*Министерство образования и молодежной политики   
Свердловской области*

*ГАПОУ СО «Асбестовский политехникум»*

*Специальность 09.02.07 «Информационные системы и программирование»*

*ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА*

*КОМПЬЮТЕРНОЙ ИГРЫ «ТЕТРИС»*

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

*КП ИС 2025.13 ПЗ*

|  |  |
| --- | --- |
| *Руководитель проекта*  *Копина М.Г.*  *\_\_\_\_\_\_\_.2025* | *Студент группы ИС-22*  *Лизякин И.В.*  *\_\_\_\_\_\_\_.2025* |

*2025*

*СОДЕРЖАНИЕ*

[введение 3](#_Toc191676808)

[1 постановка и проектирование задачи 6](#_Toc191676809)

[1.1 Описание программы 6](#_Toc191676810)

[1.2 Функциональные возможности 7](#_Toc191676811)

[1.3 Дизайн-макеты интерфейса 8](#_Toc191676812)

[2 кодирование программного продукта 12](#_Toc191676813)

[2.1 Игровой цикл 12](#_Toc191676814)

[2.2 Методы работы с конфигом 13](#_Toc191676815)

[2.3 Метод фиксирующий результат игрока 15](#_Toc191676816)

[2.4 Метод инициализации Canvas 15](#_Toc191676817)

[2.5 Логические методы и методы взаимодействия в классе GameGrid 16](#_Toc191676818)

[Заключение 18](#_Toc191676819)

[Список литературы 20](#_Toc191676820)

# введение

Объект исследования: процесс проектирования и разработки цифровой игры Tetris.

Предмет исследования: пользовательский интерфейс Windows Presentation Foundation (WPF).

Цель проекта:

* изучить функционал интерфейса Windows Presentation Foundation;
* разработать свой программный продукт;
* опубликовать приложение на GitHub.

Для достижения поставленных целей сформулированы следующие задачи:

* найти хорошие информационные источники для изучения Windows Presentation Foundation;
* воплотить оригинальный геймплей Тетрис;
* реализовать дополнительный функционал (пауза в игре, реализация русского и англоязычного интерфейса, механика сложности, звуки и музыка).

Актуальность проекта заключается в том, что выбранная игра проста в освоении, интересна в воплощении, не забывается со временем. Данная игра для многих пользователей – ностальгия, в которую хочется снова погрузиться. Даже без обучения рядовой игрок может интуитивно разобраться в управлении игры, так как в ней отсутствуют комбинации в сравнении с популярными на данный момент шутерами и файтингами.

Практическая значимость этой работы заключается в развитии навыков разработки программных модулей.

Были рассмотрены следующие реализации игры Tetris:

1. работа Никиты Юдина [1] на GitHub;
2. игра Elshad Safarov [2] в сервисе Яндекс Игры;
3. игра OttoBot [3] на YouTube.

Преимущества в игре Никиты Юдина:

* возможность менять настройки графики;
* возможность вводить чит-коды;
* присутствует разделение скорости на уровни.

Недостатки:

* отсутствие удерживаемого пространства для фигур;
* реагирование антивируса на игру.

Преимущества в игре Elshad Safarov:

* приятная визуальная составляющая;
* быстрый отклик.

Недостатки:

* асимметричный поворот;
* отсутствие удерживаемого пространства для фигур;
* отсутствие дополнительных механик.

Преимущества в игре OttoBot:

* приятная визуальная составляющая;
* наличие удерживаемого пространства для фигур;
* масштабируемость.

Недостатки:

* отсутствие дополнительных механик;
* задержка в отклике.

Для разработки игры выбрана система Windows Presentation Foundation.

Windows Presentation Foundation – технология для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем. Windows Presentation Foundation является аналогом Windows Forms – также системой для построения приложений, но гораздо более упрощенной.

Преимущества Windows Presentation Foundation в сравнении с Windows Forms:

1. Адекватная работа с разметкой. Благодаря тому, что визуальная часть программы прописывается через XAML, гораздо проще и привычнее составлять масштабируемые окна, в сравнении с Windows Forms, где это реализовано хуже.
2. Аппаратное ускорение графики. Windows Presentation Foundation позволяет перенести нагрузку с процессора на видеокарту, что в случае разработки игры имеет существенное значение.
3. Визуальные украшательства. Ключевым решением в выборе Windows Presentation Foundation стало наличие возможности настраивать градиент, прозрачность элементов и другие визуальные характеристики, что является невозможным в Windows Forms. Визуальная составляющая является важной частью компьютерной игры.

