

-Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Е. Н. Скуратенко
подпись
«_____» _____ 2021 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

Разработка обучающей игры с элементами квеста для абитуриентов,
поступающих на направление
«Прикладная информатика»

Руководитель _____ доцент, канд. физ.-мат. наук А.Н. Таскин
подпись, дата

Выпускник _____ Д. В. Зобов
подпись, дата

Консультанты
по разделам:

Экономический _____ Е. Н. Скуратенко
подпись, дата

Нормоконтролер _____ В. И. Кокова
подпись, дата

Абакан 2021

Студенту Зобову Даниилу Викторовичу

Группа ХБ 17-03

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Разработка обучающей игры с элементами квеста для абитуриентов, поступающих на направление «Прикладная информатика»

Утверждена приказом по институту № 222 от 08.04.2021 г.

Руководитель ВКР: А. Н. Таскин, доцент, канд. физ.-мат. наук, ХТИ – филиал СФУ

Исходные данные для ВКР: заказ ХТИ – филиала СФУ.

Перечень разделов ВКР:

1. Анализ предметной области проекта «Игра-квест».
2. Описание процесса разработки информационной системы «Игра-квест».
3. Оценка экономической эффективности разработки и внедрения проекта «Игра-квест».

Перечень графического материала: нет

Руководитель ВКР

подпись

А. Н. Таскин

Задание принял к исполнению

подпись

Д. В. Зобов

«08» апреля 2021 г.

РЕФЕРАТ

Выпускная квалификационная работа (ВКР) на тему «Разработка обучающей игры с элементами квеста для абитуриентов, поступающих на направление «Прикладная информатика» содержит 55 страниц текстового документа, 7 формул, 6 таблиц, 42 рисунка, 19 использованных источников.

ПРОЕКТ, ИНФОРМАЦИОННАЯ СИСТЕМА, АНАЛИЗ, ИГРА, ЗАТРАТЫ, ЭФФЕКТИВНОСТЬ, РИСК, КВЕСТ, РАЗРАБОТКА.

Объектом ВКР является процесс обучения с помощью игры.

Предмет: обучение с помощью игры.

Цель ВКР: разработать обучающую игру с элементами квеста для абитуриентов, поступающих на направление «Прикладная информатика».

Задачи ВКР: провести сбор и анализ информации о ХТИ – филиале СФУ; построить функциональную модель разработки игры; определить цель и задачи разработки системы «Игра-квест»; провести анализ существующих обучающих игр; провести теоретический анализ и выбрать программные средства разработки информационной системы; разработать обучающую игру; оценить экономическую эффективность от внедрения разработанного программного продукта.

Проанализирована основная деятельность ХТИ – филиала СФУ. С учетом данных анализа существующих программных продуктов был выбран GameMaker Studio 2. Разработан и описан программный продукт «Игра-квест». Выполнена оценка экономической эффективности и рисков проекта.

SUMMARY

The theme of the Bachelor's thesis: «Educational Game Development: Quest for Applicants of “Applied Informatics” Training Program». It contains 55 pages of a text document, 7 formulae, 6 tables, 42 figures, 19 reference items.

PROJECT, IT SYSTEM, ANALYSIS, GAME, COSTS, EFFICIENCY, RISKS, QUEST, DEVELOPMENT.

Object: learning process through games.

Subject: learning through playing.

Purpose: to develop an educational game-quest for applicants entering the “Applied Informatics” Training Program.

Objectives: to collect and to analyze information about KhTI - branch of Siberian Federal University; to build a functional model for game development; to determine the purpose and objectives of the development of the Game-Quest system; to analyze existing educational games; to carry out theoretical analysis and to select software tools for developing an IT system; to develop an educational game; to calculate economic efficiency from the implementation of the developed software product.

The main workflow activity of KhTI - branch of Siberian Federal University has been analyzed. Taking into account the data of the analysis of existing software products, GameMaker Studio 2 has been selected. The software product “Game-Quest” has been developed and described. The cost-effectiveness and risks of the project have been calculated.

English language supervisor:

signature, date

N.V. Chezybaeva
(full name)

СОДЕРЖАНИЕ

| | |
|---|----|
| ВВЕДЕНИЕ..... | 8 |
| 1 Анализ предметной области проекта «Игра-квест»..... | 9 |
| 1.1 Организационная характеристика предприятия..... | 9 |
| 1.2 Бизнес-процессы предметной области и обоснование разработки проекта..... | 12 |
| 1.3 Примеры обучающих игр..... | 12 |
| 1.4 Обоснование использования игры | 14 |
| 1.5 Обзор и обоснование использования ПО | 14 |
| 1.6 Модель разработки игры-квеста..... | 16 |
| 1.7 Выводы по аналитическому разделу | 19 |
| 2 Описание процесса разработки информационной системы «Игра-квест» | 20 |
| 2.1 Планирование и определение функций. Декомпозиция на подсистемы | 20 |
| 2.2 Запуск программного обеспечения | 21 |
| 2.3 Создание проекта в Game Maker Studio 2..... | 23 |
| 2.4 Выводы по разделу «Описание процесса разработки приложения «Игра-квест» | 41 |
| 3 Оценка экономической эффективности внедрения и разработки проекта «Игра-квест» | 41 |
| 3.1 Методика оценки экономической эффективности разработки игры- квеста | 41 |
| 3.1.1 Капитальные затраты..... | 42 |
| 3.1.2 Затраты на разработку | 42 |
| 3.1.3 Эксплуатационные затраты | 48 |
| 3.1.4 Расчет ТСО | 48 |

| | |
|---|----|
| 3.2 Оценка рисков при реализации проекта «Игра-квест»..... | 50 |
| 3.3 Выводы по разделу «Оценка экономической эффективности внедрения и разработки проекта «Игра-квест»..... | 51 |
| ЗАКЛЮЧЕНИЕ | 52 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 53 |

ВВЕДЕНИЕ

Тема игр в обучении представляет теоретический и практический интерес, потому что с развитием и расширением многих игроиндустрий в сфере развлечений и досуга игры стали очень популярны, и с помощью игр можно привлечь внимание не только подрастающего поколения, но и взрослого.

Способность играть и обучаться не только делает времяпрепровождение более увлекательным, но и полезным. Это связано с восприятием подаваемой информации по частям, которые будут понятны неопытному пользователю, и помогут лучше углубиться в изучение тех или иных научных предметов.

Проблема состоит только в отсутствии заинтересованности аудитории в целом в изучении некоторых предметов, восприятие его как сложного, непонятного обывателям процесса.

Объектом ВКР является процесс обучения с помощью игры.

Предмет ВКР – обучение с помощью игры.

Цель ВКР: разработать обучающую игру с элементами квеста для абитуриентов, поступающих на направление «Прикладная информатика».

Для достижения цели ВКР были определены следующие задачи:

1. Провести сбор и анализ информации о ХТИ – филиале СФУ.
2. Построить функциональную модель разработки игры.
3. Определить цель и задачи разработки системы “Игра-квест”.
4. Провести анализ существующих обучающих игр.
5. Провести теоретический анализ и выбрать программные средства разработки информационной системы.
6. Разработать обучающую игру.
7. Оценить экономическую эффективность от внедрения разработанного программного продукта.

1 Анализ предметной области проекта «Игра-квест»

1.1 Организационная характеристика предприятия

Заказчиком проекта «Игра-квест» является Хакасский технический институт – филиал федерального государственного автономного образовательного учреждения высшего образования «Сибирский федеральный университет», которому необходим программный продукт, предназначенный для помощи в привлечении внимания абитуриентов к направлениям подготовки ВУЗа.

Краткое наименование: ХТИ – филиал СФУ.

Юридический адрес учреждения: 655017, Республика Хакасия, г. Абакан, ул. Щетинкина, 27 (корпус “А”).

Контактный телефон: (8-3902) 22-53-55.

Главной задачей учебного заведения ХТИ – филиал СФУ является создание передовой образовательной, научно-исследовательской и инновационной инфраструктуры, продвижение новых знаний и технологий для решения задач социально-экономического развития Сибирского федерального округа, а также формирование кадрового потенциала – конкурентоспособных специалистов по приоритетным направлениям развития Сибири и Российской Федерации, соответствующих современным интеллектуальным требованиям и отвечающих мировым стандартам.

Институт имеет сертификат соответствия системы менеджмента качества требованиям ГОСТ ISO 9001-2015 (ISO 9001:2015) применительно к деятельности в области образования.

Институт осуществляет подготовку по очной, очно-заочной и заочной формам по 1 направлению специалитета и 6 направлениям бакалавриата, 2 направлениям магистратуры.

Ведется подготовка специалистов и бакалавров по следующим направлениям:

- 08.03.01 Строительство.
- 08.05.01 Строительство уникальных зданий и сооружений.
- 09.03.03 Прикладная информатика.
- 13.03.02 Электроэнергетика и электротехника.
- 15.03.05 Конструкторско-технологическое обеспечение машиностроительных производств.
- 23.03.03 Эксплуатация транспортно-технологических машин и комплексов.
- 38.03.01 Экономика.

По окончании обучения лицам, успешно прошедшим государственную итоговую аттестацию, выдается документ об образовании и о квалификации, подтверждающий получение высшего образования.

Структурная схема ХТИ – филиала СФУ продемонстрирована на рисунке 1.

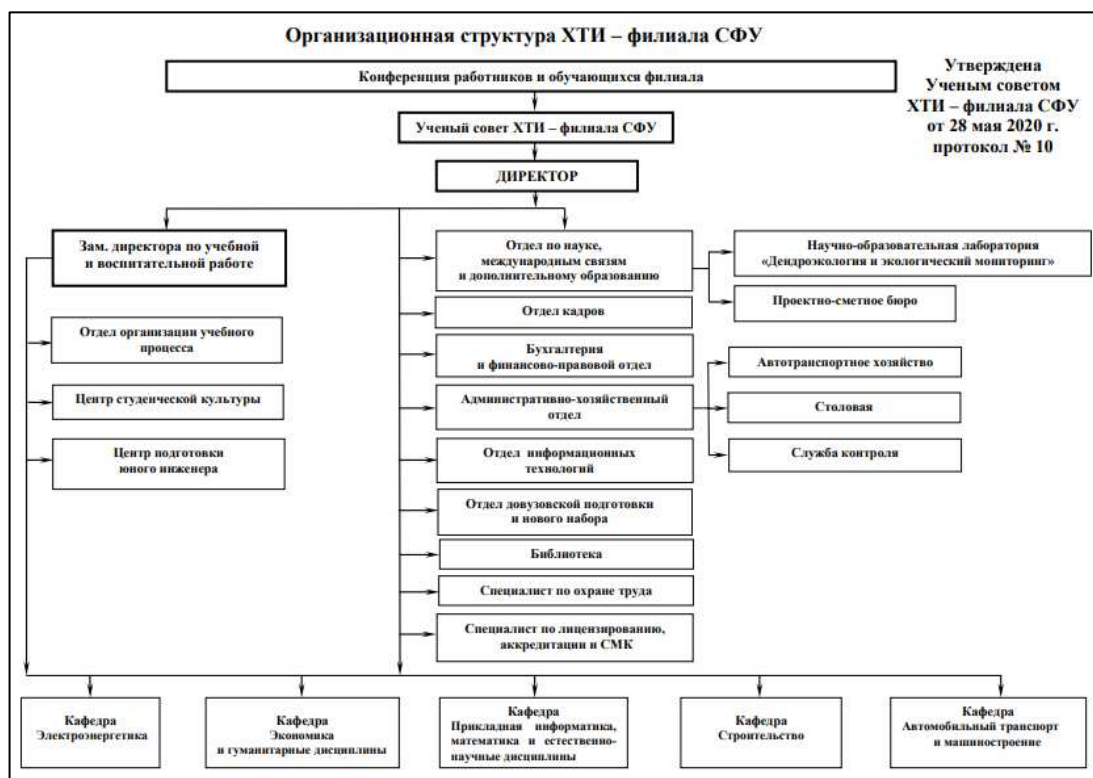


Рисунок 1 – Организационная структура ХТИ – филиала СФУ

Помимо обучения по специальности, кафедры осуществляют профориентационную деятельность, также проводят мероприятия по привлечению абитуриентов.

Высшее учебное заведение имеет в собственном распоряжении 11 компьютерных классов, оснащенных компьютерами.

На данный момент работу по привлечению абитуриентов в вуз осуществляет отдел довузовской подготовки и нового набора (ОДПиНН). Для привлечения абитуриентов и повышения их интереса к обучению и поступлению устраиваются Дни открытых дверей, олимпиады и прочие мероприятия.

В настоящее время подобные мероприятия малоэффективны. С развитием информационных технологий молодежи, и не только, все меньше причин идти куда-либо, когда можно ознакомиться с чем-либо, не выходя из дома. Достаточно просто найти информацию в сети Интернет. Поэтому следует применять новые способы, которые могут дать определенный результат.

Одним из таких способов является создание Игры-квеста, для демонстрации того, что учебный процесс можно сделать не только полезным, но и очень интересным.

Также данный проект «Игра-квест» можно использовать с целью помочь в привлечении абитуриентов и повысить интерес к образовательной деятельности института.

Для успешного ознакомления с игрой-квестом нужно лишь одно - желание приятно провести время и ознакомиться с программированием. В игре будут представлены разные уровни с одним из выбранных языков программирования, которые можно выбрать на старте игры.

1.2 Бизнес-процессы предметной области и обоснование разработки проекта

Языком программирования, выбранным для использования в качестве кода в игре, был выбран Python. Он прост в освоении и интуитивно понятен.

Данный язык программирования является многофункциональным, простым в освоении и довольно распространенным.

В игре будет 2 уровня, которые необходимо пройти при помощи взаимодействия с окружением и консолью. Консоль позволяет взаимодействовать с предметами и совершать над предметами определенные действия.

Количество уровней выбрано небольшим, потому что целью данного проекта является ознакомление пользователя с конкретным языком и развитие логического мышления. Игра покажет примерное использование кода в игре, и как именно он работает.

Игра должна быть короткая, чтобы пользователям не пришлось тратить огромное количество времени и сил на ее прохождение.

1.3 Примеры обучающих игр

Первая игра, которую можно представить, это Codecombat (рис. 2).

Данная игра обучает нескольким языкам программирования, среди которых: Python, Javascript, CoffeeScript и Lua. Позволяет изучить данные языки не только в одиночку, но и в группе.



Рисунок 2 – Окно игры Codecombat

Браузерная игра Flexbox froggy обучает основам работы с flexbox, с помощью которых можно размещать элементы на веб страницах. С помощью простых заданий и пояснений игроку нужно пройти небольшое количество уровней.

