**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**автономное образовательное учреждение высшего образования**

**«Санкт-Петербургский государственный политехнический университет Петра Великого»**

**(ФГАОУ ВО «СПбПУ»)**

**Университетский политехнический колледж**

**КУРСОВАЯ РАБОТА**

Программирование в компьютерных системах

специальность

отделение Информационных Технологий

По дисциплине: Технология разработки программного обеспечения

Тема: Информационная система «Тетрис»

Исполнитель: студент(ка) IV курса группы 42928/1

Маколов Константин Игоревич

(фамилия, имя, отчество)

Руководитель преподаватель УПК

(ученная степень, звание)

Зернова Елена Николаевна

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

(фамилия, имя, отчество)

Курсовая работа допущена к защите

Защитил(а) курсовую работу с оценкой

«\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2019 г.

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

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

«\_\_\_» 2019 г

2019 г

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc5228439)

[1 Постановка задачи 3](#_Toc5228440)

[1.1 Опорные точки зрения 3](#_Toc5228441)

[1.2 Аудирование заинтересованных лиц 4](#_Toc5228442)

[1.3 Анализ полученной информации 5](#_Toc5228443)

[1.4 Диаграмма использования 6](#_Toc5228444)

[2 Построение программы этапа проектирования 7](#_Toc5228445)

[3 Разработка программы 9](#_Toc5228446)

[4 Руководство оператора 10](#_Toc5228447)

[4.1 Модуль пользователя 10](#_Toc5228448)

[5 Методика испытаний 13](#_Toc5228449)

[5.1 Объект испытаний 13](#_Toc5228450)

[5.2 Цель испытаний 13](#_Toc5228451)

[5.3 Требования к программе 13](#_Toc5228452)

[5.4 Требования к программной документации 13](#_Toc5228453)

[6 Средства и порядок испытаний 15](#_Toc5228454)

[6.1 Требования к составу и параметрам технических средств 15](#_Toc5228455)

[6.2 Перечень этапов испытаний и проверок 16](#_Toc5228456)

[6.3 Методы испытаний 17](#_Toc5228457)

[ЗАКЛЮЧЕНИЕ 21](#_Toc5228458)

# ВВЕДЕНИЕ

Тема курсового проекта: «Тетрис»

Программа предназначена для личного использования конечными пользователями.

# Постановка задачи

## Опорные точки зрения

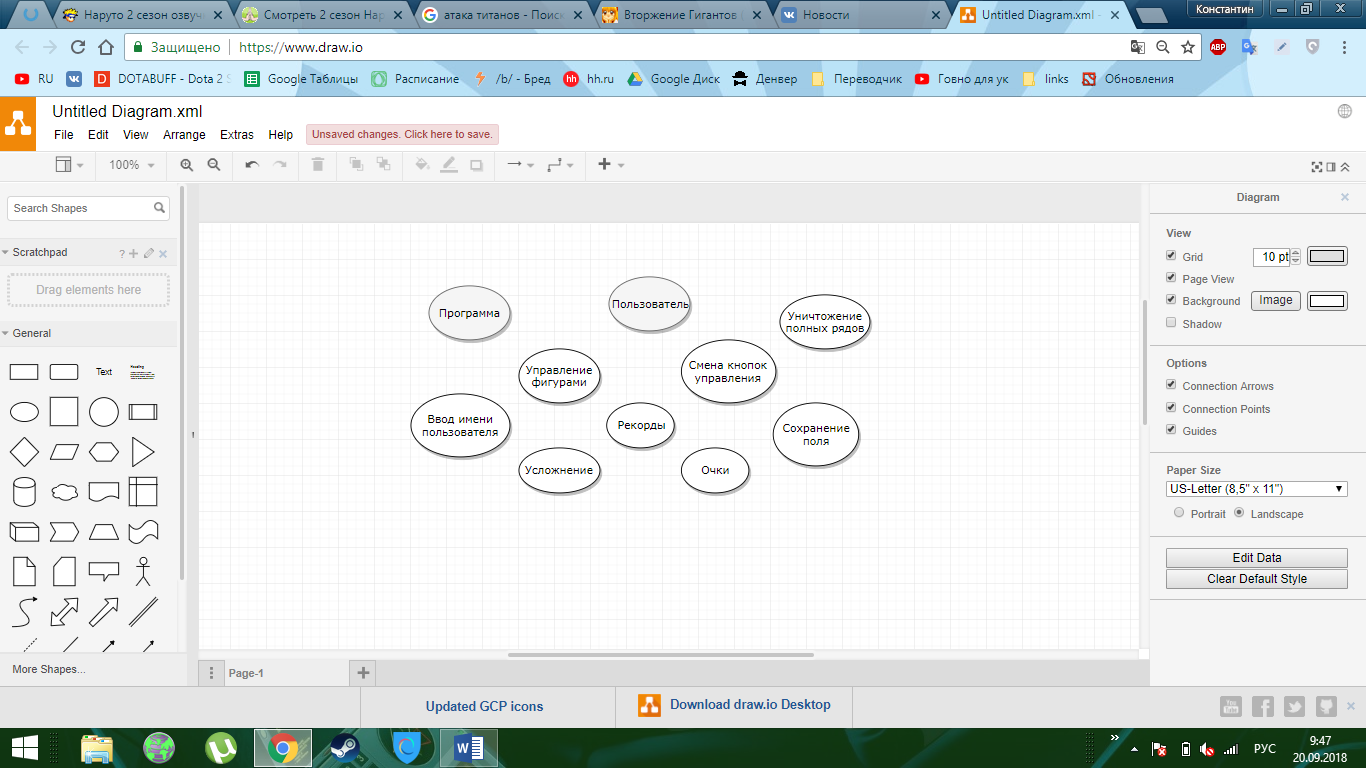


Рисунок 1 - Диаграмма идентификационных точек зрения

Таблица 1- Сервисы, соотнесённые с точками зрения

|  |  |
| --- | --- |
| **Пользователь** | **Программа** |
| Ввод имени | Вывод очков |
| Смена кнопок управления | Усложнение |
| Сохранение поля | Вывод рекордов |
| Управление фигурами | Уничтожение полных рядов |
| Процесс игры |  |

## 

Рисунок 2 - Диаграмма идентификационных точек зрения

## Аудирование заинтересованных лиц

Для решения поставленной задачи были выявлены следующие заинтересованные лица:

* Руководители фирм-разработчиков игр
* Игроки

Таблица 2 – Интервью с руководителем компании по разработке игр.

|  |  |
| --- | --- |
| Вопрос | Ответ |
| 1. Имя | Гейб Ньюэлл |
| 2. Наименование структурного подразделения | компания Valve |
| 3. Должность | генеральный директор компании Valve |
| 4. Каковы Ваши основные обязанности? | Разработка программ для игровой индустрии для различных платформ. Контроль за их реализацией. |
| 5.Хотели бы вы видеть в игре возможность добавления пользовательских блоков? | Нет, это излишне |
| 6. Какую информацию можно считать входящей, или необходимой, по вашему мнению для компьютерной игры «Тетрис»? | 1.Имя пользователя 2.Начальный уровень сложности |
| 7. Какую информацию можно считать исходящей, по вашему мнению для компьютерной игры «Тетрис»? | 1.Таблица рекордов 2.Количестов очков и рядов, набранных в данной игре |
| 8. Какой интерес или какие потребности у Вас есть относительно будущего решения (разрабатываемого ПО)? | Разрабатываемая игра должна предоставлять возможность сохранения игрового поля возможность назначить клавиши управления, а также должна периодически усложнятся (например, при достижении 100 очков уровень становится чуть сложнее) |
|  |  |
|  |  |