# постановка и проектирование задачи

## Описание программы

«Тетрис» представляет собой головоломку, построенную на использовании геометрических фигур «тетрамино».

Случайные фигурки падают сверху в прямоугольный стакан шириной 10 и высотой 20 клеток.

В полёте игрок может поворачивать фигурку на 90° с помощью кнопок "↑" и "С", а также двигать её по горизонтали с помощью кнопок "←" и "→".

Чтобы ускорить падение фигуры можно нажать на "↓" и тогда фигура принудительно снизится на 1 клетку вниз

Также можно "сбрасывать" фигурку, то есть ускорять её падение, когда уже решено, куда фигурка должна упасть с помощью клавиши "Пробел".

Фигурка летит до тех пор, пока не наткнётся на другую фигурку либо на дно стакана.

Если вы хотите сохранить текущую фигуру "на потом", вы можете нажать клавишу "Ч" и она поместится в левое окошко удерживаемых фигур.

Учтите, что вернуть фигуру обратно на поле можно только после того, как текущая фигура на поле упадет на дно стакана, а уже после этого вы сможете поменять местами новую появившуюся фигуру на поле с удерживаемой фигурой в запасе.

При нажатии кнопки "Esc" вы можете перейти в меню паузы.

Если заполнился горизонтальный ряд из 10 клеток, он пропадает и всё, что выше него, опускается на одну клетку.

Дополнительно показывается фигурка, которая будет следовать после текущей — это подсказка, которая позволяет игроку планировать действия. Темп игры постепенно ускоряется. Игрок получает очки за каждый заполненный ряд, поэтому его задача — заполнять горизонтальные ряды, не заполняя сам стакан.

Игра заканчивается, когда новая фигурка не может поместиться в стакан. При начислении очков за линии количество очков зависит от того, сколько линий убрано за один раз, и какая сложность указана в настройках. Начисление очков происходят по таким правилам (для уровня сложности "Легко"): 1 линия — 100 очков, 2 линии — 300 очков, 3 линии — 700 очков, 4 линии (то есть сделать Тетрис) — 1500 очков. При уровне сложности "Сложно" количество очков удваивается, но и скорость падения также ускоряется вдвое.

Программный продукт предназначен для пользователей любого возраста, желающих провести свой досуг занимательно. В игре присутствует соревновательный элемент. Приложение предоставляет возможность просматривать рекорды пользователей в специальной таблице лидеров. Игра бросает вызов игрокам – набрать наибольшее количество очков.

Для игроков присутствует два уровня сложности. Уровень игрок выбирает самостоятельно. После завершения игрового процесса результат игры и выбранный уровень сложности сохраняются.

## Функциональные возможности

В приложении реализован функционал манипуляции фигурами:

* перемещение фигуры влево;
* перемещение фигуры вправо;
* перемещение фигуры вниз на одну клетку;
* перемещение фигуры вниз до упора в игровое поле или другую фигуру;
* поворот фигуры по часовой стрелке;
* поворот фигуры против часовой стрелки;
* сохранение фигуры на хранение в удерживаемой области.

Также в приложении предусмотрена фиксация результата (рекорда) определенного пользователя с последующим сохранением в системе.

## Дизайн-макеты интерфейса

Дизайн-макет англоязычного интерфейса главного меню представлен на рисунке 1.