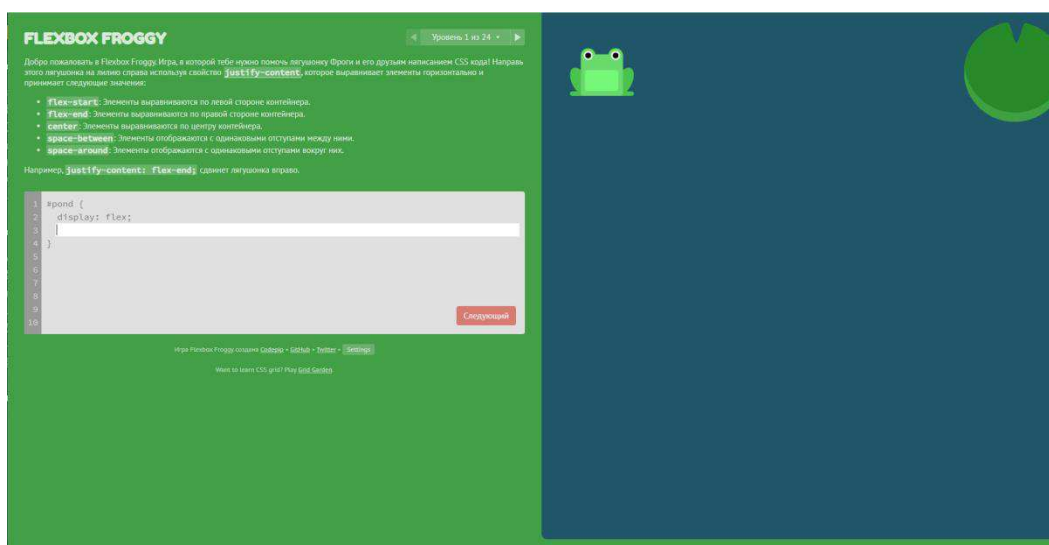


Рисунок 3 – Окно игры Flexbox froggy

Постепенно разработчики начинают использовать игры с целью обучения. Пытаясь совместить обучение и игру, что позволяет получить положительные эмоции от обучения.

1.4 Обоснование использования игры

Компьютерные игры обрели огромную популярность за небольшой промежуток времени, а также получили стремительное развитие во всех направлениях и до сих пор продолжают активно развиваться.

Компьютерная игра – программный продукт, предназначенный для развлекательных целей и приятного времяпрепровождения.

С развитием компьютерных игр появилось много разновидностей жанров, которые помогают пользователю подобрать что-то необходимое конкретно для него.

Для проекта был выбран жанр квест для того, чтобы привлечь пользователя идеями и игровым процессом.

Квест означает, что игроку предстоит провести поиск предметов и решить поставленные задачи, решив которые пользователь проходит уровни.

В качестве задач будут выступать ребусы из кода языков программирования, которые будет необходимо собрать и взаимодействовать с ними. Также будет возможность самостоятельно дописать не хватающий код.

1.5 Обзор и обоснование использования ПО

Для разработки игровых приложений используются разные специализированные программные продукты, такие как: Unity, Game Maker Studio 2, GDEVELOP, Unreal Engine.

Unity – предназначена для разработки 2D и 3D проектов в режиме реального времени. Значительно упрощает разработку будущего игрового

приложения и имеет огромную популярность у инди-разработчиков. Прост в обращении и обучении.

Game Maker Studio 2 – программный продукт, предназначенный для создания исключительно 2D проектов. Прост в освоении и не требует слишком много от разработчика. Также позволяет создать визуальный стиль и оформление для игры, что очень удобно.

GDEVELOP – программный продукт для создания игр с открытым исходным кодом, позволяющий создавать игры для Интернета (HTML5), настольных или мобильных телефонов (iOS / Android). Никаких навыков программирования или кодирования не требуется.

Unreal Engine – наверное, самый перспективный и популярный программный продукт. Помимо доступности, предоставляет огромные возможности в создании игр, но и навыков от разработчика требует немало. В основном, используется для создания больших дорогостоящих проектов.

Для анализа средств разработки игры определим критерии:

1. Стоимость.
2. Соответствие функциональных требований.
3. Сложность разработки.

Сравнение данных средств представлено в таблице 1.

Таблица 1 - Сравнение средств разработки

| Название | Unity | Game Maker Studio 2 | GDEVELOP | Unreal Engine |
|--|-----------|---------------------|-----------|---------------|
| Соответствие функциональных требований | полное | полное | полное | полное |
| Сложность разработки | средняя | низкая | низкая | высокая |
| Стоимость | бесплатно | бесплатно | бесплатно | бесплатно |

Данные перечисленные программные средства являются условно бесплатными и широко распространены. Стоимость для некоммерческого использования равна нулю и подходит данному проекту, потому что разработка происходит не с целью коммерческой выгоды.

В ходе анализа четырех доступных средств разработки было выявлено, что для проекта подойдет Game Maker Studio 2, потому что имеет огромный функционал и предназначен для 2D проектов. Из ряда преимуществ стоит выделить и то, что можно взять за основу уже готовые модели и переделать под свои нужды. Это позволит сэкономить время и средства на ПО.

1.6 Модель разработки игры-квеста

Для описания разработки игры-квеста была использована методология IDEF0.

IDEF0 – методология функционального моделирования и графическая нотация, предназначенная для описания бизнес-процессов.

Модель представлена на рисунке 4.

Входы – требования заказчика.

Выходы – готовая Игра-квест.

Управление – ГОСТ Р 51904-2002, сценарий, техническое задание.

Механизмы – Разработчик, Используемое ПО.

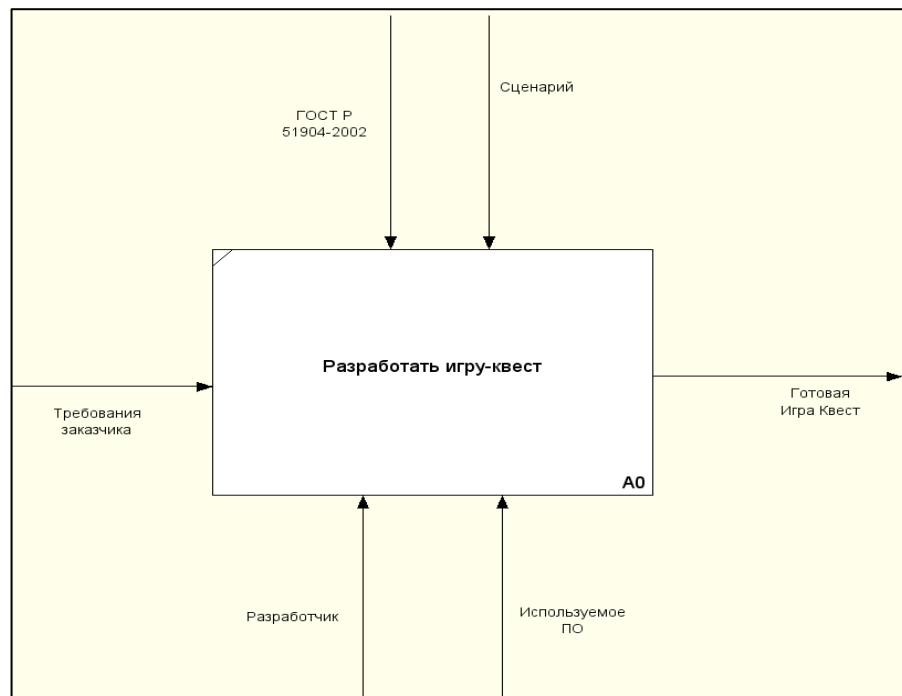


Рисунок 4 - Диаграмма IDEF0

Входом в функциональный блок «Разработать игру-квест» являются:

- Требования заказчика – влияет на составление технического задания и разработку программного продукта. Регулирует весь процесс разработки от начала и до конца.

Управлением процесса блока «Разработать игру-квест» являются:

- ГОСТ Р 51904-2002 – Стандарт разработки ПО и технического задания, контролирующий весь процесс разработки.
- Сценарий – определяет количество моделей персонажа, макет уровня и задания на уровне. Также зависит от технического задания.

Механизмами являются:

- Разработчик – создатель программы, исполняет все необходимые роли для создания проекта.
- Используемое ПО – в роли используемого ПО подразумевается Game Maker Studio 2, в котором заложено все необходимое для создания проекта.

Также необходимо знание Python для того, что добавлять задания и регулировать их.

На выходе «Готовая игра-квест» – конечный результат разработки, который должен соответствовать техническому заданию.

На рисунке 5 представлена декомпозиция функционального блока «Разработать игру-квест».

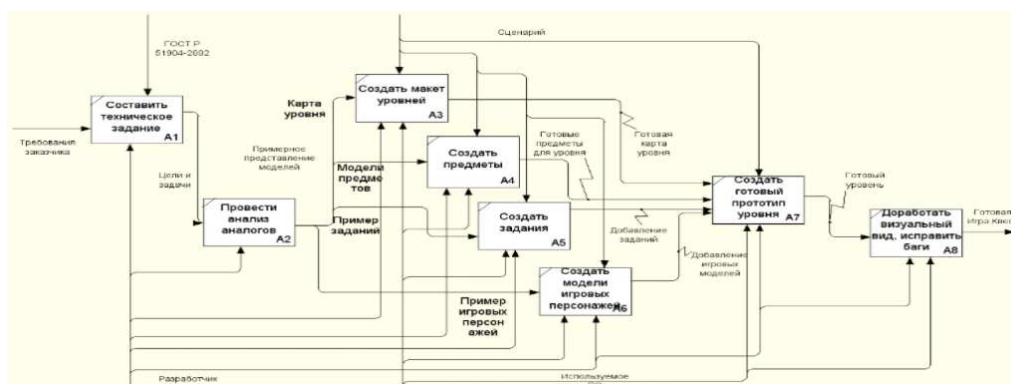


Рисунок 5 – Декомпозиция функционального блока «Разработать игру-квест»

На начальной стадии необходимо составить техническое задание. На этом этапе заказчик и разработчик оговаривают аспекты технического задания и утверждают его.

Выходом из данного блока являются цели и задачи, которые необходимо реализовать в рамках проекта.

Далее следует переход к Блоку «Провести анализ аналогов». В данном разделе разработчику необходимо провести анализ аналогов для определения некоторых аспектов разработки, чтобы создать необходимый продукт. Составляется примерный список моделей.

Итогом анализа является примерное представление моделей. Данными моделями являются: макет уровня, модели предметов, задания, модели игровых персонажей.

Создать макет уровней – создать определенное место в игре со своими ограничениями и особенностями. На макете будут располагаться предметы и персонажи.

Создать предметы – на данном этапе создаются игровые предметы, которые используются для решения ребусов уровня. Располагаются на макете уровня.

Создать задание – продумать и составить задание, которое пользователь должен будет решить.

Создать модель игровых персонажей – создание моделей с анимациями для придания проекту визуального стиля и особенностей.

Создать прототип уровня – на данном этапе на макет уровня накладываются задание, предметы и игровые персонажи в обозначенных зонах. Таким образом формируется игровой уровень, после прохождения задания которого, открывается доступ к следующему уровню.

После создания прототипа уровня и его проверки следует готовый уровень, который можно использовать в полноценной версии игры.

Далее следует полное тестирование готовой версии игры, на котором дорабатывается визуальный вид. Исправить баги – устранение технических неполадок проекта и исправление визуальных дефектов.

1.7 Выводы по аналитическому разделу

Выполнен анализ предметной области проекта «Игра-квест». Также в ходе выполнения был проведен анализ организационных характеристик Хакасского технического института – филиала СФУ, структуры его подразделений. Были выбраны программные средства для разработки. В ходе сравнения был выбран Game Maker Studio 2 для создания проекта. Определены цель и задачи проекта, также составлена диаграмма процесса разработки в методологии IDEF0.

2 Описание процесса разработки информационной системы «Игра-квест»

2.1 Планирование и определение функций. Декомпозиция на подсистемы

Диаграмма потоков данных

DFD – общепринятое сокращение от англ. data flow diagrams – диаграммы потоков данных. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.

Диаграмма потока данных понадобится для демонстрации потока данных в игре при взаимодействии с ней пользователя.

На рисунке 6 представлена диаграмма потока данных для пользователя.

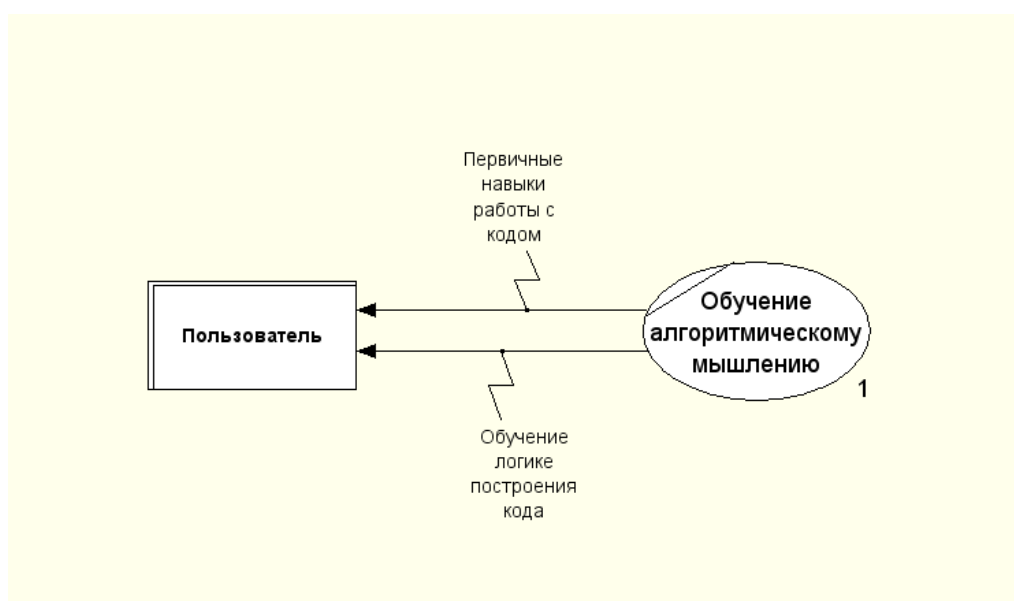


Рисунок 6 – DFD игры

На рисунке 7 показана декомпозиция диаграммы потока данных разрабатываемой игры.

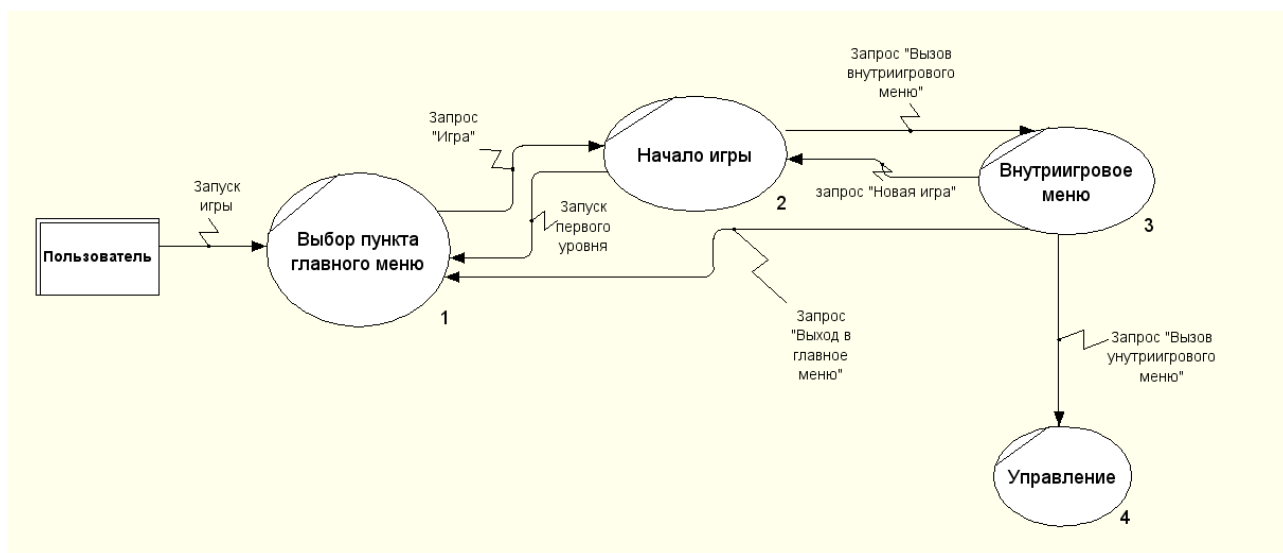


Рисунок 7 – Декомпозиция DFD игры

2.2 Запуск программного обеспечения

Перед началом разработки игры, необходимо запустить необходимое программное обеспечение. В данном случае, это Game Maker Studio 2. Запуск программы продемонстрирован на рисунке 8.