Таблица 3 – Интервью с игроком

|  |  |
| --- | --- |
| Вопрос | Ответ |
| 1. Имя | Иван Рудской |
| 2. Чем вы занимаетесь? | Видеоблогер, увлекаюсь играми |
| 3.Хотели бы вы видеть в игре возможность добавления пользовательских блоков? | Думаю, что это было бы забавно, но в этом нет необходимости |
| 4. Какую информацию можно считать входящей, или необходимой, по вашему мнению для компьютерной игры «Тетрис»? | 1.Мне хотелось бы, чтобы я мог видеть в таблице рекордов свой никнэйм => было бы хорошо добавить возможность ввода имени 2.Как заядлому геймеру мне было бы слишком просто играть на одной сложности с неопытным игроком=> необходимо выбрать начальную сложность |
| 5. Какую информацию можно считать исходящей, по вашему мнению для компьютерной игры «Тетрис»? | 1.Таблица рекордов 2.Количестов очков и рядов, набранных в данной игре |
| 6. Какой интерес или какие потребности у Вас есть относительно будущего решения (разрабатываемого ПО)? | Будет хорошо, если программа будет давать возможность изменять клавиши управления, а также сохранять игровое поле, чтобы продолжить играть в любое другое удобное время |

## Анализ полученной информации

В результате анкетирования и интервьюирования всех заинтересованных лиц были сформулированы потребности заказчика относительно разрабатываемого ПО:

* При запуске игры пользователю представляются 2 кнопки: «Старт» и «Сложность». При нажатии на кнопку «Сложность» пользователю предлагается выбрать начальную скорость падения блоков. При нажатии на «Старт» начинается игра.
* Процесс игры представляет падение фигур, составленных из 4х блоков, в контейнер размером 10х20 блоков. При составлении полного ряда из 10 блоков-ряд убирается и добавляется 10 очков. Игра завершается, когда «башня» из фигур достигает верхней клетки. После завершения игры пользователю показывается его результат и если этот результат входит в десятку лучших, то пользователю предлагается ввести своё имя и сохранить результат в таблицу рекордов.
* Программный продукт должен быть предоставлять пользователям информацию о количестве набранных очков и рекордах.
* Необходимо реализовать функции сохранения игры и усложнения, т.е. при наборе 100 очков блоки должны начать падать быстрее на 500 мс.
* Программа должна предоставлять пользователям возможность выбора удобных им для игры клавиш управления.

## Диаграмма использования

В ходе рассмотрения данной темы была спроектирована диаграмма вариантов использования (Use Case Diagram), представленная на рисунке 1.

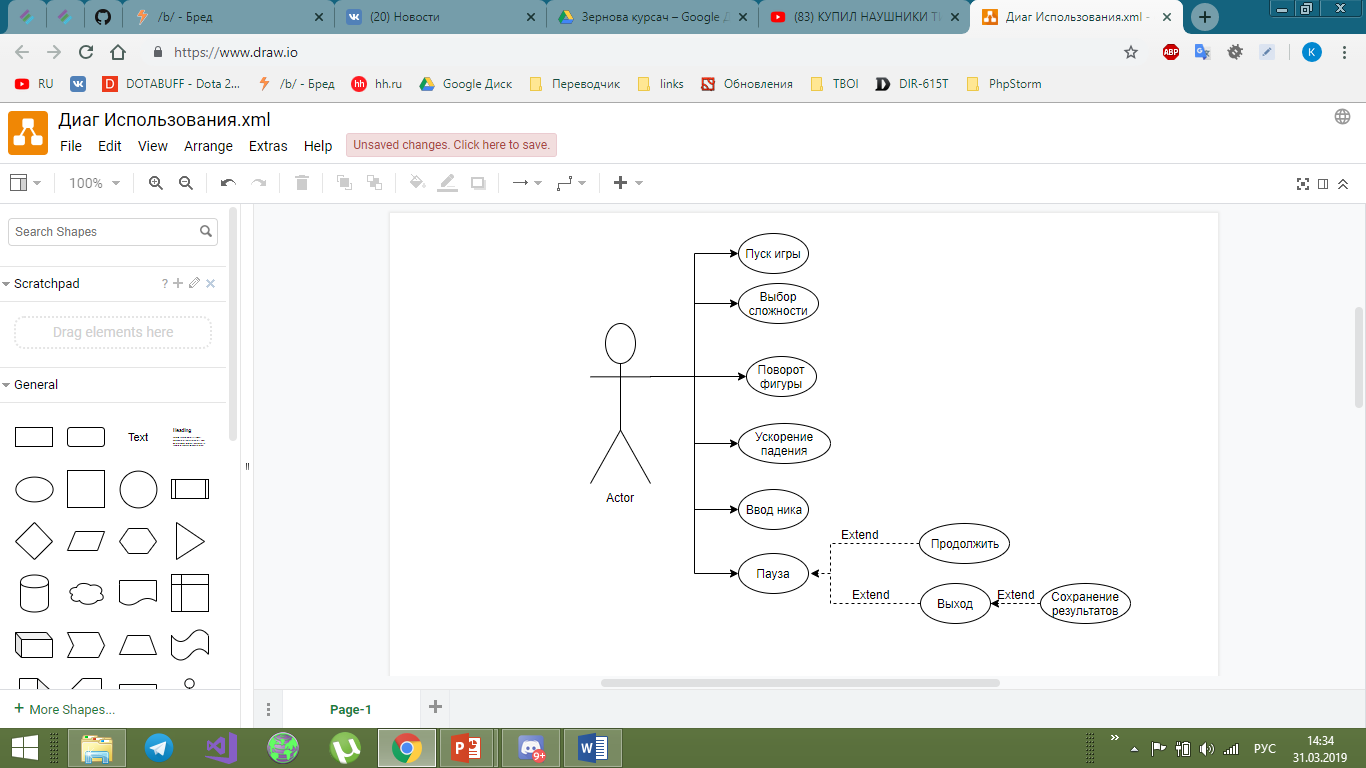


Рисунок 3 – Диаграмма вариантов использования

В программе присутствует 1 роль.

Роль «Пользователь»:

* Есть возможность запуска процесса игры;
* Есть возможность поставить игру на паузу;
* Есть возможность управления процессом игры (поворот, ускорение);
* Есть возможность выбора начальной сложности;
* Есть возможность просмотра таблицы рекордов;

# Построение программы этапа проектирования

В ходе проектирования программы была разработана диаграмма последовательности, изображённая на рисунке 4. Она отображает поток событий включения игрового меню и самого процесса игры.

