Universidad Simón Bolívar
Departamento de Computación y
Tecnología de la Información
CI-3825: Sistemas de Operación I

Prof. Fernando Torre Mora

Proyecto 1 Enero – Marzo 2024

Ajedrez Mágico Real

1 Introducción

En la saga *Harry Potter*, un juego recurrente es el Ajedrez Mágico, el cual se describe en los libros como un juego normal de ajedrez, pero en el cual las piezas son autoconscientes y a veces actúan contrario a los designios del jugador. Sin embargo, aún en estos casos, se comportan de acuerdo a las reglas del ajedrez (se mueven de la forma correcta).

Los designios del jugador son las jugadas que el jugador hace conscientemente. Las piezas actúan contrario a los designios del jugador de las siguientes maneras:

- Rehusándose a hacer la jugada indicada por el jugador (si el jugador los está enviando a una casilla "amenazada" o esta jugada deja "descuidada" otra pieza) dando objeciones vociferadas a menos que el jugador insista
- Exigiéndole verbalmente al jugador el ser usados si no están "cuidando" a otra pieza y no han sido movidos en mucho tiempo
- Reclamándole al jugador el no permitirles comer una pieza que está dentro de sus casillas "amenazadas" si el jugador pasa la oportunidad por alto por mucho tiempo
- Moverse sin permiso del jugador si pasa un tiempo todavía mayor en estos casos. Nótese que el jugador pierde el turno si no le da una contraorden a la pieza antes de que esta deje la casilla por completo.

Se le pide hacer una implementación del juego que funcione de estas maneras. Las piezas del oponente todas actuarán "contario a los designios". Para simplificar, sólo se le pide que represente dos tipos de piezas:

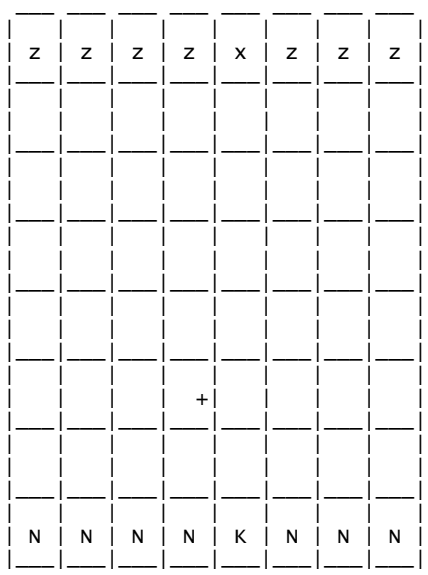
- Caballos, los cuales sólo pueden moverse en L's de 2 casillas hacia delante y 1 hacia un lado
- Y el rey de cada jugador, el cual sólo puede moverse 1 casilla por turno.

Las piezas deben funcionar igual sin importar si son del jugador o del oponente. (Se considerarán las piezas restantes para extra credit.)

Para una visualización posible de cómo podría funcionar, se sugiere ver Super Fun Action Chess.

2 Requerimientos del programa

Debe escribir un programa en C con interfaz basada en texto que permita jugar esta versión del ajedrez. El tablero de ajedrez debe imprimirse en un formato similar al siguiente



Donde

- N (abreviado de knight en inglés) representa los Caballos (♘) del jugador,
- K (abreviado de King en inglés) representa al Rey (♔) del jugador,
- z (elegido por parecer una N rotada) representa los caballos del oponente,
- x (elegido por parecer una K rotada) representa al rey del oponente,
- + representa el cursor del jugador,
- | denota la frontera vertical de las casillas y
- _ denota la frontera horizontal de las casillas.

Observe que las casillas ocupan 5 × 5 caracteres donde 1 carácter es compartido con la adyacente.

2.1 I/O

El jugador mueve el cursor con las teclas WSAD y X, donde

- W mueve el cursor un caracter hacia arriba
- S mueve el cursor un caracter hacia abajo
- A mueve el cursor un caracter hacia la izquierda
- D mueve el cursor un caracter hacia la derecha
- X selecciona la casilla

El jugador puede introducir strings de longitud arbitraria como SSSSSDDX para mover el cursor a la casilla del Rey. Si la casilla seleccionada es válida, el cursor debe cambiar a un asterisco para confirmar (es decir, debe volverse a imprimir el tablero con un asterisco en la posición del cursor en vez de un signo de suma). El jugador, entonces, puede mover el cursor hasta la casilla deseada y volver a presionar X para indicar la jugada. Si la jugada es válida, comenzará a realizarse en la pantalla, con las piezas moviéndose a razón de 1 caracter por segundo.

