МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно—Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ / Г.И. Радченко
«___»_____ 2020 г.

Разработка приложения для изучения языка C# в игровой форме

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,
к.т.н., доцент каф. ЭВМ
_____ / В.А. Парасич
«___»_____ 2020 г.

Автор работы,
студент группы КЭ-405
_____ / А.А. Самойлов
«___»_____ 2020 г.

Нормоконтролёр,
ст. преп. каф. ЭВМ
_____ / С.В. Сяськов
«___»_____ 2020 г.

Челябинск 2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно—Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ / Г.И. Радченко
«___»_____ 2020 г.

ЗАДАНИЕ

на выпускную квалификационную работу бакалавра

студенту группы КЭ—405
Самойлову Александру Андреевичу
обучающемуся по направлению
09.03.01 «Информатика и вычислительная техника»

1. Тема работы: "Приложение для изучения языка C# в игровой форме" утверждена приказом по университету от 24 апреля 2020г. №627.

2. Срок сдачи студентом законченной работы: 1 июня 2020 г.

3. Исходные данные к работе: техническое задание на разработку приложения для изучения языка C# в игровой форме.

Задачей является разработка приложения в игровой форме, в котором учебные материалы по курсу высокоуровневого программирования будут представлены в лёгкой, упрощенной форме.

Изучить технологии создания игр и приложений без использования готовых решений.

4. Перечень подлежащих разработке вопросов:

- разработка для операционной системы Microsoft Windows;
- отображение уведомлений об ошибках;
- поддержка разных языков: русский, английский;
- изменения настроек приложения;

- настройка системы на уровни подготовки: новичок, уверенный пользователь и продвинутый пользователь и соответствующее ранжирование заданий;

- обеспечить возможность ввода/корректировки заданий;

- наполнение справочным материалом.

5. Дата выдачи задания: 1 декабря 2019 г.

Руководитель работы _____ / В.А. Парасич

Студент _____ / А.А. Самойлов

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2020	
Разработка модели, проектирование	01.04.2020	
Реализация системы	01.05.2020	
Тестирование, отладка, эксперименты	15.05.2020	
Компоновка текста работы и сдача на нормоконтроль	24.05.2020	
Подготовка презентации и доклада	30.05.2020	

Руководитель работы _____ / В.А. Парасич
Студент _____ / А.А. Самойлов

АННОТАЦИЯ

Самойлов А.А. Разработка приложения для изучения языка C# в игровой форме. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2020, 70с., 17 ил., библиогр. список – 13 наим.

В рамках выпускной квалификационной работы производится детальный анализ современных технических решений и учебных материалов, выявляются их недостатки и на основе полученного опыта разрабатывается программный продукт, упрощающий учебный процесс. Организуется разработка программного комплекса с использованием таких технологий, как *C/C++, SDL 2.0, SDL image, SDL mixer, OpenGL*. Производится выборка и анализ результатов работы системы, выявление недостатков и устранение неисправностей. Доказывается способность предлагаемой архитектуры к обеспечению успешного цикла решения задач определенного класса.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Освоение любого языка программирования предполагает наличие системного и структурного мышления, которое всегда необходимо при разработке программы. Изучение основ программирования лучше начинать с простых алгоритмов, структур и языков программирования, например, с учебных задач на языке C#. Важную роль играет большое количество информации, которую нужно будет освоить при изучении.

Язык программирования C# является высокоуровневым и достаточно прост в освоении. Это оптимальный вариант для изучения основ разработки и получения технических навыков, необходимых при овладении более сложными и масштабными технологиями разработки программного обеспечения.

Целью представленной выпускной квалификационной работы является разработка учебного программного комплекса, предоставляющего учебные задачи в игровой форме. Это способствует скорости восприятия учебного материала, а также повышает мотивацию обучающегося.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

В настоящее время язык программирования C# изучается на начальных курсах университетской программы технических специальностей. Этот язык очень популярен, из-за своей простой структуры и понятного интерфейса. Благодаря своим преимуществам, он позволяет за короткие сроки разработать своё приложение для операционной системы Windows.

1.1 Постановка задачи

Задача выпускной квалификационной работы — разработать программный комплекс, предоставляющий учебные задачи в игровой форме.

1.2 Обзор аналогов

Перед началом курсового проектирования был проведен анализ и выявлены следующие программные продукты, которые зарекомендовали себя, как лучшие учебные платформы.

Codecademy Hour of Code.

Codecademy — это один из самых популярных сервисов для обучения программированию. Он привлекает своей простотой, интерактивностью и игровой системой обучения. Недавно сервис выпустил своё приложение для iOS, которое поможет сделать первые шаги в программировании. Оно позволяет в простой и игровой форме обучиться основам программирования. Речь идет не о создании собственных игр, а о понимании, что такое переменные, какие они бывают, основы написания кода и многое другое. Все занятия разделены на группы, в каждой группе есть несколько заданий. Изначально есть всего 5 заданий, но при первом включении спросят о том, стоит ли уведомлять при поступлении новых заданий. Исходя из этого, новые задания будут добавляться постепенно. С каждым новым уроком, задания становятся все сложнее и сложнее, но сохраняют свою игровую форму.[1]

CodeCombat.

CodeCombat ориентирован на учителей и учеников, но играть может каждый. Эта достаточно яркая и увлекательная платформа подойдет для практики в таких языках как *Python, JavaScript, CoffeeScript, HTML* и *CSS*. На

начальном уровне сложности будут использоваться базовые команды. Далее команды усложняются. Можно играть самому, участвовать в соревнованиях и многопользовательских вариантах игр — всё это поможет не заскучать во время отработки в этой игре полученных ранее навыков программирования. Имеется возможность играть бесплатно, но получить доступ ко всему контенту можно только по платной подписке. [2]

CSS Diner.

CSS Diner — простой, но достаточно увлекательный способ изучения основ CSS. Последовательно проходя 32 уровня, в игровой манере можно познакомиться с основами работы CSS-селекторов. Начиная с простых селекторов по классам и тэгам, уровни этой бесплатной игры постепенно усложняются, основываясь на том, что было на предыдущих уровнях. Под конец нужно будет использовать сложные структурные селекторы и их комбинации. [3]

Untrusted.

Untrusted — это приключенческая игра, которая поможет проверить и закрепить знания и навыки *JavaScript* для решения различных проблем. Это увлекательная бесплатная игра поможет отработать достаточно сложные навыки применения *JavaScript*. [4]

1.3 Недостатки существующих решений

Каждый из аналогов имеет свои преимущества, однако, среди всех имеющихся решений существуют следующие недостатки:

- отсутствие русских аналогов. Представленные решения имеют либо частичный перевод, либо полное его отсутствие, что усложняют задачу обучения для тех студентов, которые плохо разбираются в иностранном языке или не знают его вовсе;
- невозможность запускать при отсутствии интернет соединения. Так, например, большинство представленных аналогов имеют реализацию в виде веб-приложений и даже если попытаться загрузить все исходные файлы заранее, пользоваться платформой все равно будет невозможно,

т.к. она всё время обращается к серверу и, не получив ответа, перестает функционировать;

- однотипность заданий. Большинство задач, представленных на платформе, имеют одну и ту же структуру, не предоставляя разнообразных способов решения и действующих по одному и тому же алгоритму;
- отсутствие открытого исходного кода. Все аналогичные учебные ресурсы представляют собой закрытые разработки и не дают возможности как либо дополнять или изменять уже существующие модули программы.