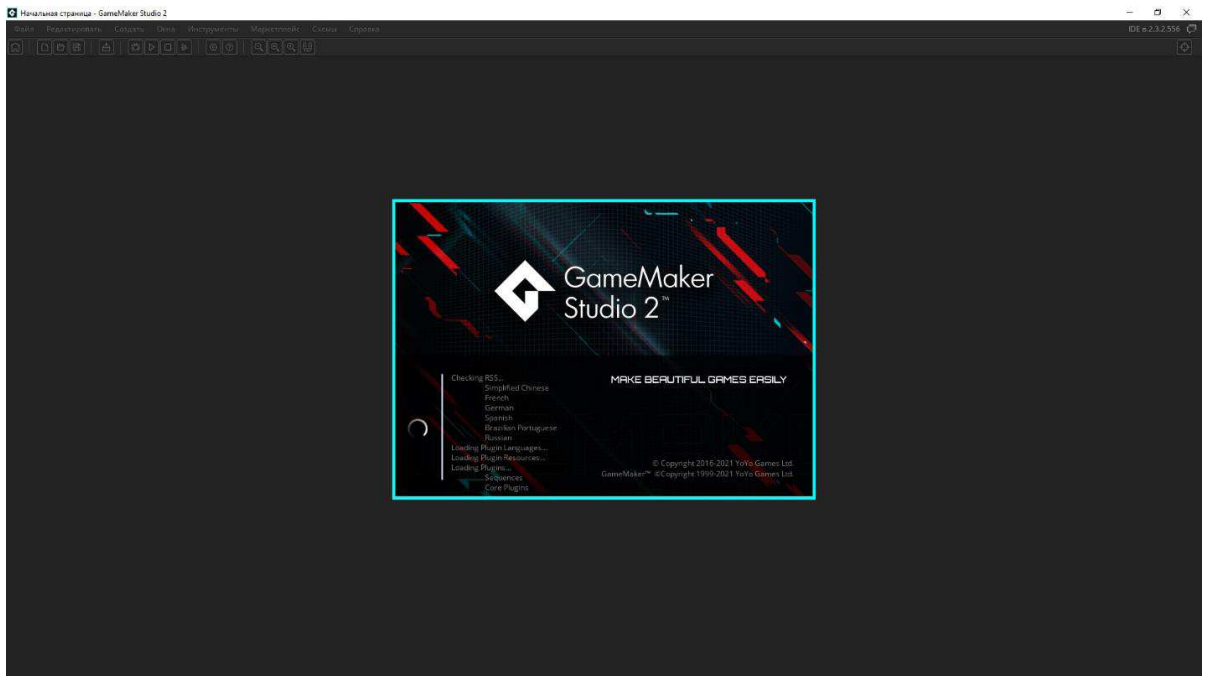


Рисунок 8 – Запуск программы Game Maker Studio 2

После запуска программы открывается доступ в главное меню (рис. 9).

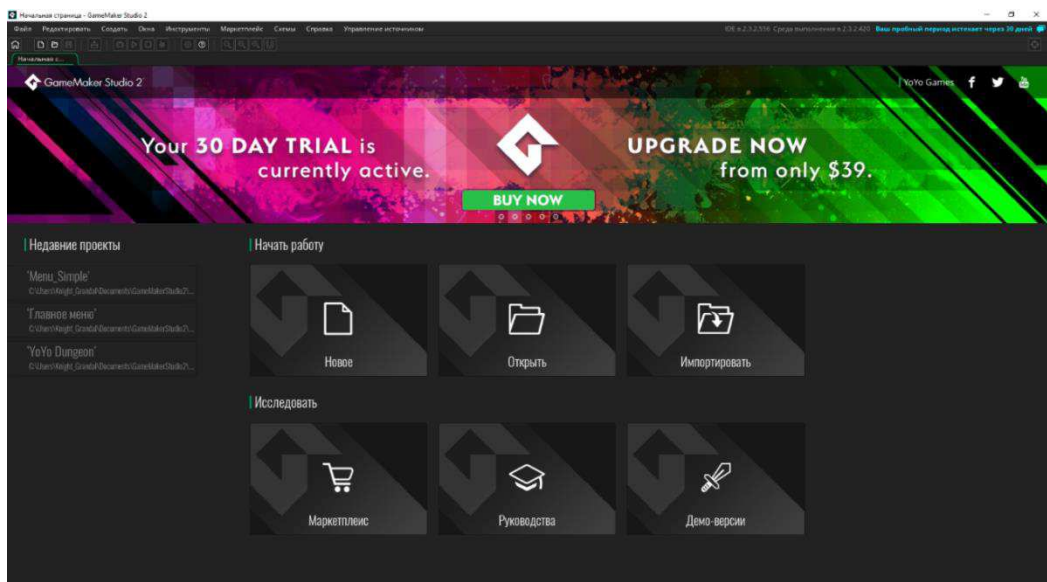


Рисунок 9 – Главное меню программы

2.3 Создание проекта в Game Maker Studio 2

После перехода по вкладке “новое” появляется возможность создать проект с помощью “Dragon and Drop” и “GameMaker Language” (далее GML) (рис. 10).

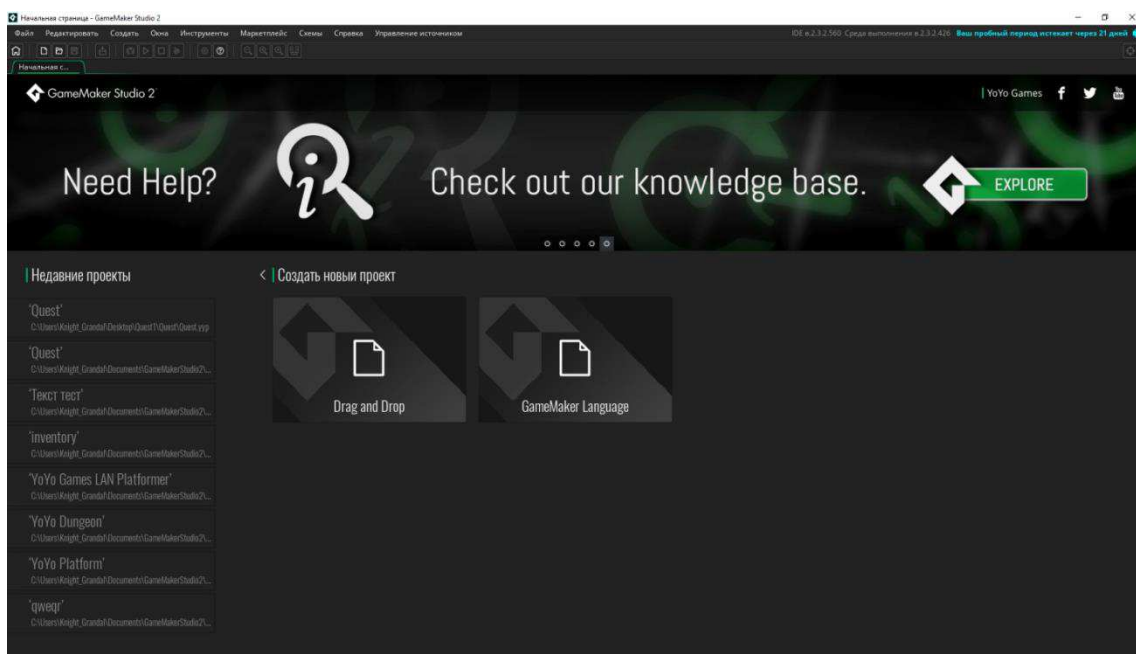


Рисунок 10 – Меню «Создание проекта»

При выборе первого варианта программа позволяет создавать объекты с предварительной настройкой и встроенным кодом, где пользователю просто необходимо их выбирать и объединять.

Второй вариант подразумевает активное использование кода и полное описание событий самостоятельно, что поможет создать уникальные функции в проекте.

В связи с некоторыми особенностями проекта был выбран второй вариант. Он поможет лучше понять логику самой программы и облегчит некоторые моменты в создании проекта. Экран разработки после выбора выглядит следующим образом (рис. 11).

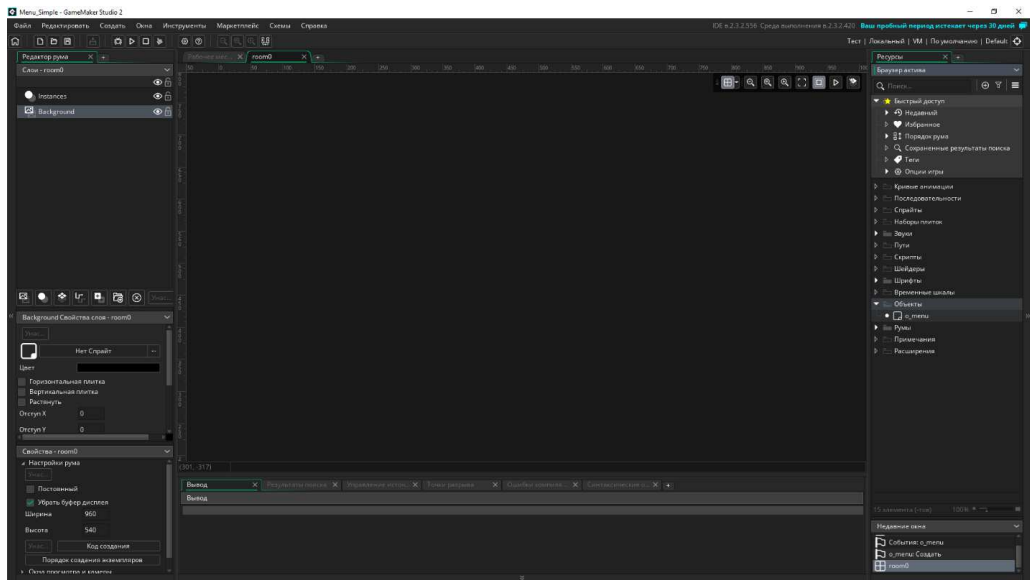


Рисунок 11 – Экран разработки

В меню разработки можно создавать объекты и проводить их настройку с помощью браузера актива, представленного на рисунке 12.

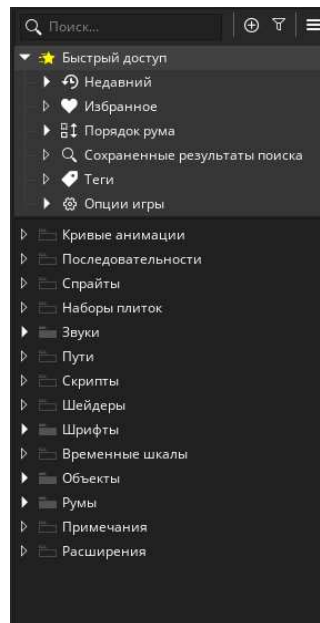


Рисунок 12 – Браузер актива

Для создания главного меню потребовалось создать объект «Меню» в котором прописывается в коде взаимодействие с кнопкой и связанными с ней событиями. Пример главного меню представлен на рисунке 13.

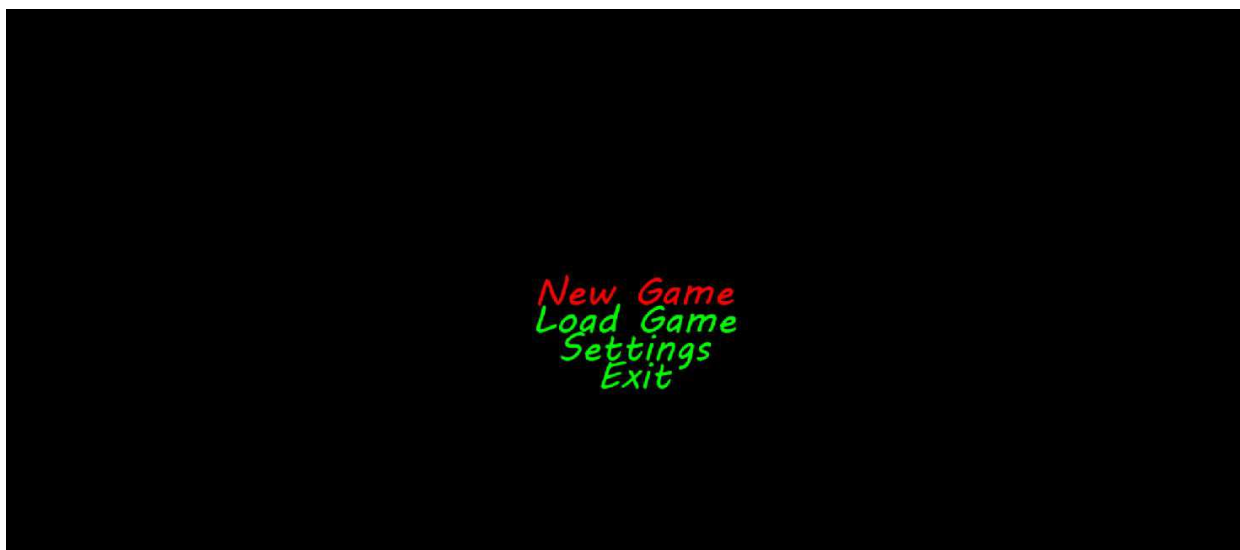


Рисунок 13 – Главное меню

Далее, описаны команды, используемые в главном меню. Для того, чтобы главное меню функционировало, необходимо добавить события: “Создать” (используется для создания переменных внутри объекта и дальнейшего обращения к ним в других событиях), “Шаг” (используется для создания управления, при котором можно переключать меню и взаимодействовать с ним), “Изобразить” (используется для запуска анимации, перехода между строками и звукового сопровождения).

