Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС)

К ЗАЩИТЕ ДОПУСТИТЬ

Заведующий кафедры КИБЭВС

А.А.Шелупанов

АВТОМАТИЗИРОВАННАЯ СИСТЕМА «BOUNCING CUBE»

Курсовая работа по дисциплине «Технологии и методы программирования»

Пояснительная записка

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Руководитель  Аспирант кафедры КИБЭВС  И.Ю.Поляков  «\_\_\_» \_\_\_\_\_ 2017 г. | РАЗРАБОТЧИК  Студент гр. 715  Н.А.Телегин  ДОКУМЕНТАТОР  Студент гр. 715  З.Ю.Халметова  ТЕСТИРОВЩИК  Студент гр. 715  А.В.Сотпа |

Томск 2017

**Реферат**

Курсовая работа, 55 страниц, 23 рисунка, 7 таблиц, 6 источников, 2 приложения.

VISUAL STUDIO, C# UNITY 3D, BOUNCING CUBE, РАЗВЛЕКАТЕЛЬНАЯ ИГРА.

Объектом разработки является автоматизированная система на языке программирования C# в среде разработки программного обеспечения Visual Studio.

Цель работы – является разработка автоматизированной системы на платформе Unity, усовершенствование навыков программирования на языке C#. Система должна быть проста для понимания пользователя.

В данной работе было реализовано мобильное приложение для операционной системы Android 5.1 и выше, которое имеет графический интерфейс.

Пояснительная записка к курсовой работе выполнена в текстовом редакторе Microsoft Word 2016.

**The abstract**

Coursework, 55 pages, 23 pictures, 7 tables, 6 sources, 2 applications.

VISUAL STUDIO, C # UNITY 3D, BOUNCING CUBE, ENTERTAINMENT GAME.

The object of development is an automated system in the programming language C# in the development environment of Visual Studio software.

The goal of the work is to develop an automated system on the Unity platform, to improve programming skills in C #. The system should be simple to understand the user.

In this paper, we implemented a mobile application for the Android 5.1 operating system and above, which has a graphical interface.

Explanatory note to the term paper is made in the text editor Microsoft Word 2016.

**Роли участников**

Роли участников распределены следующим образом:

Разработчик – Н.А.Телегин, студент группы 715;

Тестер – А.В.Сотпа, студент группы 715;

Документатор – З.Ю.Халметова, студент группы 715.

У участников, создающих систему, следующие роли:

* разработчик, создающий программное обеспечение согласно выбранному алгоритму;
* тестировщик, проверяющий программу на наличие ошибок, при этом определяя каждым тестом набор исходных данных и условий для запуска программы, набор ожидаемых результатов работы программы. То есть тестирование обеспечивает обнаружение ошибок и демонстрацию соответствия функций программы ее назначению.
* документатор, разрабатывающий все необходимые документы.

За время работы разработчик написал код.

Тестировщик создал тесты для программы и проверил ее работоспособность.

Документатор подготовил отчет о работе, включающий реферат, роли участников и краткое описание выполненной работы каждым участником, техническое задание, оценку рисков, результаты тестирования, список литературы. Также составлено руководства пользователя.

Таблица 1 - таблица по выполнению работ по созданию системы:

|  |  |  |  |
| --- | --- | --- | --- |
| Название задачи | Начало | Окончание | Участники |
| Разработка технического задания | 02.10.17 | 08.11.17 | 1,2,3 |
| Анализ рисков | 15.10.17 | 08.11.17 | 1,2,3 |
| Разработка программных модулей | 08.11.17 | 02.11.17 | 1 |
| Тестирование | 15.11.17 | 01.12.17 | 2 |
| Разработка проектной документации | 3.12.17 | 18.12.17 | 3 |
| Защита курсовой работы | 20.12.17 | 28.12.17 | 1,2,3 |

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС)

УТВЕРЖДАЮ

Зав. кафедры КИБЭВС

А.А. Шелупанов

\_\_\_\_\_\_\_\_\_\_\_\_\_

Автоматизированная система «BOUNCING CUBE»

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

На 17 листах

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Руководитель  Аспирант кафедры КИБЭВС  И.Ю.Поляков  «\_\_\_» \_\_\_\_\_ 2017 г. | РАЗРАБОТЧИК  Студент гр. 715  Н.А.Телегин  ДОКУМЕНТАТОР  Студент гр. 715  З.Ю.Халметова  ТЕСТИРОВЩИК  Студент гр. 715  А.В.Сотпа |

Томск 2017

**1 Общие сведения**

* 1. Полное наименование системы и ее условное обозначение

Полное наименование системы: автоматизированная система "BOUNCING CUBE ".

* 1. Заказчик

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР). Кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС).

* 1. Исполнитель

Студенты группы №715: Халметова З.Ю., Телегин Н.А., Сотпа А.В.

* 1. Основания разработки

Основанием для исполнения работ по созданию автоматизированной системы, предусмотренной в настоящем ТЗ, является задание на курсовую работу по дисциплине «Технологии и методы программирования».

* 1. Сведения об источниках и порядке финансирования работ

Работы по разработке курсовой работы финансировались из личных средств студентов, выполняющих данную работу.

* 1. Порядок оформления и предъявления заказчику результатов работ

Предъявление заказчику результатов работ преподносились по окончанию всех работ.

1. **Назначение и цель создания системы**

2.1 Назначение системы

Система «BOUNCING CUBE» должна выполнять развлекательные функции для человека, а также развивать навыки необходимые для прохождения игры, такие как, быстрая и правильная реакция на происходящие события, умение искать закономерности. Система предназначена для игры одного пользователя в соответствии с её описанием (см. Приложение А).

2.2 Цели создания системы

Цель создания автоматизированной системы "BOUNCING CUBE" является изучение платформы разработки игр Unity, усовершенствование навыков программирования на языке C#. Система должна быть проста для понимания пользователя.

1. **Характеристика объектов автоматизации**

Объектом автоматизации является набор игровых процессов. Данная система применяется для развлечения пользователей, совершенствования их координации движения и логического мышления.

1. **Требования к системе**

4.1 Требования к системе в целом

4.1.1 Требования к структуре и функционированию

В системе «BOUNCING CUBE» предлагается выделить следующие функциональные подсистемы:

* пользовательский интерфейс - система должна иметь интерфейс для работы пользователя с системой, предусматривающей подсистемы ввода–вывода данных.
* логический смысл (в данной подсистеме рассматривается принцип построения);
* подсистема настройки и конфигурирования игры - предназначена для корректной работы и настройки игры;
* подсистема меню - предназначена для гибкого пользования системы пользователем, обязательно наличие главного меню с минимальным количеством пунктов, а именно: главное меню, таблица рекордов, новая игра, сохранения, выход из игры, магазин;
* подсистема графического интерфейса - подсистема предназначена для осуществления удобного взаимодействия пользователя с приложением;
* подсистема звукового сопровождения – предназначена для создания приятной атмосферы игры;
* подсистема математических операций - подсистема поддерживает процесс вычисления текущего результата игрока. Сравнение полученного результата по таблице рекордов т.е. сравнение между 3-мя лучшими результатами.
* подсистема ввода/вывода - подсистема предназначена для выполнения процессов ввода сигналов с клавиатуры и вывода информации на экран (результат, окно об окончании).

Система должна поддерживать возможность сохранения игрового процесса, через поддержку работы с файловой системой. Вывод данных должен производиться посредством графического интерфейса, реализованного с помощью языка С#.

4.1.1.1 Требования к способам и средствам связи для информационного обмена между компонентами системы

Требования не оговариваются.

