI.cs***

using System.Collections;

using System.Collections.Generic;

using TMPro;

using UnityEngine;

public class BoilUI : MonoBehaviour

{

[SerializeField] private TextMeshProUGUI \_warningText;

private void Awake()

{

PushButtonFirstStand.OnButtonClick += ChangeText;

PushButtonSecondStand.OnButtonClick += ChangeText;

PushButtonTherdStand.OnButtonClick += ChangeText;

PushButtonFourthStand.OnButtonClick += ChangeText;

PushButtonSixStand.OnButtonClick += ChangeText;

}

private void ChangeText(int code)

{

if (code != 0)

{

\_warningText.text = code.ToString();

}

else

{

\_warningText.text = " ";

}

}

}

**Листинг класса *AnimateHandOnInput.cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.InputSystem;

public class AnimateHandOnInput : MonoBehaviour

{

public InputActionProperty pinchAnimationAction;

public InputActionProperty gripAnimationAction;

public Animator handAnimator;

// Update is called once per frame

void Update()

{

float triggerValue = pinchAnimationAction.action.ReadValue<float>();

handAnimator.SetFloat("Trigger", triggerValue);

float gripValue = gripAnimationAction.action.ReadValue<float>();

handAnimator.SetFloat("Grip", gripValue);

}

}

**Листинг класса *ActivateTeleportationRay.cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.InputSystem;

using UnityEngine.XR.Interaction.Toolkit;

public class ActivateTeleportationRay : MonoBehaviour

{

public GameObject leftTeleportation;

public GameObject rightTeleportation;

public InputActionProperty leftActive;

public InputActionProperty rightActive;

public InputActionProperty leftChanel;

public InputActionProperty rightChanel;

public XRRayInteractor leftRay;

public XRRayInteractor rightRay;

// Update is called once per frame

void Update()

{

bool isLeftRayHovering = leftRay.TryGetHitInfo(out Vector3 leftPos, out Vector3 leftNoraml, out int leftNormal, out bool leftValid);

leftTeleportation.SetActive(!isLeftRayHovering && leftChanel.action.ReadValue<float>() == 0 && leftActive.action.ReadValue<float>() > 0.1f);

bool isRightRayHovering = rightRay.TryGetHitInfo(out Vector3 rightPos, out Vector3 rightNoraml, out int rightNormal, out bool rightValid);

rightTeleportation.SetActive(!isRightRayHovering && rightChanel.action.ReadValue<float>() == 0 && rightActive.action.ReadValue<float>() > 0.1f);

}

}

**Листинг класса *ActivateGrapRay.cs***

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.XR.Interaction.Toolkit;

public class ActivateGrapRay : MonoBehaviour

{

public GameObject leftGrabRay;

public GameObject rightGrabRay;

public XRDirectInteractor leftDirectGrab;

public XRDirectInteractor rightDirectGrab;

// Start is called before the first frame update

void Start()

{

}

// Update is called once per frame

void Update()

{

leftGrabRay.SetActive(leftDirectGrab.interactablesSelected.Count == 0);

rightGrabRay.SetActive(rightDirectGrab.interactablesSelected.Count == 0);

}

}

ПРИЛОЖЕНИЕ В

(обязательное)

**Руководство программиста**

**1 Назначение и условия применения программы**

**1.1 Назначение и функции программы**

Разработанное приложение виртуальной реальности представляет из себя симулятор основных нештатных ситуаций, которые могут произойти в ходе эксплуатации газового бойлера. В приложении реализован интерактивный доступ к основным деталям газового бойлера, а также возможность увидеть некоторые нештатные ситуации и способы их разрешения.

