

Documento de Análisis y Diseño del Programa

Descripción del Programa

El objetivo de este programa es simular un tablero de ajedrez y calcular los movimientos válidos para un caballo desde una posición dada. El usuario podrá inicializar un tablero vacío, agregar piezas en posiciones específicas, ingresar la posición del caballo a evaluar y obtener los movimientos válidos para ese caballo.

Acciones del Programa

1. Inicializar un tablero de ajedrez vacío: La aplicación permitirá inicializar un tablero vacío de 8x8.
2. Agregar piezas al tablero: El usuario podrá agregar una cantidad específica de piezas al tablero, especificando el tipo de pieza, su color y su posición inicial en notación de ajedrez.
3. Validar posiciones: El programa verificará que las posiciones ingresadas por el usuario sean válidas y no estén ocupadas por otras piezas.
4. Ingresar y validar posición del caballo: El usuario podrá ingresar la posición del caballo a evaluar, y el programa validará que la posición sea correcta.
5. Calcular y mostrar movimientos válidos del caballo: El programa calculará todos los movimientos válidos del caballo desde la posición dada, teniendo en cuenta las piezas enemigas en el tablero.
6. Imprimir estado actual del tablero: Se imprimirá el estado actual del tablero de ajedrez con un patrón de colores alternos para las casillas.

Datos de Entrada

1. Cantidad de piezas a agregar (`N`).
2. Tipo de cada pieza (por ejemplo, alfil, peón, torre, etc.).
3. Color de cada pieza (blanco o negro).
4. Posición inicial de cada pieza (en notación de ajedrez, e.g., `a1`, `e4`).
5. Color del caballo a evaluar.
6. Posición inicial del caballo.

Variables

Variables Principales

1. ``board``: Representa el tablero de ajedrez como una matriz 2D (lista de listas) donde cada celda contiene una cadena que representa una pieza o está vacía.
 - Tipo de dato: ``list[list[str]]``
2. ``num_piezas``: Número de piezas a agregar al tablero.
 - Tipo de dato: ``int``
3. ``piece``: Tipo de pieza que se va a agregar al tablero.
 - Tipo de dato: ``str``
4. ``color``: Color de la pieza que se va a agregar al tablero.
 - Tipo de dato: ``str``
5. ``position``: Posición de la pieza en notación de ajedrez.
 - Tipo de dato: ``str``
6. ``moves``: Lista de movimientos válidos para el caballo desde una posición dada.
 - Tipo de dato: ``list[tuple[int, int]]`` (índices de matriz)
7. ``new_x``: Coordenada x de una nueva posición en el tablero.
 - Tipo de dato: ``int``
8. ``new_y``: Coordenada y de una nueva posición en el tablero.
 - Tipo de dato: ``int``
9. ``valid_moves``: Lista de movimientos válidos para el caballo, incluyendo información sobre las piezas presentes en esas posiciones.
 - Tipo de dato: ``list[tuple[tuple[int, int], str]]`` (índices de matriz y descripción del movimiento)
10. ``piezas``: Diccionario que mapea las representaciones de piezas de ajedrez a sus respectivos caracteres Unicode.
 - Tipo de dato: ``dict[str, str]``

Condiciones o Restricciones

1. Las posiciones deben estar dentro de los límites del tablero (8x8).
2. Las casillas no deben estar ocupadas por otra pieza al agregar nuevas piezas.
3. El caballo no debe moverse a una casilla ocupada por una pieza del mismo color.
4. Validar la notación de ajedrez ingresada por el usuario para asegurar su corrección.
5. El cálculo de los movimientos del caballo debe tener en cuenta la "L" de los posibles movimientos del caballo en ajedrez.

Funciones

```
def crear_tablero() -> list[list[str]]:
def posicion_a_indices(position:str) -> tuple[int, int]:
def indices_a_posicion(indices) -> str:
def validar_posicion(position:str) -> bool:
def agregar_piezas(board:list[list[str]], num_piezas:int) -> None:
def evaluate_knight_moves(board:list[list[str]], posicion_caballo:str) ->
list:

def imprimir_tablero(board:list[list[str]]) -> None:
```

-