#### 4.1.1.2 Требования к характеристикам взаимосвязей системы со смежными системами

Требования не оговариваются.

#### Требования к режимам функционирования

Разрабатываемая информационная система предназначена для работы в штатном режиме в котором подсистемы «BOUNCING CUBE» должны обеспечивать выполнение своих функций.

Диагностирование системы должно осуществляться путем проверки работоспособности основных функций.

* + - 1. Требования по диагностированию системы

Требования не оговариваются.

#### Перспективы развития, модернизации системы

Система разработана с учетом обеспечения ее дальнейшего развития и наращивания функциональности. Также будет разработан план дополнительного функционала. Также рассматриваются аспекты, позволяющие в дальнейшем осуществлять ее развитие.

### Требования к численности и квалификации персонала системы

### режиму его работы

В состав персонала, необходимого для обеспечения эксплуатации автоматизированной системы, необходимо выделить лицо, именуемое пользователем, которым является один человек, умеющий включать компьютер, пользоваться мышкой и клавиатурой, запускать программы.

#### Пользователи системы

В разрабатываемой информационной системе предусмотрена роль внешнего пользователя – пользователя системы.

* + - 1. Технический персонал системы

Требования не оговариваются.

### Требования к надежности

Надежное функционирование программы обеспечивается выполнением следующих мероприятий:

* организация бесперебойного питаниятехнических средств;
* работа пользователя без предоставления ему административных привилегий;
* наличие свободных аппаратных мощностей и памяти для функционирования;
* время восстановления, не должно превышать 2-х минут;
* постоянной проверкой программных средств на наличие компьютерных вирусов и других угроз.

При возникновении сбоев в аппаратном обеспечении, включая разряд аккумулятора устройства, информационная система восстанавливает свою работоспособность после устранения сбоев и корректного перезапуска аппаратного обеспечения (за исключением случаев повреждения рабочих носителей информации с исполняемым программным кодом).

Система обеспечивает корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях пользователю выдаются соответствующие аварийные сообщения, после чего возвращаются в рабочее состояние, предшествовавшее неверной (недопустимой) команде, группы команд или некорректному вводу данных.

### Требования безопасности

Все технические решения, использованные при создании системы, а также при определении требований к аппаратному обеспечению, соответствуют действующим нормам и правилам техники безопасности, пожарной безопасности и взрывобезопасности, а также охраны окружающей среды при эксплуатации.

### Требования к эргономике и технической эстетике

Интерфейс рассчитан на преимущественное использование монитора, т.е. управление системой осуществляется с помощью клавиатуры, мышки, кнопок, значков и т.п. элементов.

### Требования к эксплуатации, техническому обслуживанию,ремонту и хранению компонентов системы

Условия эксплуатации данного продукта требуют температурных и других ограничений состояния окружающей среды, позволяющих техническому обеспечению работать без сбоев и иных нарушений стабильности оборудования. Пользователь для корректного использования программы должен проверить соответствие параметров технических средств и требования к параметрам технических средств данной программы.

### Требования к защите информации от несанкционированногодоступа

Требования не оговариваются.

### Требования по сохранности информации при авариях

Программное обеспечение информационной системы не восстанавливает свое функционирование после аварии при корректном перезапуске системы, пользователь должен заново запустить игру. Текущий сеанс игры не сохранится.

### Требования к защите от влияния внешних воздействий

Применительно к программно-аппаратному окружению системы предъявляются следующие требования к защите от влияния внешних воздействий.

Требования к радиоэлектронной защите:

* электромагнитное излучение радиодиапазона, возникающее при работе электробытовых приборов, электрических машин и установок, приёмопередающих устройств, эксплуатируемых на месте размещения системы, не должны приводить к нарушениям работоспособности подсистем.

Требования по стойкости, устойчивости и прочности к внешним воздействиям:

* система должна иметь возможность функционирования при колебаниях напряжения электропитания в пределах от 155 до 265В;
* система должна иметь возможность функционирования в диапазоне допустимых температур окружающей среды, установленных изготовителем аппаратных средств;
* система должна иметь возможность функционирования в диапазоне допустимых значений влажности окружающей среды, установленных изготовителем аппаратных средств.

## Требования к функциональности системы

### Функции пользователей системы

Роль разработчика заключается в реализации и пополнении системы новыми функциями, а также исправление возможных ошибок в функционировании системы.

Пользователь может использовать все функции, которыми обладает система в рамках автоматизированной системы.

### Описание процессов и функций работы с системой

Процессы и функции, выполняемые при эксплуатации системы, приведены в разбивке по подсистемам: пользовательский интерфейс, подсистема настройки и конфигурирования игры, подсистема звукового сопровождения, подсистема меню, подсистема ознакомления игрой и справочная система, логический смысл, подсистема ввода/вывода, подсистема математических операций, подсистема графического интерфейса.

Процессы, выполняемые под управлением различных подсистем, реализуются на основе системных процедур, которые являются составной частью функций пользователей системы. Системные процедуры группируются в соответствии с их назначением:

* открытие приложения;
* запуск игрового процесса;
* выбор пункта меню (звук, таблица рекордов, настройки и конфигурация, выход из игры, магазин);
* игровой процесс;
* редактирование данных (перезапись рекордов);
* сохранение рекордов;
* завершение программы;

В каждой подсистеме отказ происходит, если не выполняется хотя бы одна из задач функций. Проблема решается путем перезапуска программы.

## Требования к видам обеспечения

### Требования к математическому обеспечению

Математическое обеспечение системы обеспечивает: правильный подсчет рекордов.

Подсистема поддерживает процесс вычисления текущего результата игрока. Сравнение полученного результата по таблице рекордов т.е. сравнение между 3-мя лучшими результатами.

### Требования к информационному обеспечению

Информационное обеспечение функционирования соответствует требованиям точности, непротиворечивости и актуальности.

### Требования по применению в системе языков высокого уровня

### В качестве средств разработки системы используются интегрированные продукты, которые:

* позволяют вести коллективную разработку и сопровождение системы;
* имеют функциональные возможности сбора, хранения, обработки, доступа и актуализации информации;
* позволяют использовать пиктограммы, мнемонические обозначения и меню;
* обеспечивают масштабируемость и гибкость системы;
* сокращают время разработки приложений;
* удовлетворяют по стоимости программного продукта.

Для манипулирования данными должны использоваться средства языков высокого уровня.

### Требования к программному обеспечению

Системные программные средства, используемые системой, должны быть представлены версией операционной системы Android 5.1 Lollipop и выше.

### Требования к техническому обеспечению

Компьютер, на котором будет установлена автоматизированная система должен обладать следующей минимальной конфигурацией:

# телефон с диагональю не менее 5.0 дюймов;

* процессор с минимальной тактовой частотой 1.3 ГГц;
* свободной памяти не менее 80 Мб.

### Требования к метрологическому обеспечению

Не предъявляются.

### Требования к организационному обеспечению

Не предъявляются.

### 4.3.8 Требования к методическому обеспечению

Не предъявляются.

1. **Состав и содержание работ по созданию системы**

Стадии работ над проектом выполняются в соответствии с ГОСТ 34 и перечислены в таблице 5.1.