Код в событиях представлен на рисунках 14-16.

```
oMain_menu: События
Создать
1 // @description Вставьте описание здесь
2 // Вы можете записать свой код в этом редакторе
3
4 window_set_fullscreen(true);
5
6 menu_x = room_width/2;
7 menu_y = room_height/2;
8
9 h_button = 32; // отступ
10
11 button[0] = "New Game"
12 button[1] = "Load Game"
13 button[2] = "Settings"
14 button[3] = "Exit"
15
16 buttons = array_length_id(button);
17
18 menu_index = 0;
19 last_selected = 0;
```

Рисунок 14 – Событие “Создать”

```
oMain_menu: События
Шаг
1 // @description Вставьте описание здесь
2 // Вы можете записать свой код в этом редакторе
3
4 menu_move = (keyboard_check_pressed(ord("S")) || keyboard_check_pressed(vk_down)) - (keyboard_check_pressed(ord("W")) || keyboard_check_pressed(vk_up));
5 menu_change = (keyboard_check_pressed(vk_space) || keyboard_check_pressed(vk_enter));
6
7 menu_index += menu_move;
8
9 if (menu_index < 0) menu_index = buttons - 1;
10 if (menu_index > buttons - 1) menu_index = 0;
11 if (menu_index != last_selected) audio_play_sound(snChange_menu, 1, false);
12
13 last_selected = menu_index;
14
15 if (menu_change = 1) switch(menu_index)
16 {
17     case 0:
18         room_goto(rFloor_one);
19         break;
20     case 1:
21         room_restart();
22         break;
23     /*case 2:
24         //room_goto(TEST);
25         break;
26     */
27     case 3:
28         game_end();
29         break;
30 }
31
```

Рисунок 15 – Событие “Шаг”

```
1 // @description Вставьте описание здесь
2 // Вы можете записать свой код в этом редакторе
3
4 var i = 0;
5 repeat(buttons) // повторить кнопки
6 {
7     draw_set_font(fMain_menu); //шрифт меню
8     draw_set_color(c_lime); //цвет меню
9     draw_set_halign(fa_center);
10
11     if (menu_index == i) draw_set_color(c_red);
12
13     draw_text(menu_x, menu_y + h_button * i, button[i]);
14     i++;
15 }
```

Рисунок 16 – Событие “Изобразить”

Также было создано внутриигровое меню для удобства будущих игроков (рис. 17).

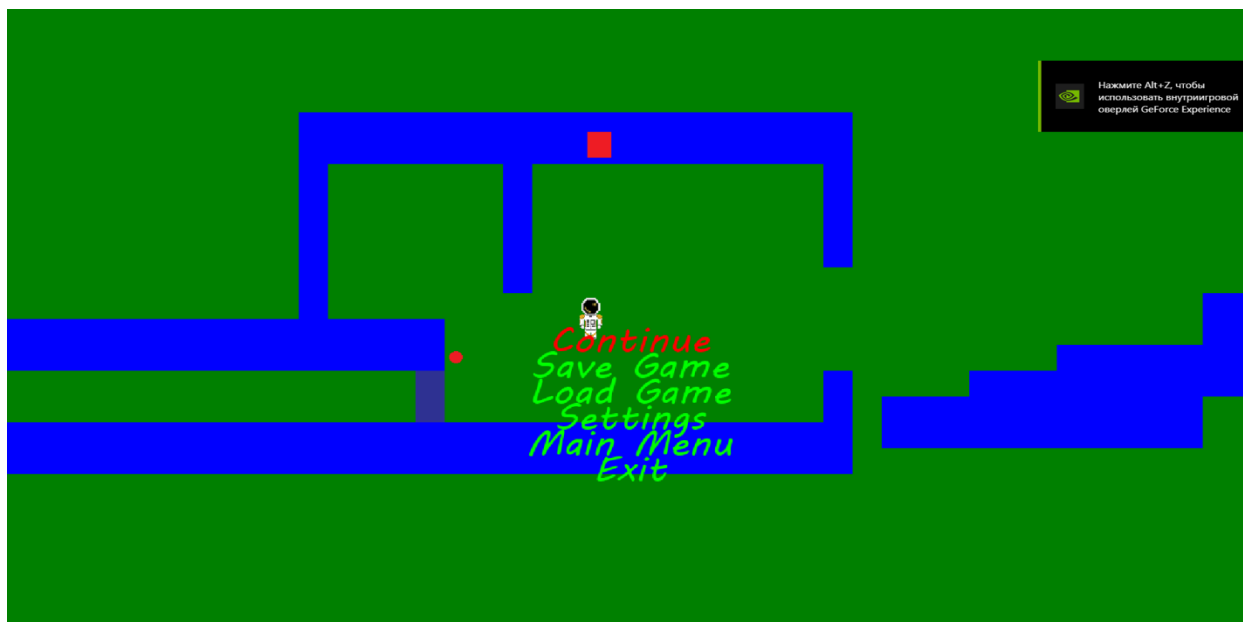
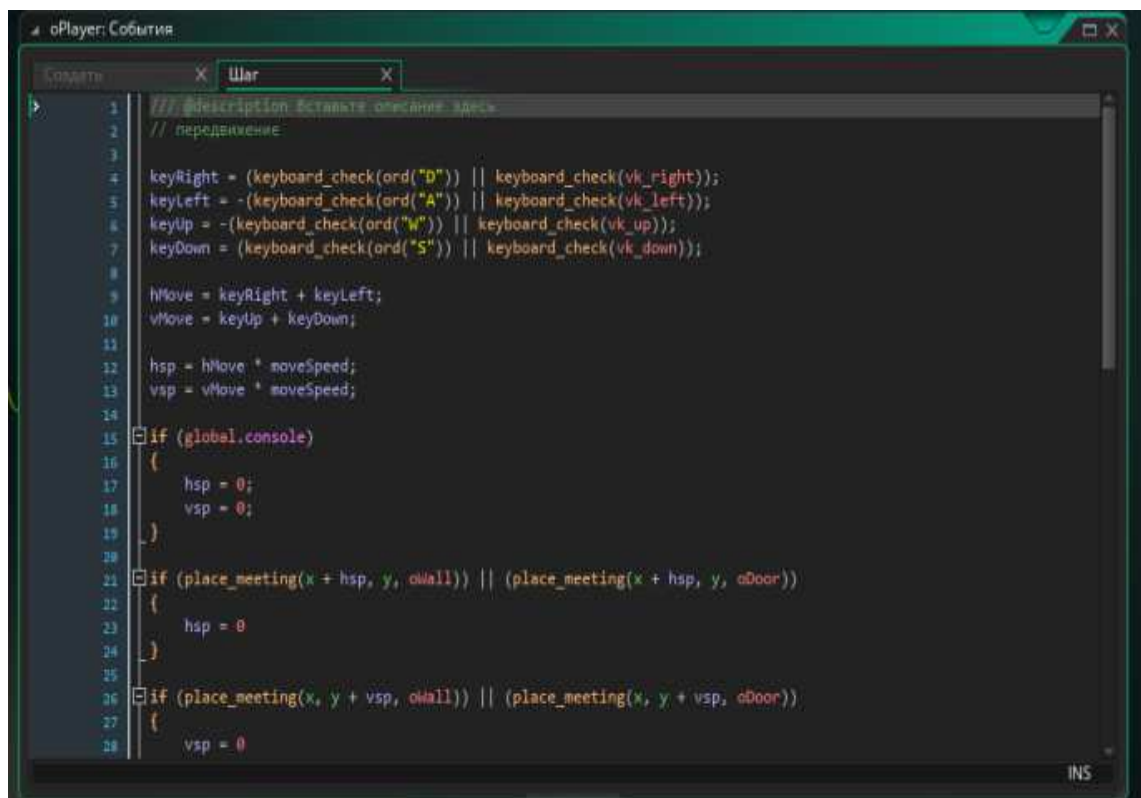


Рисунок 17 – Внутриигровое меню

Создание ключевых объектов.

Создание объектов представляет из себя создание объекта, добавление ему имени и свойств, а также создание его внешнего вида, который будет отображаться при запуске игры.

Рассмотрим создание игрока. Первоначально создается объект и присваивается имя “oPlayer” где “o” является сокращением от object. Это не является обязательным, но для удобства подобные обозначения необходимо использовать, чтобы не запутаться при обращении к объекту, так как объекту нужно будет придать внешний вид, которому тоже присваивается имя. Внешний вид объекта - это спрайт, от того и название спрайта будет “sPlayer”. Создание персонажа и код движения продемонстрированы на рисунке 18. Создание спрайта продемонстрировано на рисунке 19. После создания персонажа его необходимо разместить в первой игровой комнате, в которой и будут происходить игровые действия.

The image shows a screenshot of a code editor window titled "oPlayer: События". The code is written in C++ and defines movement logic for a player. It includes keyboard checks for 'D', 'A', 'W', and 'S' keys, calculates horizontal and vertical movement vectors, and checks for collisions with walls and doors. The code is as follows:

```
1 // @description: Вставить описание здесь
2 // передвижение
3
4 keyRight = (keyboard_check(ord("D")) || keyboard_check(vk_right));
5 keyLeft = -(keyboard_check(ord("A")) || keyboard_check(vk_left));
6 keyUp = -(keyboard_check(ord("W")) || keyboard_check(vk_up));
7 keyDown = (keyboard_check(ord("S")) || keyboard_check(vk_down));
8
9 hMove = keyRight + keyLeft;
10 vMove = keyUp + keyDown;
11
12 hsp = hMove * moveSpeed;
13 vsp = vMove * moveSpeed;
14
15 if (global.console)
16 {
17     hsp = 0;
18     vsp = 0;
19 }
20
21 if (place_meeting(x + hsp, y, oWall) || (place_meeting(x + hsp, y, oDoor)))
22 {
23     hsp = 0;
24 }
25
26 if (place_meeting(x, y + vsp, oWall) || (place_meeting(x, y + vsp, oDoor)))
27 {
28     vsp = 0;
```

Рисунок 18 – Создание персонажа и код движения

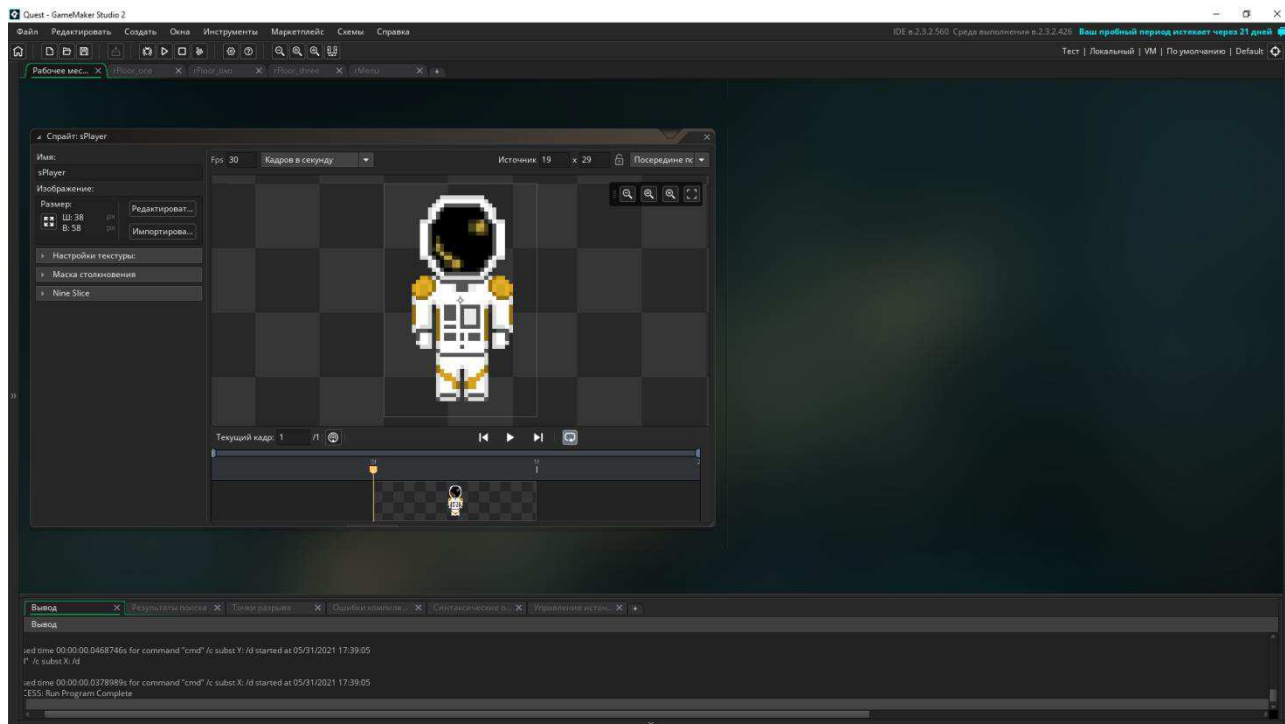


Рисунок 19 – Спрайт персонажа

Подобным образом были созданы кнопка, дверь и консоль. При взаимодействии с кнопкой дверь открывается и закрывается. При активации консоли открывается окно, в котором можно печатать текст. Пример работы представлен на рисунках 20 и 21. Код активации кнопки представлен на рисунке 22. Окно консоли представлено на рисунке 23. Код, используемый для вызова консоли и работы в ней, продемонстрирован на рисунках 24 и 25.

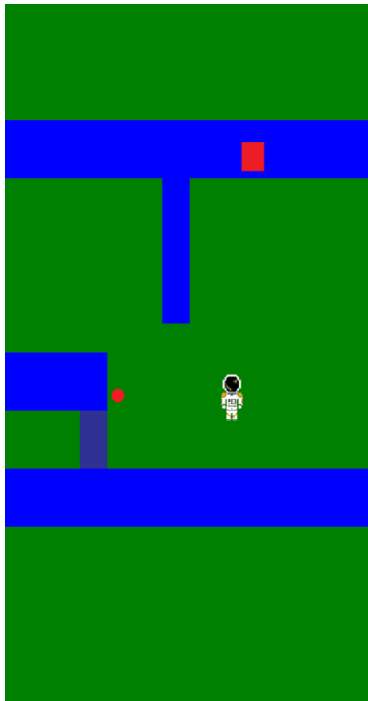


Рисунок 20 – Закрытая дверь

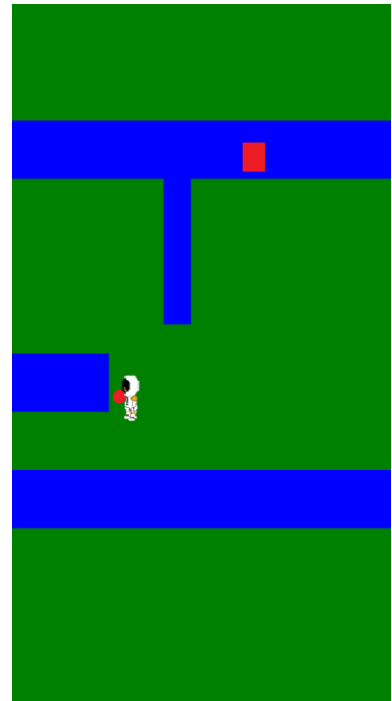


Рисунок 21 – Открытая дверь

```
oBottom: События
Создать Шаг
1 // @description Вставьте описание здесь
2 // Вы можете записать свой код в этом редакторе
3
4
5
6 if (distance_to_object(oPlayer) < 13)
7 {
8     if (keyboard_check_released(ord("F")))
9     {
10         door_open = !door_open;
11     }
12 }
13
14 if (door_open)
15 {
16     instance_deactivate_object(oDoor);
17 }
```