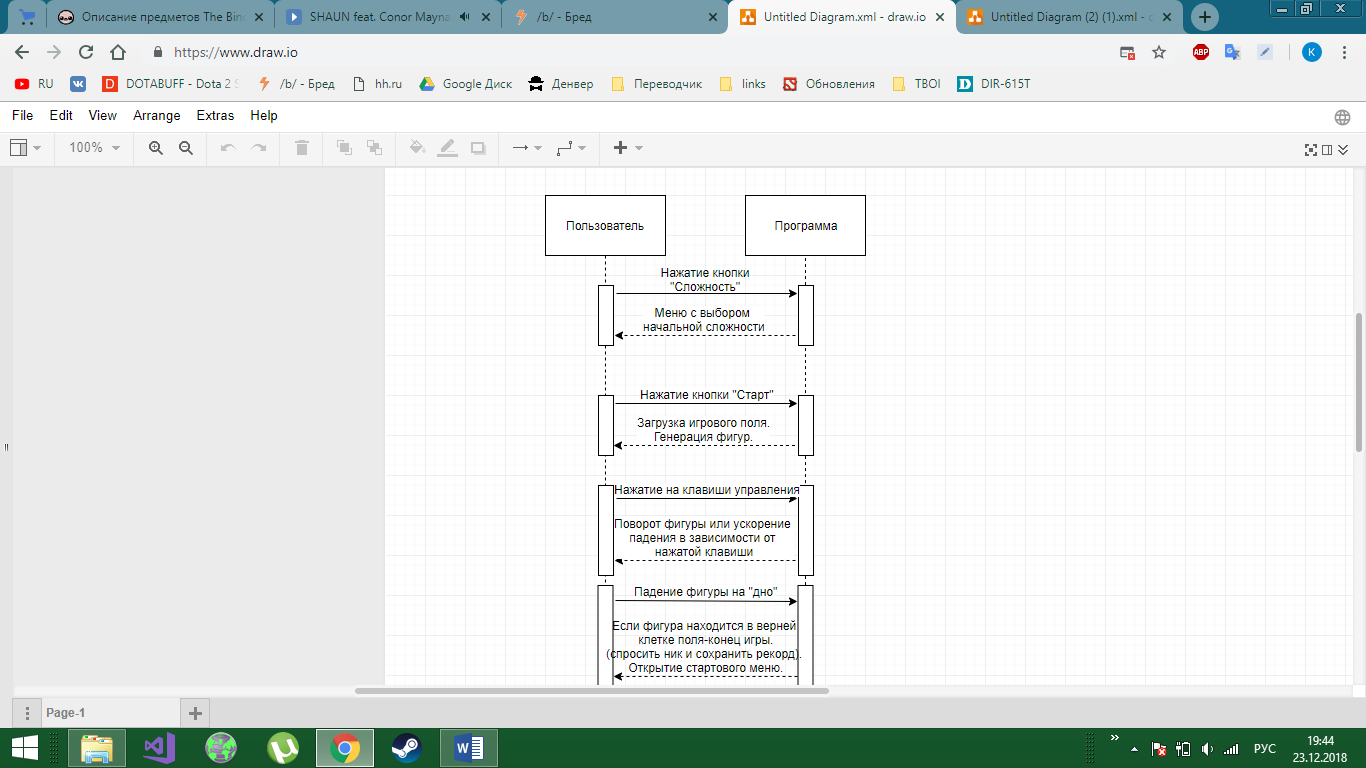
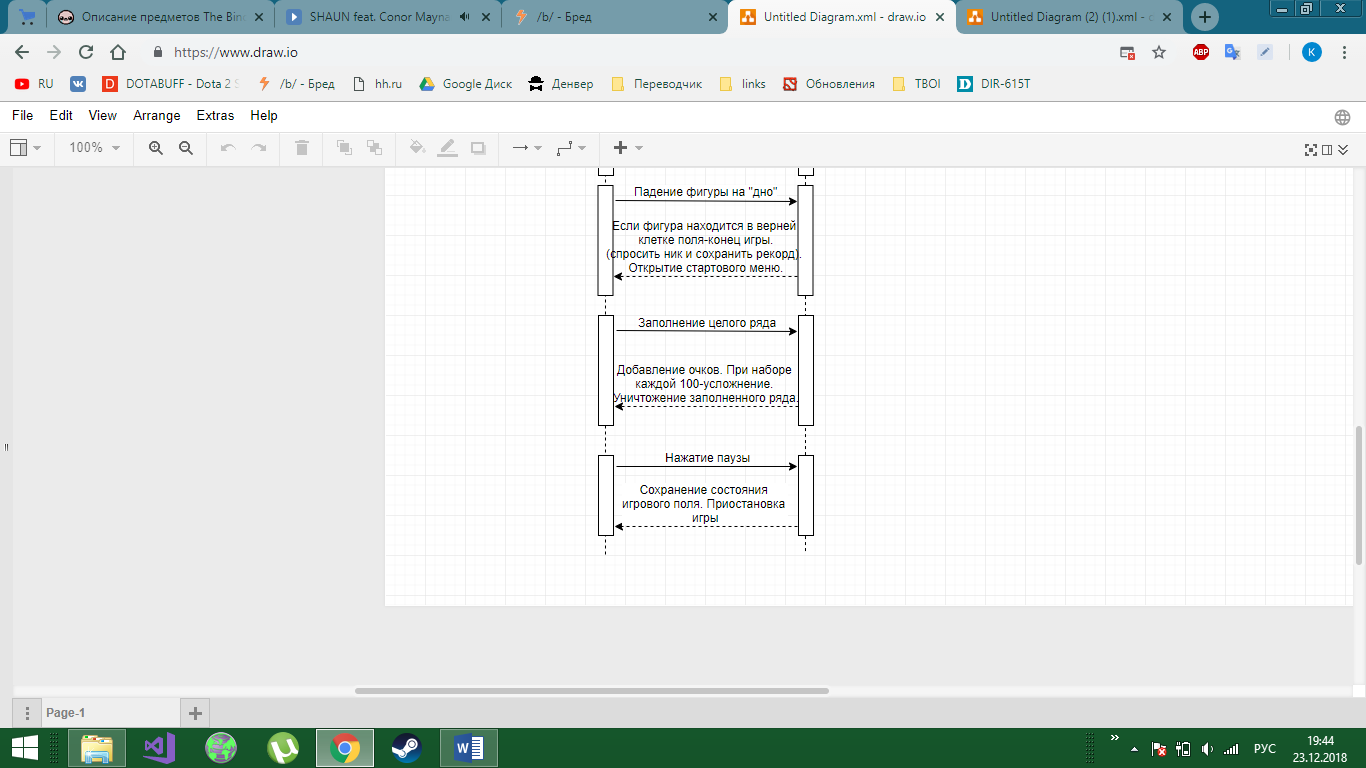
  


Рисунок 4 – Диаграмма последовательности

Также была построена диаграмма активностей. Диаграмма наглядно отражает взаимосвязь между ролями: пользователем и программой.