Таблица 5.1 - Стадии и этапы работ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание этапа и стадии | Срок | Форма отчетности | Итоги |
| Составление технического задания | 10.10-18.10 | ТЗ | Согласованное техническое задание (ТЗ) |
| Оценка рисков | 30.10.17-30.10.17 | ТЗ | Составлена оценки рисков |
| Составление системы взаимодействия | 15.11.17-30.11.17 | ПЗ | Составленная система взаимодействия |
| Выработка алгоритма | 1.12.17 - 10.12.17 | ПЗ | Составленный алгоритм |
| Работа с графикой | 10.12.17 - 15.12.17 | ПЗ | Графический интерфейс для данной программы |
| Создание программы | 15.12.17 - 17.12.17 | ПЗ | Код для работы программы и взаимодействия пользователя с интерфейсом |
| Тестирование программы | 17.12.17- 19.12.17 | Тестирование системы на наличие ошибок | Отчет о тестировании |

Продолжение таблицы 5.1 – Стадии и этапы работ

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание этапа и стадии | Срок | Форма отчетности | Итоги |
| Защита работы | 20.12.17 - 28.12.17 | ТЗ, ПЗ | Защищенная работа |
|  |  |  |  |
|  |  |  |  |

1. **Порядок контроля и приемки системы**

6.1 Виды, состав, объемы методы испытаний системы и ее составных частей

6.1.1 Перечень этапов испытаний и проверок

# Приемка программы организуется и проводится преподавателями кафедры КИБЭВС.

* Во время приемки программы проверяется выполнение всех функций программы;
* программа должна быть полностью работоспособна;
* испытания проводятся путем всестороннего использования программы;
* тестирование работоспособности кнопки и своевременной реакции программы;
* программа должна распознавать нажатия определенных кнопок на экране и своевременно реагировать на них должным образом;
* тестирование процесса пользования программой.

Проверка на корректность отображения данных.

6.2 Предварительные испытания

# Запуск программы;

# обработка входных данных;

# выдача результата;

# сохранение результата в файл.

Эти испытания проводятся разработчиком на этапе тестирования программы.

6.3 Приемочные испытания

# Запуск программы;

# проверка работоспособности программы, путем набора входных данных;

# сохранение результата в файл;

# вывод результата на экран устройства;

# обнуление результатов.

Приемочные испытания проводятся во время сдачи проекта разработчиком совместно с заказчиком. Все подсистемы испытываются одновременно на корректность взаимодействия подсистем, влияние подсистем друг на друга, т.е. испытания проводятся комплексно.

6.4 Общие требования к приемке работ по стадиям

Сдача-приемка осуществляется представителями Заказчика и Исполнителя.

Все создаваемые в рамках настоящей работы программные изделия передаются Заказчику, как в виде готовых модулей, так и в виде исходных кодов, представляемых в электронной форме на стандартном машинном носителе (на компакт-диске).

6.5 Статус приемочной комиссии (государственная, межведомственная, ведомственная)

Статус приемочной комиссии определяется заказчиком до проведения испытаний.

**7 Требования к документированию**

Список необходимой документации к данному программному проекту:

# лист утверждения (ГОСТ 19.104-78);

# техническое задание (ГОСТ 19.201-78.);

# пояснительная записка (ГОСТ 19. 106-78.);

# руководство пользователя (ГОСТ 19. 106-78.);

# схемы алгоритмов, программ, данных и систем (ГОСТ 19.701-90);

Вся документация оформляется согласно ОС ТУСУР 01-2013. Содержание технического задания соответствует ГОСТ 34.602-89 «Комплекс стандартов на автоматизированные системы. Техническое задание на автоматизированные системы». Руководство пользователя соответствует РД 50-34.698-90 «Автоматизированные системы требования к содержанию документов». Пояснительная записка оформляется согласно ГОСТ 19.404-79 «Пояснительная записка».

7.1 Общие требования к приёмке работы

Работоспособность программы и разработанная программная документация проверяется комиссией, состоящей из преподавателей кафедры КИБЭВС ТУСУРа.

Проверка программы и программной документации комиссией должна быть проведена не позднее 30 декабря 2017 года.

1. **Источники разработки**

Источником разработки программы является:

* UNITY documentation.

СОСТАВИЛИ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование организации, предприятия | Должность исполнителя | Фамилия, имя, отчество | Подпись | Дата |
| ТУСУР | студент | Халметова З.Ю. |  |  |
| ТУСУР | студент | Телегин Н.А. |  |  |
| ТУСУР | студент | Сотпа А.В. |  |  |

СОГЛАСОВАНО

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование организации, предприятия | Должность исполнителя | Фамилия, имя, отчество | Подпись | Дата |
| ТУСУР | Аспирант каф. КИБЭВС | Поляков И.Ю. |  |  |

Оглавление

[1 Введение](#_Toc501541429) 24

[2 Цель системы 24](#_Toc501541430)

[3 Технический проект 25](#_Toc501541431)

[3.1 Структура системы](#_Toc501541432) 27

[3.2 Управляющие связи](#_Toc501541433) 27

[3.3 UML диаграмма подсистем](#_Toc501541434) 27

[4 Оценка рисков](#_Toc501541436) 27

[4.1 Выявление рисков 28](#_Toc501541437)

[4.2 Снижение влияния рисков **.**](#_Toc501541438)28

[5 Результаты тестирования](#_Toc501541439) 29

[5.1 Тестирование компонентов **.**](#_Toc501541440)32

[5.2 Функциональное тестирование **.**](#_Toc501541441)33

[5.3 Анализ производительности **.**](#_Toc501541443)44

[6 Заключение **.**](#_Toc501541444)47

[Список использованных источников **.**](#_Toc501541445)48

[Приложение А (справочное) **.**](#_Toc501541446)49

[Приложение Б(обязательное) **.**](#_Toc501541447)54

[Приложение В](#_Toc501541447) 55

1 Введение

Целью является развлечение, получение удовольствия.  
Задачей является создание оптимальных условий труда, создающих у человека чувство удовлетворения от проделанной работы. При работе с программным обеспечением чувство удовлетворения возникает, если пользователь успешно и без лишних усилий выполняет с помощью программы ту или иную задачу. Компьютерные игры являются весьма специфическим видом ПО, в котором удовлетворение пользователя обусловлено целым рядом специфических факторов. [1]

Компьютерные игры ориентируются на развитие у игроков определенных знаний, навыков, способностей. Как правило, в компьютерных играх от игрока требуется:

* владение средствами управления, быстрота и точность манипуляций;
* быстрая и правильная реакция на происходящие события;
  + чувство времени, умение выдерживать заданные временные интервалы;
  + способность следить за несколькими объектами одновременно;
  + знание географии игрового поля, законов игрового мира;
  + знание конкретной предметной области, которая моделируется в игре;
  + умение искать закономерности. [2]

Автоматизированная система была реализована на языке С#.

2 Оценка рисков

2.1 Выявление рисков

Таблица 2.1.1 – Выявление рисков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Идентификатор | Источник | Условия возникновения | Вероятность, % | Последствия | Влияние | Связанные риски |
| Неверно составленное ТЗ | Подготовительный этап | Неправильно установленные цели заказчика | 24,8 | 3 | 74,4 | 2 |
| Ошибки при разработке | Реализация | Ошибки в разработке алгоритма программы | 39,6 | 4 | 158,4 | 3, 6 |
| Разработка неправильного графического интерфейса | Реализация | Однопоточность | 18,6 | 2 | 37,2 | 2 |
| Ошибки при тестировании | Тестирование | Отсутствие тестов на значимых функциях системы | 34,4 | 4 | 137,6 | 5 |
| Неисправленные ошибки в программе | Тестирование | Недоработанная программа на основе тестов, проведенных тестировщиком | 47,6 | 4 | 190,4 | 4 |

Продолжение таблицы 2.1.1 – Выявление рисков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Идентификатор | Источник | Условия возникновения | Вероятность, % | Последствия | Влияние | Связанные риски |
| Задержки сроков выполнения работ | Защита | Недоработанные реализация и тестирование, недоделанная пояснительная записка | 20,2 | 3 | 60,6 | 3, 4, 5 |
| Аппаратные сбои | Защита | Отключение электричества, сбои служб, не входящих в разрабатываемую систему | 20 | 2 | 40 | - |

Была введена шкала оценки последствий рисков:

1 – Риск не представляет угрозы для проекта, можно пренебречь. Риск может привести к частным ошибкам.

