

Консольная шахматная игра с ботом

1. Особенности:

- Отображение шахматной доски в ASCII-графике
- Полная поддержка всех шахматных фигур
- Обработка и проверка пользовательских ходов
- Случайные ходы со стороны бота
- Проверка шаха и мата
- Сохранение истории ходов в файл

2. Структура проекта

2.1. Основные модули и классы

Метакласс PieceMeta

- Регистрация всех подклассов фигур в PieceMeta.registry
- Используется для автоматической генерации доступных фигур

Класс BasePiece

- Абстрактный базовый класс для всех шахматных фигур
- Атрибуты:
 - symbol — символ для отображения (например, 'P' для пешки)
 - color — 'white' или 'black'
- Методы:
 - moves(position, board) — абстрактный метод для возвращения доступных ходов
 - __str__() — возвращает символ фигуры в верхнем/нижнем регистре в зависимости от цвета

Фигуры (наследники BasePiece)

Каждая фигура реализует метод `moves(position, board)` и определяет свою механику движения:

- Pawn (пешка)
- Rook (ладья)
- Knight (конь)
- Bishop (слон)
- Queen (ферзь)
- King (король)

Функция `linear_moves()`

- Универсальный помощник для расчета линейных ходов
- Используется слоном, ладьей и ферзём

3. Игровая логика

Класс `Game`

Основной класс, управляющий шахматной логикой.

3.1. Атрибуты:

- `board` — двумерный массив (8x8), содержащий фигуры или `None`
- `history` — список строк с описанием всех сделанных ходов
- `turn` — текущий цвет, который должен ходить ('white' или 'black')

3.2. Методы:

- `init_board()`
 - Устанавливает начальное положение фигур на доске
- `print_board()`
 - Печатает шахматную доску в терминале
- `move_piece(from_pos, to_pos)`

- Перемещает фигуру, обновляет board, сохраняет ход в history
- Форматирует описание хода, включая взятие фигур
- **get_all_valid_moves(color)**
 - Возвращает все возможные ходы для указанного цвета
 - Игнорирует ходы, которые оставляют короля под шахом
- **in_check(color, board)**
 - Проверяет, находится ли король указанного цвета под атакой на данной доске
- **is_checkmate(color)**
 - Проверяет, является ли текущая ситуация матом
- **user_turn()**
 - Ожидает ввода пользователя
 - Парсит и проверяет корректность хода
 - Ввод в формате: e2 e4
 - Обрабатывает команду help
- **bot_turn()**
 - Делает случайный допустимый ход за чёрных
- **save_history()**
 - Сохраняет историю всех ходов в файл game_history.txt (в режиме дозаписи)
- **play()**
 - Главный цикл игры
 - Чередует ходы пользователя и бота
 - Отображает доску и историю после каждого хода
 - Завершает игру при мате, выводит победителя и сохраняет историю

4. Интерфейс

Пользовательский ввод

- Формат хода: <буква><цифра> <буква><цифра> (например, e2 e4)
- Дополнительная команда: help

Вывод

- ASCII-доска
- История ходов
- Сообщения о шахе/мате и победителе

5. Сохранение данных

- Все ходы сохраняются в файл game_history.txt с пометкой "New Game" перед началом каждой партии.

6. Дополнительные требования

- Поддержка шаха и мата
- Игровая логика не допускает ходов, оставляющих короля под шахом
- Реализация всех фигур и их правил хода
- Поддержка базового ИИ (рандомные ходы)
-