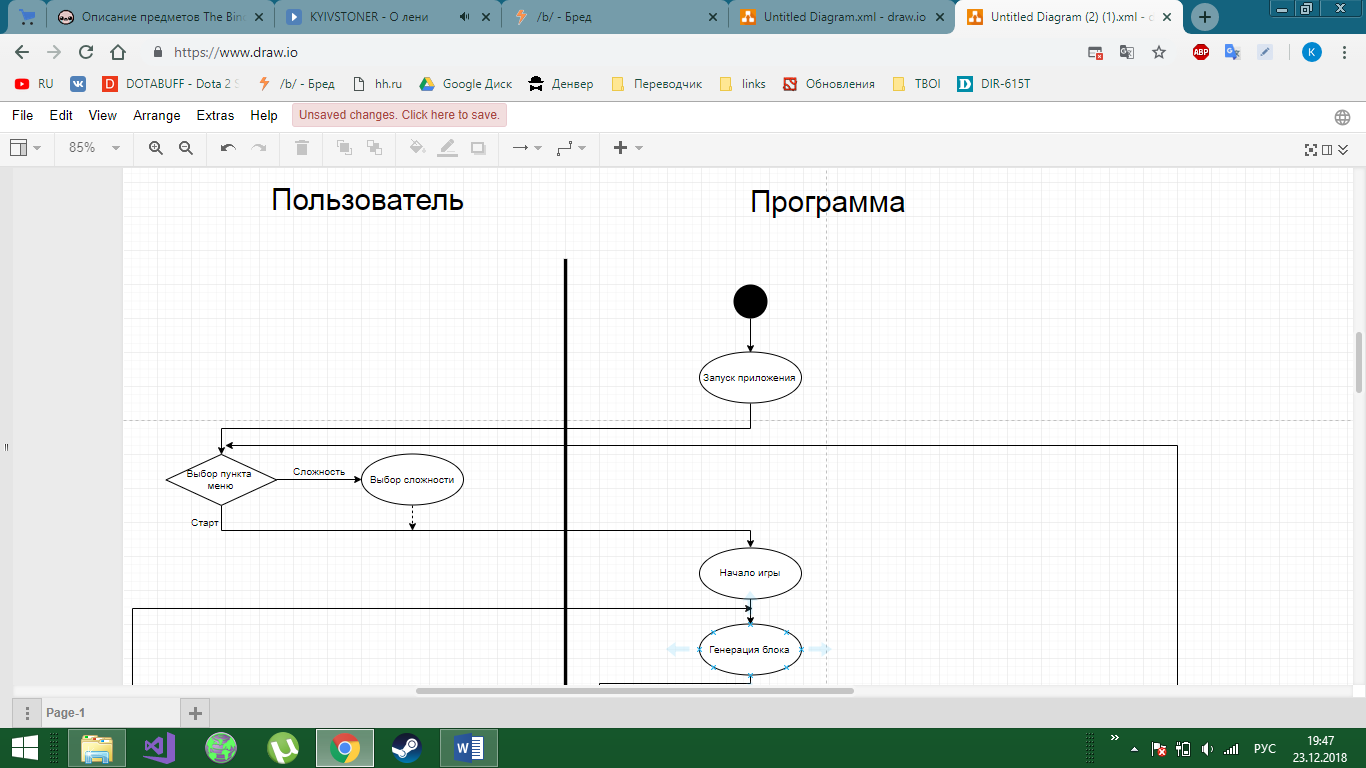
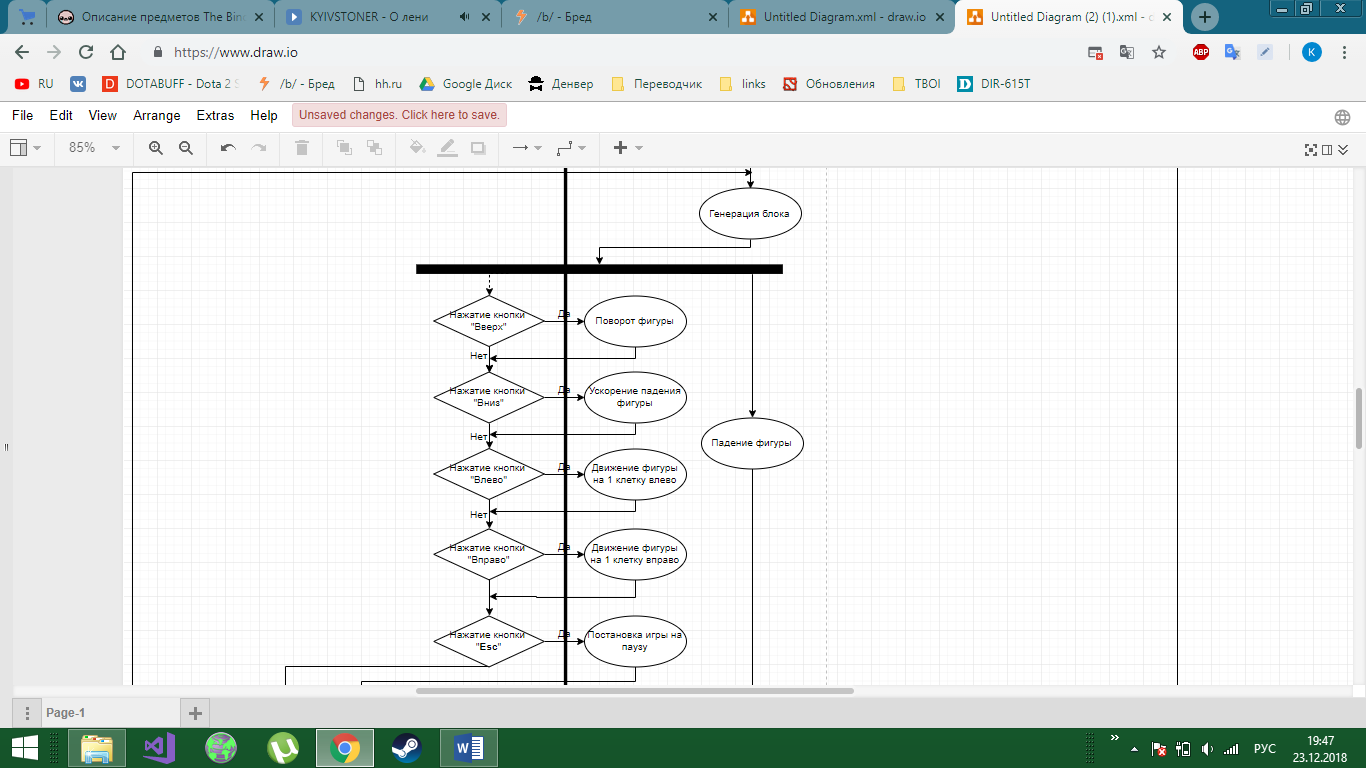