2 – Риск представляет угрозу для проекта, но последствия могут быть устранены после ввода системы в эксплуатацию без использования дополнительных ресурсов.

3 – Риск представляет угрозу для проекта, но последствия могут быть устранены после ввода системы в эксплуатацию, для этого потребуются дополнительные ресурсы.

4 – Риск представляет угрозу для проекта, последствия риска могут быть устранены только до ввода системы в эксплуатацию.

5 – Риск представляет угрозу для системы в целом, последствия риска не могут быть устранены.

В таблице 2.1.2 производится оценка последствий риска

Таблица 2.1.2 – Оценка последствий риска [3]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № эксперта  № риска | Эксперт №1 | Эксперт №2 | Эксперт №3 | Эксперт №4 | Эксперт №5 | Усредненная оценка |
| Риск №1 | 3 | 3 | 4 | 3 | 3 | 3,2≈3 |
| Риск №2 | 4 | 3 | 4 | 4 | 3 | 3,6≈4 |
| Риск №3 | 2 | 1 | 2 | 3 | 2 | 2 |
| Риск №4 | 4 | 4 | 3 | 3 | 4 | 3,6≈4 |
| Риск №5 | 5 | 4 | 4 | 4 | 4 | 4,2≈4 |
| Риск №6 | 3 | 4 | 3 | 2 | 3 | 3 |
| Риск №7 | 2 | 3 | 2 | 1 | 2 | 2 |

В таблице 2.1.3 производится оценка вероятности риска

Таблица 2.1.3 – Оценка вероятности риска

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № эксперта  № риска | Эксперт №1 | Эксперт №2 | Эксперт №3 | Эксперт №4 | Эксперт №5 | Усредненная оценка |
| Риск №1 | 0,35 | 0,2 | 0,21 | 0,18 | 0,3 | 0,248 |

Продолжение таблицы 2.1.3 – Оценка вероятности риска [3]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № эксперта  № риска | Эксперт №1 | Эксперт №2 | Эксперт №3 | Эксперт №4 | Эксперт №5 | Усредненная оценка |
| Риск №2 | 0,3 | 0,36 | 0,42 | 0,5 | 0,4 | 0,396 |
| Риск №3 | 0,2 | 0,14 | 0,1 | 0,19 | 0,3 | 0,186 |
| Риск №4 | 0,26 | 0,55 | 0,3 | 0,23 | 0,38 | 0,344 |
| Риск №5 | 0,4 | 0,6 | 0,5 | 0,48 | 0,4 | 0,476 |
| Риск №6 | 0,1 | 0,2 | 0,23 | 0,18 | 0,3 | 0,202 |
| Риск №7 | 0,18 | 0,2 | 0,2 | 0,3 | 0,12 | 0,2 |

Таким образом, следующие риски требуют вмешательства: 1, 2, 4, 5, 6.

2.2 Меры по снижению возможности реализации рисков

2.2.1 Риск «Неверно составленное ТЗ»

Для того чтобы снизить вероятность возникновения риска необходимо:

* написать техническое задание в соответствии с предъявляемыми требованиями и темой работы;
* вносить правки в техническое задание до полного соответствия требованиям и выбранной темой;

2.2.2 Риск «Разработка неправильного графического интерфейса»

Для того чтобы снизить вероятность возникновения риска нужно:

* удостовериться в том, чтобы разработчик программировал на том языке программирования, с которым хорошо ознакомлен;
* разобраться с предметной областью и материалами, связанными с темой курсовой работы;
* уметь работать со сторонними библиотеками;
* иметь возможность обратиться к более компетентным программистам с возникшими вопросами;
* знать английский язык.

2.2.3 Риск «Ошибки при тестировании»

Чтобы снизить вероятность возникновения риска необходимо:

* возможность помочь тестеру разобраться с программным кодом разработчиком;
* своевременно уведомлять разработчика о выявленных ошибках;
* отслеживать своевременное исправление выявленных ошибок.

2.2.4 Риск «Неисправленные ошибки в программе»

Чтобы снизить вероятность возникновения риска необходимо:

* выполнение программистом отладки кода;
* покрыть все важные функции тестами;
* проделать повторные тесты после внесенных исправлений;

2.2.5 Риск «Задержки сроков выполнения работ»

Чтобы снизить вероятность возникновения риска необходимо:

* вовремя исправлять найденные ошибки разработчиком;
* вовремя делать тесты;
* вовремя оформлять документацию.

Для расчета коэффициента конкордации использовалась следующая формула:

|  |  |
| --- | --- |
| ,  где –количество критериев,  – количество экспертов. | (1) |
| , | (2) |

* 1. Способы снижения вероятности возникновения риска

Чтобы снизить вероятность возникновения риска «выход за рамки графика», можно применить следующие рекомендации:

* организовать еженедельные собрания сотрудников для контроля результатов работы;
* хранить данные на нескольких носителях;
* увеличить количество тестов;
* решить вопросы и неточности с заказчиком;
* увеличить штат сотрудников, либо иметь менее узкоспециализированный персонал;
* провести анкетирование сотрудников.

Чтобы снизить вероятность возникновения риска «противоречивые требования заказчика», можно применить следующее:

* сделать план диалога с заказчиком;
* решить вопросы и неточности с заказчиком.

Чтобы снизить вероятность возникновения риска «болезнь персонала», можно применить следующее:

* проводить меры по повышению иммунитета сотрудников (вакцинация);
* установить нормированный рабочий день;
* принимать менее узкоспециализированных сотрудников;
* увеличение количества сотрудников.

Такие действия помогут уменьшить размеры потерь, снизить вероятности риска и изменить его последствия.

**3-Описание алгоритма**

* 1. Вербальное описание

В ходе выполнения курсовой работы было разработано приложение для развлечения в досуговое время. Предусмотрен графический интерфейс пользователя, обеспечивающий удобное манипулирование приложением пользователю.

Класс CubeJump реализует игровой процесс данной игры. В нем реализованы методы, позволяющие определять положение игрового куба в пространстве, определять клики игрока по игровому полю и их длительность, вести счет игры, передавать команды для создания новых платформ и удаления старых платформ.

В классе SpawnBlocks реализовано создание и удаление игровых платформ, основного игрового куба.

Класс Score позволяет вести подсчет очков, набранных игроком.

Данный Класс - GameArrangment отвечает за визуализацию состояний игры. То есть, в процессе игры будут отображаться счет, максимальный рекорд, при этом не будут отображаться кнопки главного меню, названия игры.

В классе Buttons реализованы методы, позволяющие запускать отдельные части игры в зависимости от нажатия на кнопки.

Класс ShopBG отвечает за визуализацию и управление игровым магазином.

