Universidad Rafael Landivar
Pensamiento Computacional
Sección 04
Ing. Luis Pedro Ovalle Arrecis

PROYECTO NO.2
MOVIMIENTOS DE CABALLO EN AJEDREZ

María Alejandra Sarahi Santos Pretzanzin 1110824

Guatemala 20 de mayo de 2024

¿Qué acciones debe poder hacer su programa?

Solicitar Cantidad de Piezas

El programa pide al usuario ingresar el número de piezas de ajedrez que desea colocar en el tablero, asegurándose de que sea un número positivo.

Ingresar Piezas en el Tablero

Para cada pieza, el programa solicita:

- El tipo de pieza, por ejemplo: peón, caballo, alfil, torre, dama, rey.
- El color de la pieza: blanco o negro.
- La posición en el tablero utilizando la notación estándar de ajedrez, por ejemplo: a1, e4.
- El programa valida que el tipo de pieza sea correcto, que el color sea blanco o negro, y que la posición sea válida y esté libre en el tablero.

Ingresar Pieza Caballo a Observar

El usuario especifica la posición inicial del caballo que se va a evaluar, con las mismas validaciones que para las otras piezas, pero asegurando que el tipo de pieza sea un caballo.

Calcular y mostrar Todos los Posibles Movimientos válidos del Caballo

- Basándose en la posición inicial del caballo, el programa calcula todos los posibles movimientos siguiendo las reglas de movimiento en "L" del caballo en ajedrez.
- Para cada movimiento posible, el programa verifica si la casilla está vacía o contiene una pieza del equipo contrario, y muestra esta información.

Imprimir el Tablero de Ajedrez

Tras ingresar todas las piezas y antes de mostrar los movimientos del caballo, el programa imprime el estado actual del tablero. Utiliza colores alternos para diferenciar las casillas y coloca las piezas en sus respectivas posiciones.

¿Con qué datos va a trabajar? ¿Qué información debe pedir al usuario?

Defina sus datos de entrada.

1. Cantidad de Piezas:
    - Descripción: Número entero que representa la cantidad de piezas de ajedrez que el usuario desea ingresar al tablero.
    - Tipo de Dato: Entero (`int`).
    - Validación: Debe ser un número positivo mayor que cero.
2. Tipo de Pieza:
    - Descripción: Tipo de la pieza de ajedrez que se va a colocar en el tablero.
    - Tipo de Dato: Cadena de texto (`string`).
    - Validación: Debe coincidir con uno de los tipos válidos (peon, caballo, alfil, torre, dama, rey). El programa verifica que la entrada pertenezca a esta lista.
3. Color de la Pieza:
    - Descripción: Color de la pieza de ajedrez.
    - Tipo de Dato: Cadena de texto (`string`).
    - Validación: Debe ser "blanco" o "negro".
4. Posición de la Pieza:
    - Descripción: Ubicación de la pieza en el tablero de ajedrez, usando la notación estándar del ajedrez (por ejemplo, a1, h8).
    - Tipo de Dato: Cadena de texto (`string`).
    - Validación: Debe ser una posición válida dentro del tablero (letras de 'a' a 'h' y números de '1' a '8') y la casilla debe estar libre.

Información adicional solicitada al usuario:

- Para la pieza del caballo a evaluar, se pide la misma información que para las otras piezas (tipo, color y posición), aunque el tipo ya está predefinido como caballo.
- Se asegura que el tipo sea caballo y que la casilla esté libre o contenga una pieza del equipo contrario para evaluar los posibles movimientos.

Cómo se le solicita la información al usuario:

- El programa utiliza mensajes claros y directos para solicitar cada dato, indicando los formatos o las opciones válidas.
- Se emplean bucles y condiciones para validar las entradas y asegurarse de que sean correctas antes de proceder con las siguientes acciones.