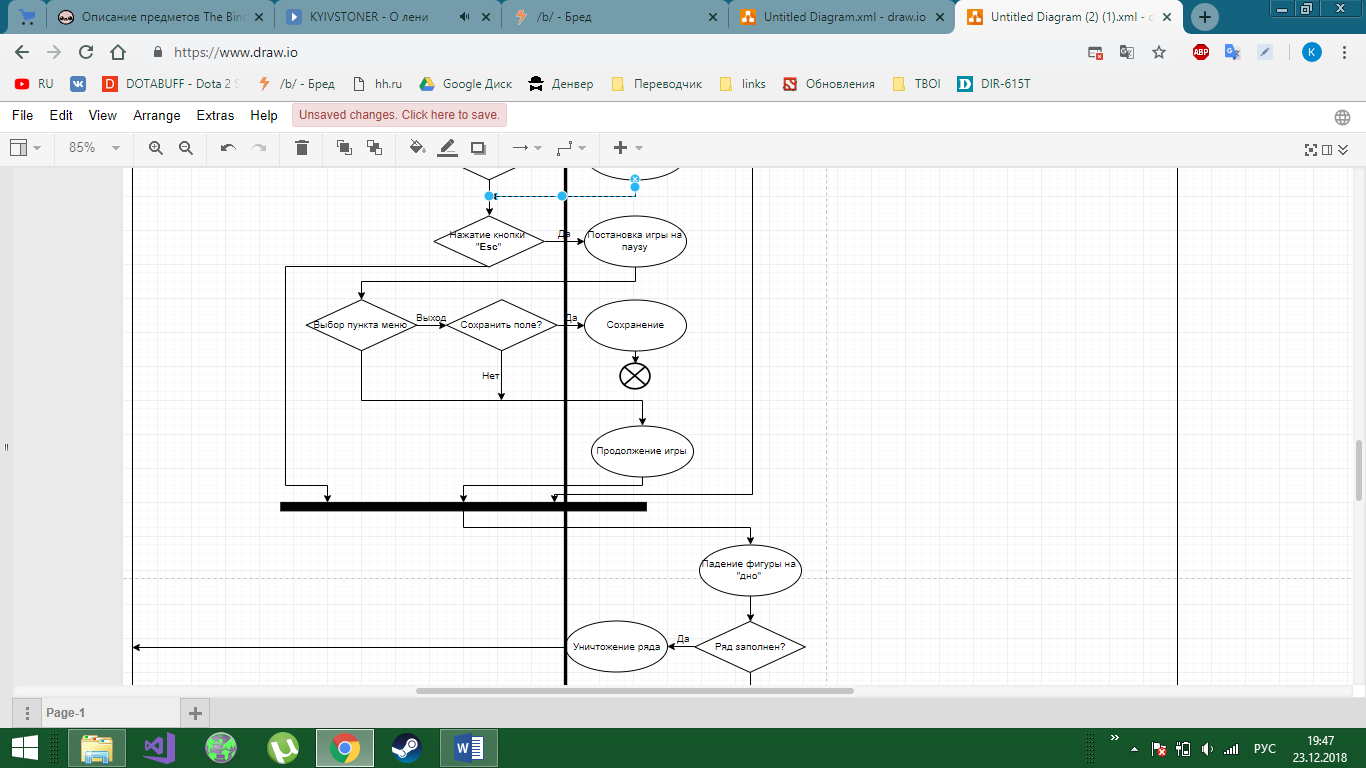
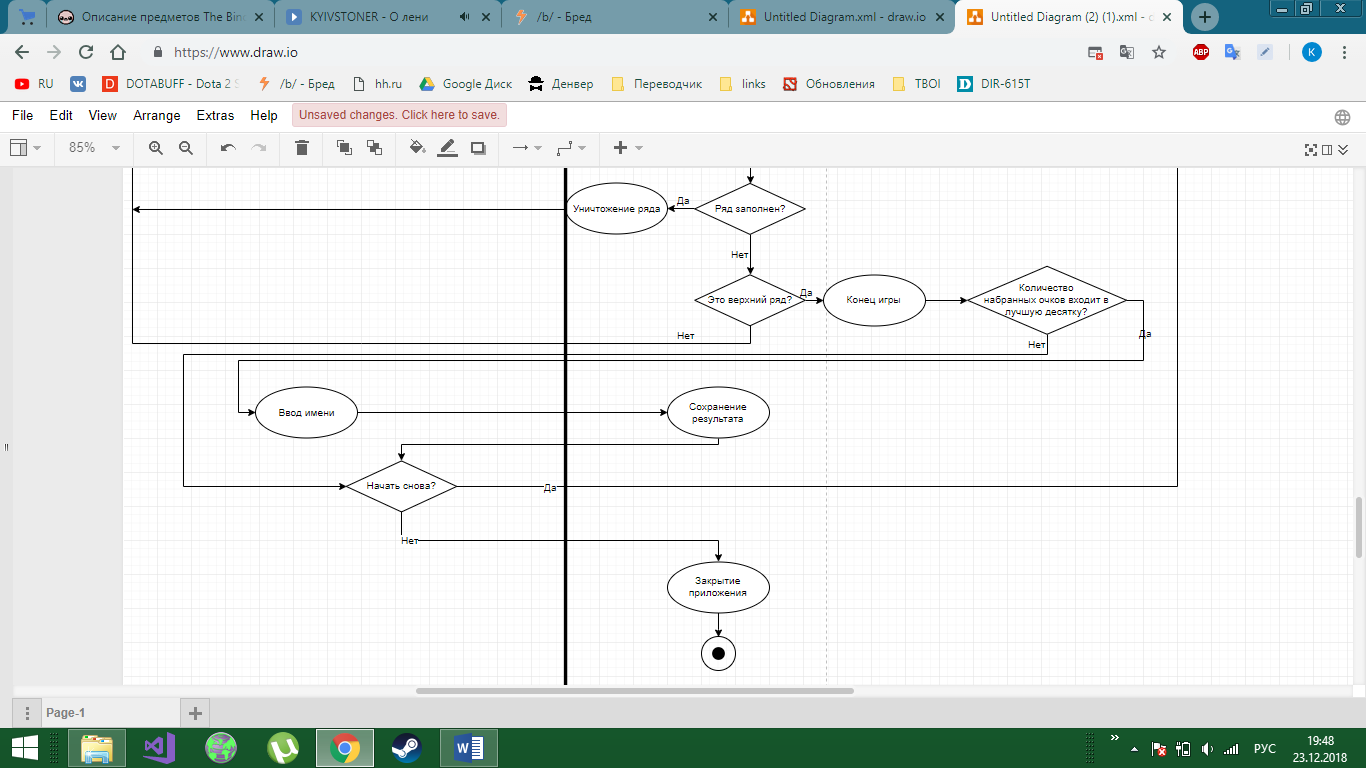
   

