

ENUNCIADO DEL PROYECTO 1

Entrega

Martes 7 de Octubre del 2014 (Martes de la Semana 5) antes de las 11:59 pm en el espacio de su grupo en Aula Virtual.

1. Descripción General

Para este proyecto, se debe entregar un programa que realice las siguientes operaciones sobre un tablero de ajedrez de 8x8: Verificar validez de movimiento, Hacer movimiento y advertir posible ataque enemigo. En este caso se hará una versión simplificada del juego de ajedrez que solamente tendrá en cuenta a los peones.

Siguiendo los principios de Encapsulamiento y Modularidad se debe implementar el TAD Tablero.

Su programa se ejecutará mediante el comando especificado a continuación. Su módulo principal debe tener el mismo nombre.

```
1 python3 chess.py <entrada> <salida>
```

Siendo <Entrada>el archivo que contiene las instrucciones que debe realizar el programa, y salida es el archivo en donde se escribirá la salida.

2. Modelo Abstracto

A continuación se muestra una especificación incompleta del TAD Tablero. El primer paso del proyecto es completar la especificación formal y sustituir la explicación en lenguaje natural de cada operación por la pre- y post-condición.

2.1. Especificación de TAD Tablero

```
1 Modelo de Representacion
2   tablero[8][8]; # Tablero.
3   ...
4
5 Invariante de Representacion
6   ...
7
8 Operaciones
9   load(in archivo,nombre: string, in/out tableros)
10      # Lee desde el archivo el tablero, y lo coloca en
11      # el diccionario junto con los otros tableros.
12
13   print(in nombre: string in tableros)
14      # Imprime en el tablero indicado en nombre
15
16   check(in origen,destino,nombre: string, in tableros)
17      # Imprime la palabra VALID o INVALID segun sea el caso.
18
19   move(in origen,destino,nombre: string, in/out tableros)
20      # Modifica el tablero indicado en el diccionario de tableros
21
22   checkThreat(in jugador: string, in tableros)
23      # Imprime la palabra YES o NO segun sea el caso.
```

2.2. Formato de Instrucciones

A continuación se coloca una tabla en donde se especifica la instrucciones que debe procesar el programa.

Sintaxis	Semántica
LOAD <archivo><nombre>	Lee desde el archivo el tablero en el formato especificado en Formato de Entrada
PRINT <nombre>	Imprime en el archivo de salida el tablero indicado.
CHECK <casilla origen><casilla destino><nombre>	Imprime la palabra VALID o INVALID según sea el caso.
MOVE <casilla origen><casilla destino><nombre>	Realiza el movimiento de la pieza indicada en caso de ser válido. Si el movimiento es inválido, imprime la palabra INVALID y no se realiza ningún movimiento en el tablero.
CHECK THREAT <jugador><nombre>	Imprime la palabra YES o NO según sea el caso. En jugador recibirá una P mayúscula para verificar si las blancas están amenazadas o una p minúscula para las negras.

2.3. Formato de Salida

Al imprimir los tableros de ajedrez, deberán estar encabezados por la palabra Tablero: seguido del nombre del tablero. Luego se imprime el tablero representando a los espacios vacíos con el caracter numeral (#), los peones de las blancas con la letra P mayúscula y los peones de las negras con la letra p minúscula. Al final se debe imprimir una secuencia de guiones con 8 caracteres de ancho. El tablero debe imprimirse imaginando que está jugando desde el lado de las blancas. A continuación se coloca un ejemplo en donde se imprime un tablero llamado Semifinal, y luego se imprime uno llamado Finales.

Tablero: Semifinal

```
#####  
p#p###p#  
#p#p####  
####pp#p  
###P####  
#P##P###  
P#P###PP  
#####
```

Tablero: Finales

```
#####  
p#p#ppp#  
#p#p####  
#####p  
#####  
#P##P###  
P#PP##PP  
#####
```

Para el caso de la instrucción CHECK, en la salida se debe imprimir la palabra VALID o INVALID correspondiente a un tablero válido o no válido respectivamente.

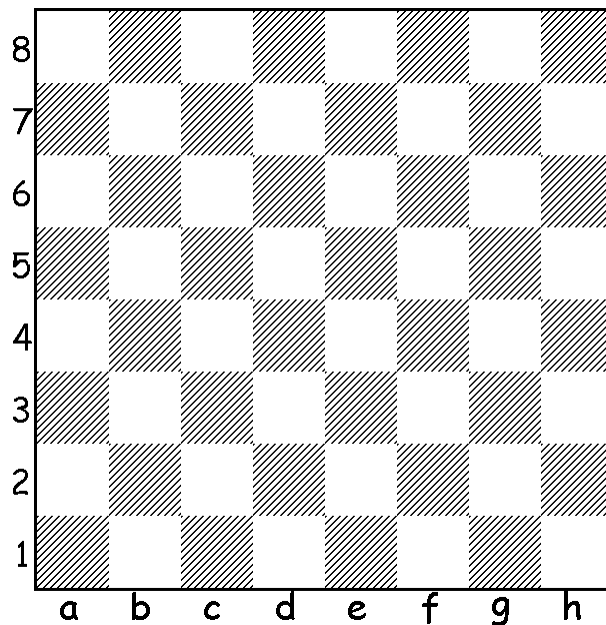
Para la instrucción CHECK THREAT, en la salida se imprimirá la palabra YES o NO en el de estar amenazado por las piezas enemigas. Estar amenazado significa que en un solo movimiento de las piezas enemigas, éstas son capaces de eliminar una de las piezas aliadas.

