*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](#_Toc190344033)

[1 постановка и проектирование задачи 5](#_Toc190344034)

[1.1 Описание программы 5](#_Toc190344035)

[1.2 Функциональные возможности 5](#_Toc190344036)

[1.3 Дизайн-макеты интерфейса 6](#_Toc190344037)

[2 кодирование программного продукта 10](#_Toc190344038)

[2.1 Игровой цикл 10](#_Toc190344039)

[2.2 Методы работы с конфигом 11](#_Toc190344040)

[Заключение 13](#_Toc190344041)

[Список литературы 14](#_Toc190344042)

# 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).

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

Были рассмотрены следующие реализации игры Tetris:

1. работа Никиты Юдина [№] на GitHub
2. игра Elshad Safarov [№] в сервисе Яндекс Игры
3. игра OttoBot [№] 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:

* приятный визуал;
* быстрый отклик.

Недостатки:

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

Преимущества в игре OttoBot:

* приятный визуал;
* наличие удерживаемого пространства для фигур;
* масштабируемость.

Недостатки:

* отсутствие дополнительных механик.

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

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.

## Дизайн-макеты интерфейса

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

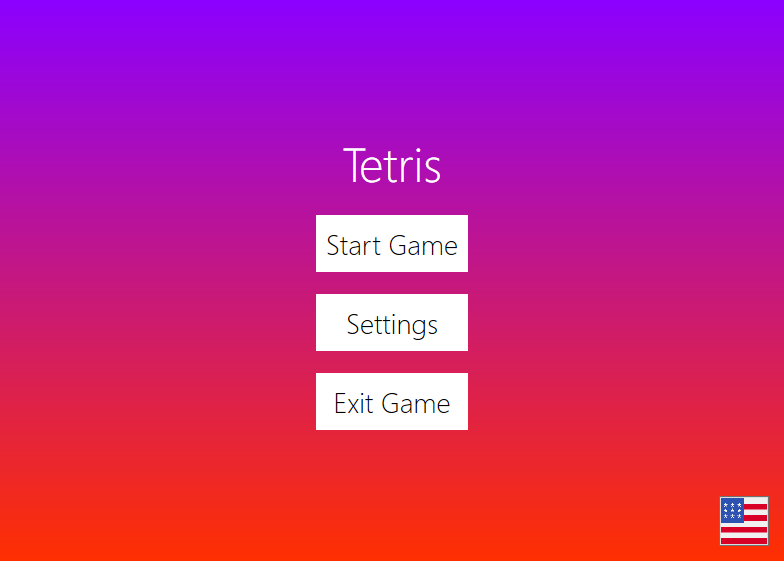


Рисунок 1 – Дизайн-макет англоязычного интерфейса главного меню.

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

Рисунок 2 – Дизайн-макет русскоязычного интерфейса главного меню.

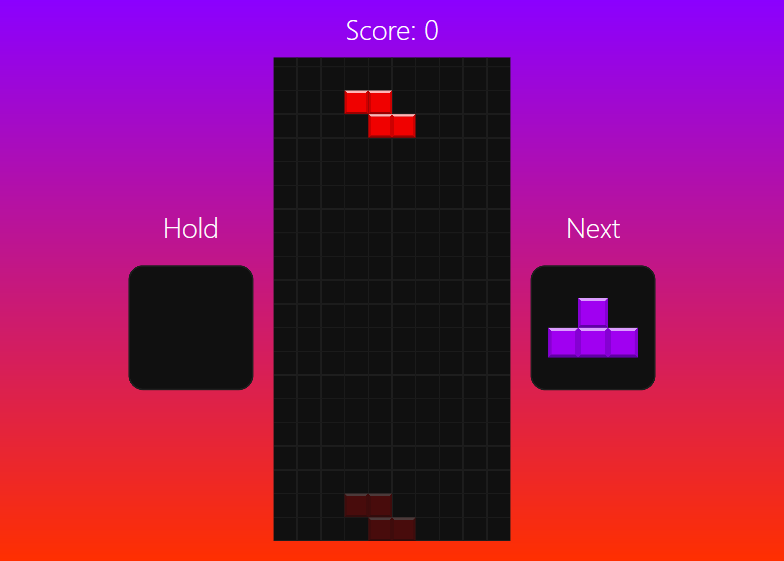
Дизайн-макет англоязычного интерфейса игрового поля представлен на рисунке 3.