Рисунок 5 – Диаграмма активности

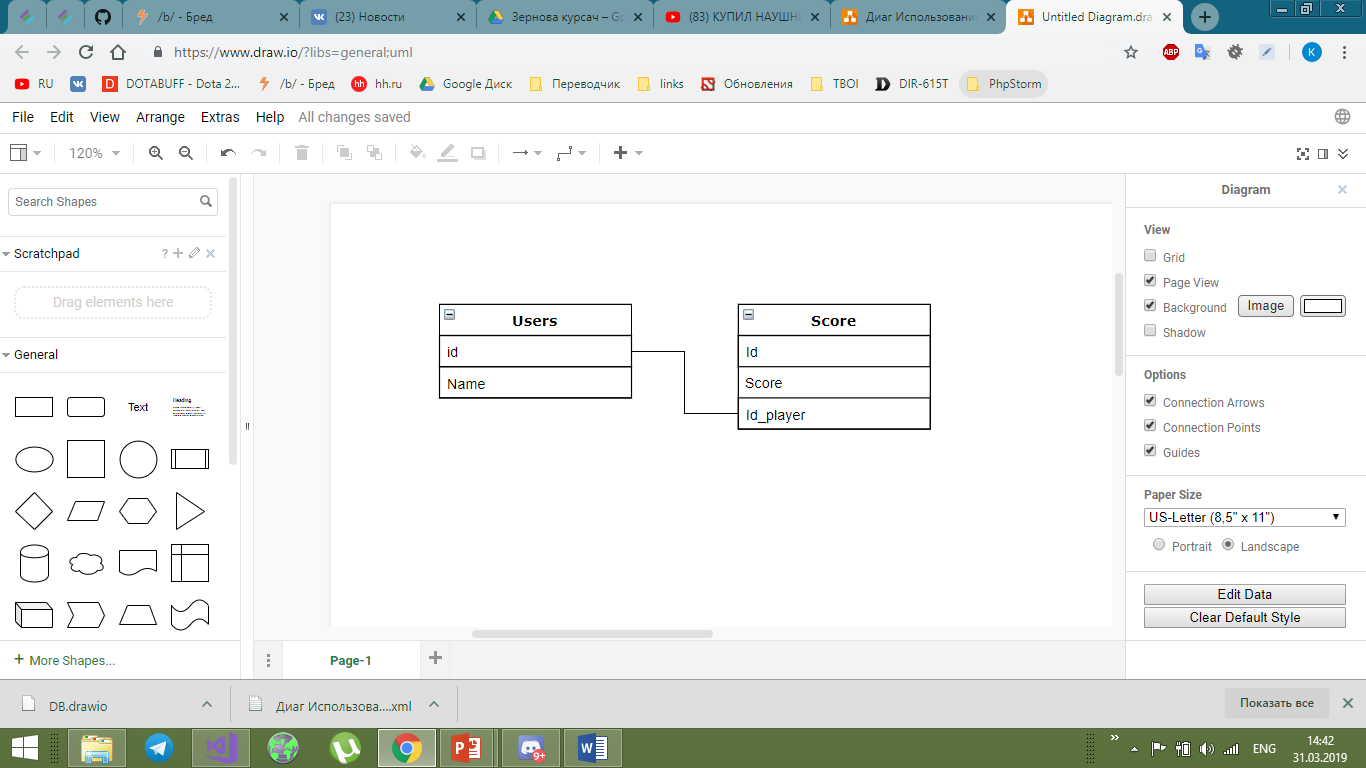
# Разработка программы

Программа была написана на языке с# с использованием технологии WPF.

Для работы программы была создана база данных, которая включает в себя следующие сущности:

* Сущность «Users» необходима для хранения пользователей;
* Сущность «Score» используется для хранения результатов;

Схема базы данных изображена на рисунке 6.

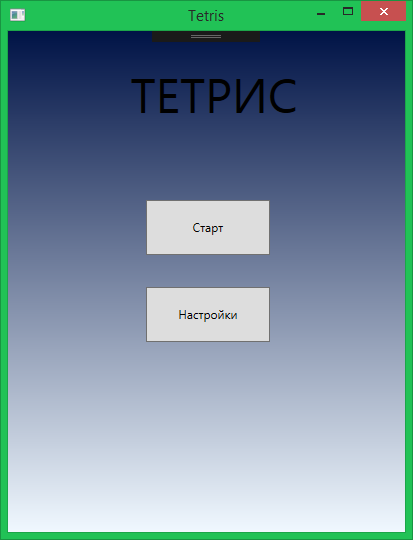
  
Рисунок 6 – База данных

# Руководство оператора

## Модуль пользователя

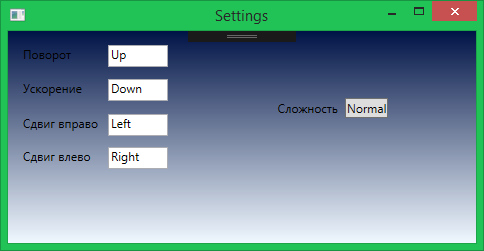
* + 1. Перед игрой
       1. Главное меню

При запуске приложения появляется главное меню (рисунок 7), в котором можно начать игру, нажав на кнопку «Старт», либо настроить сложность игры и кнопки управления, нажав на кнопку «Настройки»:

  
Рисунок 7 – Главная форма пользователя

* + - 1. Настройки

Если вы нажмете кнопку «Настройки» перед вами появится окно настроек (рисунок 8), а главное меню уйдет на задний план.

  
Рисунок 8 – Меню настроек

В этом меню можно настроить кнопки управления игрой, а также выбрать сложность.

* + 1. Меню игры

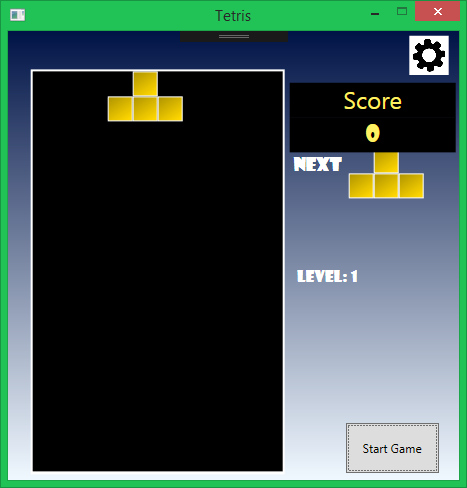
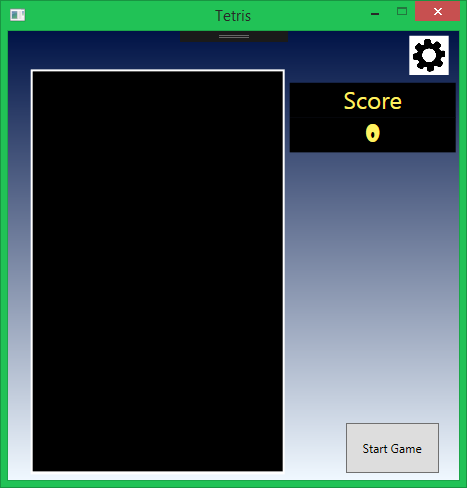
В меню игры мы видим игровое поле, кнопка «Start Game\Pause Game» , поле вывода следующего блока, меню настроек, поле вывода очков (Рисунок 9).

Рисунок 9 –Меню игры

* + 1. Завершение игры
       1. Меню результатов

Когда игра закончена отображается окно результатов в котором находятся: поле ввода имени и окно рекордов (Рисунок 10).

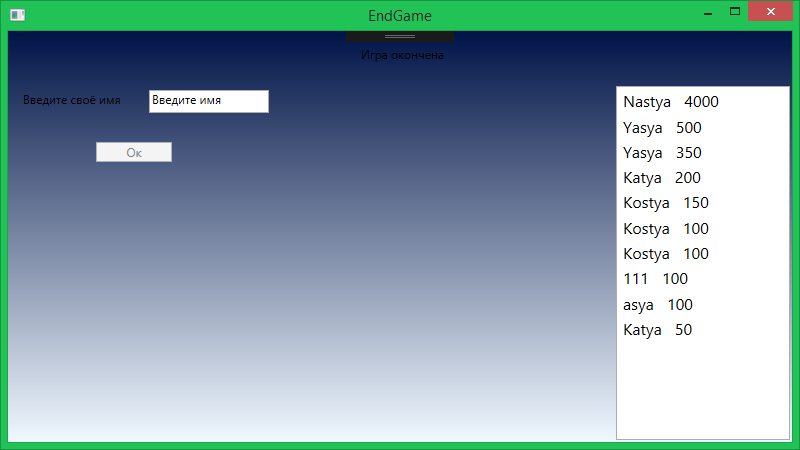


Рисунок 10 – Окно «Меню результатов»

# Методика испытаний

## Объект испытаний

* Полное наименование программной разработки: «Тетрис».
* Основное назначение программного обеспечения заключается в создании программного приложения «Тетрис»: пользователь имеет возможность игры в тетрис, которая заключается в падении блоков из четырех блоков на дно стакана, с последующим их удалением при заполнении ряда.
* Наименование темы разработки – «Тетрис».