Рисунок 22 – Код кнопки

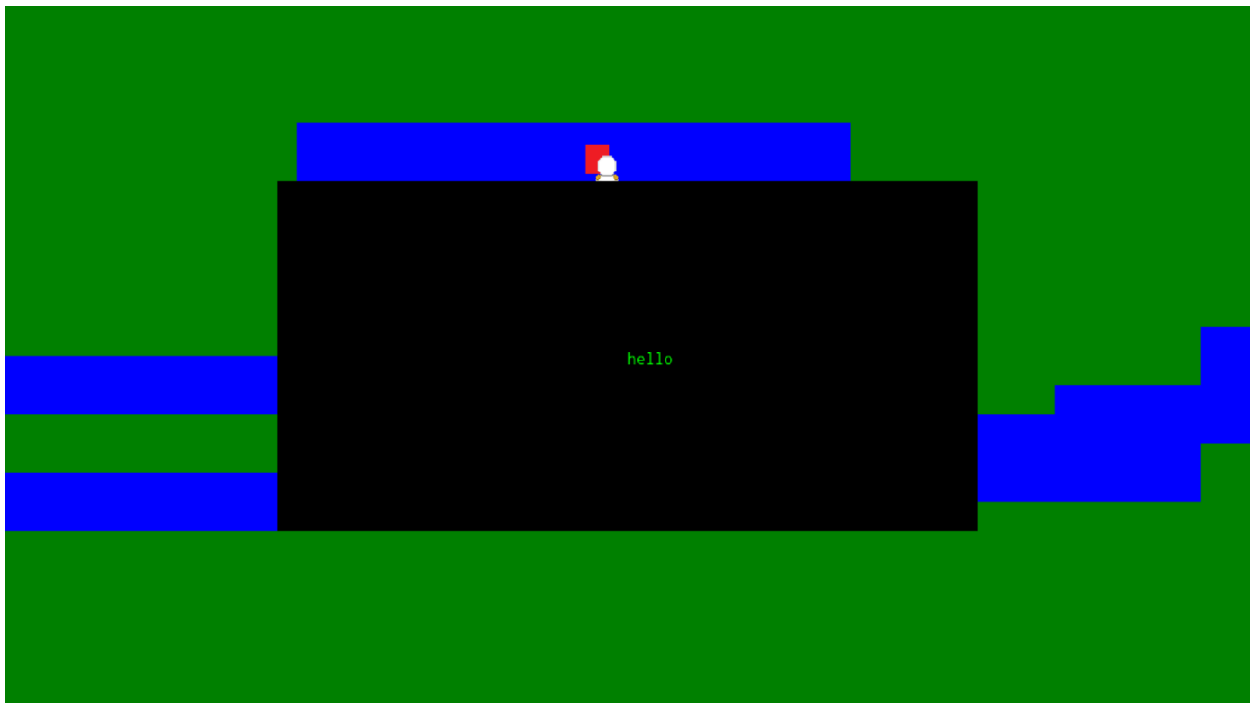
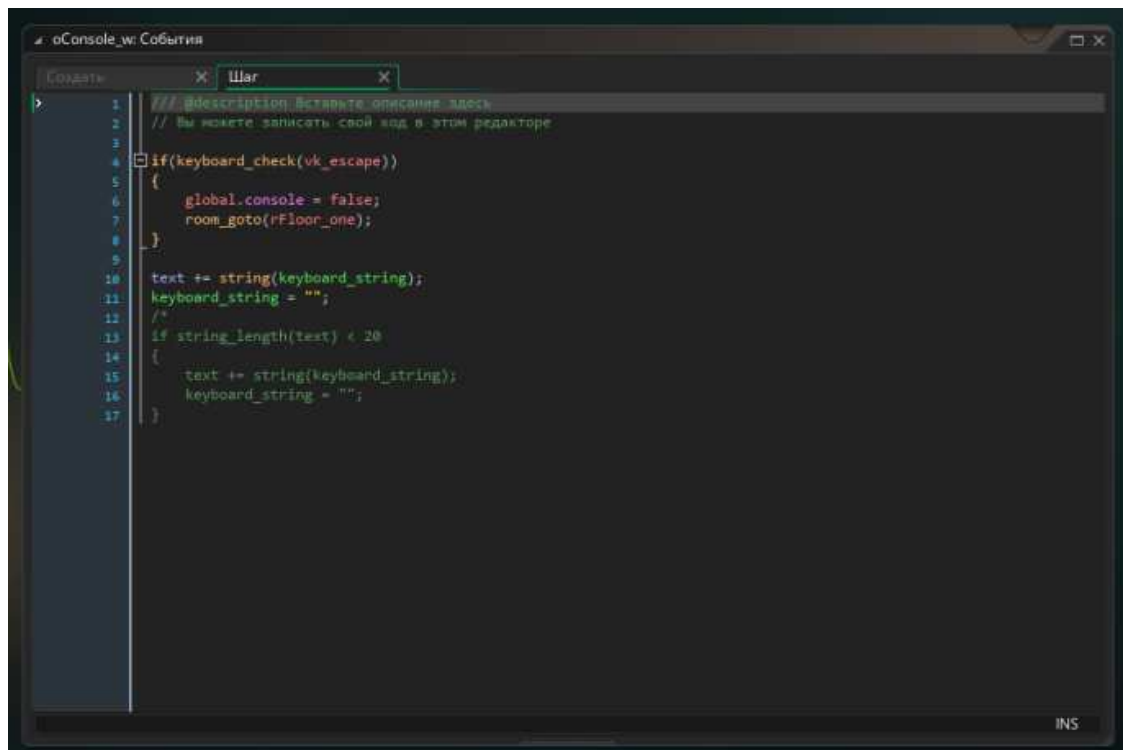


Рисунок 23 – Активированная консоль

```
oConsole: События
События Шар
1  /// @description Вставьте описание здесь
2  // Вы можете записать свой код в этом редакторе
3
4  if (distance_to_object(oPlayer) < 15)
5  {
6      if (keyboard_check_pressed(ord("F"))) global.console = true;
7  }
8
9  if (global.console)
10 {
11     if (!sprite_exists(global.screenShot_2)) // проверяет, есть ли скриншот и делает его, если отсутствовал
12     {
13         global.screenShot_2 = sprite_create_from_surface(application_surface, 0, 0, view_wport[3], view_hpor
14     }
15     instance_deactivate_object(oPlayer);
16     room_goto(rConsole);
17 }
18 else
19 {
20     if (sprite_exists(global.screenShot_2))
21     {
22         sprite_delete(global.screenShot_2);
23     }
24 }
25
```

Рисунок 24 – Код вызова консоли



```
1 // @description Вставьте описание здесь
2 // Вы можете записать свой код в этом редакторе
3
4 if(keyboard_check(vk_escape))
5 {
6     global.console = false;
7     room_goto(rFloor_one);
8 }
9
10 text += string(keyboard_string);
11 keyboard_string = "";
12 /*
13 if string_length(text) < 20
14 {
15     text += string(keyboard_string);
16     keyboard_string = "";
17 }
```

Рисунок 25 – Код для ввода символов в консоли

Также был реализован переход из одной игровой комнаты в другую, представленный на рисунках 26-28.

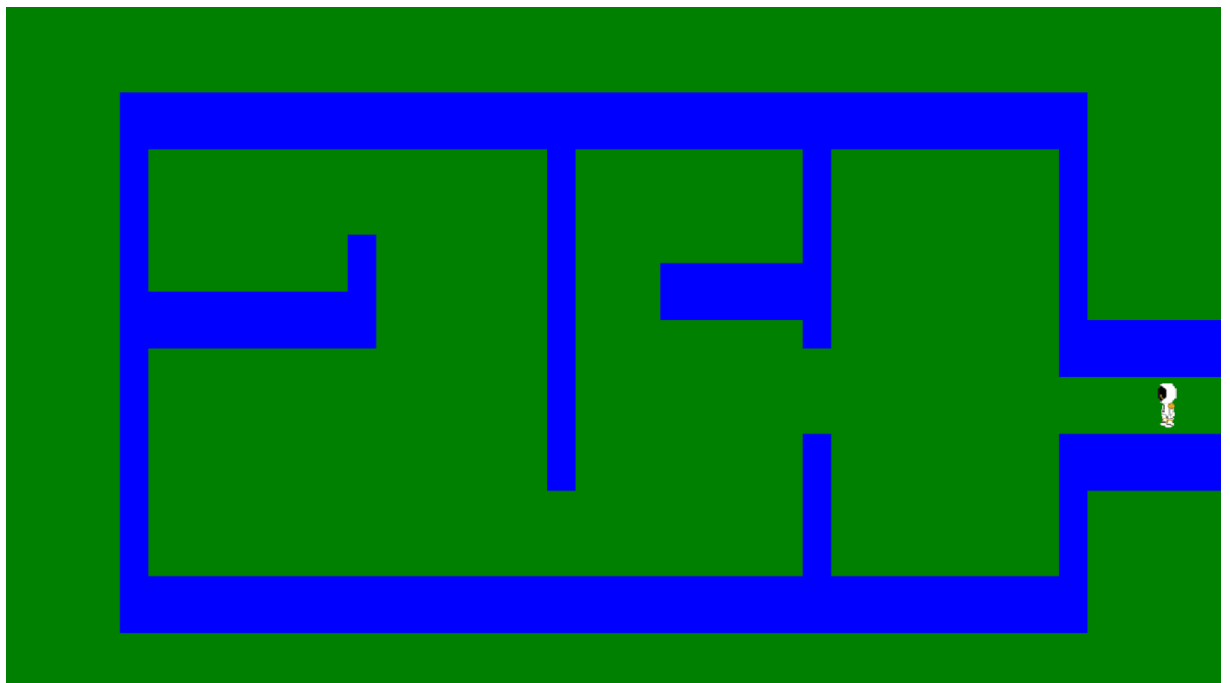


Рисунок 26 – Вторая тестовая комната

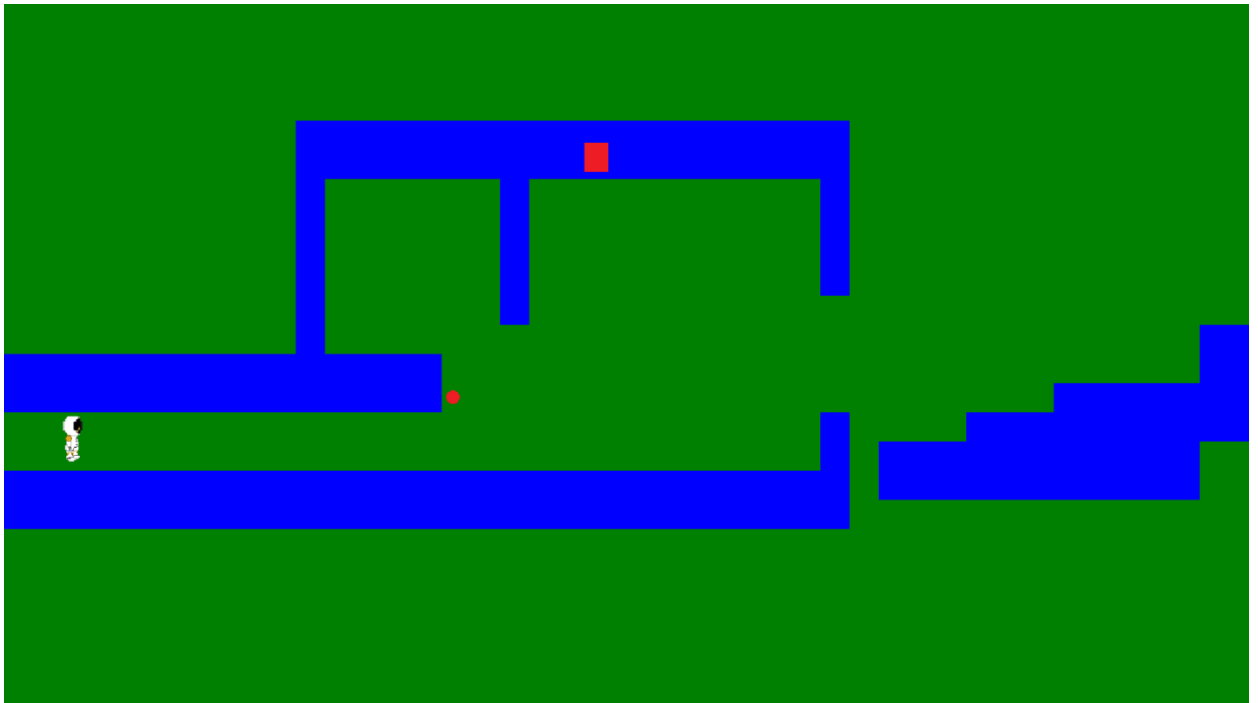


Рисунок 27 – Первая тестовая комната

```
oNext_room: Создать
Создать
1 // @description Вставьте описание здесь
2 // Вы можете записать свой код в этом редакторе
3
4 visible = false;
5
6 pen_x = 0;
7 pen_y = 0;
8 pen_room = room;
```

Рисунок 28 – Код для перехода между комнатами

Создание первого уровня.

На данный момент первый уровень завершен. Основные механики игры были добавлены, но не доработаны. Рисунок первой комнаты (рис. 29).

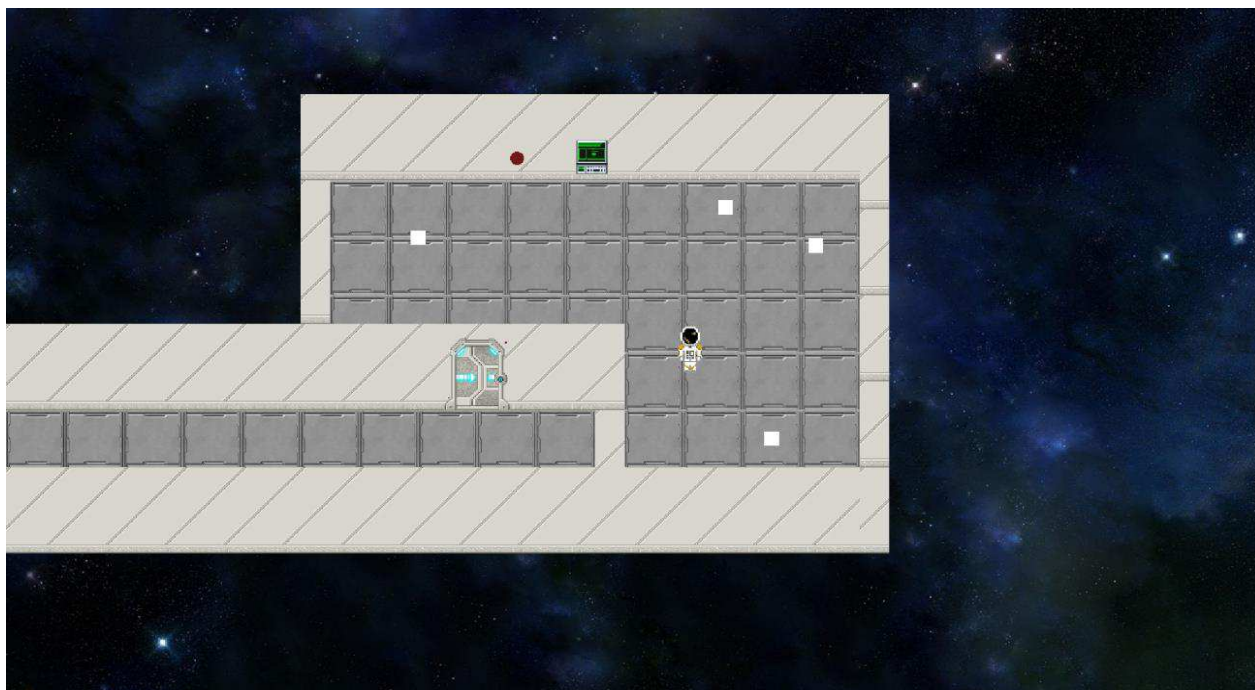


Рисунок 29 – Первая комната

Первый уровень представляет из себя помещение с загадкой, разгадав которую можно перейти в следующую комнату.