Рисунок 3 – Дизайн-макет англоязычного интерфейса игрового поля.

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

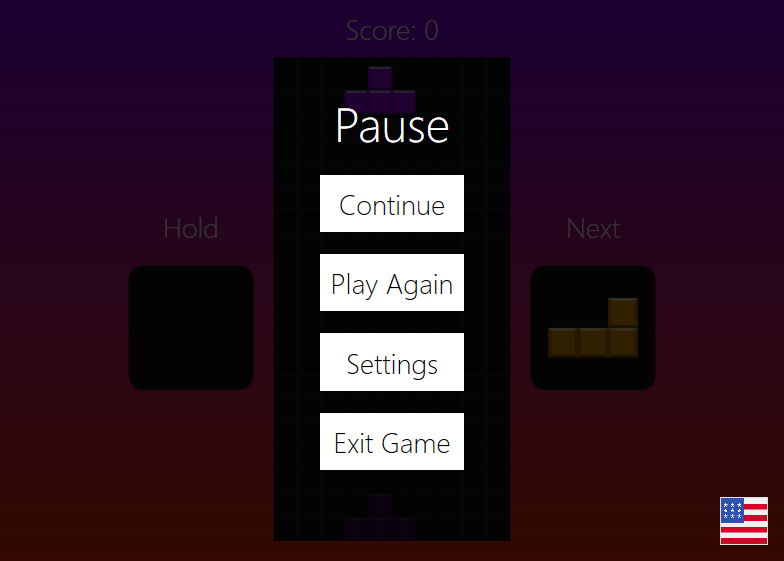


Рисунок 4 – Дизайн-макет англоязычного интерфейса меню паузы.

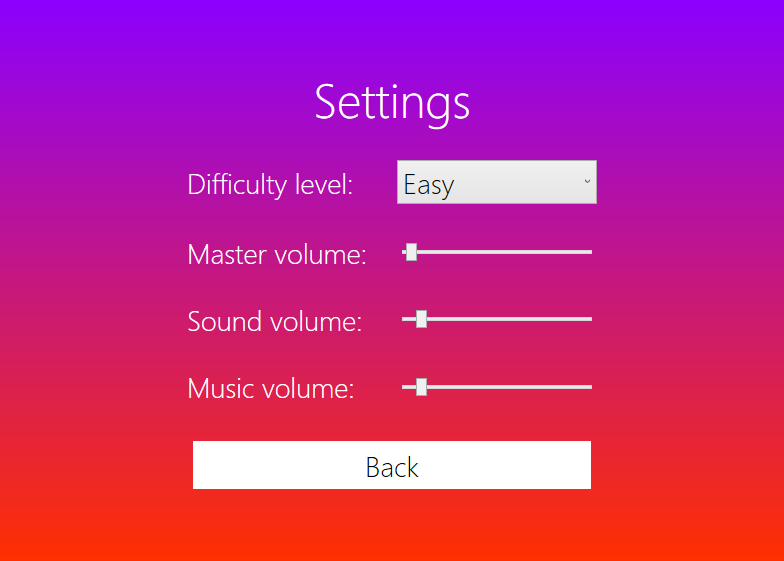
Дизайн-макет англоязычного интерфейса меню настроек представлен на рисунке 5.

Рисунок 5 – Дизайн-макет англоязычного интерфейса меню настроек.