## Цель испытаний

Целью проводимых по настоящей программе и методике испытаний является предварительное определение функциональной работоспособности и нагрузочной устойчивости ПО.

Программа приемочных испытаний должна удостоверить работоспособность по в соответствии с функциональным предназначением, нагрузочной устойчивостью.

## Требования к программе

* + 1. Требования к функциональным характеристикам:
* Автоматическая генерация блока;
* Возможность пользователя управлять сложностью игры.
* Давать информацию о следующем блоке, который будет сгенерирован программа должна обеспечивать выдачу чека в конце операции;
* Давать пользователю взаимодействовать с блоками (вращать, двигать и ускорять падение);   
  + 1. Требования к надежности:
* использование лицензированного программного обеспечения;
* проверка программы на наличие вирусов;
* организация бесперебойного питания.

## Требования к программной документации

* + 1. Перечень руководящих документов, на основании которых проводятся испытания

Испытания Системы проводятся на основании следующих руководящих документов:

* утверждённое Техническое задание на разработку Системы;
* настоящая Программа и методика приемочных испытаний.  
  + 1. Место и продолжительность испытаний

Предположительное место проведения испытаний – фирма, создающая игры.

* + 1. Организации, участвующие в испытаниях

В приемочных испытаниях участвуют представители следующих организаций:

* Фирма, работающая в игровой сфере (Заказчик);
* Разработчик (Исполнитель);

Конкретный перечень лиц, ответственных за проведение испытаний системы, определяется Заказчиком.

* + 1. Перечень предъявляемых на испытания документов

Для проведения приемочных испытаний Исполнителем предъявляются следующие документы:

* Техническое задание на разработку Системы
* Пояснительная записка к Техническому проекту Системы
* Описание комплекса технических средств
* Общее описание Системы
* Структура баз данных Системы
* Ведомость эксплуатационных документов
* Руководство оператора Системы
* Программа и методика испытаний

# **Средства и порядок испытаний**

## Требования к составу и параметрам технических средств

* + 1. Для нормального функционирования данной информационной системы необходим компьютер, клавиатура, мышь и следующие технические средства:
* процессор Intel или другой совместимый;
* объем свободной оперативной памяти ~500Мб;
* объем необходимой памяти на жестком диске ~1Гб;
* .NET Framework версии 4.5 и выше.
  + 1. Требования к информационной и программной совместимости

Для полноценного функционирования данной системы необходимо наличие операционной системы выше Microsoft Windows XP или совместимой. Язык интерфейса – русский.

* + 1. Требования к маркировке и упаковке

Программа должна поставляться на диске в виде исполняемого (еxе) файла, документации и проекта. На диске должна быть наклейка с надписью "Тетрис ". Диск должен быть упакован в пластиковую коробочку или бумажный кейс.

* + 1. Требования к транспортировке и хранению

Диск с программой должен храниться вдали от электромагнитных полей и не подвергаться механической деформации. Место и условия хранения должны соответствовать санитарным требованиям отрасли. Сроки хранения устанавливаются в соответствии с гарантийными сроками поставщика магнитных носителей.

Основные требования к транспортировке – создание условий, исключающих механические повреждения магнитного носителя.

## Перечень этапов испытаний и проверок

В процессе проведения приемочных испытаний должны быть протестированы следующие подсистемы ПО:

* подсистема отображения информации;
* подсистема управления содержанием;
* подсистема хранения данных;
* подсистема обмена данными.

Все подсистемы испытываются одновременно на корректность взаимодействия подсистем, влияние подсистем друг на друга, т.е. испытания проводятся комплексно.

Приемочные испытания включают проверку:

* полноты и качества реализации функций, указанных в ТЗ;
* выполнения каждого требования, относящегося к интерфейсу Системы;
* средств и методов восстановления работоспособности Системы после отказов;
* комплектности и качества эксплуатационной документации.
* полноты действий, доступных пользователю, и их достаточность для функционирования Системы;
* возможности работы пользователей без специальной подготовки;
* реакции системы на ошибки пользователя;
* проверку наличия в эксплуатационной документации рекомендаций по восстановлению работоспособности и полноты их описания.

## Методы испытаний

* + 1. Испытания на основе тест-кейсов

Таблица 4 – Тест-кейс отображения всех окон

|  |  |  |
| --- | --- | --- |
| **Действие** | **Ожидаемый результат** | **Результат теста:**   * **пройден** * **провален** * **заблокирован** |
| Запуск программного обеспечения | На экране отобразится страница с кнопками «Старт» и «Настройки». | Пройден |
| Нажатие кнопки «Старт» | Откроется окно с полем для игры и кнопками «Пауза» и «Настройки». | Пройден |
| Окно «Настройки» | Откроется окно настроек | Пройден |

Для выполнения тест-­кейсов этой категории необходимо запустить программу, нажать кнопку «Старт» и в открывшемся окне производить необходимые тесты.

Таблица 5 – Тест-кейс процесса игры

|  |  |  |
| --- | --- | --- |
| **Действие** | **Ожидаемый результат** | **Результат теста:**   * **пройден** * **провален** * **заблокирован** |
| Кнопка «Start Game» | Запускается генерация блоков и начинается процесс игры | Пройден |
| Кнопка «Stop Game» | Кнопка появляется во время процесса игры. При нажатии игра ставится на паузу. | Пройден |
| Заполнить полностью 1(или более) горизонтальный ряд | Ряды должны исчезнуть и в окне «Score» отобразиться новый результат на 50 больший предыдущего. | Пройден |
| Заполнить «Стакан» тетриса до верхнего ряда | Игра прекращается, открывается окно результатов | Пройден |

Для выполнения тест-­кейсов этой категории необходимо запустить программу, нажать кнопку «Старт», в открывшемся окне нажать «Start game» и закончить игру (Заполнить все вертикальные ряды). После этого откроется окно результатов, в котором необходимо проводить тесты.

Таблица 6 - Тест-кейс окна результатов

|  |  |  |
| --- | --- | --- |
| **Действие** | **Ожидаемый результат** | **Результат теста:**   * **пройден** * **провален** * **заблокирован** |
| Нажатие кнопки «Ок» с неизмененным полем имени | Кнопка «Ок» блокируется и не даёт пользователю это сделать. | Пройден |
| Нажатие кнопки «Ок» с пустым полем имени | Кнопка «Ок» блокируется и не даёт пользователю это сделать. | Пройден |
| Нажатие кнопки «Ок» с заполненным полем имени | Кнопка «Ок» блокируется, предотвращая повторное добавление, таблица рекордов обновляется, и если пользователь попал в список рекордсменов, отобразить его результат в таблице рекордов | Пройден |
| Заполнить «Стакан» тетриса до верхнего ряда | Игра прекращается, открывается окно результатов | Пройден |

* + 1. Испытания на основе Unit tests

Для проверки корректности выполняемых методов были написаны unit-tests, которые покрывают наиболее важные методы для нормального функционирования приложения.