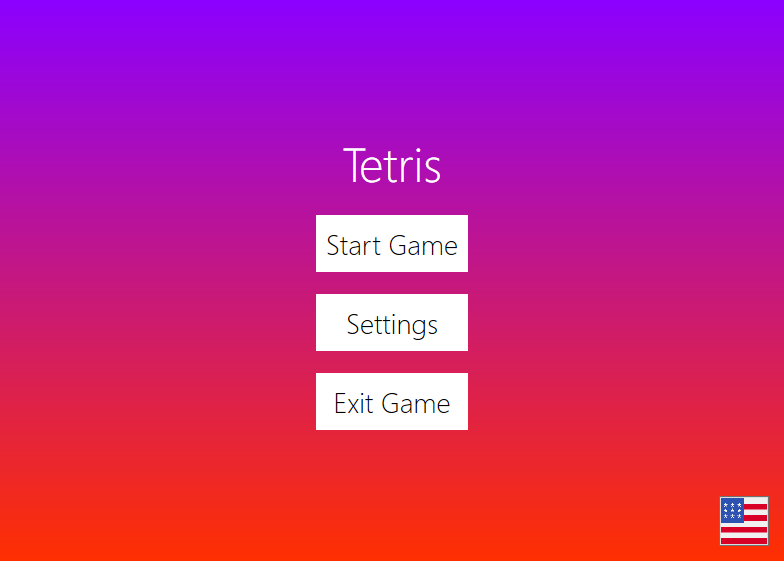


Рисунок 1 – Дизайн-макет англоязычного интерфейса главного меню.

Дизайн-макет русскоязычного интерфейса игрового поля представлен на рисунке 2.

Рисунок 2 – Дизайн-макет русскоязычного интерфейса главного меню.

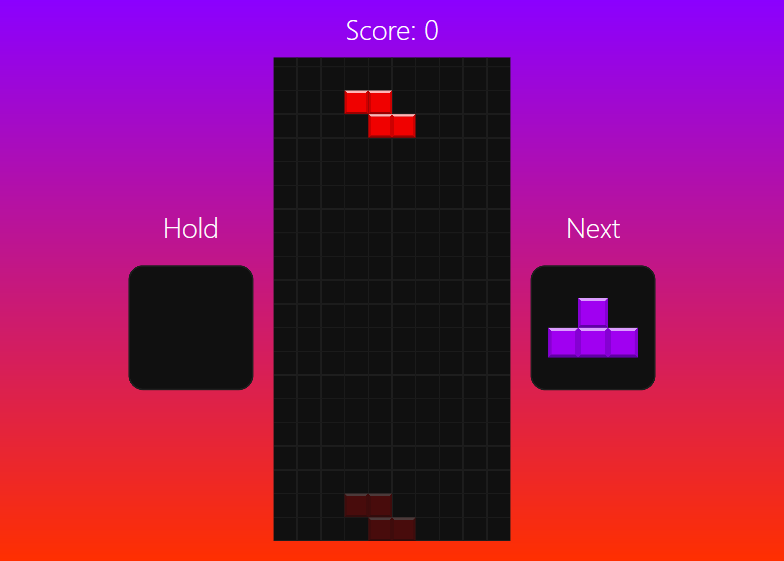
Дизайн-макет англоязычного интерфейса игрового поля представлен на рисунке 3.

Рисунок 3 – Дизайн-макет англоязычного интерфейса игрового поля.

Дизайн-макет англоязычного интерфейса меню паузы представлен на рисунке 4.

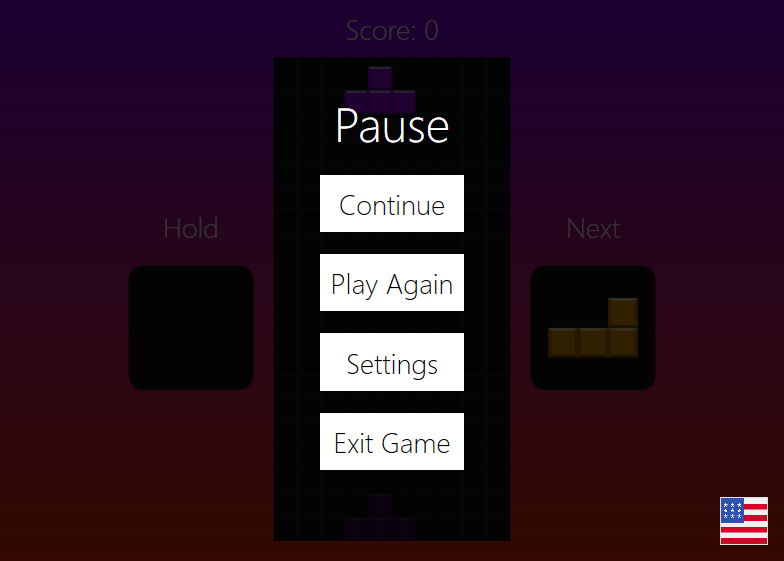


Рисунок 4 – Дизайн-макет англоязычного интерфейса меню паузы.