Класс ScrallObjects позволяет перемещать объекты в пространстве на заданные координаты. [6]

* 1. Диаграмма классов

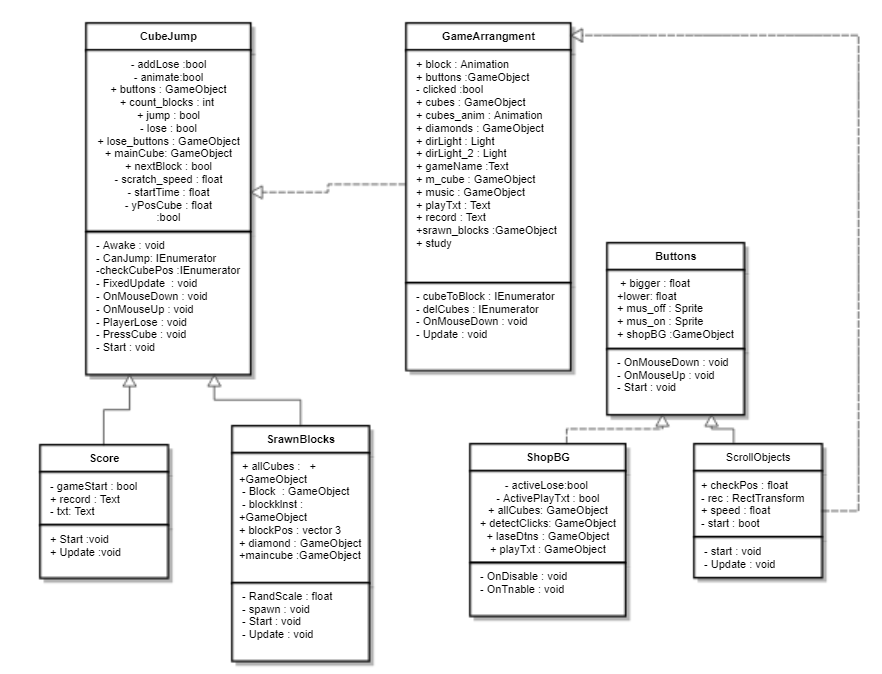


Рисунок 3 - Диаграмма классов [6]

* 1. Блок схемы системы

В приложении Б приведены блок-схемы для общего взаимодействия пользователя с приложением.

Блок схемы системы приведены в приложении Б.

1. Результаты тестирования

Существуют два принципа тестирования программы:

* функциональное тестирование (тестирование «черного ящика»);
* структурное тестирование (тестирование «белого ящика»).

При тестировании «черного ящика» исследуется работа каждой функции на всей области определения, рассматриваются системные характеристики программ, игнорируется их внутренняя логическая структура.

Тестирование «белого ящика» основано на анализе управляющей структуры программы. Объектом тестирования является внутреннее поведение программы. Программа считается полностью проверенной, если проведено исчерпывающее тестирование маршрутов (путей) ее графа управления. В этом случае формируются тестовые варианты, в которых:

* гарантируется проверка всех независимых маршрутов программы;
* проходятся все ветви True, False для всех логических решений;
* выполняются все циклы (в пределах границ их диапазонов);
* анализируется правильность внутренних структур данных.

Исчерпывающее тестирование – тестирование, проверяющее все наборы исходных данных, все варианты их обработки. Оно включает большое количество тестовых вариантов.

В обоих случаях исчерпывающее тестирование затруднительно. При этом даже исчерпывающее тестирование маршрутов при выборе принципа «белого ящика» не гарантирует соответствия программы исходным требованиям к ней. Есть вероятность пропуска ошибок. Также в программе могут быть пропущены некоторые маршруты. Однако в принципе «черного ящика» есть несколько преимуществ:

* простота восприятия;
* более быстрое создание теста.

4.1 Обнаружение ошибок

В ходе тестирования программы были обнаружены ошибки в ее работе, после чего программа была отправлена на доработку. Ниже представлена таблица 4.1.1, которая была составлена из обнаруженных ошибок.

Таблица 4.1.1. – Ошибки программы и их исправления

|  |  |
| --- | --- |
| Обнаруженные ошибки | Решение проблемы |
| Программа не фиксировала нажатия по бокам сенсора дисплея смартфона. | Разработка зависимости для поля фиксации нажатия сенсора дисплея от размеров дисплея. |
| В программе после просмотра рекорда, рекорд не убирался, при старте игры. | Были переписаны условия отображения рекорда. |
| В разделе shop были доступны кнопки, расположенные в главном меню, т.е. их могли прожимать. | Кнопки главного меню стали недоступны в разделе shop. |
| Куб улетал с реактивной скоростью за пределы игрового поля, если время нажатия на сенсор дисплея был ничтожно малым. | Был установлен минимальный временный, дефолтный, порог нажатия на сенсор дисплея. |
| Иллюстрация куба в магазине увеличился в размерах, если игрок несколько раз входил и выходил из shop. | Пересмотрены условия возврата размеров кубиков в shop. |
| Ошибка выключения звука игры. | Не решена, поскольку данная ошибка возникает на этапе сборки проекта в.apk файл из-за неполадки компилятора. |

4.2 Проведение Unit – тестов

Для тестирования игры использовались стандартные средства тестирования платформы Unity, а именно Unity test tools, который включает в себя Test Runner, EditMode Test C# Script.

4.2.1 Тест сбора алмазов

Тестировался метод сбора, получения алмаза в игре. Вначале создаем объект класса CollectDiamond, после чего создаем объект - колайдер, который представляет собой область, в которой создается алмаз. Далее вызываем метод нашей созданной игровой сущности, на вход которой подаем созданный колайдер. На выходе ожидаем, что колайдер будет уничтожен, и что счетчик количества алмазов будет равен 1.

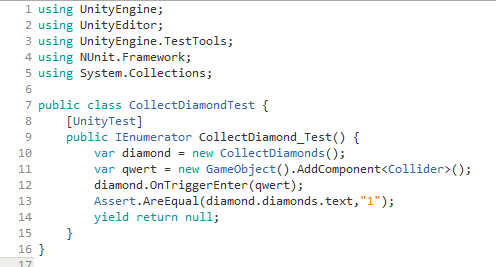


Рисунок 4.2 - Код unit – теста №1

проверка результата.png

Рисунок 4.3 - Результат unit – теста №1

4.2.2 Тест на удаление блока

Тестировался метод удаления блока – платформы на который куб прыгает. По аналогии создается объект класса DeleteBlock, после чего колайдер, в котором создается блоки - платформы. Далее вызываем метод нашей созданной игровой сущности, на вход которой подаем созданный колайдер. На выходе ожидаем, что колайдер будет уничтожен.

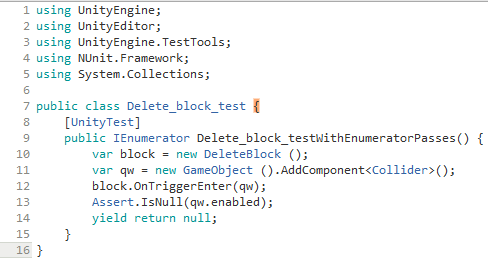


Рисунок 4.4 - Код unit – теста №2

проверка результата теста удаление бдлока.png

Рисунок 4.5 - Результат unit – теста №2

4.2.3 Тест на выдвижение объекта

Тестировался метод для выдвижения объекта. Создаем объект, в который передаем ссылку на необходимый нам класс. Вызываем функцию с входными значениями по позиции равное -100, и по скорости 5f. На выходе ожидаем что позиция нашего созданного объекта будет равно 0 (см. рисунок4.6).