Для помощи в решении были разбросаны записки с кодом, которые игрок должен прочесть и использовать. Пояснение к коду содержится в других записках. Пример записки представлен на рисунке 30, код записки – на рисунке 31.

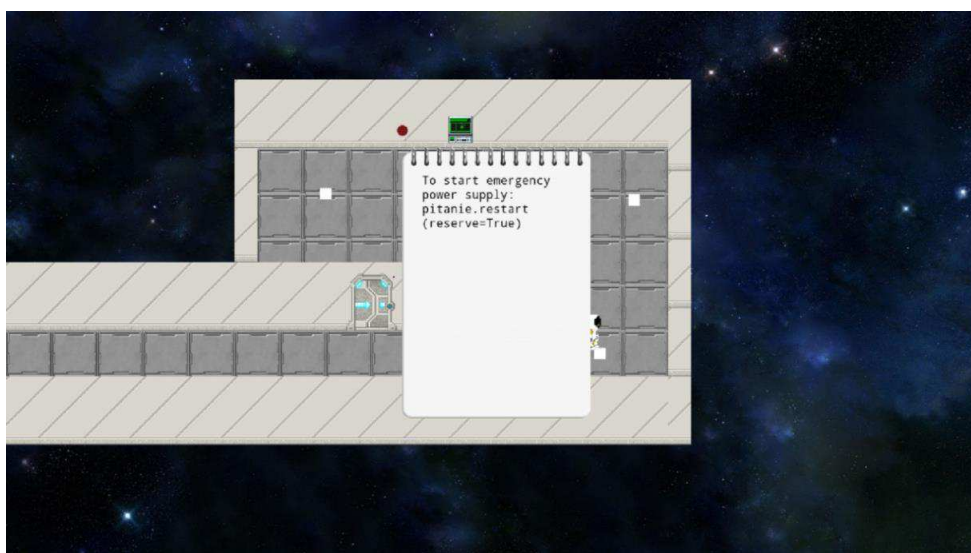
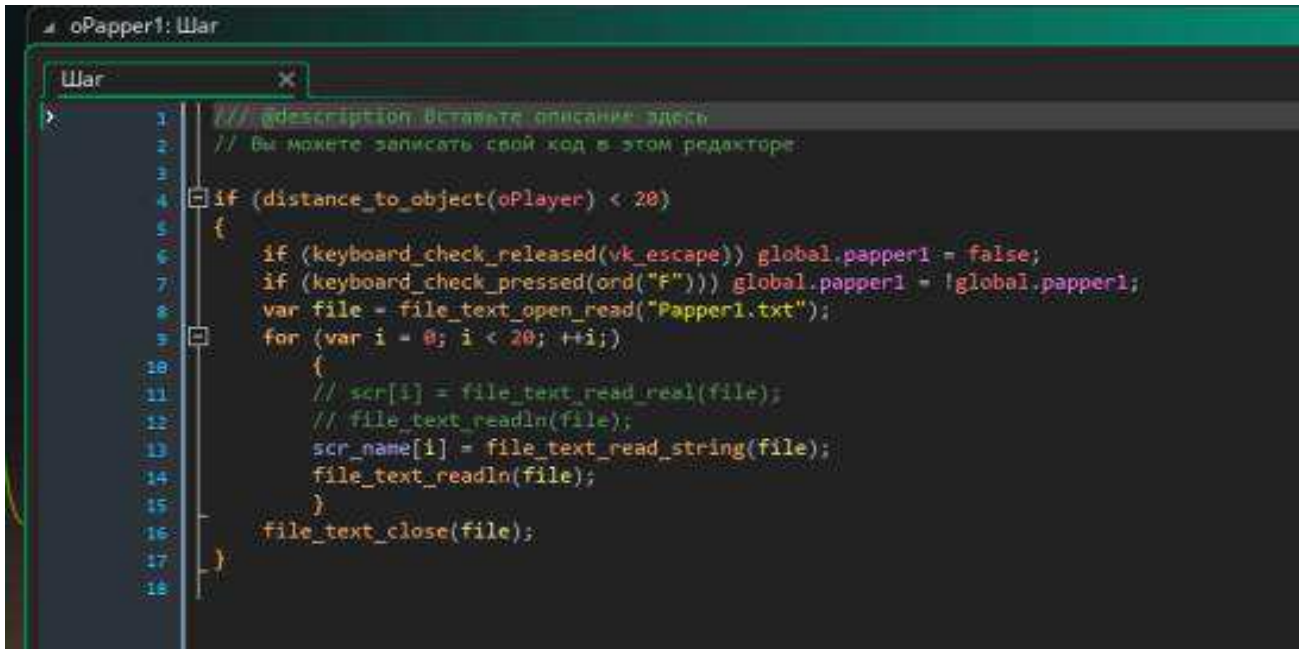


Рисунок 30 – Записка



```
oPapper1: Шар
Шар
1  /// @description Оставить описание здесь
2  // Вы можете записать свой код в этом редакторе
3
4  if (distance_to_object(oPlayer) < 20)
5  {
6      if (keyboard_check_released(vk_escape)) global.papper1 = false;
7      if (keyboard_check_pressed(ord("F"))) global.papper1 = !global.papper1;
8      var file = file_text_open_read("Papper1.txt");
9      for (var i = 0; i < 20; ++i)
10     {
11         // scr[i] = file_text_read_real(file);
12         // file_text_readln(file);
13         scr_name[i] = file_text_read_string(file);
14         file_text_readln(file);
15     }
16     file_text_close(file);
17 }
18
```

Рисунок 31 – Код записки

Консоль на первом уровне связана с кнопкой для открытия двери. Введя правильно код, она загорается, и открывается дверь. Пример консоли и кода для открытия двери представлен на рисунке 32.



Рисунок 32 – Работа в консоли

После успешного ввода загорается кнопка, нажав на которую открывается дверь (рис. 33-34).

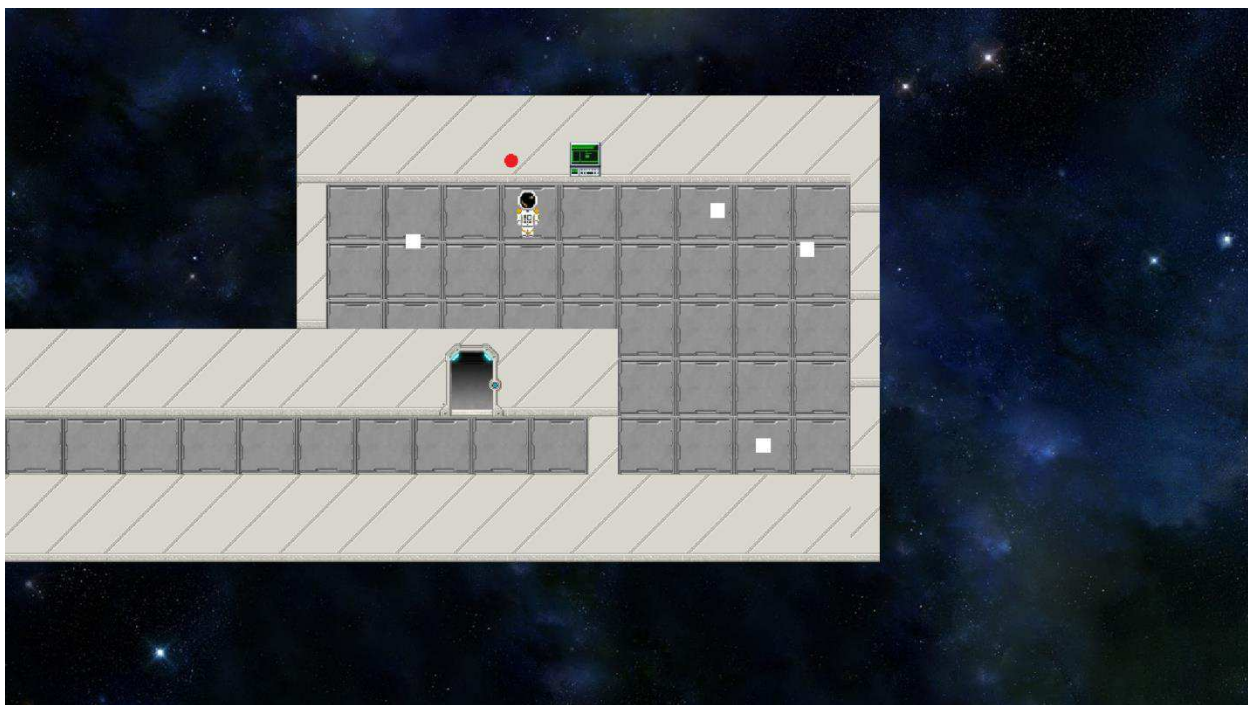
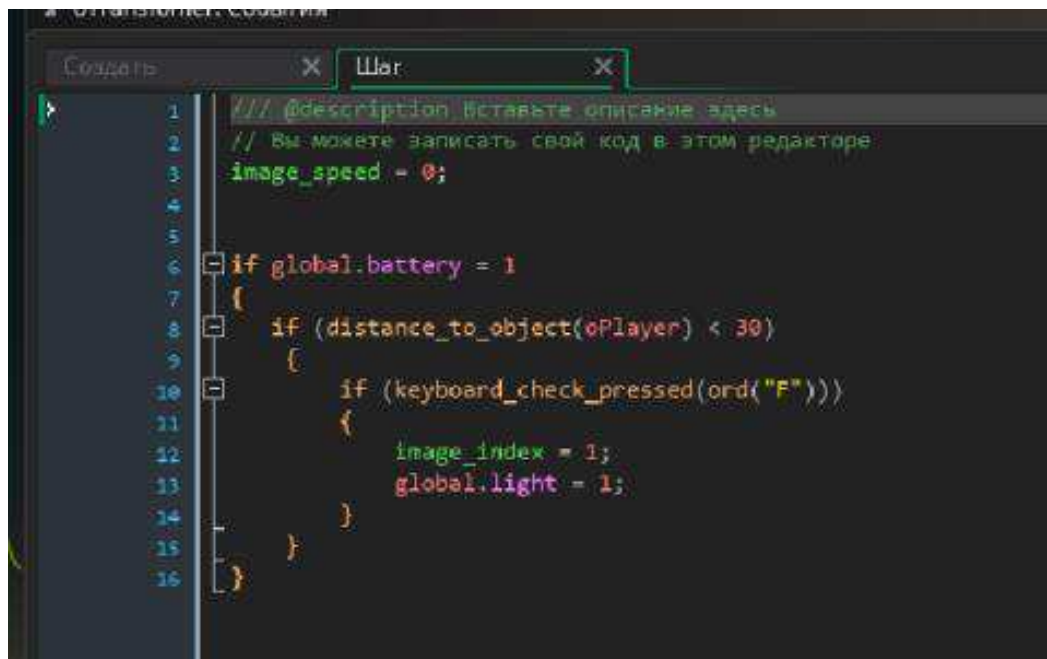


Рисунок 33 – Открытие двери



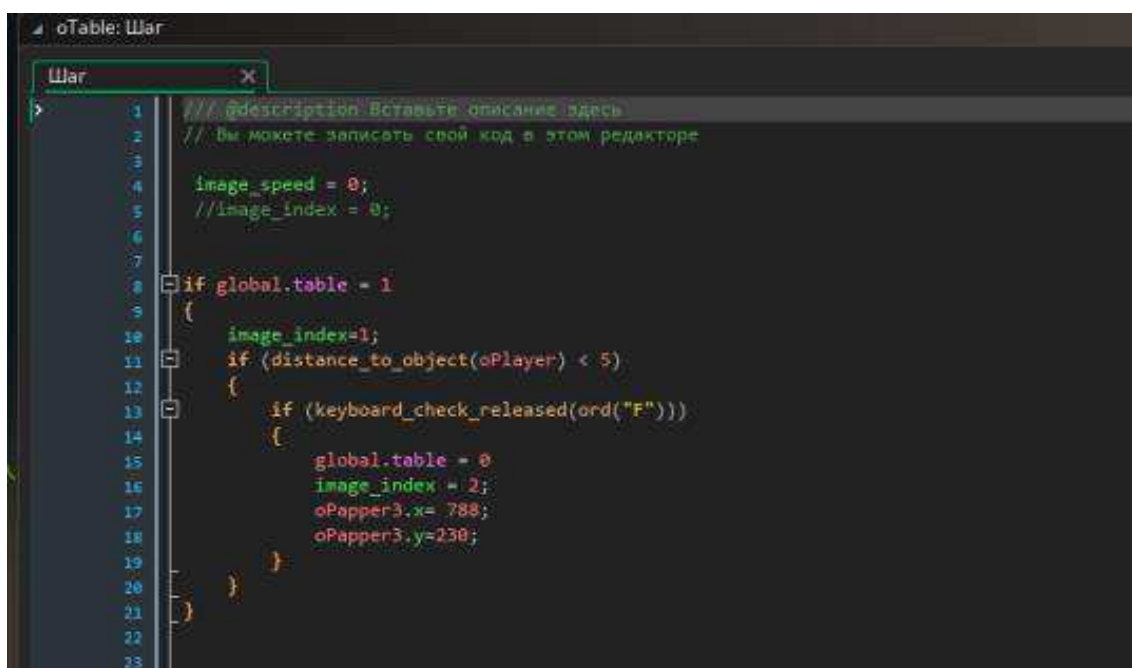
Рисунок 34 – Финальная версия первой комнаты

Для первой комнаты были добавлены игровые предметы, которые помогут игроку. С помощью консоли с ними можно взаимодействовать, о чем указано в спрятанных записках. Примеры кода предметов представлены на рисунках 35-36.



```
1  /// @description Вставьте описание здесь
2  // Вы можете записать свой код в этом редакторе
3  image_speed = 0;
4
5
6  if global.battery = 1
7  {
8      if (distance_to_object(oPlayer) < 30)
9      {
10         if (keyboard_check_pressed(ord("F")))
11         {
12             image_index = 1;
13             global.light = 1;
14         }
15     }
16 }
```

Рисунок 35 – Код трансформатора



```
1  /// @description Вставьте описание здесь
2  // Вы можете записать свой код в этом редакторе
3
4  image_speed = 0;
5  //image_index = 0;
6
7
8  if global.table = 1
9  {
10     image_index=1;
11     if (distance_to_object(oPlayer) < 5)
12     {
13         if (keyboard_check_released(ord("F")))
14         {
15             global.table = 0
16             image_index = 2;
17             oPapper3.x= 788;
18             oPapper3.y=230;
19         }
20     }
21 }
22
23
```

Рисунок 36 – Код стола

Если возникнут проблемы с взаимодействием, то в игре предусмотрена подсказка с клавишами управления (рис. 37).