Таблица 1 - Сравнение аналогов

Решение	Бесплатный доступ	Поддержка русского языка	Модуль языка C#	Открытый исходный код	Игровая форма
Codecademy Hour of Code	+	+	+	—	—
CodeCombat	Подписка	+	—	—	+
CSS Diner	+	—	—	—	+
Untrusted	+	—	—	—	+

1.4 Выбор технологических решений

При разработке проекта использовался ряд технологий, среди которых *C/C++, SDL 2.0, SDL image, SDL mixer, OpenGL*, компилятор Microsoft .NET v2.0.5.

SDL (Simple DirectMedia Layer) — это свободно распространяемое программное обеспечение, представляет собой мультимедийную библиотеку, реализующую программный интерфейс к графической подсистеме. Активно используется при разработке кроссплатформенных программ, а также игр.[5][6]

SDL mixer — подбиблиотека, предоставляющая функции для организации сложного аудио, в основном, сведение звука из нескольких источников. [7][8]

SDL image — подбиблиотека, предназначенная для поддержки различных растровых форматов изображений. [9][10]

OpenGL — это открытый и мобильный стандарт в области компьютерной графики. На данный момент он является одним из самых популярных

графических стандартов во всём мире. Одно из основных преимуществ данной библиотеки — программы, написанные с помощью *OpenGL* можно переносить практически на любые платформы, получая при этом одинаковый результат. Если устройство поддерживает какую-то функцию, то эта функция выполняется аппаратно, если нет, то библиотека выполняет её программно. [11]

Simple DirectMedia Layer лишён данных недостатков и предоставляет разработчику полный функционал, включающий изменения самой среды разработки в том числе.

Microsoft .NET компилятор — это готовое техническое решение, которое представляет собой специальную программу, предназначенную для трансляции всех модулей программы, написанных на языке программирования C# в эквивалентные программные уровни на промежуточном языке программирования CIL (Common Intermediate Language), для дальнейшей их сборки операционной системой Windows.

2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

При постановке задачи, проектировании технического задания и в разборе существующих аналогов был выявлен и определен ряд требований, которым должен соответствовать разрабатываемый программный продукт. Они разделяются на функциональные и нефункциональные.

2.1 Функциональные требования

Одним из самых важных моментов при постановке задачи является определение функциональных требований, т. е. функций, необходимых пользователю, которые нужно реализовать для достижения цели, а также программных решений, которые стоит учесть при проектировании для оптимизации затрат времени на разработку.

Итак, основными функциональными требованиями являются:

- разработка для операционной системы Microsoft Windows;
- отображение уведомлений об ошибках;
- поддержка разных языков: русский, английский;
- возможность изменения настроек приложения;

- настройка системы на уровни подготовки: новичок, уверенный пользователь и продвинутый пользователь и соответствующее ранжирование заданий;
- обеспечить возможность ввода/корректировки заданий;
- наполнение справочным материалом.

Для наглядного отображения структуры работы приложения можно использовать диаграммы вариантов использования бизнес процессов, которые изображены на рисунках 1 и 2.

2.2 Нефункциональные требования

Среди нефункциональных требований одними из основных являются:

- возможность сохранения игрового процесса;
- влияние выполняемых команд на объекты игрового мира;
- наличие игрового персонажа;
- возможность выбора уровня подготовки и типа задания;
- возможность отключения звука;
- возможность подсчёта очков и времени прохождения.

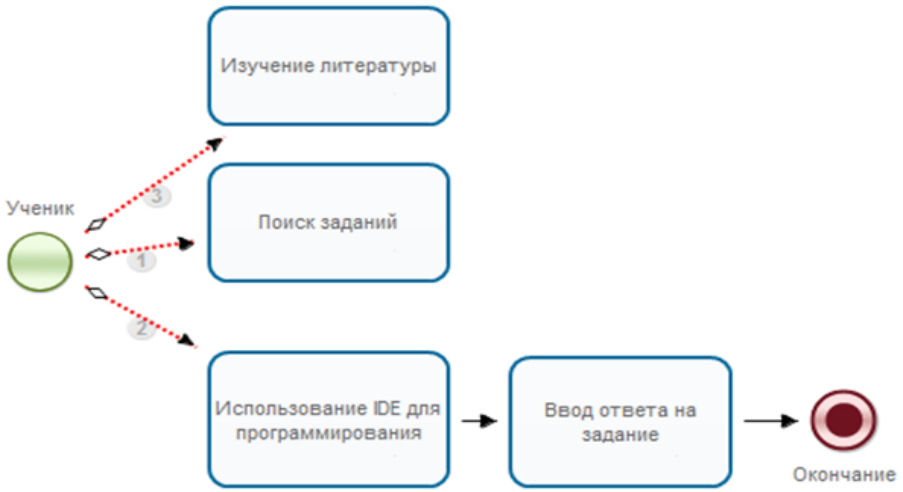

Рисунок 1 — Существующие бизнес процессы

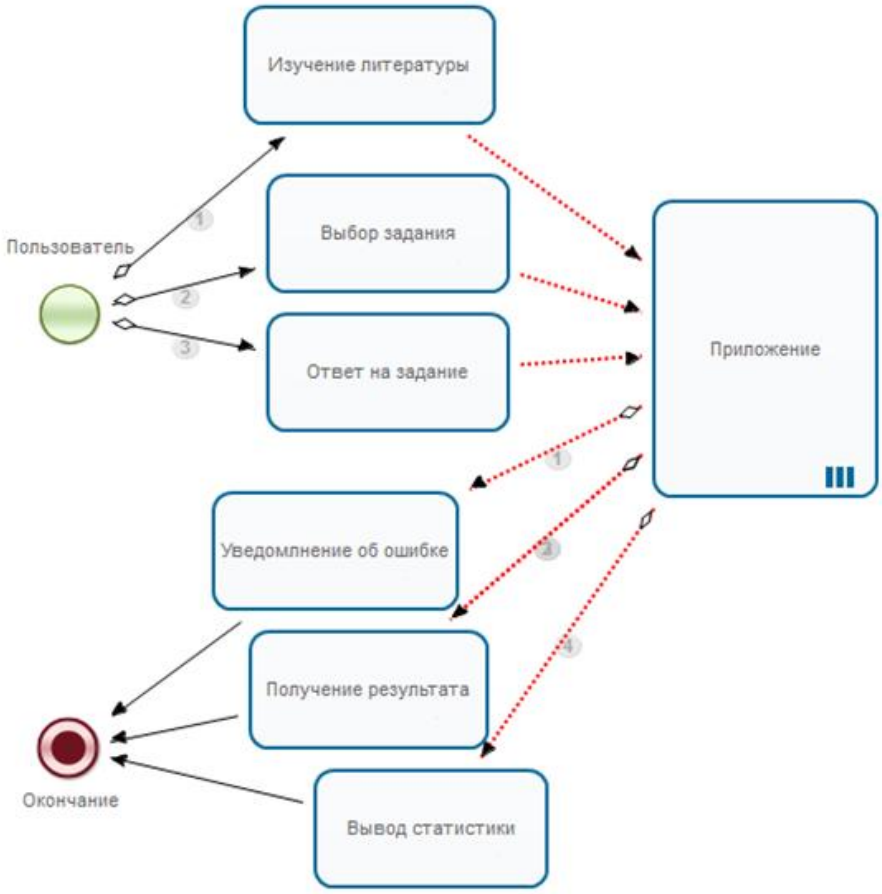

Рисунок 2 — Бизнес процессы разрабатываемого проекта

3 ПРОЕКТИРОВАНИЕ

Разрабатываемый проект логически можно разделить на следующие модули:

- модуль разработки;
- модуль интерфейса;
- модуль игрового мира;
- модуль интерпретации;
- модуль инициализации;
- модуль деинициализации;
- внешние подключаемые модули.

