

Breakthrough



Содержание

01 Титул и план

02 Требования и архитектура

03 Правила и ввод

04 Persistence и JSON

05 Game Core

06 UI и демо



01

Титул и план

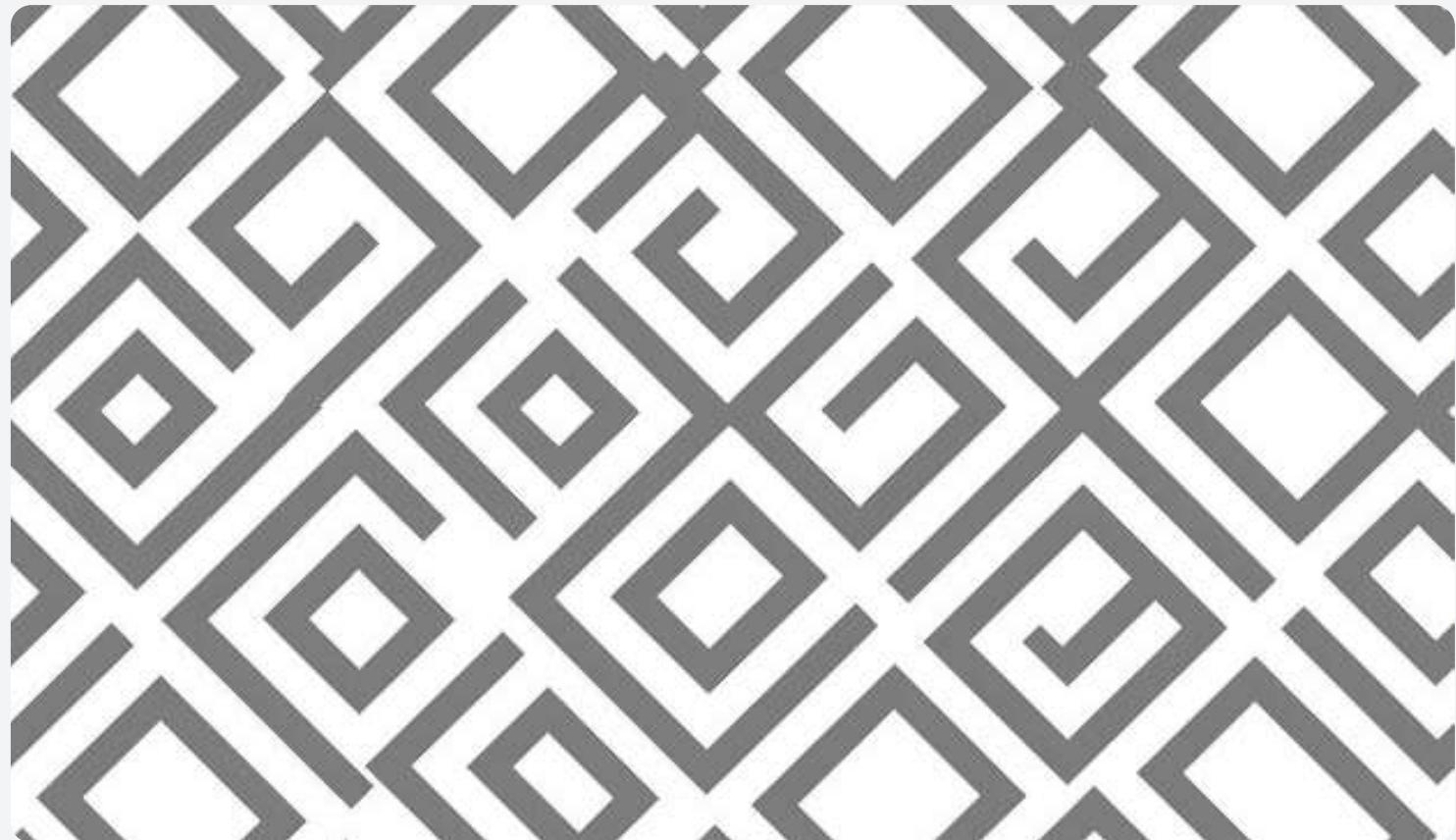
BREAKTHROUGH

Консольная реализация настольной
игры, .NET 9

 Команда из 3-х участников

 Кроссплатформенность: Win / macOS /
Linux

 Язык: C# (.NET 9)



План Доклада

1

2

3

4

5

6

7

Требования и запуск Структура проекта

Правила и ввод

UI и команды

Persistence

Демо

Вклад участников



02

Требования и
архитектура

Требования и Запуск

Предусловия

- ✓ .NET SDK 9
- ☐ Кроссплатформенность: Windows, macOS, Linux

Команды Запуска

```
# Проверка версии  
$ dotnet --version
```

```
# Сборка и запуск (Windows)  
$ dotnet build  
$ dotnet run --project  
Breakthrough.Console\Breakthrough.Console.csproj
```

Структура Репозитория



Breakthrough.Console

Главный проект, точка входа.



Domain

Логика игры: Board, Game, Move, Rules.



UI

Меню, отрисовка, ввод.



Persistence

Сохранения и таблица рекордов (JSON).

Breakthrough.sln

Корневой файл решения

Заметки докладчика: Показать дерево каталогов, объяснить разделение обязанностей, подчеркнуть читаемость архитектуры.