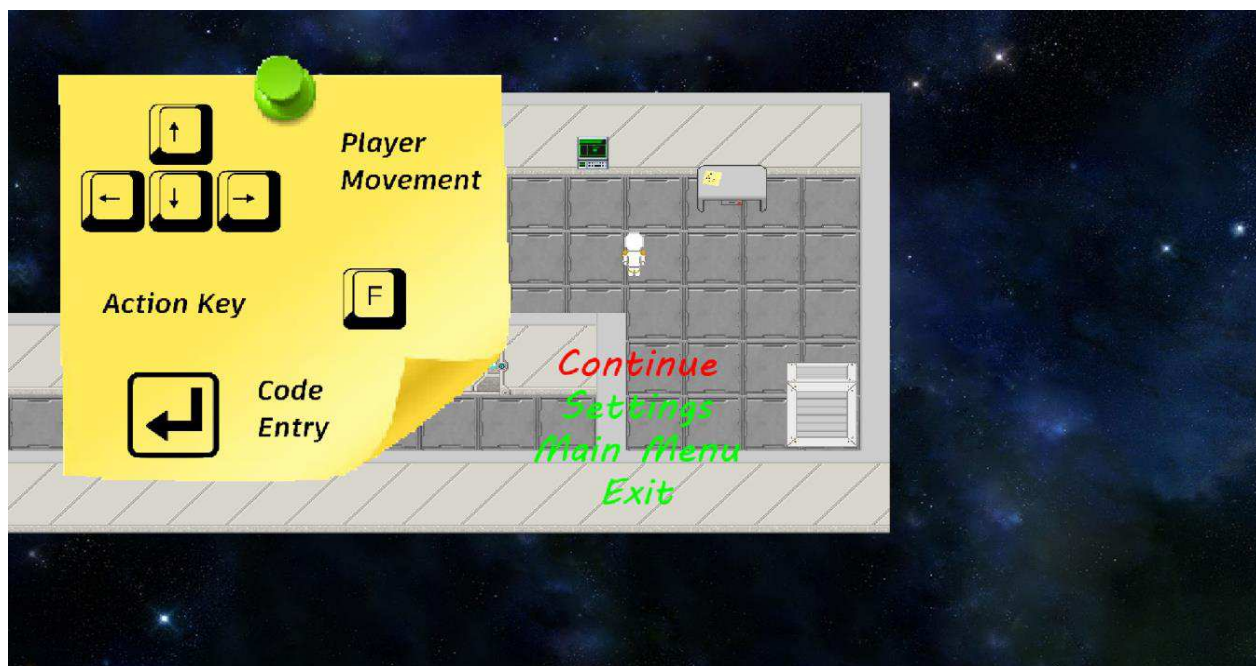


Рисунок 37 – Управление

Создание второго игрового уровня.

Второй уровень является продолжением первого, но уже без указаний, как и что работает. На уровне размещена только одна подсказка, по которой игрок сможет сориентироваться и применить навыки, полученные в первой комнате. Вторая игровая комната развивает алгоритмическое мышление и закрепляет знания (рис. 38-39).

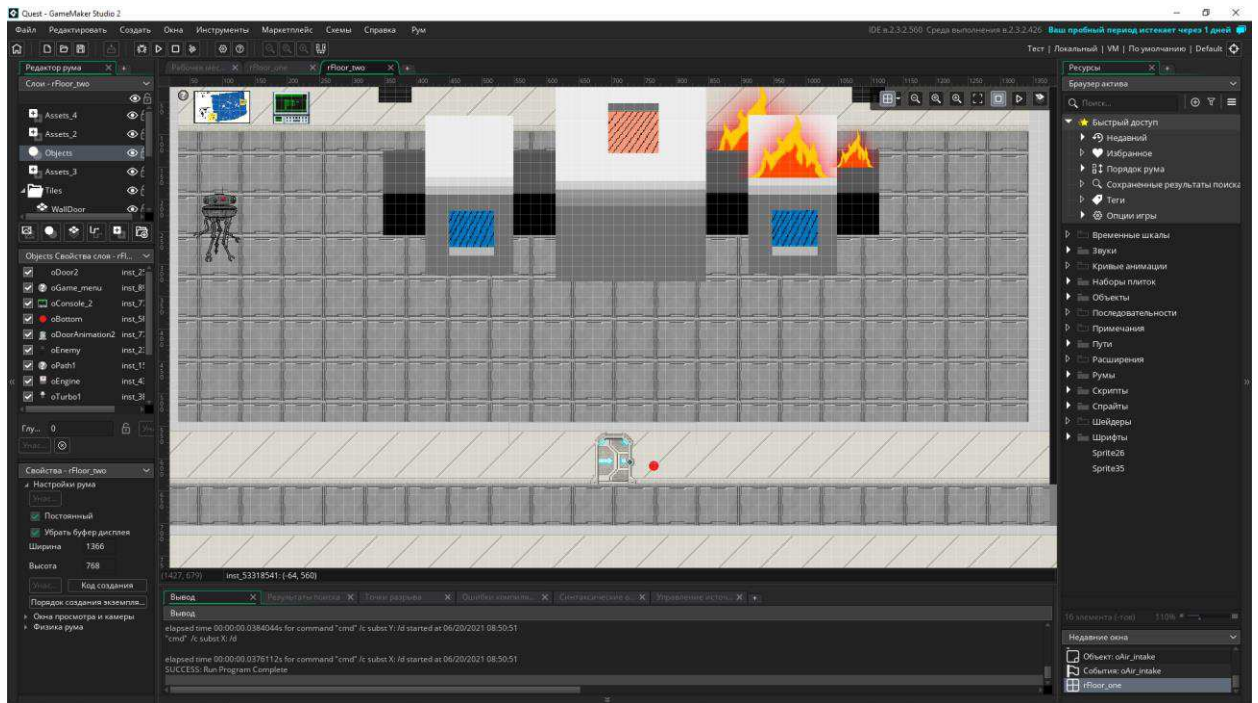


Рисунок 38 – Вторая игровая комната

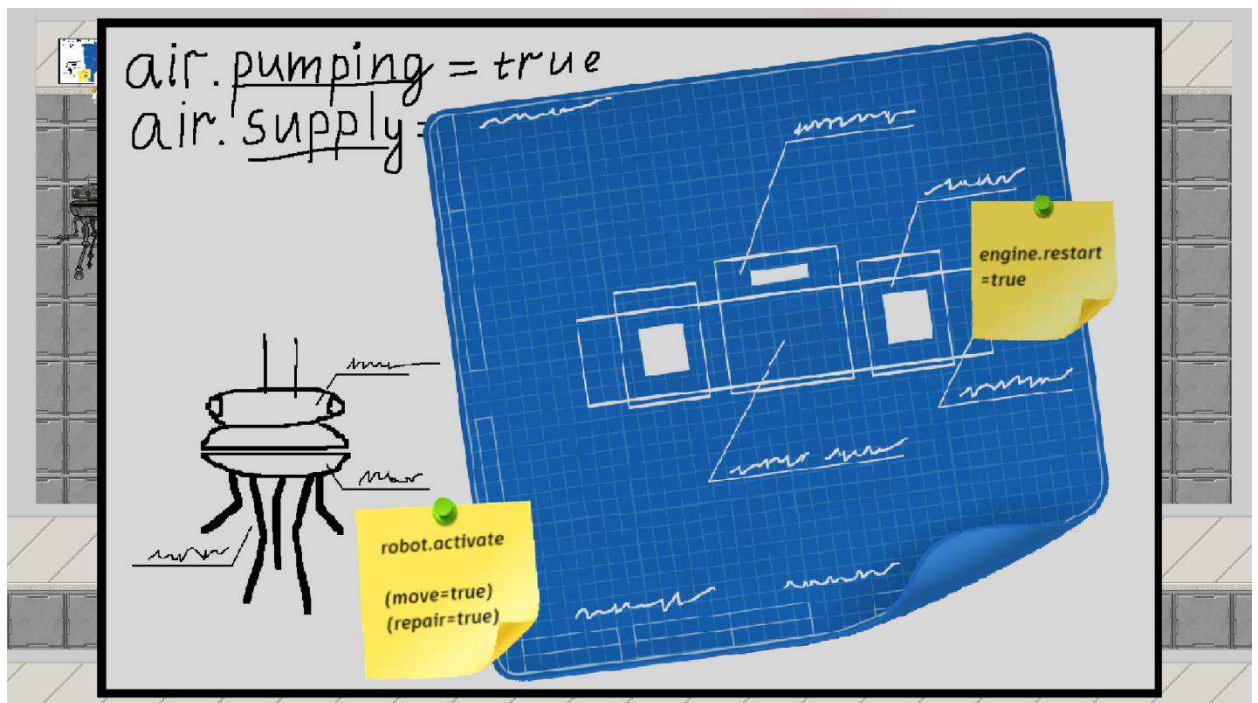


Рисунок 39 – Подсказка во второй комнате

Код предметов используемых в данной комнате, продемонстрирован ниже на рисунках 40-41.

```
oAir_intake: События
Создать x Шар x
1 /// @description Вставьте описание здесь
2 // Вы можете записать свой код в этом редакторе
3
4 if (global.air_intake = true)
5 {
6     image_speed = 1;
7     if image_speed > 0
8     {
9         if (image_index >= image_number - 1) image_speed = 0;
10    }
11 }
12
13 if (global.air_intake = false)
14 {
15     image_speed = -1;
16     if image_speed < 0
17     {
18         if (image_index <= image_number - image_number) image_speed = 0;
19    }
20 }
```

Рисунок 40 – Код вентиляции

```
oTurbo2: События
Создать x Шар x
1 /// @description Вставьте описание здесь
2 // Вы можете записать свой код в этом редакторе
3
4 if (global.robot_work) image_speed = 1;
5
6 if (global.engine = true)
7 {
8     image_index = image_number - 1;
9     image_speed = 0;
10 }
```

Рисунок 41 – Код двигателя

2.4 Выводы по разделу «Описание процесса разработки приложения «Игра-квест»

Проведен разбор выбранного приложения для работы над проектом и его предварительная настройка.

Продемонстрировано создание спрайтов и объектов, находящихся в игровых комнатах.

Выполнены создание двух уровней с разными заданиями и демонстрация игровых механик.

Показан пример использования кода для прохождения уровня (рис.32).

3 Оценка экономической эффективности внедрения и разработки проекта «Игра-квест»

3.1 Методика оценки экономической эффективности разработки игры-квеста

ТСО (Total cost ownership) – это общая величина совокупных затрат на разработку программного продукта. Помогает представить, из чего состоит продукт и как изменяется его стоимость с течением времени.

ТСО рассчитывается по формуле

$$ТСО=DE+IC1+IC2, \tag{1}$$

где DE – прямые затраты;

IC – косвенные затраты первой и второй группы.

3.1.1 Капитальные затраты

Расчет капитальных затрат производится по формуле

$$K = K_{\text{пр}} + K_{\text{тс}} + K_{\text{лс}} + K_{\text{по}} + K_{\text{ио}} + K_{\text{об}} + K_{\text{оэ}}, \quad (2)$$

где $K_{\text{пр}}$ – затраты на всю разработку программного продукта;

$K_{\text{тс}}$ – затраты на технические средства управления;

$K_{\text{лс}}$ – затраты на создание линий связи, интернет-соединения;

$K_{\text{по}}$ – затраты на программные средства для использования готового продукта;

$K_{\text{ио}}$ – затраты на формирование информационной базы;

$K_{\text{об}}$ – затраты на обучение персонала;

$K_{\text{оэ}}$ – затраты на опытную эксплуатацию.

3.1.2 Затраты на разработку

Подсчет затрат на разработку программного продукта производится по формуле

$$K_{\text{пр}} = K_{\text{зп}} + K_{\text{ипс}} + K_{\text{свт}} + K_{\text{проч}}, \quad (3)$$

где $K_{\text{зп}}$ – затраты на заработную плату разработчиков;

$K_{\text{ипс}}$ – затраты на инструментальные программные средства для проектирования;

$K_{\text{свт}}$ – затраты на средства вычислительной техники для проектирования;

$K_{\text{проч}}$ – прочие затраты разработки.

Для расчета заработной платы разработчика возьмем минимальную оплату труда по России, составляющую 12 792 рублей, и увеличим на 30%

(северная надбавка), и еще на 30% (районный коэффициент для Хакасии). В итоге, заработная плата работника составляет:

$$\text{ЗП} = 12\,792 + (12\,792 * 0,3) + (12\,792 * 0,3) = 20467 \text{ рублей.}$$

Расчет заработной платы разработчика приведен в таблице 2.

Таблица 2 – Расчет заработной платы разработчика

| Состав заработной платы | Сумма, рублей |
|-------------------------|---------------|
| Оклад | 12792 |
| Районный коэффициент | 3838 |
| Северный коэффициент | 3838 |
| НДФЛ (13%) | 2661 |
| Итого | 20467 |

Также при расчете заработной платы разработчика следует учитывать сумму обязательных отчислений во внебюджетные фонды (30,2% от суммы начисленной заработной платы):

$$K_{\text{зп}} = 20467 * 1,302 = 26648 \text{ рубля.}$$

Сроки по реализации проекта:

1. Анализ предметной области, консультация с заказчиком – 5 дней.
2. Разработка прототипа – 15 дней.
3. Тестирование – 2 дня.
4. Доработка – 7 дней.
5. Введение в эксплуатацию – 1 день.

Далее происходит оценка затрат на программное обеспечение для реализации и работы разработанного программного продукта.

Затраты на программное обеспечение для разработчика в данном проекте рассчитываются из суммы затрат на программное обеспечение. Их стоимость указана в таблице 3:

Таблица 3 – Затраты на программное обеспечение

| Наименование | Назначение | Стоимость, рублей | Срок использования |
|--------------------------|---------------------------|---|---------------------------|
| Microsoft Windows 10 Pro | Начальный интерфейс | 3 950 | 30 дней |
| Microsoft Office 2019 | Для создания документации | 2 429 | Для создания документации |
| Game Maker Studio 2 | Для создания проекта | бесплатно для некоммерческой деятельности | Для создания проекта |

ОС Windows и Microsoft Office будут актуальны еще один год. Для создания проекта необходимо, примерно, один месяц, из чего следует, что стоимость данных продуктов делим на 12.

$$K_{\text{ипс}} = (3950 + 2429)/12 = 532 \text{ рубля.}$$

Затраты на технические средства рассчитываются из стоимости комплектующих персонального компьютера, на котором разрабатывался проект. Стоимость комплектующих указана в таблице 4:

Таблица 4 – Затраты на технические средства

| № | Наименование оборудования | Количество | Стоимость, рублей | Нормированный срок службы |
|---|-----------------------------|------------|-------------------|---------------------------|
| 1 | Процессор Core i5 9400F BOX | 1 | 11 899 | 5 лет |

Окончание таблицы 4

| | | | | |
|----|---|-------|-------|--------|
| 2 | Видеокарта MSI GeForce GTX 1660 | 1 | 20700 | 4 года |
| 3 | Материнская Плата MSI H310M GAMING ARCTIC | 1 | 5350 | 4 года |
| 4 | Kingston HyperX FURY Blue Series 4 ГБ | 2 | 5000 | 4 года |
| 5 | Корпус Zalman Z3 Plus черный | 1 | 3650 | 4 года |
| 6 | Блок питания High Power Super GD 850W | 1 | 7400 | 4 года |
| 7 | 1 ТБ Жесткий диск WD Purple | 1 | 3900 | 4 года |
| 8 | 128 ГБ SSD-накопитель Goodram CX400 | 1 | 1550 | 4 года |
| 9 | Монитор Acer V227QAbi | 1 | 7100 | 4 года |
| 10 | Клавиатура A4Tech Bloody B120 | 1 | 2500 | 5 лет |
| 11 | Мышь проводная A4Tech Bloody P81 черный | 1 | 1500 | 5 лет |
| | | Итого | 68049 | |

Длительность разработки проекта составляет 30 дней, поэтому полную стоимость комплектующих использовать нельзя, и теперь нужно рассчитать амортизацию оборудования на срок эксплуатации.