3.1 Архитектура предлагаемого решения

Структура разрабатываемой в данной выпускной квалификационной работе программы схематично изображена на рисунке 3. В основном блоке программы расположены 4 основных модуля.

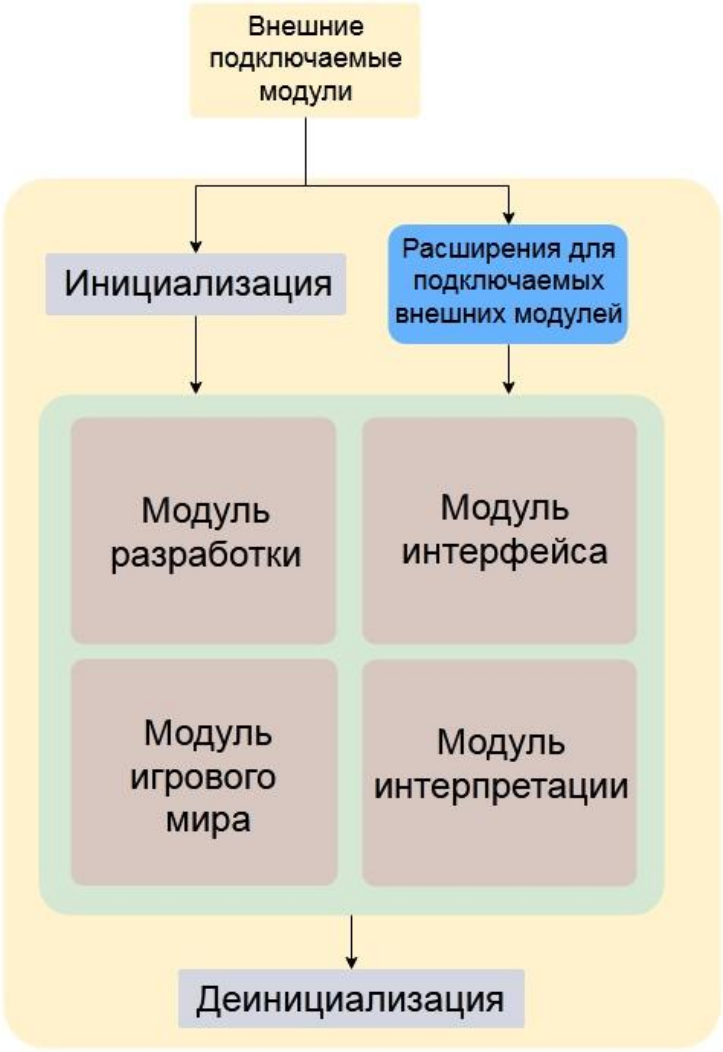

Рисунок 3 — Архитектура предлагаемого решения

3.2 Модуль разработки

Модуль разработки представляет из себя отдельную подпрограмму, которая функционирует внутри игровой платформы. Она предоставляет функционал разработчика, редактора уровней, позволяет обновлять, модернизировать и дополнять игровой мир.

Рисунок 4 — Архитектура модуля разработки

3.3 Модуль интерфейса

Модуль интерфейса необходим для того, чтобы предоставить пользователю все инструменты взаимодействия с игровым миром, в том числе игровое поле, редактор кода или область с заданием, в которой можно набирать текст ответа.

Все возможности модуля интерфейса разделяются на две группы. Первая группа для пользователей, помимо основных функций, предоставляет следующий функционал:

- возможность отображения уведомлений об ошибках;
- поддержка разных языков: русский, английский;
- возможность сохранения игрового процесса.

Вторая группа, для администратора-разработчика:

- наличие консоли;
- наличие игрового редактора уровней и заданий.

Архитектура модуля интерфейса отображена на рисунке 5.

Рисунок 5 — Архитектура модуля интерфейса

3.4 Модуль игрового мира

Модуль игрового мира представляет собой объекты, текстуры, игрока, задания, модели, анимации, спрайты, игровые звуки и другие элементы, которые относятся к игровому миру. Модуль игрового мира схематично изображен на рисунке 6.

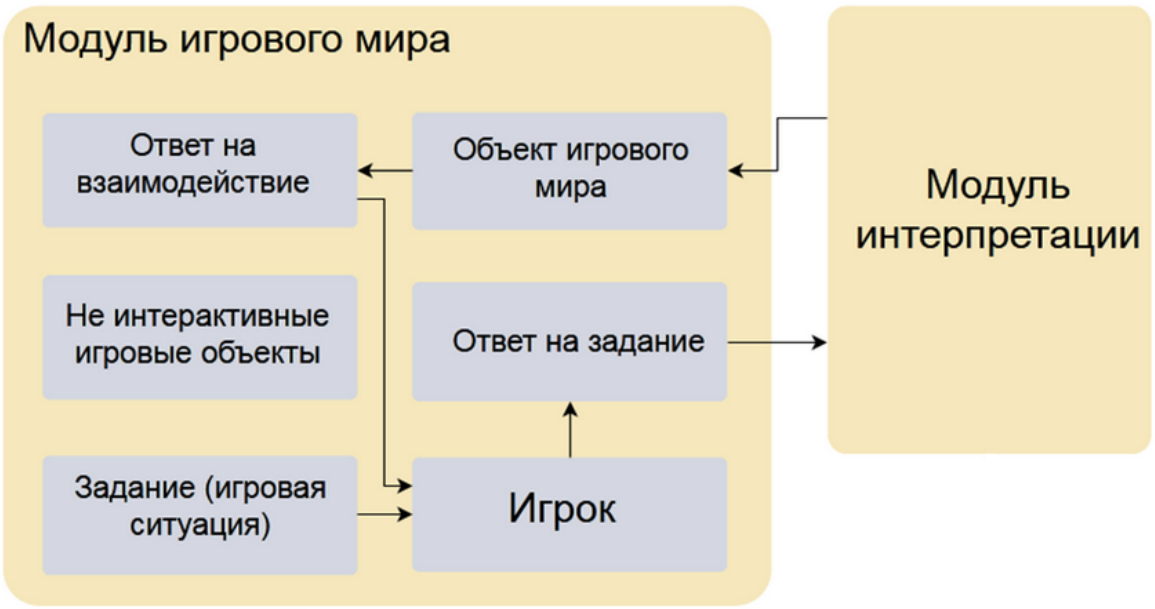

Рисунок 6 — Архитектура модуля игрового мира

3.5 Модуль интерпретации

Модуль интерпретации предназначен для проверки программного кода, если такой предусмотрен учебным заданием. В случае, если в коде допущена ошибка, то на экране появляется уведомление о том, что задание выполнено неверно.

3.6 Модули инициализации и деинициализации

Модули инициализации и деинициализации отвечают за запуск программного обеспечения, его корректную работу и правильное завершение программы.

3.7 Внешние подключаемые модули и расширения

Внешние подключаемые модули — это дополнительные программные продукты, которые необходимы для корректной работы учебной платформы, среди них такое программное обеспечение, как *SDL 2, openGL*, Конвертатор *iconv*[12] для конвертации в *utf8* или *ANSI* символов, *SDL 2 Image, SDL 2 TTF, SDL Mixer*.