**1.2 Необходимое техническое обеспечение**

Для корректной работы программного средства необходимо соблюдение следующих требований:

* поддерживаемые операционные системы *Windows* версии десять и выше;
* наличие стандартной клавиатуры и мыши;
* процессор *Intel* *Core* *i*5 4590 (4-го поколения), четырехъядерный (или более поздней версии), либо *AMD* *Ryzen* 5 1400 3,4 ГГц (настольный), четырехъядерный (или более поздней версии);
* оперативная память 8 ГБ *DDR*3 (или выше);
* свободное пространство на диске не менее 10 ГБ;
* графический процессор *NVIDIA* *GTX* 1060 (или более поздней версии) с поддержкой *DX*12 дискретного *GPU*, либо *AMD* *RX* 470/570 (или более поздней версии) с поддержкой DX12 дискретного *GPU*;
* графический драйвер *Windows* *Display* *Driver* *Model* (*WDDM*) 2.2;
* порт отображение графики *HDMI* 2.0 или *DisplayPort* 1.2;
* для отображения должен быть подключенный внешний или интегрированный дисплей *VGA* (800x600) (или более поздней версии);
* *usb*-подключение *USB* 3.0;
* подключение *Bluetooth* (для контроллеров движения) должно быть *Bluetooth* 4.0.

**2 Характеристики программы**

При запуске приложении с подключённым гарнитуром виртуальной реальности, пользователь оказывается в помещении, стилизованном под кухню, в котором находится газовый бойлер

**3 Обращение к программе**

Приложение запускается путем открытия исполняющегося файла *UnityVr.exe*, который находится в каталоге с игрой.

**4 Входные и выходные данные**

Входной информацией являются данные, которые предоставляют игроки путём нажатия на кнопки контроллера. Выходной информацией являются кадры на экране с результатом игрового процесса.

**5 Сообщения**

В ходе выполнения приложения никаких сообщений не предусмотрено.

ПРИЛОЖЕНИЕ Г

(обязательное)

**Руководство системного программиста**

**1 Общие сведения о программе**

**1.1 Назначение и функции программы**

Разработанное приложение виртуальной реальности представляет из себя симулятор основных нештатных ситуаций, которые могут произойти в ходе эксплуатации газового бойлера. В приложении реализован интерактивный доступ к основным деталям газового бойлера, а также возможность увидеть некоторые нештатные ситуации и способы их разрешения.

**1.2 Необходимое техническое обеспечение**

Для корректной работы программного средства необходимо соблюдение следующих требований:

* поддерживаемые операционные системы *Windows* версии десять и выше;
* наличие стандартной клавиатуры и мыши;
* процессор *Intel* *Core* *i*5 4590 (4-го поколения), четырехъядерный (или более поздней версии), либо *AMD* *Ryzen* 5 1400 3,4 ГГц (настольный), четырехъядерный (или более поздней версии);
* оперативная память 8 ГБ *DDR*3 (или выше);
* свободное пространство на диске не менее 10 ГБ;
* графический процессор *NVIDIA* *GTX* 1060 (или более поздней версии) с поддержкой *DX*12 дискретного *GPU*, либо *AMD* *RX* 470/570 (или более поздней версии) с поддержкой DX12 дискретного *GPU*;
* графический драйвер *Windows* *Display* *Driver* *Model* (*WDDM*) 2.2;
* порт отображение графики *HDMI* 2.0 или *DisplayPort* 1.2;
* для отображения должен быть подключенный внешний или интегрированный дисплей *VGA* (800x600) (или более поздней версии);
* *usb*-подключение *USB* 3.0;
* подключение *Bluetooth* (для контроллеров движения) должно быть *Bluetooth* 4.0.

**2 Структура программы**

Программа представляет из себя приложение виртуальной реальности, где помещение имеет интерьер кухни, в котором располагается газовый бойлер. Пользователь может нажимать на кнопки, расположенные по всей площади помещения, и может увидеть внутренне содержимое бойлера, ознакомиться с принципом его работы, а также увидеть основные аварийные ситуации и способы их разрешения.

**3 Настройка программы**

Настройка ПО осуществляется путем запуска исполняемого файла.

**4 Проверка программы**

Отсутствие сообщения об ошибках свидетельствует, о корректной работе приложения.

**6 Сообщения системному программисту**

Для успешного запуска программы на ПЭВМ необходимо установить все необходимые компоненты через портал смешанной реальности *Windows*. Для этого необходимо подсоединить очки виртуальной реальности к компьютеру и привязать контроллеры при помощи *Bluetooth*.

ПРИЛОЖЕНИЕ Д

(обязательное)

**Руководство пользователя**

**1 Введение**

Руководство пользователя информирует пользователя базовыми знаниями по эксплуатации игрового приложения.

Приложение предназначено для организации процесса обучения по обслуживанию и эксплуатации газового бойлера *Aristone* *cares* *xc*-*hs*.