По статистике, приведенной в таблице 4, можно сделать вывод, что при нормальной работе настольного компьютера его срок службы составит 4 года.

$$A_{\text{год}} = C_{\text{б}} * N_{\text{ам}}, \quad (4)$$

где $A_{\text{год}}$ – амортизация за год использования;

$C_{\text{б}}$ – балансовая стоимость;

$N_{\text{ам}}$ – норма амортизации.

$$A_{\text{пр}} = A_{\text{год}}/K_{\text{рдг}} * K_{\text{дэ}}, \quad (5)$$

где $A_{\text{пр}}$ – проектная амортизация;

$K_{\text{рдг}}$ – количество рабочих дней в 2021 году;

$K_{\text{дэ}}$ – количество дней эксплуатации.

Минимальный срок эксплуатации некоторых комплектующих равен 4-м годам, но следует учесть вероятность поломки некоторых деталей с последующей заменой. Данными комплектующими выступают процессор и материнская плата.

$$A_{\text{год}} = 68049 * 0.25 = 17012 \text{ рублей.}$$

$$K_{\text{свт}} = A_{\text{пр}} = 17012/12 = 1417 \text{ рублей.}$$

За данность принято, что на прочие затраты необходимо оставлять не меньше 3% от общих расходов на программное обеспечение и вычислительную технику.

$$K_{\text{проч}} = (26648 + 1417 + 532) * 0.03 = 858 \text{ рублей.}$$

$$K_{\text{пр}} = 26648 + 1417 + 532 + 858 = 29455 \text{ рублей.}$$

Состав данного пункта затрат по компонентам изображен на рисунке 42.

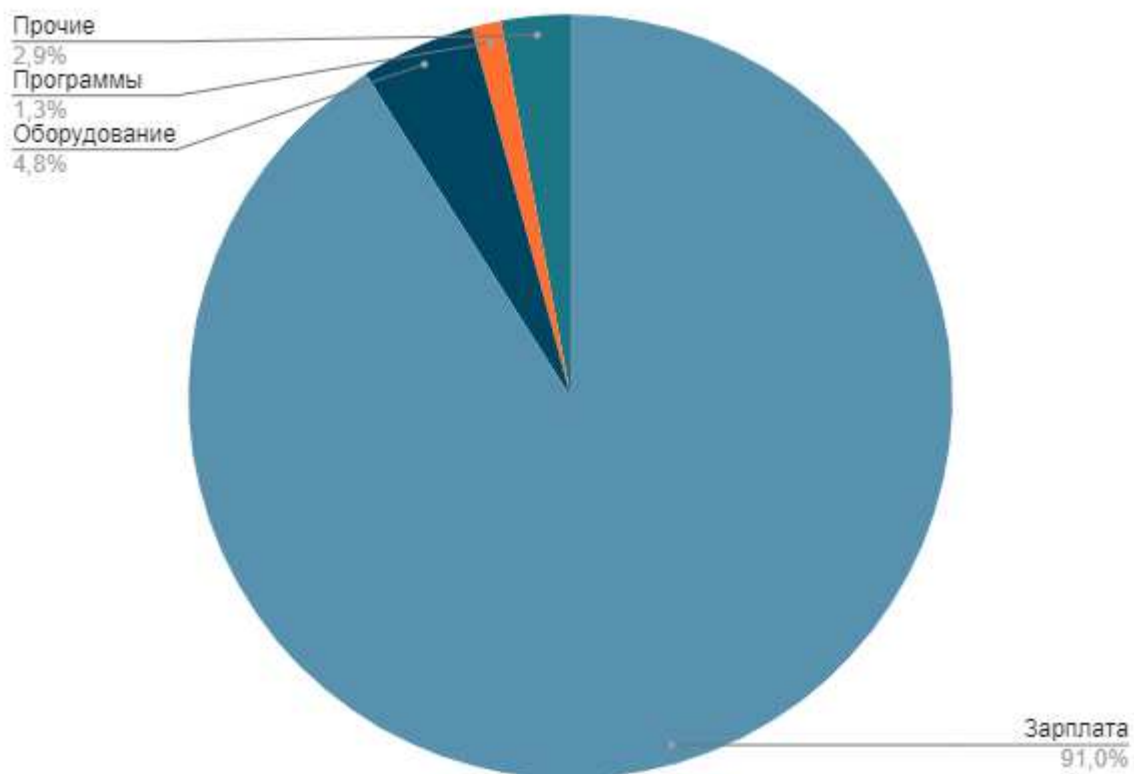


Рисунок 42 – Состав затрат на разработку

Исходя из диаграммы, можно сделать вывод, что самым затратным пунктом является заработная плата разработчика на весь срок разработки и составляет 91% от всей стоимости.

$K_{тс}$ – затраты на технические средства управления должны быть равны нулю, потому что данная программа будет функционировать на устройствах пользователей.

$K_{лс}$ – затрат на создание линий связи, интернет-соединения просто нет, потому что программный продукт не требует подключения к сети.

$K_{по}$ – затрат на программные средства для использования готового продукта нет, так как приложение полностью автономно.

$K_{ио}$ – затраты на формирование информационной базы отсутствуют, потому что они бесплатны;

$K_{об}$ – затраты на обучение персонала отсутствуют, потому что обучение персонала входит в обязанности разработчика;

$K_{оз}$ – затраты на опытную эксплуатацию уже включены в заработную плату разработчика.

Все пункты капитальных затрат, кроме затрат на разработку, равны нулю, и потому $K=K_{пр}$.

3.1.3 Эксплуатационные затраты

Расчет эксплуатационных затрат происходит по формуле 6:

$$C = C_{зп} + C_{ао} + C_{то} + C_{гс} + C_{ни} + C_{проч}, \quad (6)$$

где $C_{зп}$ - зарплата персонала, работающего с информационной системой;

$C_{ао}$ - амортизационные отчисления;

$C_{то}$ - затраты на техническое обслуживание;

$C_{гс}$ - затраты на использование глобальных сетей;

$C_{ни}$ - затраты на носитель информации;

$C_{проч}$ - прочие затраты.

Особенность данного программного продукта в том, что он является полностью автономным и никаких средств для эксплуатации не требует.

Так как все расходы равны нулю, то следует, что $C = 0$.

3.1.4 Расчет ТСО

Прямые затраты рассчитываются по формуле (7)

$$DE = DE1 + DE2 + DE3 + DE4 + DE5 + DE6 + DE7 + DE8, \quad (7)$$

где DE1 – капитальные затраты. По результатам расчетов равняются 29455 рублей;

DE2 – расходы на управление информационными технологиями. Равняются 0;

DE3 – расходы на техническую поддержку автоматизированного обеспечения и программного обеспечения. Равняются 0;

DE4 – расходы на разработку прикладного программного обеспечения внутренними силами. Равняются 0;

DE5 – расходы на аутсорсинг. Тут представляется возможность привлечения людей со стороны для тестирования проекта. Затраты на это равны 1500 руб;

DE6 – командировочные расходы. Равняются 0;

DE7 – расходы на услуги связи. Равняются 0;

DE8 – другие группы расходов. Равняются 0.

DE = 29286 рубля,

TCO = DE + DE5 = 29455 + 1500 = 30955 рубля.

Состав совокупной стоимости владения системой проекта представлен в таблице 5.

Таблица 5 – Совокупная стоимость владения

| Виды затрат | Планируемая сумма, руб. |
|--|-------------------------|
| Зарплата сотрудника | 26648 |
| Затраты на средства вычислительной техники | 1417 |
| Затраты на инструментальные программные средства | 532 |
| Прочие затраты на разработку | 858 |

3.2 Оценка рисков при реализации проекта «Игра-квест»

Проект подвержен малому количеству рисков, которые так или иначе могут сказаться на разработке данного проекта. Риски проекта продемонстрированы в таблице 6.

Таблица 6 – Оценка рисков проекта

| № | Описание риска | Вероятность возникновения риска | | | Степень воздействия риска на проект | | |
|--------------------------|-----------------------------|---------------------------------|----------|-----------------|-------------------------------------|------------|---------|
| | | маловероятно | вероятно | весьма вероятно | неопасный | допустимый | опасный |
| Внешние факторы риска | | | | | | | |
| 1 | Реализационный риск | | | + | | | + |
| 2 | Человеческий фактор | | | + | | + | |
| Внутренние факторы риска | | | | | | | |
| 1 | Снижение качества продукции | | + | | + | | |

Данный проект разрабатывается в очень нестабильный период, поэтому вероятность риска постоянно растет.

3.3 Выводы по разделу «Оценка экономической эффективности внедрения и разработки проекта «Игра-квест»»

При выполнении данной части выпускной квалификационной работы были произведены расчеты всех затрат по методике ТСО.

Проект не имеет доходов в связи с чем экономический эффект не был рассчитан.

Целесообразность проекта обусловлена привлечением абитуриентов путем прохождения игры квеста для поступления в ХТИ – филиал СФУ.

ЗАКЛЮЧЕНИЕ

1. Проведено ознакомление с основной деятельностью Хакасского технического института - филиала СФУ.
2. Построена структурная схема предприятия.
3. Сформулированы цели и задачи игры, а также обоснован выбор жанра.
4. Обоснован выбор программных средств для разработки. В данном проекте выбран Game Maker Studio 2.
5. Описана поэтапная разработка игры.
6. Проведена оценка планируемой совокупности стоимости разработки проекта.
7. Составлен перечень возможных рисков и выполнена их оценка.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Выполнение и защита выпускной квалификационной работы по направлению 09.03.03 «Прикладная информатика» [Электронный ресурс] : метод. указания / сост. Е. Н. Скуратенко, В. И. Кокова, И. В. Янченко ; Сиб. федер. ун-т, ХТИ – филиал СФУ. – Абакан : ХТИ – филиал СФУ, 2017.– Режим доступа: https://e.sfu-kras.ru/pluginfile.php/1368122/mod_resource/content/1/Met_1050.pdf.
2. ГОСТ Р 51904-2002. ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ [Электронный ресурс]: Электронный фонд правовых и нормативно-технических документов. – Режим доступа: <https://docs.cntd.ru/document/1200030195>.
3. Движок Unity – особенности, преимущества и недостатки [Электронный ресурс]: Игровой портал Cubiq. – Режим доступа: <https://cubiq.ru/dvizhok-unity/>.
4. Десятка лучших движков для создания своих собственных игр [Электронный ресурс] / Хабр. – 2015. – 10 апреля. – Режим доступа: <https://habr.com/ru/post/257019/>.
5. Информация о ХТИ – филиале СФУ [Электронный ресурс]: Сайт Хакасского технического института – филиала «Сибирского Федерального Университета». – Режим доступа: <http://khti.sfu-kras.ru/>.
6. Качаем скиллы, отдыхая. Топ-10 лучших игр для программистов [Электронный ресурс]: JavaRush. – 2020. – 2 марта. – Режим доступа: <https://javarush.ru/groups/posts/2571-kachaem-skillih-otdihkhaja-top-10-luchshikh-igr-dlja-programmistov>.
7. Компьютерная игра [Электронный ресурс]: Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Компьютерная_игра.
8. Методология DFD [Электронный ресурс]: Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/DFD/>

9. Методологии IDEF0 [Электронный ресурс]: Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/IDEF0>.
10. О Unity: Обзор движка [Электронный ресурс]: ITkeys. – Режим доступа: <https://itkeys.org/about-unity>.
11. Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности СТО 4.2-07-2014, Красноярск: ИПК СФУ, 2014. – 60 с. [Электронный ресурс]. – Режим доступа: <http://about.sfu-kras.ru/node/8127>.
12. Солозобов, О. GDevelop [Электронный ресурс] / О. Солозобов // Рейтинг топ-100 технологий для создания MVP стартапа. – Режим доступа: <https://8d9.ru/program/gdevelop>.
13. Управление IT-проектом. Курсовая работа [Электронный ресурс]: методические указания / Е. Н. Скуратенко, И. В. Янченко, В. И. Кокова; Сиб. федер. ун-т: ХТИ – филиал СФУ, 2018. – Режим доступа: http://89.249.130.59/docs/Met_1082.pdf.
14. Хмелева, А. 11 доступных движков для тех, кто хочет начать создавать свои игры [Электронный ресурс] / А. Хмелева // Компьютерра. – 2020. – 16 апреля. – Режим доступа: <https://www.computerra.ru/265077/eti-instrumenty-pomogut-vam-nachat-sozdavat-videoigry/>.
15. Что такое движок Unreal Engine и зачем на него переходить? [Электронный ресурс] / Ain. – 2020. – 22 мая. – Режим доступа: <https://ain.ua/2020/05/22/chto-takoe-dvizhok-unreal-engine/>.
16. Construct 2. Создавать собственные игры еще никогда не было так просто [Электронный ресурс]: Медиа портал Uptodown. – Режим доступа: <https://construct-2.ru.uptodown.com/>.
17. GameMaker Studio. Создавайте собственные игры для любых платформ [Электронный ресурс]: Медиа портал Uptodown. – Режим доступа: <https://gamemaker-studio.ru.uptodown.com/windows>.

18. GameMaker Studio 2 Manual. Главная страница [Электронный ресурс]. – Режим доступа: <https://manual-ru.yoyogames.com/#t=Content.htm>.
19. Flexbox Froggy Уровень 1. [Электронный ресурс]: Flexbox Froggy. – Режим доступа: <https://flexboxfroggy.com/#ru>

Выпускная квалификационная работа выполнена мной самостоятельно.
Использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в одном экземпляре.

Библиография 19 наименований.

Один экземпляр сдан на кафедру.

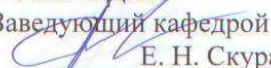
« ____ » _____ 2021 г.

_____ Зобов Даниил Викторович

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой

Е. Н. Скуратенко
подпись
« 21 » июня 2021 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

Разработка обучающей игры с элементами квеста для абитуриентов,
поступающих на направление
«Прикладная информатика»

Руководитель  21.06.2021 доцент, канд. физ.-мат. наук А.Н. Таскин
подпись, дата

Выпускник  21.06.2021 Д. В. Зобов
подпись, дата

Консультанты
по разделам:

Экономический  Е. Н. Скуратенко
подпись, дата

Нормоконтролер  В. И. Кокова
подпись, дата

Абакан 2021

Студенту Зобову Даниилу Викторовичу

Группа ХБ 17-03

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Разработка обучающей игры с элементами квеста для абитуриентов, поступающих на направление «Прикладная информатика»

Утверждена приказом по институту № 222 от 08.04.2021 г.

Руководитель ВКР: А. Н. Таскин, доцент, канд. физ.-мат. наук, ХТИ – филиал СФУ

Исходные данные для ВКР: заказ ХТИ – филиала СФУ.

Перечень разделов ВКР:

1. Анализ предметной области проекта «Игра-квест».
2. Описание процесса разработки информационной системы «Игра-квест».
3. Оценка экономической эффективности разработки и внедрения проекта «Игра-квест».

Перечень графического материала: нет

Руководитель ВКР



подпись

А. Н. Таскин

Задание принял к исполнению



подпись

Д. В. Зобов

«08» апреля 2021 г.