Внешние модули взаимодействуют с инициализацией, т.е. проводят первые манипуляции и настраивают сами внешние модули для дальнейшего использования. Также сюда включаются и ресурсы, которые будут использоваться в основном блоке, осуществляется подготовка подмодулей, которые используется в основном цикле.

4 РЕАЛИЗАЦИЯ

Программа предполагает разделение на три уровня подготовки — 3 уровня сложности: начальный, уверенный, продвинутый. Каждый из уровней сложности имеет свой перечень заданий, в зависимости от уровня подготовки пользователя. Каждый разработчик может добавлять задания в раздел, для которого оно предназначались.

За прохождение уровня игроку начисляется определённое количество очков, а также показывается локальная статистика прохождения этого задания. После прохождения всех заданий данного уровня сложности показывается статистика модуля подготовки. Во вкладке "статистика" главного меню можно посмотреть суммарную статистику по всем уровням сложности.

Игроку предоставляется возможность выбрать задание из представленного перечня в соответствии с желаемым уровнем сложности. Предполагается такой список задач на выбор:

- работа с консолью;
- арифметические операции;
- условные операторы;
- работа с массивами;
- операторы циклов;
- работа со строками;
- использование методов и т.д.

После завершения курса обучения, игрок может увидеть статистику выполнения заданий, которая включает в себя подсчёт затраченного на выполнения задач времени, количество совершённых ошибок, количество полученных бонусных очков.

В начале каждого уровня расположены игровые экраны с подсказками и приветственным текстом, а также экран взаимодействия с игровым миром, с помощью которого осуществляется выполнение поставленной задачи.

Войдя в режим написания кода игроку предоставляется текст с заданием, поле ввода кода и отдельное окно с выходными данными. При совершении синтаксической ошибки игроку выводится соответствующее сообщение с указанием строки, в которой совершена. Отладка и отслеживание ошибок осуществляется за счёт компилятора Microsoft.Net.

Когда программа прошла тест на синтаксические ошибки, в окне выходных данных игроку выдаётся результат и оценка ответа — правильно решена задача или нет. Если задача решена верно, то выходные значения действуют на игровые объекты, например, передвижение, изменение размера или высоты, отрисовка текста на специальном экране, который эмулирует консоль.

После выполнения поставленной задачи, в игровом поле исчезает преграда, которая позволяет игроку дойти до конца уровня и получить очки. После завершения уровня игрок может посмотреть игровую статистику.

4.1 Проверка синтаксических ошибок

Для проверки синтаксических ошибок вводимого игроком текста используется компилятор Microsoft .Net. Для взаимодействия приложения с ним используются два файла с расширением *.bat*. .BAT — это пакетный файл, содержащий сценарий (набор команд). В первом файле запуск компилятора, во втором — запуск скомпилированного исполняющего файла.

Листинг 4.1.1 — Сценарии .bat файла compile.bat

```
C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe /utf8output /nologo
/out:resource\compiler\main.exe resource\compiler\main.cs >resource\compiler\error.txt
```

Листинг 4.1.2 — Сценарии bat файла compile.bat

```
resource\compiler\main.exe >resource\compiler\output.txt
```

Из представленных листингов видно, что вывод исполняющих файлов будет перенаправлен в файл, который можно будет прочитать из приложения. Для запуска этих *.bat* файлов используется WinAPI (Windows API — общее наименование набора базовых функций интерфейсов программирования приложений операционных систем семейств *Microsoft Windows*).

Листинг 4.1.3 — Запуск compile.bat

```
void CInterp::Compile()
{
    remove("resource/compiler/error.txt");
    remove("resource/compiler/main.exe");
    WinExec("resource\\compiler\\compile.bat", SW_HIDE);
}
```

Листинг 4.1.4 — Запуск make.bat

```
void CInterp::Make()
{
    remove("resource/compiler/output.txt");
    WinExec("resource\\compiler\\make.bat", SW_HIDE);
}
```

Для компиляции и исполнения итоговой программы необходимо время. Для того, чтобы не останавливать основное приложение, для каждого кадра будет проверяться существование выходных файлов *error.txt* и *output.txt*. Также учитывается тот факт, что программа может зациклиться, долго выполняться или выходные данные вовсе будут отсутствовать. Если выходные данные не появляются 10 секунд, то процесс выполнения выходного исполняющего файла

сбрасывается. Листинг реализации функции компиляции приведён в приложении А — листинг А.1.

4.2 Проверка на правильный результат

После того, как пользователь ввёл своё решение задачи и дождался окончание процесса проверки на синтаксические ошибки, то полученный результат выполнения данного решения сравнивается с эталонным, который находится в файле **название уровня*_good.txt*. Например, для нахождения корней квадратного уравнения нужно получить два числа (первый и второй корень). В начале считывается результат, который был получен игроком исходя из написанной им программы. Затем считывается файл с правильным ответом, после чего эти значения сравниваются. Если значения не совпадают, тогда полученный ответ является неправильным. Листинг реализации функции проверки на правильный ответ приведён в приложении А — листинг А.2.

4.3 Реализация взаимодействия игрока с объектами игрового мира

Игровые уникальные особенности достигаются за счет взаимодействия игрока с игровым миром с помощью команд, например, сортировка массива, которую можно представить в игровой ситуации в виде ступенек разной высоты для того, чтобы дойти до выхода из уровня, или вывод текста на игровую консоль, которая представлена в виде экрана. Листинг реализации функций взаимодействия со ступеньками приведён в приложении А — листинг А.3.

Также игрок может управлять своим персонажем на карте с помощью стандартного управления, например, с помощью клавиш управления курсором. Отличительной чертой разрабатываемого приложения является фокус на написании реального кода, а не только поиске алгоритма прохождения пути.

В качестве элемента дополнительной мотивации в игре присутствуют бонусы, которые открываются за решение задач. Также они отображаются в статистике игрока. Их можно встретить на некоторых уровнях и они находятся на них определённое время и вскоре исчезают. Этот элемент определяет на сколько быстро справился игрок с поставленной задачей(смог он их успеть подобрать или нет).

4.4 Реализация вывода текстовой информации

В процессе программирования большую часть времени занимает работа с текстом: набор программного кода, просмотр справок и документации, вывод информации о синтаксических ошибках и т.п. Именно поэтому в данном проекте много внимания уделяется взаимодействию пользователя с программным продуктом, чтению и вводу текста.

Первое, что делает программа при запуске - загружает дополнительные модули в память компьютера. В том числе к этим модулям относятся файлы шрифтов, как правило, это файлы с расширением *.ttf*. Для их правильного импорта в проект и корректной работы используется библиотека *SDL 2 ttf*, однако, в силу того, что графическое отображение текста данным инструментом не оптимально - оно требует слишком много ресурсов - применяется перевод шрифтов текста в текстуры, а затем загрузка в видеопамять. Эту задачу легко можно выполнить с помощью OpenGL, а для экономии памяти и лучшей производительности будем использовать лишь 256 символов таблицы ASCII кодировки ANSI Windows 1251. Стоит отметить, что шрифты формата *.ttf* могут иметь кодировку отличную от Windows 1251, поэтому для считывания каждого символа будем переводить их в необходимую нам кодировку с помощью библиотеки *iconv*. Листинг реализации загрузки шрифтов приведён в приложении А — листинг А.4.