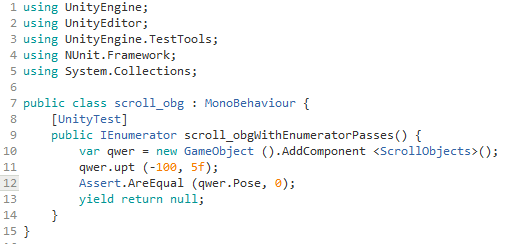


Рисунок 4.6 – Код unit – теста №3

проверка теста 3.png

Рисунок 4.7 – Результат unit – теста №3

4.2.4 Тест проверки счета

Тестировался метод подсчета счета. Создаем объект класса Score, после чего вызываем метод Update. Данный метод не принимает на вход никаких параметров, поэтому на вход его ничего не подаем, но он использует общедоступную переменную, куда записывается количество удачных прыжком кубика с платформы на платформу. На выходе ожидаем что счет изменится и станет равным 1.

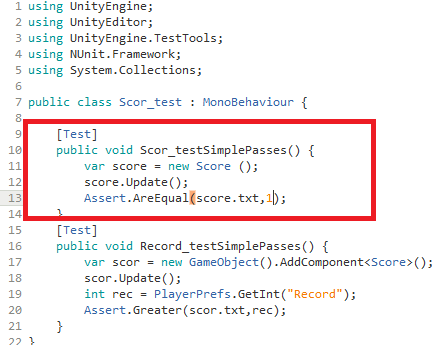


Рисунок 4.8 – Код теста 4



Рисунок 4.8 – Результат теста 4

4.2.5 Тест проверки рекорда

Тестировался метод проверки счета. Создаем объект класса Score, после чего вызываем метод Update. Данный метод не принимает на вход никаких параметров, поэтому на вход его ничего не подаем, но он использует общедоступную переменную, куда записывается количество удачных прыжком кубика с платформы на платформу. Так же объявляем переменную rec целого типа, куда записываем текущее хранимое значение рекорда. На выходе ожидаем что количество очков который набирает игрок будет больше чем текущее количество очков в рекорде.

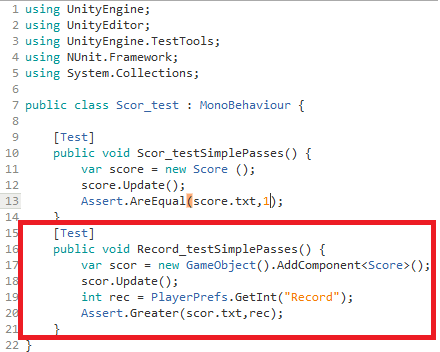


Рисунок 4.9 – Код теста 5



Рисунок 4.10 –Результат теста 5

## **Заключение**

В ходе курсовой работы было написано приложение для развлечения пользователей, совершенствования их координации движения и логического мышления, была произведена ошибка риска и проведён анализ по их устранению ошибок, разработаны блок-схемы по ГОСТ, UML диаграммы, составлен реферат, описан пользовательский интерфейс, составлено руководство пользователя, составлено техническое задание.

Список использованных источников

1 Unity3d. [Электронный ресурс]. – Режим доступа: http://www.unity3d.ru/ (дата обращения: 21.12.2017).

1. ОС-ТУСУР 01-2013 [Электронный ресурс.] - Режим доступа: http://www.tusur.ru/export/sites/ru.tusur.new/ru/education/documents/inside/tech\_01-2013\_new.pdf (дата обращения: 5.10.2017).

# Язык программирования С# / The C Programming Language Керниган, Брайан Уилсон; Ритчи, Деннис Макалистэйр, 2007 год, 256 страниц.

1. Форум программистов и сисадминов Киберфорум. [Электронный ресурс]. – Режим доступа: http://www.cyberforum.ru/ (дата обращения: 20.12.2017).
2. Книга Unity Game Development Essentials. Автор: Ryan Henson Creighton, 2012 год, 209 страниц.

# [UML-проектирование систем реального времени параллельных и распределенных приложений](http://forcoder.ru/uml/uml-proektirovanie-sistem-realnogo-vremeni-parallelnyh-i-raspredelennyh-prilozhenij-1140). <http://forcoder.ru/uml/> [Электронный ресурс] (дата обращения: 18.12.2017).

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС)

К ЗАЩИТЕ ДОПУСТИТЬ

Заведующий кафедры КИБЭВС

А.А.Шелупанов

АВТОМАТИЗИРОВАННАЯ СИСТЕМА «BOUNCING CUBE»

Курсовая работа по дисциплине «Технологии и методы программирования»

Руководство пользователя

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Руководитель  Аспирант кафедры КИБЭВС  И.Ю.Поляков  «\_\_\_» \_\_\_\_\_ 2017 г. | РАЗРАБОТЧИК  Студент гр. 715  Н.А.Телегин  ДОКУМЕНТАТОР  Студент гр. 715  З.Ю.Халметова  ТЕСТИРОВЩИК  Студент гр. 715  А.В.Сотпа |

Томск 2017

Оглавление

1. [Введение 41](#_bookmark27)
   1. [Область применения 41](#_bookmark28)
   2. [Краткое описание возможностей 41](#_bookmark29)
   3. [Уровень подготовки пользователя 41](#_bookmark30)
   4. [Перечень эксплуатационной документации 41](#_bookmark31)
2. [Назначение и условия применения 41](#_bookmark32)
   1. [Виды деятельности, функции 41](#_bookmark33)
   2. [Программные и аппаратные требования к системе 42](#_bookmark34)
3. [Подготовка к работе 42](#_bookmark35)
   1. [Состав дистрибутива 42](#_bookmark36)
   2. [Запуск системы 42](#_bookmark37)
   3. [Проверка работоспособности системы 43](#_bookmark38)
4. [Описание операций 43](#_bookmark39)
   1. [Наименование операций 43](#_bookmark40)
   2. [Условия выполнения операции 43](#_bookmark41)
   3. [Подготовительные действия 43](#_bookmark42)
   4. [Основные действия 44](#_bookmark43)
   5. [Заключительные действия 44](#_bookmark44)
   6. [Ресурсы, расходуемые на операцию 44](#_bookmark45)
5. [Аварийные ситуации. Восстановление приложения 44](#_bookmark46)
6. [Рекомендации по освоению 44](#_bookmark47)

**1 Введение**

* 1. Область применения

Объектом разработки является автоматизированная система на языке программирования C# в среде разработки программного обеспечения Visual Studio.

* 1. Краткое описание возможностей

Процессы и функции, выполняемые при эксплуатации системы, приведены в разбивке по подсистемам: пользовательский интерфейс, подсистема настройки и конфигурирования игры, подсистема звукового сопровождения, подсистема меню, подсистема ознакомления игрой и справочная система, логический смысл, подсистема ввода/вывода, подсистема математических операций, подсистема графического интерфейса.

Процессы, выполняемые под управлением различных подсистем, реализуются на основе системных процедур, которые являются составной частью функций пользователей системы.

* 1. Уровень подготовки пользователя

В состав персонала, необходимого для обеспечения эксплуатации BOUNCING CUBE, необходимо выделить лицо, именуемое пользователем, которым является один человек, умеющий включать компьютер, пользоваться мышкой и клавиатурой, искать и перемещать файлы, запускать программы.

Требования к квалификации и режиму работы персонала не предъявляются.

* 1. Перечень эксплуатационной документации

Перечень эксплуатационных документов, с которыми необходимо ознакомиться:

* «Руководство пользователя».

1. **Назначение и условия применения**

* 1. Программные и аппаратные требования к системе

Системные программные средства, используемые системой, должны быть представлены версией операционной системы Android 5.1 Lollipop и выше.

