*Министерство образования и молодежной политики   
Свердловской области*

*ГАПОУ СО «Асбестовский политехникум»*

*Специальность 09.02.07 «Информационные системы и программирование»*

*ТЕМА КУРСОВОГО*

*ПРОЕКТИРОВАНИЯ*

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

*КП ИС 2025.01 ПЗ*

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

*2025*

*СОДЕРЖАНИЕ*

[введение 3](#_Toc188277250)

[1 постановка и проектирование задачи 5](#_Toc188277251)

[1.1 Описание программы 5](#_Toc188277252)

[1.2 Функциональные возможности 5](#_Toc188277253)

[1.3 Прототипы интерфейса 6](#_Toc188277254)

[2 кодирование программного продукта 8](#_Toc188277255)

[2.1 Вывод всех опубликованных объявлений 8](#_Toc188277256)

[2.2 Вывод объявлений по критериям фильтрации и поиска 9](#_Toc188277257)

[Заключение 10](#_Toc188277258)

[Список литературы 11](#_Toc188277259)

# введение

Объект исследования: процесс проектирования и разработки цифровой игры Tetris.

Предмет исследования: пользовательский интерфейс WPF

Цель проекта:

* изучить функционал интерфейса WPF;
* разработать программный продукт Тетрис;
* научиться создавать готовый продукт.

Для достижения поставленных целей сформулированы следующие задачи:

* найти хорошие информационные источники и учебники WPF;
* воплотить оригинальный геймплей Тетрис;
* реализовать различные механики и меню;
* выпустить проект в релиз.

Для меня практическая значимость этой работы состоит в том, что на основе данной игры, я улучшу свои навыки в разработке программных модулей.

Проанализировав готовые варианты Tetris, я обратил внимание, что много любительских работ имеют посредственную визуальную составляющую и скучный, однообразный геймплей. Однако хочется выделить работу Никиты Юдина(1) на GitHub, где есть возможность менять графику, вводить чит-коды, а также менять способ перемещения. А еще видео от OttoBot(2), у которого я перенял основной геймплей и визуальную часть.

# постановка и проектирование задачи

## Описание программы

Программный продукт предназначен для совершеннолетних и несовершеннолетних пользователей, желающих провести свой досуг, играя в игры. Игры должна бросать игрокам вызов – набрать наибольшее количество очков. В зависимости от игрового опыта для разных игроков присутствует разные уровни сложности. Приложение предоставляет возможность просматривать рекорды пользователей через специальную таблицу лидеров. После выхода из игры все результаты и настройки сохраняются.

## Функциональные возможности

В приложении необходимо реализовать функционал манипуляции фигурами.

Функции манипуляции:

* 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 интерфейса главного меню представлен на рисунке 1.

Рисунок 1 – Дизайн-макет интерфейса главного меню.

Дизайн-макет интерфейса игрового поля представлен на рисунке 2.

Рисунок 2 – Дизайн-макет интерфейса игрового поля.

Дизайн-макет интерфейса меню паузы представлен на рисунке 3.

Рисунок 3 – Дизайн-макет интерфейса меню паузы.

Дизайн-макет интерфейса меню настроек представлен на рисунке 4.

Рисунок 4 – Дизайн-макет интерфейса меню настроек.

Дизайн-макет интерфейса таблицы лидеров представлен на рисунке 5.

Рисунок 5 – Дизайн-макет интерфейса таблицы лидеров.

Дизайн-макет интерфейса обучения представлен на рисунке 6.

Рисунок 6 – Дизайн-макет интерфейса обучения.

# кодирование программного продукта

## Метод игрового забега

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;

}

# 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;
* Таблицу 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 я составил список положительных и негативных аспектов моей игры:

Достоинства:

* Нормально;

Недостатки:

* Небольшие задержки при нажатии.

# Список литературы

Информация, размещенная на странице в Интернете, является электронным ресурсом удаленного доступа. Описание электронных ресурсов локального и удаленного доступа регулируется ГОСТ 7.82 — 2001 «Библиографическая запись. Библиографическое описание электронных ресурсов». Вот ссылка на выдержки из документа и примеры: <http://new.gramota.ru/spravka/letters/85-rubric-79>

1. Работа Никиты Юдина — <https://github.com/yudinikita/tetris-oop>
2. Видео урок OttoBot — <https://www.youtube.com/watch?v=jcUctrLC-7M>
3. Учебник WPF —