Благодаря использованию описанных выше технических решений, возможно многократно вызывать отрисовку текста в любом месте программного кода без особых потерь производительности. Листинг реализации отображения текста приведён в приложении А — листинг А.5.

Для отображения большого объема текста, в качестве меры дополнительной оптимизации, используются списки отображения. Списки отображения — это группа команд OpenGL, сохраненных для последующего выполнения. Когда список отображения вызывается для выполнения, команды обрабатываются в порядке их появления в списке. Списки отображения хранятся в видеопамяти, что освобождает шину памяти видеокарты от постоянной пересылки команд.

4.5 Реализация поддержки мультиязычности

Для поддержки русского языка, как отмечалось выше, была использована кодировка ANSI Windows 1251. Windows 1251 состоит из двух частей — общие символы и символы кириллицы. В зависимости от выбора языка в приложении использовался тот или иной файл, который хранит в себе информацию о строках текста в английском и в русском вариантах. Чтение из файла происходит напрямую в память.

Для отображения более объёмного текста, например, для вывода справок, используются списки отображения, то есть чтение реализуется сразу в видеопамять. Листинг реализации поддержки мультиязычности приведён в приложении А — листинг А.6.

4.6 Реализация ввода и отображения текста в игровых окнах

Для удобства ввода и отображения информации во время игры, были созданы так называемые "игровые" окна. Их можно перемещать как окна в операционной системе Windows, чтобы их было удобно расположить. Всего окон 3:

- окно ввода текста программы;
- окно просмотра выходных данных или ошибок;
- окно с заданием, которое нужно выполнить, чтобы пройти уровень.

Для окна просмотра выходных данных были использованы списки отображения, так как получаемый результат может быть большим. Для окна ввода текста программы было использовано обычное отображение текста, так как это окно часто изменяется пользователем.

В первое окно пользователем вводится текст, то есть пишется программа, которую он может запустить и посмотреть результат во втором окне. Таким образом, обрабатывается каждое нажатие клавиш и передаётся в строку, которая затем отображается. Листинг реализации ввода текста приведён в приложении А — листинг А.7.

4.7 Итоги разработки

В процессе разработки было реализовано 14 модулей: *Console, Editor, Editor level, Extension, Interpritator, Localizator, Menu, Menu Page, Menu Table, Pad, App, Main, Player, World.*

Также были использованы 6 внешних подключаемых библиотек: *openGL, iconv, SDL2, SDL2_mixer, SDL2_ttf, SDL_image.*

Весь программный код состоит из 9095 строк кода. Список назначений модулей:

1. Console — консоль разработчика, в которую можно ввести специальные команды для быстрого доступа к какому-либо объекту приложения (например, изменение языка приложения). Листинг реализации консоли разработчика приведён в приложении А — листинг А.8.

2. Editor — редактор списка уровня. В данном модуле разработчику предоставляется функционал для создания новых игровых уровней, включающий в себя параметры уровня сложности, название и последовательность прохождения заданий.

3. Editor Level — непосредственно сам редактор уровня. С его помощью него разработчик создаёт игровой мир, наполняет его бонусами, объектами, триггерами.

4. Extension — модуль работы с внешними подключаемыми библиотеками.

5. Interpritator — в данном модуле осуществляется работа с компилятором, считывание синтаксических ошибок и проверка правильности результата выполнения задачи.

6. Localizator — осуществляет поддержку мультиязычности, т.е. русского и английского языков.

7. Menu — главное меню игры, в которых можно увидеть статистику, настройки, выбрать уровень или посмотреть справочный материал. Состоит из двух подмодулей: *Menu Page* и *Menu Table.*

8. Menu Page — представляет из себя страницу меню, в котором размещается контент (кнопки, ползунки).

9. Menu Table — предназначен для заглавного выборочного меню, которое всегда находится на экране. Оно осуществляет выбор страниц меню. Листинг реализации меню выбора страницы приведён в приложении А — листинг А.9.

10. Pad — модуль по работе с вводом ответа на задание и получения выходных данных. Представлен в виде оконных форм.

11. App — главный модуль игры. Осуществляет цикл, за который происходит расчёт параметров и отрисовка объектов.

12. Main — в данном модуле находятся основные функции и определения, которые используется во всей программе. Также осуществляется порядок инициализации всех остальных модулей посредством вызовов конструкторов.

13. Player — модуль игрока. Здесь осуществляется взаимодействие игрока с объектами игрового уровня, передвижение и отрисовка персонажа. Листинг реализации модуля игрока приведён в приложении А — листинг А.10.

14. World — модуль игрового уровня, который содержит триггеры, бонусы, объекты, задания, производит их отрисовку и проверку на столкновения.

5 ТЕСТИРОВАНИЕ

5.1 Условия выполнения программы

Для установки и запуска программы в операционной системе должны быть установлены драйвера для графического адаптера (видеокарты). Также обязательным условием является установка компилятора Microsoft .NET v2.0.5. Все остальные необходимые модули и библиотеки будут загружаться вместе с приложением.

5.2 Минимальные системные требования

Минимальные системные требования были сформированы исходя из используемых ресурсов системы с помощью программ Windows Task Manager и Process Explorer[13].

- процессор архитектурой x86 1.44 ГГц;
- видеокарта с 128 МБ видеопамяти;
- операционная система Windows 7 и выше;
- свободное пространство на накопителе 50 МБ;
- оперативная память 128 МБ.

5.3 Запуск программы

Для установки и запуска программы необходимо скачать и запустить архив с исходными данными, а затем запустить файл CodeSharp.exe.

Алгоритм работы программы следующий. При запуске приложения открывается главное меню с пунктами выбора, среди которых:

- выбор уровня;
- настройки;
- подсказки;
- статистика;
- выход.

Скриншот главного меню изображен на рисунке 7.

Рисунок 7 — Главное меню программы

Каждый из пунктов меню активен и выполняет свою индивидуальную функцию, на рисунках 8-11 представлены параметры некоторых из пунктов меню.

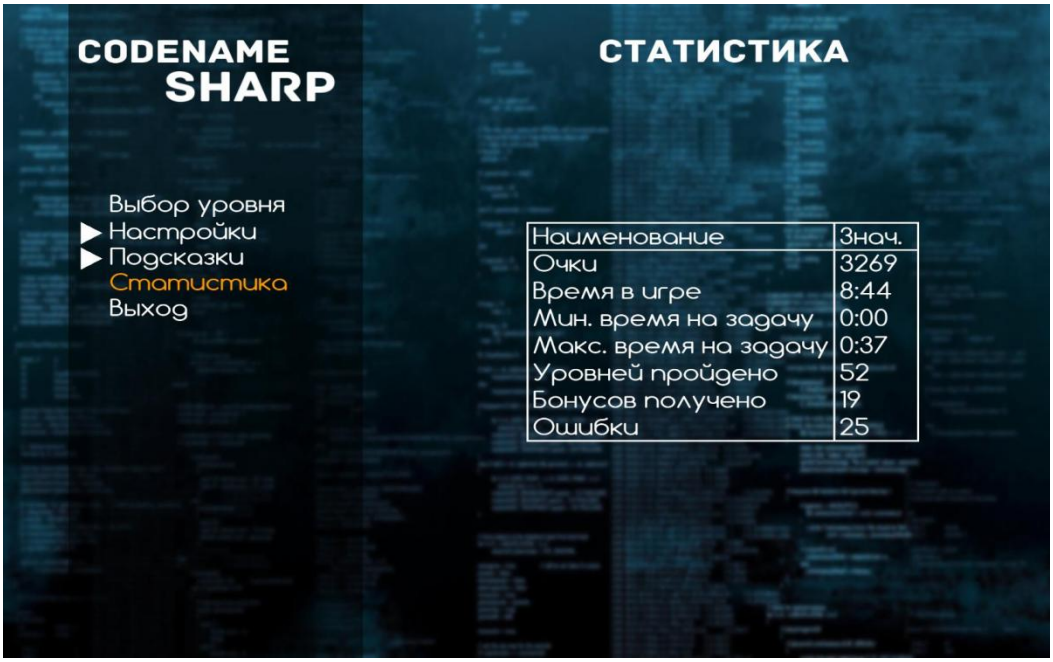

Рисунок 8 — Пункт меню "Статистика"

Раздел “Подсказки” содержит справочный материал по языку программирования C#, он разделен на 3 уровня подготовки:

- для новичка;
- для уверенного пользователя;
- для продвинутого пользователя.