En caso contrario, debe imprimirse el mensaje vociferado de la pieza, y el cursor debe volver a ser un signo +.

Nótese que los strings de entrada no necesariamente deben terminar en X; puede introducirse

SSSS⏎
SDD⏎
X

Con el mismo resultado. Sin embargo, cada vez que se presione Enter (⏎) debe imprimirse el tablero con el cursor en la posición actual.

Si el cursor se detiene sobre un caracter imprimible (que no es un espacio en blanco) decimos que el cursor "tapa" a ese carácter, reemplazándolo sólo por esa impresión.

2.1.1 Validación

La casilla es válida para seleccionar si, y solo si, contiene una pieza del jugador. La selección es inválida si se intenta seleccionar una frontera entre dos o más casillas

Una orden es válida si

- El jugador está en su turno
- Le indica a la pieza una casilla de destino válida (acorde con sus capacidades de movimiento) incluyendo la casilla actual (esto indica una contraorden, obligando a la pieza a permanecer en la casilla actual)
- La casilla destino no está ocupada por una pieza del mismo jugador
- Debido a que los caballos "saltan" por encima de las piezas para llegar a su casilla destino, no se le puede ordenar detenerse en medio de su movimiento (el rey, debido a que solo puede moverse una casilla, no tiene "medio" de su movimiento)

2.1.2 Turnos y contraórdenes

Un turno dura 10 segundos (debido a que este es el tiempo que tomaría mover un caballo a razón de 1 carácter por segundo). Esto significa que, si el jugador da una orden, deberá esperar 10 segundos para dar la siguiente.

Si una pieza se empieza a mover por iniciativa propia, el jugador puede dar la contraorden hasta, e incluyendo, el momento en que la pieza esté en la frontera de la casilla. En lo que la pieza entre en la siguiente casilla, el jugador habrá perdido el turno y deberá esperar los 10 segundos.

Al recibir una contraorden, la pieza debe permanecer en su posición en ese momento. En consecuencia, las piezas no siempre estarán centradas en sus casillas, incluso pudiendo "tapar" la frontera entre casillas. Nótese que el jugador puede así cancelar la selección de esa pieza sin perder el turno.

2.1.3 Comer y Ganar

Si la casilla destino está ocupada por una pieza del jugador contrario al llegar, la pieza es comida. Si cualquier pieza de un jugador se come al rey contrario, ese jugador habrá ganado y el juego termina.

Nótese que si un caballo "salta por encima" del rey, no se considera que el rey sea comido ya que esa no era la casilla destino.

2.2 Objeciones de las piezas

Al recibir una orden válida, las piezas podrán objetar la jugada. Una pieza objeta una orden si:

- La orden los envía a una casilla “amenazada”
- La pieza está “cuidando” a otra pieza

Una casilla es “amenazada” si existe una pieza oponente que la tiene dentro de las casillas válidas para moverse. Una pieza *x* es “cuidada” por una pieza *y*, si la casilla de la pieza *y* es una de las casillas válidas para moverse de la pieza *x* y ambas pertenecen al mismo jugador.

Recuerden, sin embargo, que si el jugador le da a la misma pieza la misma orden, la pieza debe acatarla.

2.3 Iniciativa de las piezas

Cada pieza tiene una paciencia – un tiempo el cual están dispuestas a esperar sin órdenes –, la cual es proporcional a su distancia de la piezas enemigas. Al inicio de la partida, las piezas con la menor paciencia son los reyes, ya que también están más cerca del rey enemigo.

La paciencia se calcula multiplicando la distancia en casillas por 100 dividido entre el valor de la pieza enemiga: los caballos valen 3 y el rey vale 10.

Por lo tanto, para los reyes, el rey contrario se encuentra a 7 casillas, dándoles una paciencia de $7\times100\div10=70$ segundos por parte del rey enemigo; y, para los caballos más cercanos que se encuentran a 6 casillas, una paciencia de $6\times100\div3=200$ segundos. Se toma siempre la paciencia menor.