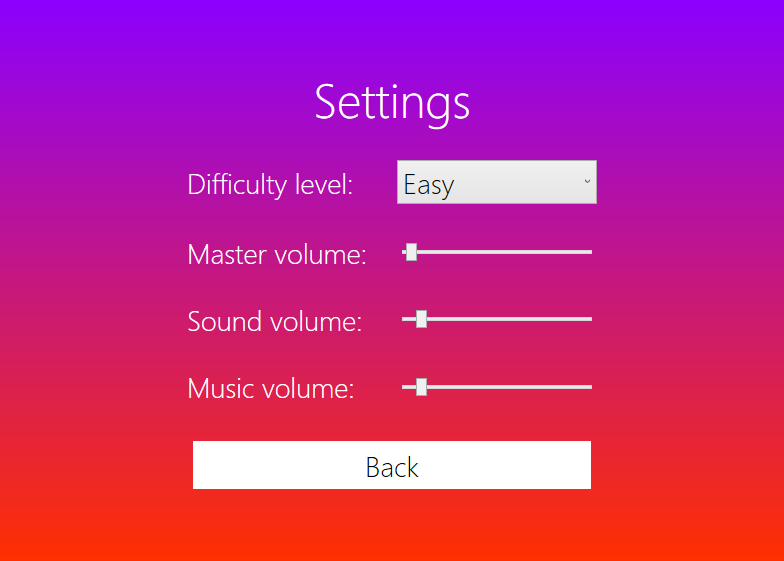
Дизайн-макет англоязычного интерфейса меню настроек представлен на рисунке 5.

Рисунок 5 – Дизайн-макет англоязычного интерфейса меню настроек.

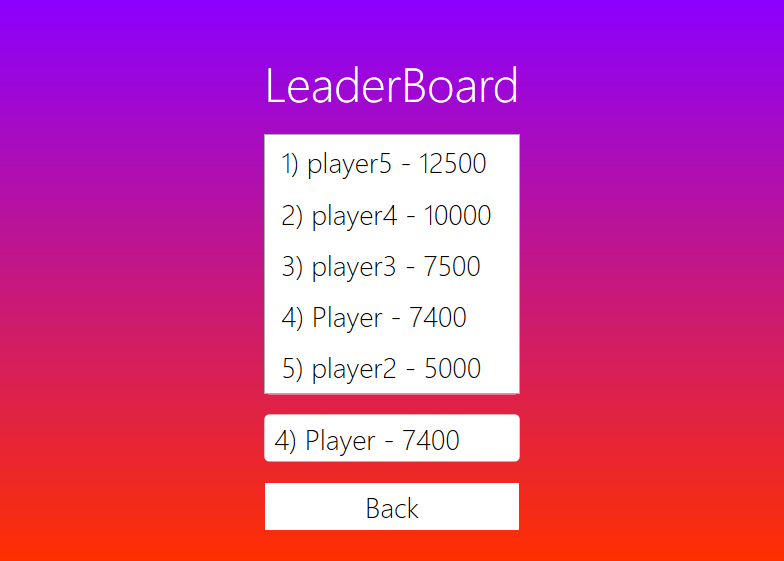
Дизайн-макет англоязычного интерфейса таблицы лидеров представлен на рисунке 6.

Рисунок 6 – Дизайн-макет англоязычного интерфейса таблицы лидеров.

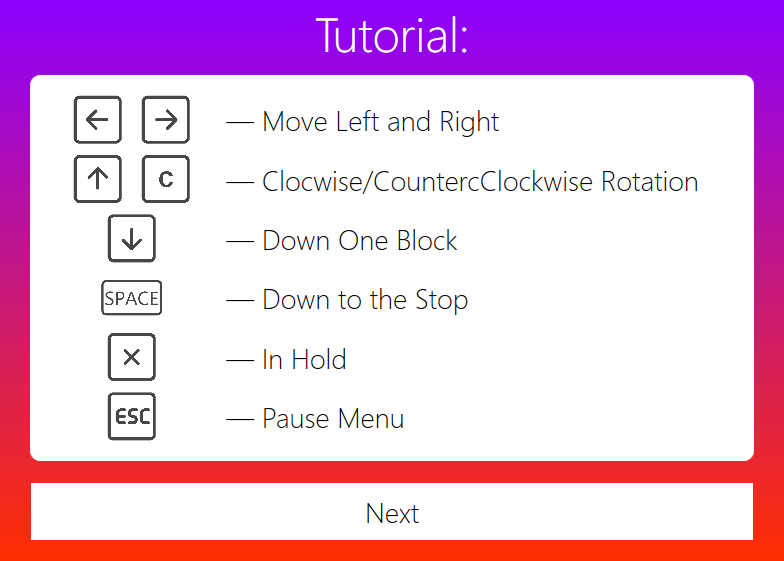
Дизайн-макет англоязычного интерфейса обучения представлен на рисунке 7.