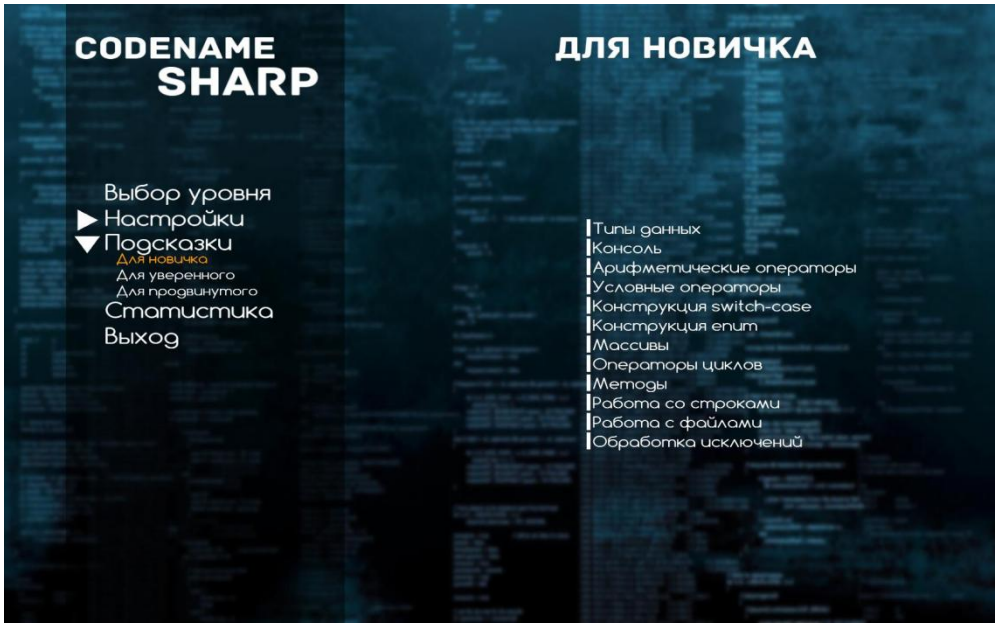

Рисунок 9 — Пункт меню "Подсказки"

Пункт меню “Настройки” включает в себя изменяемые параметры графики, звука и самой игры, в зависимости от требований пользователя. Так, например, можно изменить громкость музыки и звука или вовсе отключить их, также, имеется доступ к свойствам отображения видео.

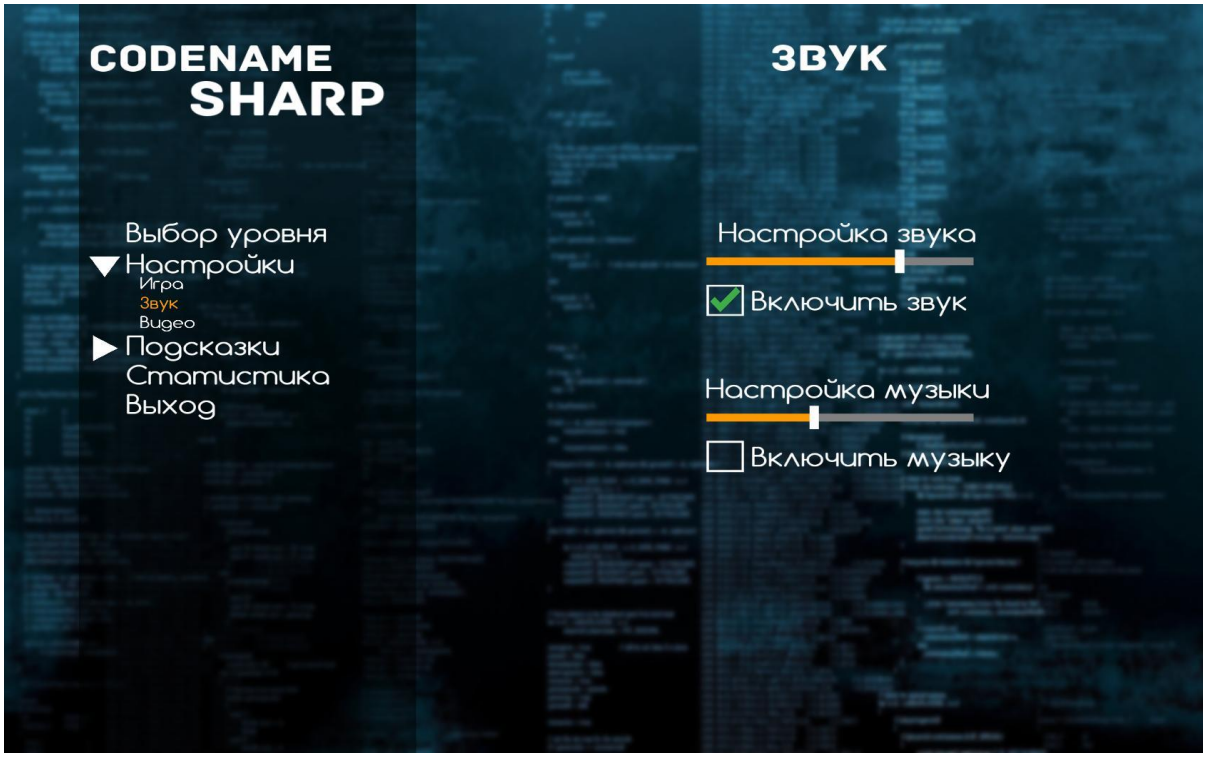

Рисунок 10 — Пункт меню "Настройки"

В соответствии с требованиями к выполнению проекта в рамках выпускной квалификационной работы была также реализована поддержка английского языка, который можно переключить в пункте меню “Настройки” - “Игра”.

Рисунок 11 — Пункт меню "Настройки" — "Игра"

Скриншот с выбором задания показан на рисунке 12.

Рисунок 12 — Выбор уровня

5.4 Работа программы

Программа представляет собой обучающую платформу с игровыми элементами. Все приложение разделено по уровням, в соответствии с уровнем подготовки и знания языка C# пользователем. Существует 3 уровня сложности:

- новичок;
- уверенный пользователь;
- продвинутый пользователь.

На рисунке 12 представлен скриншот программы при запущенном задании 1 сложности “Новичок”.