¿Qué variables utilizará para almacenar la información? Defina el nombre y el tipo de dato que utilizará para gestionar estos datos principales

Variables fundamentales para el funcionamiento

1. **`TAMANODELTABLERO`**
    - **Tipo de Dato:** Entero (`int`)
    - **Descripción:** Almacena el tamaño del tablero de ajedrez, que es un valor constante de 8 (8x8 casillas).
2. **`PIEZAS_VALIDAS`**
    - **Tipo de Dato:** Lista de cadenas (`list` de `str`)
    - **Descripción:** Contiene los tipos de piezas de ajedrez válidos que se pueden ingresar en el tablero (peon, caballo, alfil, torre, dama, rey).
3. **`tablero`**
    - **Tipo de Dato:** Lista de listas de cadenas (`list` de `list` de `str`)
    - **Descripción:** Representa el tablero de ajedrez, donde cada elemento es una cadena que puede ser vacía (indicando casilla vacía) o contener la abreviación de la pieza y su color.
4. **`cantidad_piezas`**
    - **Tipo de Dato:** Entero (`int`)
    - **Descripción:** Almacena la cantidad de piezas que el usuario desea ingresar al tablero. Se solicita al inicio del programa.
5. **`tipo`**
    - **Tipo de Dato:** Cadena (`str`)
    - **Descripción:** Almacena el tipo de la pieza que se está ingresando al tablero. Se verifica contra `PIEZAS_VALIDAS`.

6. **`color`**
    - **Tipo de Dato:** Cadena (`str`)
    - **Descripción:** Almacena el color de la pieza (blanco o negro) y se utiliza para verificar la colocación correcta en el tablero según las reglas del ajedrez.
7. **`posicion`**
    - **Tipo de Dato:** Cadena (`str`)
    - **Descripción:** Almacena la posición de la pieza en el tablero en notación estándar del ajedrez (ej., a1, e4). Se valida para asegurar que es una posición correcta dentro del tablero.
8. **`movimientos_caballo`**
    - **Tipo de Dato:** Lista de cadenas (`list` de `str`)
    - **Descripción:** Almacena las posiciones válidas a las que el caballo puede moverse desde su posición actual, según las reglas de movimiento de esta pieza en el ajedrez.
9. **`letra`**, **`numero`**
    - **Tipo de Dato:** Cadena (`str`) y Entero (`int`)
    - **Descripción:** Variables auxiliares utilizadas para descomponer la posición del tablero en componentes de columna (letra) y fila (número), facilitando la manipulación y validación de movimientos.

¿Qué condiciones o restricciones debe tomar en cuenta? ¿Qué cálculos debe hacer?

Condiciones y Restricciones:

1. **Validación de Tipo de Pieza:**
    - **Condición:** El tipo de pieza ingresado debe estar en la lista de tipos válidos (peon, caballo, alfil, torre, dama, rey).
    - **Restricción:** No se permite ingresar un tipo de pieza que no sea reconocido por el juego de ajedrez.
2. **Validación de Color de Pieza:**
    - **Condición:** El color ingresado debe ser 'blanco' o 'negro'.
    - **Restricción:** No se acepta ningún otro color fuera de estas dos opciones.
3. **Validación de Posición en el Tablero:**
    - **Condición:** La posición debe consistir en una letra de 'a' a 'h' y un número de '1' a '8'.

- **Restricción:** No se puede ingresar una posición fuera de estos límites o una posición que ya esté ocupada por otra pieza.

4. **Movimientos del Caballo:**
    - **Condición:** El caballo se mueve en forma de "L", lo cual significa dos casillas en una dirección y una en una dirección perpendicular.
    - **Restricción:** El caballo no puede moverse a una casilla ocupada por una pieza del mismo color.