Рисунок 7 – Дизайн-макет англоязычного интерфейса обучения.

# кодирование программного продукта

Ниже приведены интересные примеры участков кода:

## Игровой цикл

private async Task GameLoop(CancellationToken token)

{

try

{

Draw(gameState);

//В главном меню

while (!IsGameStarted)

{

await Task.Delay(500);

}

SetLanguage();

GameField.Visibility = Visibility.Visible;

int difficulty = SettingsMenu.DifficultyModificator;

//В игре

while (!gameState.GameOver && !token.IsCancellationRequested)

{

//Подсчет времени падения

int delay = Math.Max(\_minDelay, \_maxDelay - (gameState.ClearedRows \* \_delayDecrease \* difficulty));

await Task.Delay(delay);

//В меню паузы

while (IsGamePaused && !token.IsCancellationRequested)

{

PauseMenu.Visibility = Visibility.Visible;

await Task.Delay(500);

difficulty = SettingsMenu.DifficultyModificator;

}

PauseMenu.Visibility = Visibility.Hidden;

//Чтобы при нажатии в тайминг блок дважды вниз не спускался

if (\_isKeyDownPressed)

{

\_isKeyDownPressed = false;

}

else

{

gameState.MoveBlockDown();

}

Draw(gameState);

}

}

catch (OperationCanceledException)

{

} //Необходимо при перезапуске игры с меню паузы

WindowRegistration window = new WindowRegistration();

window.Owner = this;

window.Show();

//Для меню конца игры

FinalScoreText.Text = $"{\_scoreText}{gameState.Score}";

GameOverMenu.Visibility = Visibility.Visible;

}

## Методы работы с конфигом

public static void CreateConfigFile(string filename)

{

using (StreamWriter sw = File.CreateText(filename))

{

sw.WriteLine("difficulty = 1");

sw.WriteLine("master\_volume = 0.5");

sw.WriteLine("music\_volume = 1.0");

sw.WriteLine("sound\_volume = 1.0");

}

}

private static T GetValueFromConfigFile<T>(string field)

{

using (var sr = new StreamReader("config.txt"))

{

while (!sr.EndOfStream)

{

string line = sr.ReadLine()?.Trim();

if (string.IsNullOrEmpty(line))

continue;

string[] parts = line.Split(" = ");

if (typeof(T) != typeof(string))

{

if (parts[0].Equals(field, StringComparison.OrdinalIgnoreCase))

{

return (T)Convert.ChangeType(parts[1], typeof(T));

}

}

else

{

if (parts[0].Equals(field, StringComparison.OrdinalIgnoreCase))

{

return (T)(object)parts[1];

}

}

}

}

return default(T);

}

private static void SetValueToConfigFile<T>(string field, T value)

{

string filename = "config.txt";

using (var sr = new StreamReader(filename))

using (var sw = new StreamWriter(filename + ".tmp", false))

{

while (!sr.EndOfStream)

{

string line = sr.ReadLine();

if (line != null)

{

if (sr.EndOfStream)

{

if (line.Split(" = ")[0] == field)

{

sw.Write(field + " = " + value);

}

else

{

sw.Write(line);

}

}

else

{

if (line.Split(" = ")[0] == field)

{

sw.WriteLine(field + " = " + value);

}

else

{

sw.WriteLine(line);

}

}

}

}

}

File.Delete(filename);

File.Move(filename + ".tmp", filename);

}

## Метод фиксирующий результат игрока

public static void UpdateTable(string name, int score)

{

//Высчитывание результатов и внесение в таблицу

User player = new User(name, score);

try

{

//Список рекордсменов (Если нет - сработает обработка исключений)

LeaderBoard.UpdateLeaderBoardList();

//Позиция игрока в списке (Если игрока в списке нет - сработает отработка исключений)

LeaderBoard.GetCurrentUser(out User playerInList1, out int index1);

if (player.Score > playerInList1.Score)

{

LeaderBoard.RewriteLineInList(index1, player);