Рисунок 13 — Игровое поле задания 1, сложность "Новичок"

Игровое поле представлено в виде 2D платформера. Пользователю в управление дается персонаж, которому необходимо выполнять задания, в соответствии с сюжетом игры — проходить уровень.

Управление персонажем осуществляется с помощью клавиш клавиатуры:

- перемещение вверх-влево-вниз-вправо - клавиши WASD соответственно или клавиши стрелок;
- взаимодействие с предметами окружающего мира осуществляется за счет клавиш ENTER, “E”.

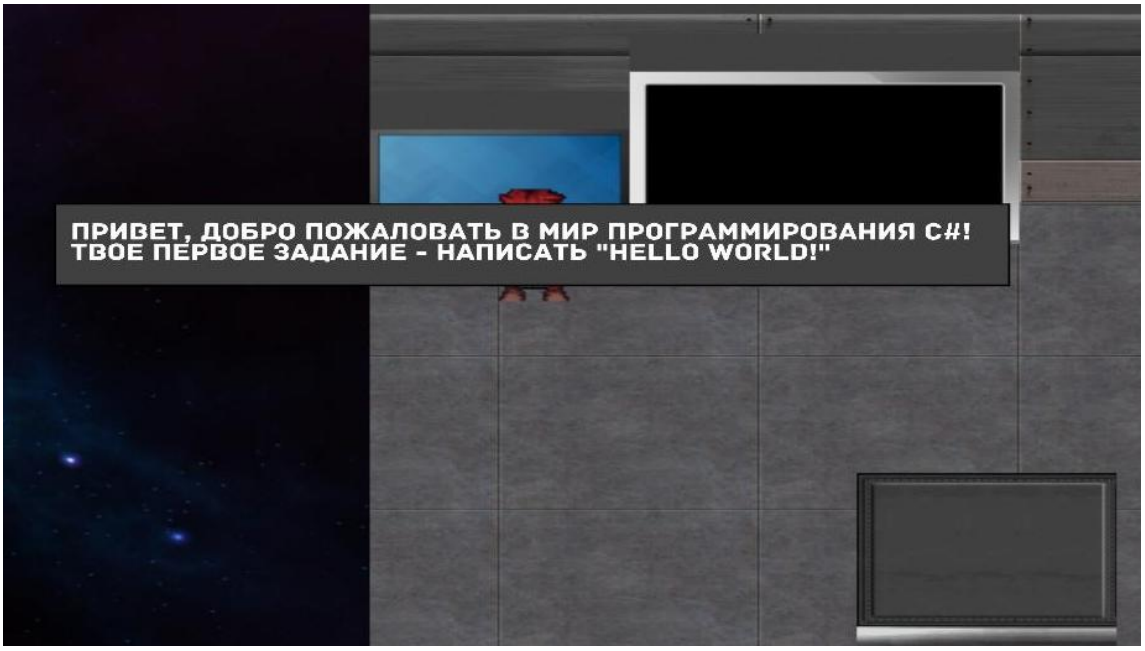

Рисунок 14 — Внутриигровое приветствие

Следующий шаг — решение задачи — осуществляется с помощью второго терминала, нужно подойти к нему и нажать клавишу взаимодействия (клавиша “E” или клавиша “ENTER”). При активации терминала откроются окна компилятора, задания и выходных данных.

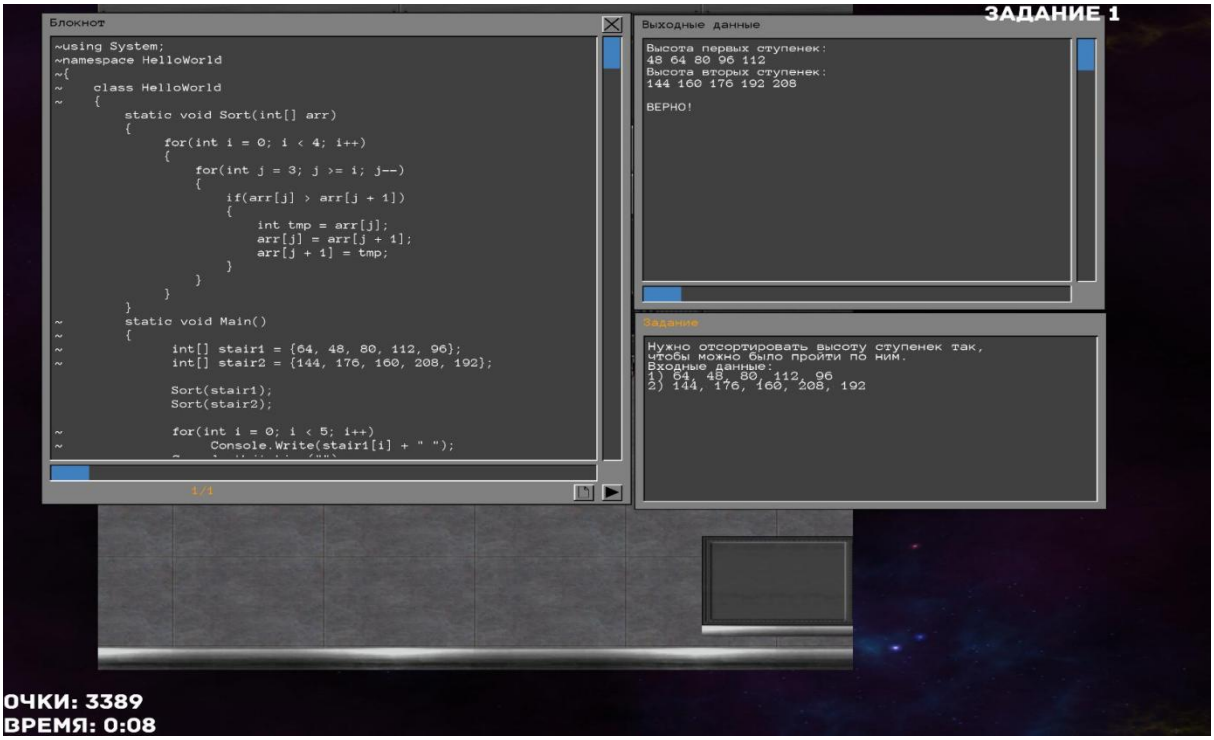

Рисунок 15 — Активация задания

После успешного выполнения задачи игроку показывается статистика прохождения уровня.