2.4. Formato de Entrada

Muy similar al formato de salida, el archivo de entrada sólo contendrá un tablero, el cual tiene una representación muy parecida al formato de salida. A continuación se coloca un ejemplo:

```
#####  
p#p#ppp#  
#p#p####  
#####p  
#####  
#P##P###  
P#PP##PP  
#####
```

Para el caso de las instrucción MOVE, en las que se recibe una casilla, a continuación se coloca una imagen de un tablero con los nombres de las casillas:



La entrada se dará con la primera letra en mayúscula seguido del numero correspondiente. Por ejemplo, la casilla A1 es la casilla de la esquina inferior izquierda.

2.5. Validez de los movimientos

El peón tiene dos tipos de movimientos válidos. Avanzar y Capturar. Avanzar consiste simplemente en 1 o 2 pasos en el eje de las columnas. El peón puede avanzar dos pasos únicamente cuando esta en su posición inicial (fila 2 para las blancas y fila 7 para las negras). De resto puede avanzar solamente 1 casilla.

El movimiento de captura se realiza es en diagonal hacia la columna inmediata, ya sea izquierda o derecha capturando una pieza adversaria.

Nótese que no se tomará en cuenta la captura del peón al paso.

Es decir, para saber si una jugada es válida (Función CHECK) se deben verificar las siguientes lista de condiciones:

- <Casilla origen>y <Casilla destino>son posiciones válidas en el tablero.
- En <Casilla origen>existe una pieza.
- <Casilla destino>está libre si es un movimiento en columna. Si es una captura, que la pieza que ocupa la casilla destino sea del color opuesto.
- No hay una pieza entre <Casilla origen>y <Casilla destino>.

- Todos los tableros dados tendrán los peones en posiciones válidas, es decir, no habrán piezas blancas en la fila 1, ni piezas negras en la fila 8.
- El movimiento debe cumplir con la explicación dada de un movimiento de peón.

3. Requerimientos de Entrega

1. La entrega consiste en los siguientes artefactos. Se recomienda elaborarlos en el orden propuesto.
 - Especificación completa con el modelo abstracto del TAD Tablero. Puede ser escrita a mano y escaneada, pero en todo caso debe ser perfectamente legible.
 - Especificación con el modelo concreto de su escogencia del TAD Tablero. Igual que el punto anterior.
 - Implementación del proyecto.
 - Archivo `readme.txt` con instrucciones breves para correr el proyecto, documentación de problemas y/o errores y cualquier otro comentario que considere pertinente.
2. Se evaluará positivamente la limpieza y buena estructura del código. En particular procure separar las operaciones del TAD de la lógica de entrada-salida de su programa. Se recomienda que la implementación del TAD Matriz esté en su propio archivo `.py`, distinto del módulo principal.
3. El código debe estar adecuadamente documentado, lo cual incluye nombre de variables y métodos auto-explicativos en la medida de lo posible. Recuerde que el propósito de la documentación no es repetir información ya expresa en el código.
4. La entrega del proyecto será directamente en el espacio de su grupo en el Aula Virtual. Deposite un archivo comprimido que contenga los archivos de código fuente, así como otros que sean necesarios para la ejecución de la aplicación. El nombre del archivo debe ser `Proy1-XX.tar.gz`, donde XX es el número de su grupo.
5. Todos sus archivos de código deben tener un encabezado con los nombres y números de carnet de ambos miembros del equipo.
6. La entrega del proyecto debe realizarse ANTES de la fecha y hora indicadas al inicio del enunciado. Se recomienda realizar la entrega con antelación y considerar imprevistos como fallas de conectividad.

Se proveerá una muestra de archivos de entrada y la salida esperada junto con este enunciado. Igualmente se adjuntará código con un ejemplo básico de lectura y escritura de archivos.

El no cumplimiento de los requerimientos podrá resultar en el rechazo de su entrega.

4. Preguntas y consultas

Se recomienda hacer preguntas con la mayor antelación posible. A fin de realizar las consultas de manera pública y persistente, coloque sus preguntas en el foro del Aula Virtual.