Cálculos

1. **Cálculo de Posiciones Válidas para Movimientos del Caballo:**
    - **Descripción:** Se calculan todas las posibles posiciones a las que un caballo puede moverse desde su posición actual. Esto se logra sumando o restando 1 o 2 a las coordenadas actuales del caballo en formas que representan los movimientos en "L".
2. **Índices de Matriz para Manipulación del Tablero:**
    - **Descripción:** La posición ingresada por el usuario (como 'a1', 'e4', etc.) se convierte en índices de matriz para manipular el tablero. La letra se traduce a un índice de columna y el número a un índice de fila.
3. **Verificación de Casillas Libres o Enemigas:**
    - **Descripción:** Antes de añadir un movimiento a la lista de movimientos válidos del caballo, el programa verifica si la casilla objetivo está vacía o contiene una pieza del equipo contrario. Esto implica verificar el último carácter del contenido de la casilla en la matriz del tablero.
4. **Control de Entradas del Usuario:**
    - **Descripción:** Se implementan bucles y condiciones para asegurar que todas las entradas del usuario cumplan con las reglas definidas, solicitando reingresos en caso de errores.

¿Qué funciones implementará?

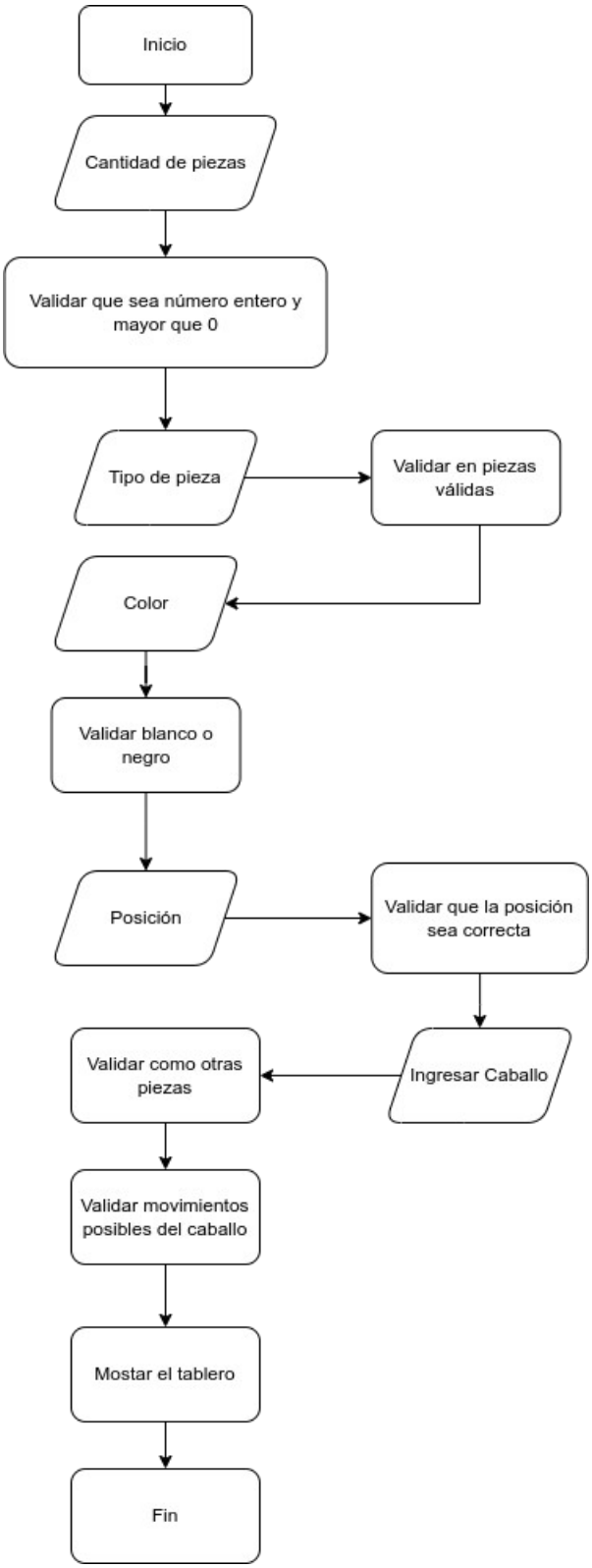
1. **`piezas_a_validar(tipo)`**
    a. **Propósito:** Verifica si el tipo de pieza ingresado por el usuario es válido.
    b. **Parámetros:** `tipo` (cadena) - El tipo de pieza ingresado por el usuario.
    c. **Retorno:** Booleano - `True` si el tipo de pieza es válido, `False` en caso contrario.
2. **`la_posicion_es_valida(posicion)`**
    - **Propósito:** Comprueba si la posición ingresada por el usuario es válida según la notación del ajedrez y está dentro de los límites del tablero.
    - **Parámetros:** `posicion` (cadena) - La posición en notación estándar del ajedrez.
    - **Retorno:** Booleano - `True` si la posición es válida, `False` en caso contrario.
3. **`color_a_validar(color)`**
    - **Propósito:** Verifica si el color ingresado por el usuario es uno de los dos colores permitidos en el ajedrez.
    - **Parámetros:** `color` (cadena) - El color ingresado por el usuario.
    - **Retorno:** Booleano - `True` si el color es válido, `False` en caso contrario.
4. **`tablero_juego(tablero)`**
    - **Propósito:** Imprime el tablero de ajedrez mostrando las piezas en sus respectivas posiciones, utilizando colores alternos para las casillas.
    - **Parámetros:** `tablero` (lista de listas) - La matriz que representa el tablero de ajedrez.
    - **Retorno:** Ninguno - Imprime el tablero directamente.
5. **`usuario_ingresar_la_pieza(es_caballo=False)`**
    - **Propósito:** Solicita al usuario ingresar detalles de una pieza para colocarla en el tablero, con una opción específica si la pieza es un caballo.
    - **Parámetros:** `es_caballo` (booleano) - Indica si la pieza a ingresar es el caballo.
    - **Retorno:** Tipo, color y posición de la pieza ingresada.
6. **`que_movimientos_del_caballo_hay(posicion, tablero, color)`**
    - **Propósito:** Calcula y devuelve una lista de todos los movimientos válidos para un caballo desde una posición dada, considerando las piezas en el tablero.
    - **Parámetros:**
        - `posicion` (cadena) - La posición inicial del caballo.