### Требования к техническому обеспечению

Компьютер, на котором будет установлена автоматизированная система должен обладать следующей минимальной конфигурацией:

# телефон с диагональю не менее 5.0 дюймов;

* процессор с минимальной тактовой частотой 1.3 ГГц;
* свободной памяти не менее 80 Мб.

1. **Подготовка к работе**

Список необходимых файлов программы BOUNCING CUBE:

* BOUNCING CUBE.apk.
  1. Запуск системы

Для того, чтобы запустить АС BOUNCING CUBE необходимо запустить файл BOUNCING CUBE.apk.

* открытие приложения;
* запуск игрового процесса;
* выбор пункта меню (звук, таблица рекордов, настройки и конфигурация, выход из игры, магазин);
* игровой процесс;
* редактирование данных (перезапись рекордов);
* сохранение рекордов;
* завершение программы;

В каждой подсистеме отказ происходит, если не выполняется хотя бы одна из задач функций. Проблема решается путем перезапуска программы.

* 1. Проверка работоспособности системы

Программное обеспечение работоспособно, если в результате действий пользователя, изложенных в подразделе 3.2, на экране монитора отобразилось главное окно клиентской программы, успешно завершились действия по сжатию или распаковке файла, при этом аварийного закрытия программы не произошло.

1. **Описание операций**
   1. Наименование операций

* Запуск игры;
* сохранение в файл;
* завершение игры.
  1. Условия выполнения операции

Программа запущена, успешно функционирует.

* 1. Подготовительные действия

Запуск программы.

* 1. Основные действия

Выполнение операций выбора файла, его сжатия или извлечения файла из него в соответствии с 3.2.

* 1. Заключительные действия

Заключительные действия отсутствуют.

* 1. Ресурсы, расходуемые на операцию

Требуемыми ресурсами для выполнения операции являются описанные в подразделе 2.2.

1. **Аварийные ситуации. Восстановление программы**

При сбое в работе аппаратуры восстановление нормальной работы системы должно производиться после:

* перезагрузки операционной системы;
* запуска исполняемого файла системы BOUNCING CUBE.

При ошибках в работе аппаратных средств (кроме носителей данных и программ) восстановление функции системы возлагается на ОС.

При ошибках, связанных с программным обеспечением, восстановление работоспособности возлагается на ОС.

При неверных действиях пользователей система не выполняет свои функции.

1. Рекомендации по освоению

Для успешного освоения программы АС необходимо иметь начальные навыки работы с ПК и изучить следующее:

* Настоящее «Руководство пользователя».

Приложение А

(Справочное)

Руководство пользователя

Для запуска игры выберите приложение BOUNCING CUBE, кликните один раз по сенсору экрана. Далее пользователь увидит начальное меню игры, изображено на рисунке А.1.



Рисунок А.1- Главное меню

Также пользователю предлагаются такие параметры как настройки, таблица рекордов, выход из игры, а также магазин.

В настройках приведены дополнительные функции такие как: звук, дополнительные расширения google и twitter, изображено на рисунке А.2.



Рисунок А.2 – Меню настройки

В таблице рекордов можно увидеть какое место занимает пользователь по окончанию игры.



Рисунок А.3 – Рекорды игры

В магазине пользователь может выбрать цвет куба, но для этого он должен получить бриллиант во время игры, на что в итоге обменивает куб, на один из бриллиантов. Бриллианты можно зарабатывать в процессе игры, если куб пройдет 4 этапа, то ему зачисляется один бриллиант, в итоге, чем больше он заработает очков, тем больше вариантов он может приобрести в магазине, как показано на рисунке А.4.

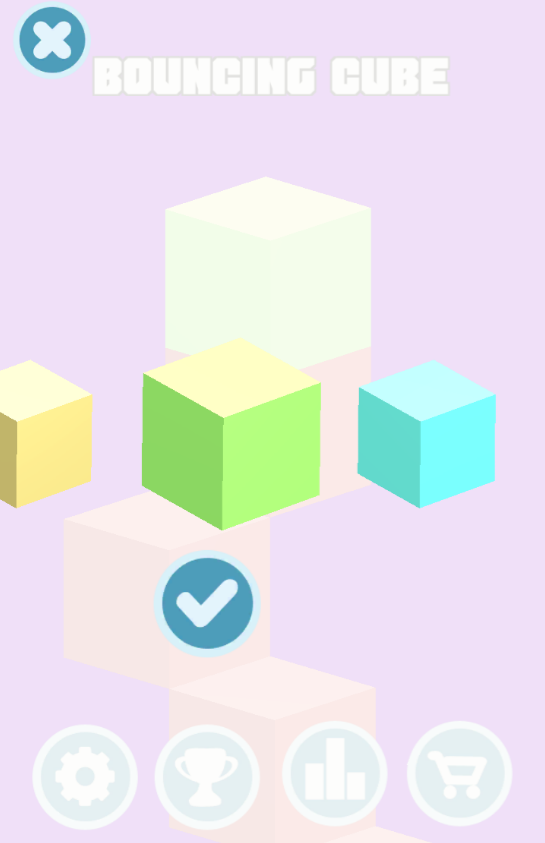


Рисунок А.4 – Выбор цвета куба из магазина

Если пользователь проиграл, то ему предлагается вариант продолжить игру, либо выбрать один из вышеперечисленных параметров, а также выход из игры, как показано на рисунке А.5. В любой момент времени пользователь может покинуть игру по своему желанию, в следствии чего игра будет автоматически закрыта. Сам процесс игры изображен на рисунке А.6. Пользователю предлагается выбирать клик в области экрана, тем самым куб будет менять свое положение.

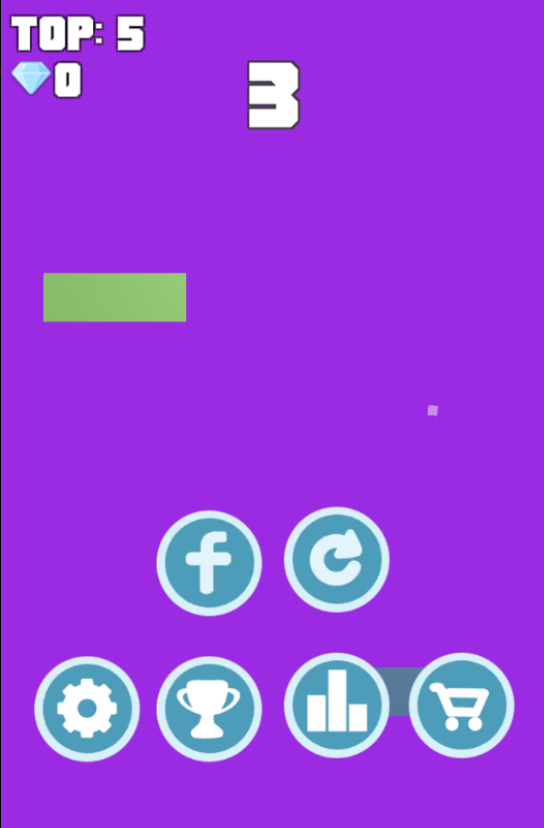


Рисунок А.5 – Конец игры

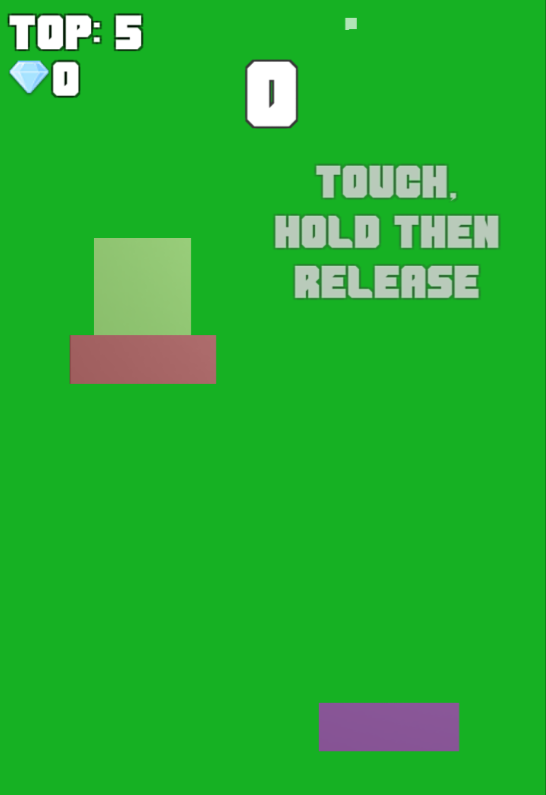


Рисунок А.6 – Процесс игры

Приложение Б

(Обязательное)