Программное средство обладает следующим функционалом:

* возможность передвигаться по помещению, где расположен газовый бойлер;
* изучить основные детали бойлера, «подержать их в руках»;
* увидеть анимацию ключевых аварийных ситуаций;
* узнать основные способы устранения нештатных ситуаций.

Ход использования приложения:

Для использования программного средство пользователь должен быть ознакомлен с:

– настоящим руководством пользователя;

– правилами использования ПК.

**2 Назначение и условия применения**

Разработанное приложение виртуальной реальности представляет из себя симулятор основных нештатных ситуаций, которые могут произойти в ходе эксплуатации газового бойлера. В приложении реализован интерактивный доступ к основным деталям газового бойлера, а также возможность увидеть некоторые нештатные ситуации и способы их разрешения.

Для корректной работы программного средства необходимо соблюдение следующих требований:

* поддерживаемые операционные системы *Windows* версии десять и выше;
* наличие стандартной клавиатуры и мыши;
* процессор *Intel* *Core* *i*5 4590 (4-го поколения), четырехъядерный (или более поздней версии), либо *AMD* *Ryzen* 5 1400 3,4 ГГц (настольный), четырехъядерный (или более поздней версии);
* оперативная память 8 ГБ *DDR*3 (или выше);
* свободное пространство на диске не менее 10 ГБ;
* графический процессор *NVIDIA* *GTX* 1060 (или более поздней версии) с поддержкой *DX*12 дискретного *GPU*, либо *AMD* *RX* 470/570 (или более поздней версии) с поддержкой DX12 дискретного *GPU*;
* графический драйвер *Windows* *Display* *Driver* *Model* (*WDDM*) 2.2;
* порт отображение графики *HDMI* 2.0 или *DisplayPort* 1.2;
* для отображения должен быть подключенный внешний или интегрированный дисплей *VGA* (800x600) (или более поздней версии);
* *usb*-подключение *USB* 3.0;
* подключение *Bluetooth* (для контроллеров движения) должно быть *Bluetooth* 4.0.

**3 Подготовка к работе**

Для запуска приложения необходимо запустить скомпилированный исполняемый файл *UnityVr.exe* из каталога с игрой.

Если все инструкции соблюдены, и приложение не выдает никаких сообщений об ошибках, значит, программа работает исправно.

**4 Описание операций**

При запуске с подключённым гарнитуром, пользователь оказывается в помещении с газовым бойлером. Вид помещения приведён на рисунке Д.1.