* TestScore() – Тест, проверяющий работу метода подсчета очков
* TestColor() – Тест, проверяющий работу метода покраски фигуры тетрамино
* TestReturn() – Тест, проверяющий работу метода для сброса необходимых для работы данных

[TestClass]

public class UnitTest1

{

MainWindow win = new MainWindow();

[TestMethod]

//метод для проверки корректности работы

//метода, считающего и возвращающего набранные очки

public void TestScore()

{

int scoreNormal=50;

for(int i=1;i<=3;++i)

Assert.AreEqual(scoreNormal, win.getScore()/i);

}

[TestMethod]

//метод для проверки метода, красящего фигуры тетромино

public void TestColor()

{

Color clr = new Color();

LinearGradientBrush kek = new LinearGradientBrush();

Assert.AreEqual(kek.GetType(), win.getGradientColor(clr).GetType());

}

[TestMethod]

//метод для проверки метода, обнуляющего значения

public void TestReturn()

{

win.downPos = 5;

win.leftPos = 5;

win.isRotated = true;

win.rotation = 99999;

win.reset();

Assert.AreEqual(0, win.downPos);

Assert.AreEqual(3, win.leftPos);

Assert.AreEqual(false, win.isRotated);

Assert.AreEqual(0, win.rotation);

}

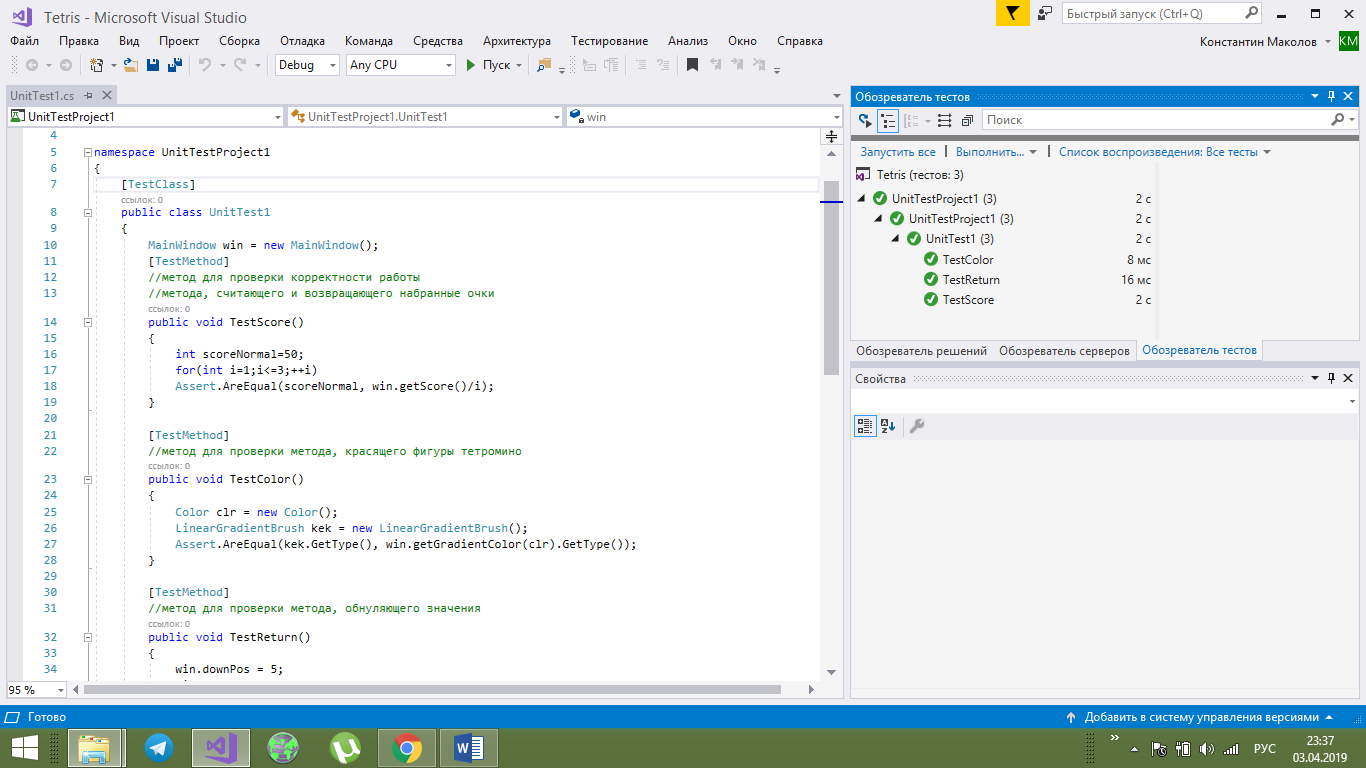
 }

Рисунок 11 – Выполненные UnitTests

* + 1. Испытания на основе User stories
       1. User stories для пользователя
    - Как пользователь я могу назначить клавиши управления.
    - Как пользователь я могу настроить сложность игры.
    - Как пользователь я могу играть в игру «Tetris».
    - Как пользователь я могу менять сложность игры.
    - Как пользователь я могу просматривать свой результат.
    - Как пользователь я могу видеть следующий блок
    - Как пользователь я могу просматривать таблицу рекордов
    - Как пользователь я могу добавлять свой результат в таблицу рекордов
    1. Аудирование
       1. Требования к функциональным характеристикам

Выполнено:

* Программа должна случайно генерировать блоки заранее определенной формы
* Программа должна предоставлять возможность поставить игру на паузу
* Программа должна позволять пользователю просмотра рекордов
* Должна сохранять результат
* Должна давать возможность настройки сложности
* Должна давать возможность настройки клавиш управления

Не выполнено:

* Программа должна предоставлять возможность сохранения игрового поля;
  + - 1. Полнота покрытия тестами

С помощью тестов были проверены следующие требования:

* программа должна после каждого убранного ряда добавлять пользователю 50\*n очков, где n-уровень сложности

**(результат теста:** пройден);

* программа должна окрашивать разные блоки в разные цвета

**(результат теста:** пройден);

* программа должна выполнять сброс всех данных в конце игры

**(результат теста:** пройден);

* + - 1. Юзабилити

В программе все интуитивно понятно. Программой пользоваться удобно, но во время процесса игры следующий блок отображается только после небольшой задержки. А также, путем некоторых манипуляций, можно добавить свой результат в таблицу несколько раз за одну игру.

* + - 1. Дизайн программы

Элементы расположены удобно и не вызывают затруднений в поиски той или иной кнопки. Все окна в программе выполнены в одинаковом стиле.

* + - 1. Понятность и полнота Руководства оператора

Руководство оператора написано довольно подробно и для каждого из пунктов предоставлены соответствующие скриншоты. Каждый пункт содержит исчерпывающее описание порядка действий при работе с программой. Так же руководство оператора написано простым языком, который не вызывает затруднений в понимании.

* + - 1. Оценка уровня защищенности данных

Так как программы не использует никаких личных данных пользователя, то о защищённости данных говорить не приходится.

* + - 1. Оценка качества написания кода

Код написан хорошо, без использования GOTO. Переменные названы логично и не составляют труда в понимании. В коде нет магических чисел, код чистый. Код содержит лишь незначительное количество комментариев. Код использует большое количество методов, ссылающиеся друг на друга, из-за этого программа трудночитаема, но при этом названия методов отображают выполняемые ими функции

# ЗАКЛЮЧЕНИЕ

Результатом разработки проекта является программная модель, которая будет использоваться для личного использования.

Программа соответствует требованиям, перечисленным в техническом задании, и полностью готова к использованию.