}

}

catch

{

LeaderBoard.AddLineInList(player);

}

LeaderBoard.UpdateLeaderBoardList();

}

## Метод инициализации Canvas

private Image[,] InitGameCanvas(GameGrid grid)

{

Image[,] imageControls = new Image[grid.Rows, grid.Columns];

int cellSize = 25;

for (int r = 0; r < grid.Rows; r++)

{

for (int c = 0; c < grid.Columns; c++)

{

Image imageControl = new Image

{

Width = cellSize,

Height = cellSize

};

Canvas.SetTop(imageControl, (r - 2) \* cellSize + 10);

Canvas.SetLeft(imageControl, c \* cellSize);

GameCanvas.Children.Add(imageControl);

imageControls[r, c] = imageControl;

}

}

return imageControls;

}

## Логические методы и методы взаимодействия в классе GameGrid

public bool IsInside(int r, int c)

{

return r >= 0 && r < Rows && c >= 0 && c < Columns;

}

public bool IsEmpty(int r, int c)

{

return IsInside(r, c) && grid[r, c] == 0;

}

public bool IsRowFull(int r)

{

for (int c = 0; c < Columns; c++)

{

if (grid[r,c] == 0)

{

return false;

}

}

return true;

}

public bool IsRowEmpty(int r)

{

for (int c = 0; c < Columns; c++)

{

if (grid[r,c] != 0)

{

return false;

}

}

return true;

}

private void ClearRow(int r)

{

for (int c = 0; c < Columns; c++)

{

grid[r, c] = 0;

}

}

private void MoveRowDown(int r, int numRows)

{

for (int c = 0; c < Columns; c++)

{

grid[r + numRows, c] = grid[r, c];

grid[r, c] = 0;

}

}

public int ClearFullRows()

{

int cleared = 0;

for (int r = Rows - 1; r >= 0; r--)

{

if (IsRowFull(r))

{

ClearRow(r);

cleared++;

}

else if (cleared > 0)

{

MoveRowDown(r, cleared);

}

}

return cleared;

}

# Заключение

В процессе разработки было воплощено множество механик и изучено несколько полезных способов реализации игровых решений для дальнейших проектов. В частности, помимо основного поля были добавлены различные меню:

* главное меню;
* меню обучения;
* меню паузы;
* меню настроек;
* таблица лидеров.

А также была создана новая механика сложности. Из-за нехватки времени не удалось реализовать музыкальное сопровождение для игры, несмотря на то что аудиофайлы уже находятся в проекте. В ближайшем будущем музыка и звуки будут добавлены. Также планируется создание соревновательного режима как в игре Tetris 99 от Nintendo, а также реализация темной темы для светочувствительных пользователей.

Проведя опрос, был составлен список положительных и негативных аспектов игры:

Достоинства:

* красивый лаконичный интерфейс, приятное цветовое решение;
* возможность изменения сложности игрового процесса;
* фиксация результата (рекорда) в таблице лидеров под собственным именем;
* понятное, интуитивное управление;
* плавное перемещение фигур по стакану;
* наличие простого обучения.

Недостатки:

* небольшие задержки при нажатии;
* отсутствие звукового сопровождения;
* отсутствие популярной темной темы в приложении;
* отсутствие кооперативного режима игры;
* иногда долго не появляются конкретные фигуры.

# Список литературы

1. Юдин Никита. Листинг приложения [Электронный ресурс]. – режим доступа: <https://github.com/yudinikita/tetris-oop>
2. OttoBotCode. [Электронный ресурс]. – режим доступа: <https://www.youtube.com/watch?v=jcUctrLC-7M>
3. Сайт Metanit [Электронный ресурс]. – режим доступа: <https://metanit.com/sharp/wpf/1.php>
4. Справочник и самоучитель WPF [Электронный ресурс]. – режим доступа: <https://wpf-tutorial.com/>
5. Elshad Safarov. [Электронный ресурс]. – режим доступа: [https://yandex.ru/games/app/376930](https://yandex.ru/games/app/376930%23info)
6. Официальный продукт Tetris 99 от Nintendo. [Электронный ресурс]. – режим доступа: https://www.nintendo.com/us/store/products/tetris-99-switch/?srsltid=AfmBOoruHuB1xAYACgL4KflcFehBaiqBUQWTdIc9gmjNJW1Dq1HL35gO