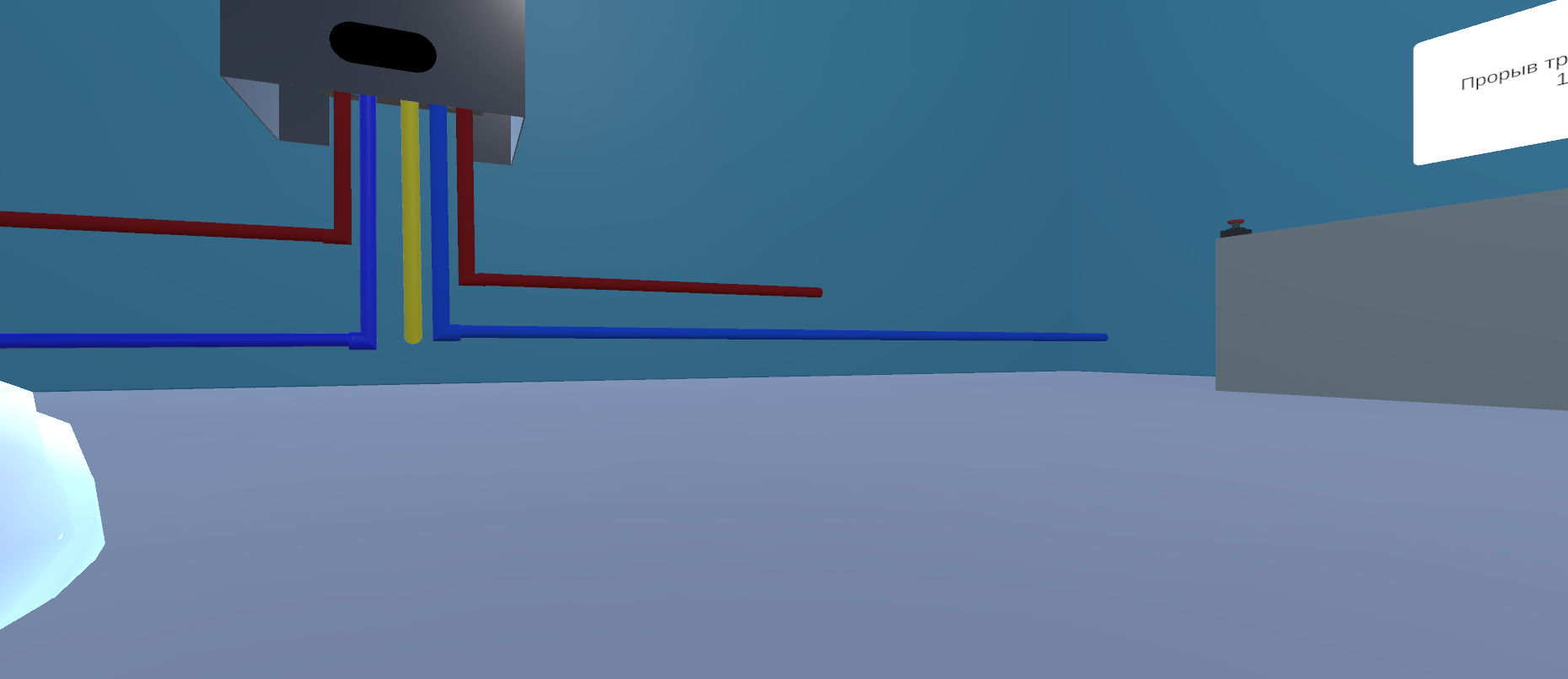


Рисунок Д.1 – Общий вид на старте игрового приложения виртуальной

реальности

Пользователь может увидеть в бойлер в более детальном виде. Панель с кнопкой для получения детального вида приведена на рисунке Д.2.

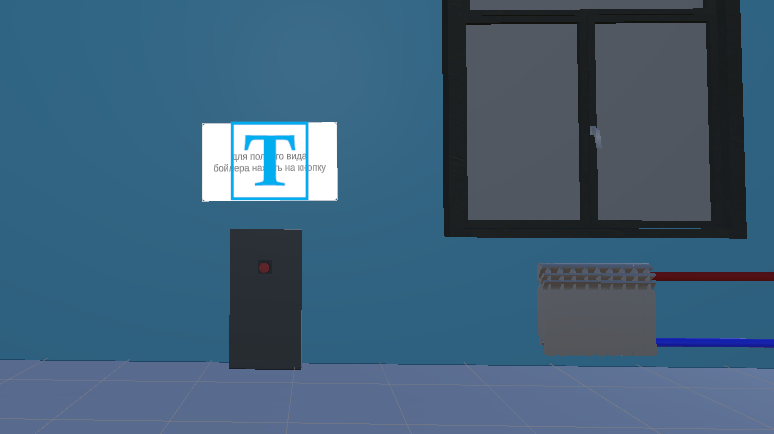


Рисунок Д.2 – Кнопка для получения детального вида котла

Пользователь может увидеть общий вид работы котла. На рисунке Д.3 показан котёл во время имитации работы.

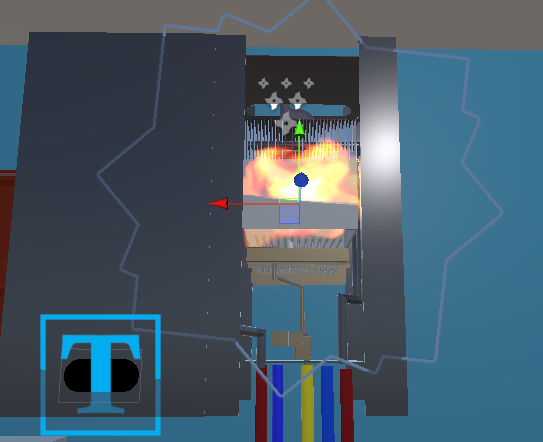


Рисунок Д.3 – Имитация работы бойлера

Для получения информации об аварийных ситуациях, в помещении находятся стенды, где пользователь может изучить макеты неисправных деталей, прочитать информации об аварийной ситуации и возможных способах её решения, а также, нажав на кнопку, которая находится возле каждого стенда, увидеть анимацию вызванной аварийной ситуации внутри газового бойлера. На рисунке Д.4 показаны стенды с информацией об аварийных ситуациях газового бойлера.