Cuando se agota la paciencia, la pieza debe imprimir un reclamo, identificándose claramente. El reclamo dado al jugador depende de si existe una pieza que se puedan comer o no. Si la hay, ese es el reclamo. Si no la hay, el reclamo es simplemente que quieren ser usados.

La pieza reclama dos veces antes de tomar la iniciativa. Si la pieza tiene un oponente que pueda comerse, ese es su movimiento. En cualquier otro caso, elije uno de sus movimientos posibles al azar.

Pueden moverse por iniciativa propia cualquier número de piezas dentro del mismo turno; sin embargo, sólo se contarán los 10 segundos de la primera.

El tiempo de la paciencia no corre si la pieza está “cuidando” a otra, pero comienza a correr desde el inicio si la pieza cuidada se mueve.

3 Funcionamiento del programa

El programa debe crear al menos dos procesos hijo: uno para el jugador y otro para el oponente. Estos procesos deben, a su vez, crear un hilo para cada pieza. Pueden crear hilos adicionales.

Los hilos deben dormir por el tiempo de su paciencia. Si el jugador le emite una orden al hilo (aunque esa orden sea “seleccionar/deseleccionar”), el hilo debe interrumpir su dormida para cumplir con la orden y luego comenzar una nueva dormida.

Para cumplir con la orden, el hilo debe actualizar su posición en el tablero 1 caracter, dormir 1 segundo, actualizarla al siguiente carácter, dormir un segundo, y así sucesivamente hasta haber completado el movimiento. Note que si el hilo vuele a ser interrumpido, su manejo de la interrupción dependerá de si está en la casilla inicial del movimiento o no.

Si una pieza es comida, su hilo debe terminar y el hilo llamador debe realizar el join.

Se recomienda que el tablero, indicando las posiciones de todas las piezas, esté almacenado en el proceso padre y que los hijos le envíen actualizaciones y solicitudes de lectura por *pipes*.

4 Informe

Debe escribir un informe que contenga

- Introducción, indicando la estructura del informe
- Decisiones de diseño, indicando las estructuras de datos utilizadas y la estructura del árbol de procesos e hilos
- Decisiones de implementación, indicando las dificultades encontradas y/o arreglos realizados sobre la marcha
- Conclusiones, recomendaciones y lecciones aprendidas; incluyendo el estado de jugabilidad del programa

5 Requerimientos de la entrega

Todos sus códigos deben estar debidamente documentado y seguir las buenas prácticas de programación en Unix.

El proyecto debe ser subido al Moodle de la materia en la sección marcada como "  Proyecto 1". Sólo deberá efectuar una entrega por grupo.

6 Evaluación

El proyecto tiene una ponderación de 15 puntos. Se asignarán

- 6 puntos por código
  - 1 punto por su manejo de procesos
  - 1 punto por su manejo de hilos
  - 1 punto por su comunicación entre procesos
  - 1 punto por su comunicación entre hilos
  - 1 punto por sus estructuras de datos
  - 1 punto por sus rutinas de impresión y programa principal
- 5 puntos por ejecución
  - 1 punto por procesar correctamente los movimientos del cursor
  - 1 punto por el formato de impresión del tablero
  - 1 punto por imprimir con la frecuencia adecuada para que el juego sea jugable
  - 1 punto por el movimiento correcto de las piezas
  - 1 punto por la "autoconsciencia" de las piezas (iniciativa y mensajes vociferantes)
- 4 puntos por el informe
  - 1 punto por su introducción
  - 1 puntos por sus decisiones de diseño
  - 1 punto por sus decisiones de implementación
  - 1 puntos por sus conclusiones

El programa debe correr sin errores.

7 Extra Credit

Se considerará para puntos extra credit implementar piezas adicionales. A este particular, se recomienda investigar los valores de las piezas, ya que a una pieza sólo debe importarle dejar de cuidar una pieza de valor igual o mayor. Todas las demás piezas sí pueden ser detenidas en medio de su movimiento

En ningún caso se considerará para puntos extra más puntos de los que vale el 50% del proyecto.