Блок-схемы алгоритма

При старте игры мы попадаем в главное меню, которое содержит элементы управления, а именно – кнопки: рекорды, магазин, настройки, выход из игры. Пока мы находимся в главном меню, игра будет считывать любой клик в области экрана. Если координаты клика будут попадать в диапазон координат одной из кнопок, то будет выполнено действие, соответствующее этой кнопке. Если координаты клика не будут попадать в область координат кнопок, то будет запущен игровой процесс. В начале игрового процесса воспроизводится анимация перемещения главного куба на начальную платформу. Далее программа будет считывать клики пользователя по области экрана и запоминать длительность этих кликов. После принятия клика в зависимости от длительности будет совершено перемещение главного куба в игровом пространстве. Если после этого куб остается на текущей платформе, то выбирается один из двух вариантов, либо увеличение счета игры на единицу, либо поражение. Если куб попадает на следующую платформу, то игровой счет увеличивается на единицу.

Если не выполняется ни одно из предыдущих условий, то игроку засчитывается поражение.

На рисунке Б.1 приведена основная блок-схема игры.

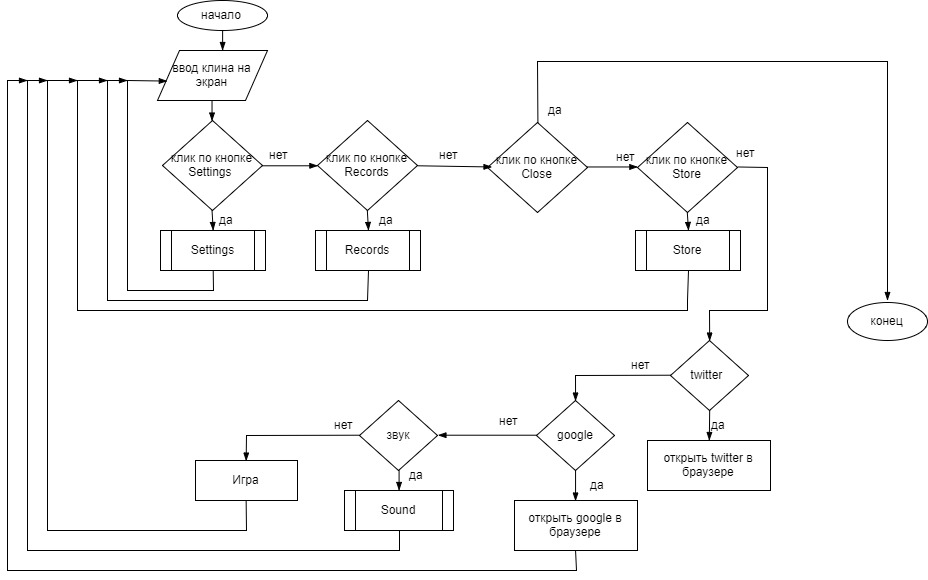


Рисунок Б.1 – Блок-схема алгоритма игры

На рисунке Б.2 показана блок – схема функций Игра и Sound

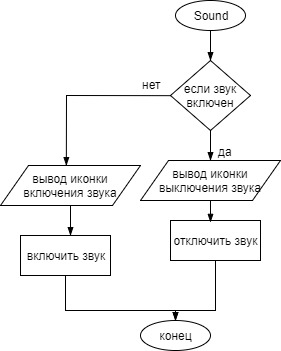
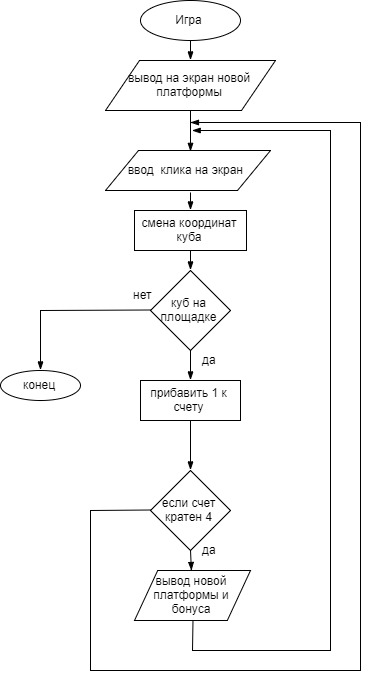


Рисунок Б.2 – Блок –схема функций Игра и Sound

На рисунке Б.3 показана блок – схема функции Store

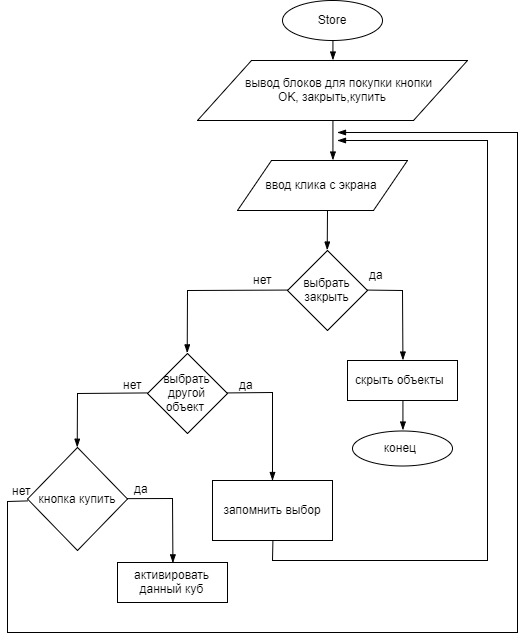


Рисунок Б.3 – Блок –схема функции Store

На рисунке Б.4 показана блок – схема функций Setting и Records

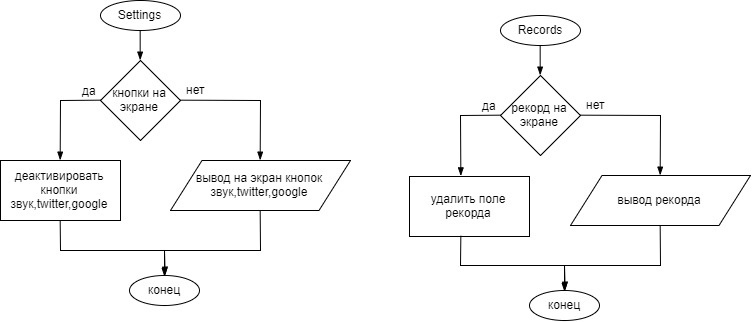


Рисунок Б.4 – Блок –схема функций Settings, Records

Приложение В

Электронный носитель содержит следующие файлы: листинг программы, тесты, пояснительную записку, техническое задание, файл BOUNCING CUBE.apk.