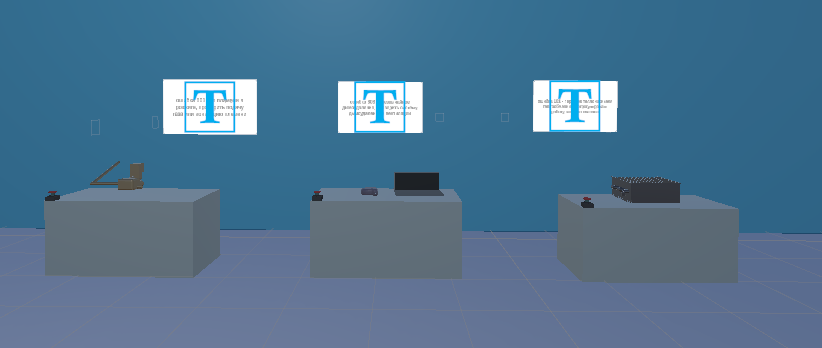


Рисунок Д.4 – Общий вид стенда с информацией об аварийных ситуациях

**5 Аварийные ситуации**

Чтобы избежать ошибок при использовании данного игрового приложения, необходимо соблюдать порядок действий и условия использования, описанные в разделе 3 данного руководства пользователя.

В случае непредвиденного «зависания» программы рекомендуется завершить процесс в диспетчере задач и запустить снова. Или можно подождать некоторое время, так как существует возможность зависания программы за счет наличия библиотек, используемых в игровом приложении.

**6 Рекомендации по освоению**

Игровое приложение необходимо запускать на компьютере, на котором выполнены все необходимые настройки по разворачиванию приложения виртуальной реальности, а также компьютер соответствует предъявляемым системным требованиям. В противном случае, стабильная работа приложения не гарантируется.

ПРИЛОЖЕНИЕ Е

(справочное)

**Текст публикации**

**П.В. Климанский, В.В. Комраков**

*(ГГТУ им. П.О. Сухого, Гомель)*

**СТРУКТУРА АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА**

**МЕНЕДЖЕРА-ЛОГИСТА В ПУ «СВЯЗЬИНФОРМСЕРВИС» РУП**

**«БЕЛОРУСНЕФТЬ»**

Деятельность всех предприятий сегодня связана с необходимостью доставки грузов. Сегодня процесс транспортировки груза может осложняться высокой стоимостью логистического цикла, долгим документооборотом и сохранностью груза. Для осуществления грузоперевозок по территории Беларуси РУП «Белоруснефть» учитывается региональный фактор, поскольку в разных районах различный спрос на грузоперевозки, а также климатическую и дорожную обстановку.

Для разработки приложения применялся объектно-ориентированный язык *C*#. Данный язык позволяет реализовать производительные приложения любой сложности с удобным интерфейсом. Для графического интерфейса пользователя использован *WPF* (*Windows* *Presentation* *Foundation*). Платформа *WPF* позволяет создавать приложения под множество операционных систем семейства *Windows*.

Одним из достоинств языка *C*# является возможность организации удобного и быстрого доступа к базе данных, которая используется в разработанном приложении. Для доступа к данным использована технология *Entity* *Framework* *Core* (*EF* *Core*), который представляет собой объектно-ориентированную, легковесную и расширяемую технологию от компании Microsoft для доступа к данным. *EF* *Core* является *ORM*-инструментом (*object*-*relational* *mapping* – отображения данных на реальные объекты). Таким образом, через *EF* *Core* можно работать с любой системой управления базой данных (СУБД), если для нее имеется нужный провайдер.

Разработанный программный продукт позволит РУП «Белоруснефть» сделать процесс грузоперевозок более быстрым и прозрачным. Возможность учёта спроса на грузоперевозки по каждому району позволит избежать простоя техники, и перебросить в другой район при острой необходимости, что сократит издержки при перевозке грузов.