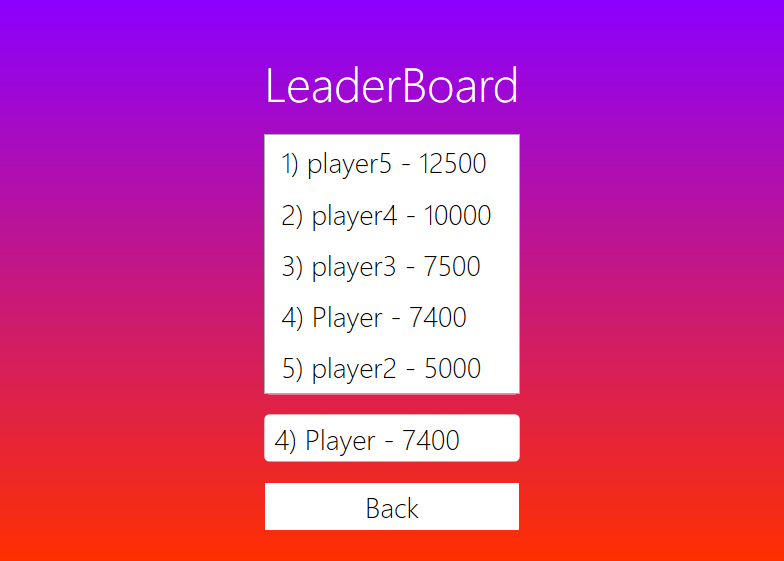
Дизайн-макет англоязычного интерфейса таблицы лидеров представлен на рисунке 6.

Рисунок 6 – Дизайн-макет англоязычного интерфейса таблицы лидеров.

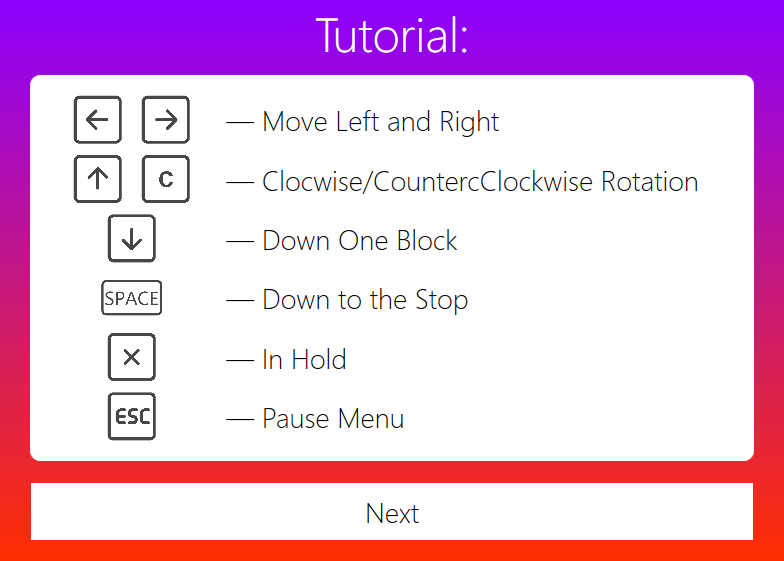
Дизайн-макет англоязычного интерфейса обучения представлен на рисунке 7.

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

//Высчитывание результатов и внесение в таблицу

User player = new User(LeaderBoard.Name, gameState.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 playerInList, out int index);

if (player.Score > playerInList.Score)

{

LeaderBoard.RewriteLineInList(index, player);

}

}

catch (Exception ex)

{

LeaderBoard.AddLineInList(player);

}

LeaderBoard.UpdateLeaderBoardList();

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

{

TypeConverter converter = TypeDescriptor.GetConverter(typeof(T));

if (converter != null && converter.CanConvertFrom(typeof(string)))

{

return (T)converter.ConvertFromString(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);

}

# Заключение

В процессе разработки я многому научился и много что воплотил. В частности, помимо основного поля я добавил различные меню:

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

А также добавил новую механику сложности. Из-за нехватки времени я не успел добавить музыку и звуки в игру, несмотря на то что аудиофайлы уже находятся в проекте. Из-за личной любви к проекту я планирую в будущем всё-таки добавить звуки в игру.

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

Достоинства:

* красивый интерфейс;
* понятное управление;
* наличие обучения.

Недостатки:

* небольшие задержки при нажатии;
* иногда долго не появляются конкретные фигуры.

# Список литературы

1. Работа Никиты Юдина – <https://github.com/yudinikita/tetris-oop>
2. Видео урок OttoBot – <https://www.youtube.com/watch?v=jcUctrLC-7M>
3. Сайт Metanit – <https://metanit.com/sharp/wpf/1.php>
4. Сайт WPF tutorial – <https://wpf-tutorial.com/>
5. Игра Elshad Safarov – [https://yandex.ru/games/app/376930](https://yandex.ru/games/app/376930%23info)