

Шахматы

1. Особенности

- Отображение шахматной доски в ASCII-графике.
- Поддержка всех шахматных фигур и их правил хода.
- Проверка корректности пользовательских ходов.
- Случайные ходы со стороны бота.
- Проверка шаха и мата.
- Сохранение истории ходов в файл.

2. Структура проекта

2.1. Основные структуры данных

- **Фигура** — кортеж или словарь с ключами:
 - `symbol` — символ фигуры (например 'P' для пешки).
 - `color` — 'white' или 'black'.
- **Доска** — двумерный массив 8x8, в котором хранится либо фигура, либо None.
- **История** — список строк, где каждый элемент описывает сделанный ход.
- **Состояние игры** — словарь с ключами:
 - `board` — текущее состояние доски.
 - `history` — список ходов.
 - `turn` — чей ход ('white' или 'black').

2.2. Основные функции

Работа с доской

- `init_board() -> list[list]`
Создаёт и возвращает начальное расположение фигур.

- `print_board(board: list[list]) -> None`
Отображает доску в ASCII-графике.

Ходы

- `get_moves(piece: dict, position: tuple[int,int], board: list[list]) -> list[tuple]`
Возвращает все возможные ходы для фигуры на указанной позиции.
- `linear_moves(position, board, directions) -> list[tuple]`
Универсальная функция для расчёта ходов по линиям (используется для ладьи, слона и ферзя).
- `move_piece(state: dict, from_pos: tuple, to_pos: tuple) -> dict`
Перемещает фигуру, обновляет доску и историю.
- `get_all_valid_moves(board: list[list], color: str) -> list[tuple]`
Возвращает все допустимые ходы для цвета, исключая те, что оставляют короля под шахом.

Проверка шаха и мата

- `in_check(board: list[list], color: str) -> bool`
Проверяет, находится ли король указанного цвета под атакой.
- `is_checkmate(board: list[list], color: str) -> bool`
Проверяет, является ли ситуация матом.

Игрок и бот

- `user_turn(state: dict) -> dict`
Обрабатывает ввод игрока (e2 e4), проверяет корректность и возвращает обновлённое состояние.
- `bot_turn(state: dict) -> dict`
Делает случайный допустимый ход за бота.

История

- `save_history(history: list[str]) -> None`
Сохраняет ходы в файл `game_history.txt`, добавляя пометку "New Game".

Главный цикл

- `play()`
Запускает игру: чередует ходы пользователя и бота, печатает доску и историю, проверяет шах/мат, завершает игру.

3. Интерфейс

- Ввод игрока:
 - Формат: <буква><цифра> <буква><цифра> (например, e2 e4).
 - Дополнительная команда: `help`.
- Вывод:
 - ASCII-доска.
 - История ходов.
 - Сообщения о шахе, мате и победителе.

4. Сохранение данных

- Все ходы сохраняются в `game_history.txt`.
- Перед новой партией добавляется строка "New Game".

5. Дополнительные требования

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