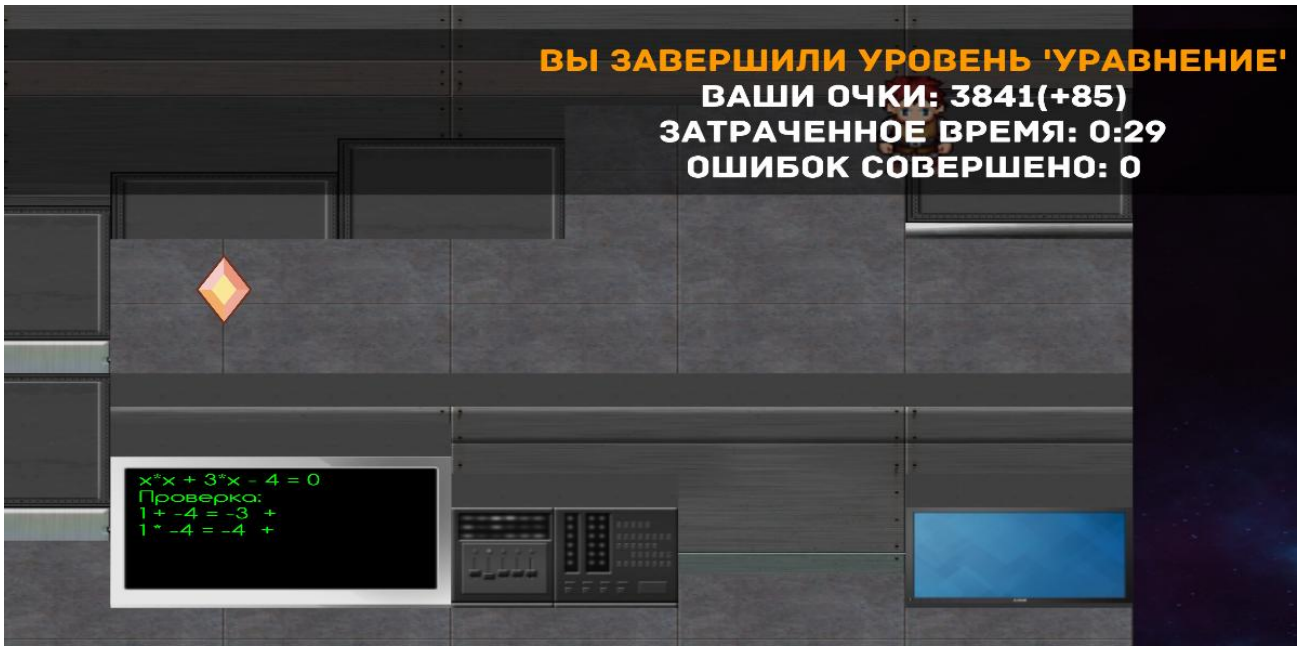

Рисунок 16 — Локальная статистика

После выполнения всех задач, игроку показывается общая статистика за все уровни.

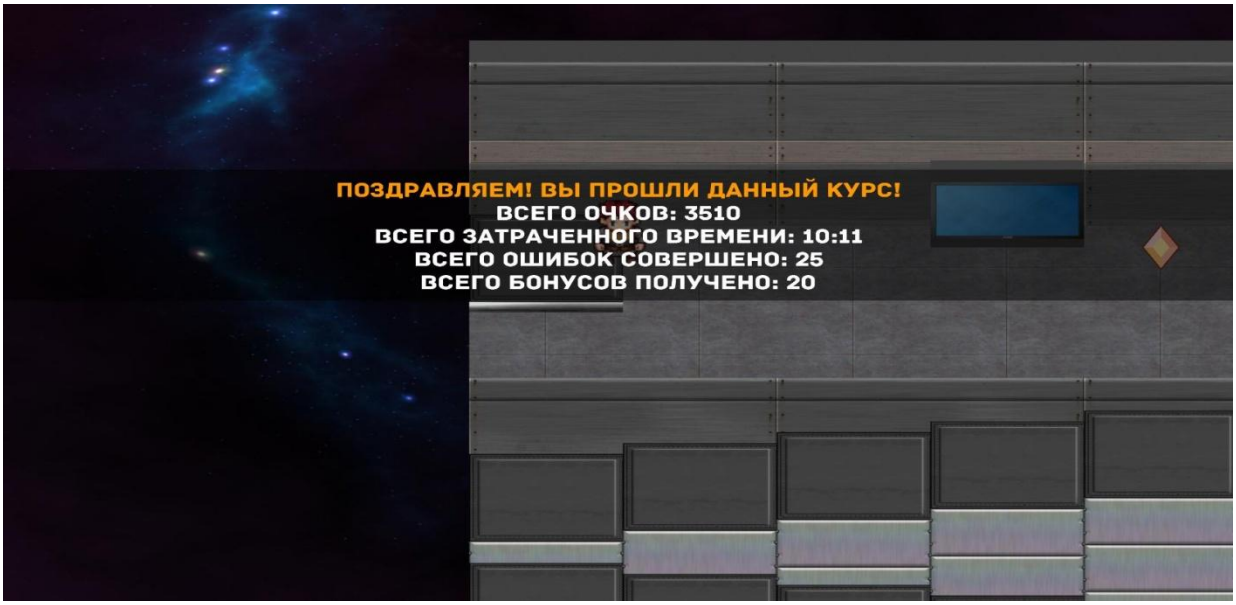

Рисунок 17 — Общая статистика

ЗАКЛЮЧЕНИЕ

В представленной выпускной квалификационной работе были рассмотрены подробности проектирования приложения для изучения языка программирования C# в игровой форме. Был проведен поиск и анализ исходных данных, а также поставлены задачи для разработки. Опираясь на анализ доступных технологий и сред разработки, была выбрана среда разработки и необходимые программные средства. На основе выбранных технологий было разработано программное решение, которое отвечает всем предъявленным в техническом задании требованиям.

За время дипломного проектирования были изучены технологии разработки приложений на языках программирования *C/C++,* библиотек *SDL 2.0,* а также библиотека для работы с компьютерной графикой *OpenGL*. Также подробно были изучены и проанализированы методики составления обучающих задач, учтены возможные индивидуальные особенности обучающихся, проведена научно-исследовательская работа.

Основываясь на опыте, полученном во время изучения новых технологий разработки и методик анализа, было разработано Windows-приложение Codename Sharp, которое полностью удовлетворяет поставленной в данном дипломном проекте задаче. Дальнейшее развитие программного продукта предполагает увеличение количества задач, внедрение новых типов заданий, а также, в связи с тем, что исходный код программы является открытым и свободно распространяемым в сети интернет, имеется возможность приглашения других разработчиков к работе над проектом и улучшению существующих модулей системы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Codecademy полный список / Страница скачивания // URL: https://www.codecademy.com/catalog/subject/all (дата обращения 12.01.2020).
2. CodeCombat официальный сайт / Страница скачивания // URL: https://codecombat.com/ (дата обращения 12.01.2020).
3. CSS Dinner онлайн / Страница скачивания // URL: https://flukeout.github.io/ (дата обращения 12.01.2020).
4. Игры для тренировки навыков программирование / Страница скачивания // URL: https://itvdn.com/ru/blog/article/dev-games (дата обращения 12.01.2020).
5. Страница скачивания SDL 2 / Страница скачивания // URL: https://www.libsdl.org/download-2.0.php (дата обращения 10.01.2020).
6. Документация SDL 2 / Страница скачивания // URL: https://wiki.libsdl.org/FrontPage (дата обращения 10.01.2020).
7. Страница скачивания SDL 2 mixer / Страница скачивания // URL: https://www.libsdl.org/projects/SDL_mixer/ (дата обращения 16.01.2020).
8. Документация SDL 2 mixer / Страница скачивания // URL https://www.libsdl.org/projects/SDL_mixer/docs/index.html (дата обращения 18.01.2020).
9. Страница скачивания SDL 2 image / Страница скачивания // URL: https://www.libsdl.org/projects/SDL_image/ (дата обращения 19.01.2020).
10. Документация SDL 2 image / Страница скачивания // URL: https://sdl.beuc.net/sdl.wiki/SDL_image (дата обращения 19.01.2020).
11. Документация openGL / Страница скачивания // URL: https://www.opengl.org/documentation/ (дата обращения 10.01.2020).
12. LibIconv for Windows / Страница скачивания // URL: http://gnuwin32.sourceforge.net/packages/libiconv.htm (дата обращения 15.01.2020).
13. Process Explorer / Страница скачивания // URL: https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer (дата обращения 20.05.2020).