03

Правила и ввод

Правила Игры

- ▣ Доска 8×8; по 16 фишек (2 ряда) у каждого игрока.
- ↑ Белые ходят снизу вверх; чёрные — сверху вниз; и есть возможностьходить по диагонали.
- Ход: на 1 клетку вперёд (если пусто). Взятие: на 1 клеткудиагонально (если есть фишкаСоперника).
- ▣ Победа: любая фишкадостигает противоположногокрая.

	A	B	C	D	E	F	G	H	
8	■	■	■	■	■	■	■	■	8
7	■	■	■	■	■	■	■	■	7
6									6
5									5
4									4
3									3
2	○	○	○	○	○	○	○	○	2
1	○	○	○	○	○	○	○	○	1
	A	B	C	D	E	F	G	H	

Формат Ввода и Команды

Формат Хода

a2 a3

```
Breakthrough (Прорыв)
Ход: white (White) | Ходов: 0
Команды: :menu, :exit, :help, :moves, :save [файл], :load [файл], :scores
Ход: например a2 a3
```

```
a b c d e f g h
8 B B B B B B B B 8
7 B B B B B B B B 7
6 . . . . . . . . 6
5 . . . . . . . . 5
4 . . . . . . . . 4
3 . . . . . . . . 3
2 W W W W W W W W 2
1 W W W W W W W W 1
 a b c d e f g h
```

> a2 a3

Команды

:help

:menu

:exit

:moves

:save [файл]

:load [файл]

:scores

...



04

Persistence и JSON

Сохранения и Таблица Рекордов (JSON)

Пример `save.json`

```
{ "WhiteName": "white",
  "BlackName": "black", "Turn": 0,
  "PlyCount": 2, "Rows": [
    "BBBBBBBBBB", "BBB.BBBB",
    "...B....", "...", "...", "W.....",
    ".WWWWWWWW",
    "WWWWWWWW" ] }
```

Пример `scores.json`

```
[ {"PlayerName": "Alex", "Moves": 12, "Date": "..."},  
 {"PlayerName": "Sam", "Moves": 15, "Date": "..."},  
 {"PlayerName": "Kim", "Moves": 18, "Date": "..."} ]
```

```
Breakthrough (Прорыв)
Ход: white (White) | Ходов: 0
Команды: :menu, :exit, :help, :moves, :save [файл], :load [файл], :scores
Ход: например a2 a3
```

```
   a b c d e f g h
8 B B B B B B B B 8
7 B B B B B B B B 7
6 . . . . . . . . 6
5 . . . . . . . . 5
4 . . . . . . . . 4
3 . . . . . . . . 3
2 W W W W W W W W 2
1 W W W W W W W W 1
   a b c d e f g h
```

```
> a2 a3
```

Визуализация в Консоли

- А ASCII-графика: `W` — белые, `B` — чёрные, `.` — пустая.
- Цветовая подсветка фигур (при поддержке терминалом).
- Подсказки и ошибки: понятные сообщения для пользователя.



05

Game Core

Участник 1: Game Core

Реализация логики игры



Правила

Ход вперёд, взятие по диагонали,
проверка победы.



Модели

Board.cs, Game.cs, Move.cs, Piece.cs.



Механика

Генерация доступных ходов для
команды :moves.

Примеры Ходов

Корректные

a2 a3 — свободная клетка вперёд.

b2 a3 — взятие по диагонали.

Некорректные

a2 a4 — шаг больше чем на 1.

a2 a3 — если a3 занята.



06

UI и демо

Участник 2: UI & UX

- ☰ Главное меню: новая игра, правила, рекорды, выход.
- ❖ Модули: ConsoleMenu.cs, ConsoleRenderer.cs, InputReader.cs, Program.cs.
- 👤 Ответственность: удобство ввода, подсказки, ошибки.

==== Breakthrough (Прорыв) ====

- 1) Новая игра
- 2) Как играть
- 3) Таблица рекордов
- 4) Выход



Демо UI: Шаг за Шагом

1

`dotnet run ...` — запуск приложения.

2

Выбрать "Новая игра", ввести имена.

3

Показать ввод хода `a2 a3`.

4

Показать `:moves`, `:help`, `:save demo.json`.



07

Persistence и
документация

Участник 3: Persistence & Docs

Сохранение

- FileGameStorage.cs (IGameStorage)
- FileScoreStorage.cs (IScoreStorage)
- Автосохранение после победы

Документация

- Подробный README
- Инструкция по запуску
- Примеры команд

Заметки докладчика: Показать фрагмент README, подчеркнуть кроссплатформенные пути.

Сценарии Save / Load / Scores



:save / :load

Сохранение и загрузка партии. Файл по умолчанию `save.json`.



Автозапись

После победы результат автоматически добавляется в `scores.json`.



:scores

Просмотр топ-результатов. Лучший результат — меньше ходов.

Заметки докладчика: Продемонстрировать цикл сохранения-загрузки, показать структуру scores.json, объяснить критерий лучшего результата.



08

Завершение

СПАСИБО!

Вопросы?



Заметки докладчика: Перечислить три основных достижения проекта, напомнить о кроссплатформенности и open-source.