    - `tablero` (lista de listas) - El tablero de ajedrez.
    - `color` (cadena) - El color del caballo.
- **Retorno:** Lista - Lista de cadenas que representan las posiciones válidas de movimiento.

7. **`main()`**

- **Propósito:** Función principal que coordina la ejecución del programa, solicitando la entrada de piezas, colocándolas en el tablero, gestionando la entrada del caballo y mostrando sus movimientos válidos.
- **Parámetros:** Ninguno.
- **Retorno:** Ninguno - Controla el flujo principal del programa

Algoritmo que implementará en el programa, descrito mediante el Diagrama de Flujo elaborado en Draw.io, para mostrar la lógica de las diferentes acciones. Realizar un diagrama de flujo por cada función y un diagrama general del programa principal.



Referencias

tuplas en Python. (s/f). El Libro De Python. Recuperado el 11 de mayo de 2024, de https://ellibrodepython.com/tuplas-python

Associative arrays. (s/f). Brilliant.org. Recuperado el 11 de mayo de 2024, de https://brilliant.org/wiki/associative-arrays/

j2logo. (2020, abril 10). Convertir a mayúsculas y minúsculas en Python. J2LOGO. https://j2logo.com/python/convertir-a-mayusculas-y-minusculas-en-python/

Londoño, P. (2022, septiembre 28). Listas en Python: qué son, cómo crearlas y ordenarlas. Hubspot.es. https://blog.hubspot.es/website/lista-python

Python ord(). (s/f). Programiz.com. Recuperado el 11 de mayo de 2024, de https://www.programiz.com/python-programming/methods/built-in/ord

(S/f). Shiksha.com. Recuperado el 11 de mayo de 2024, de https://www.shiksha.com/online-courses/articles/ord-and-chr-functions-in-python/#:~:text=What%20is%20the%20ord(),the%20character)%20as%20an%20output.