

Introducción

Solitaire Chess® es un juego de lógica y estrategia producido por la empresa *ThinkFun*™. Es un juego para un único jugador, donde el objetivo es utilizar las reglas de movimiento y captura de las fichas de ajedrez para generar estrategias de juego. Se proveen tarjetas con instrucciones para la ubicación predefinida de las fichas. Una vez ubicadas, se procede a capturar y eliminar todas menos una de las piezas, usando los movimientos permitidos en el ajedrez. Cada movimiento debe generar la captura de una pieza, con lo que la estrategia debe plantearse con cuidado para lograr el objetivo.

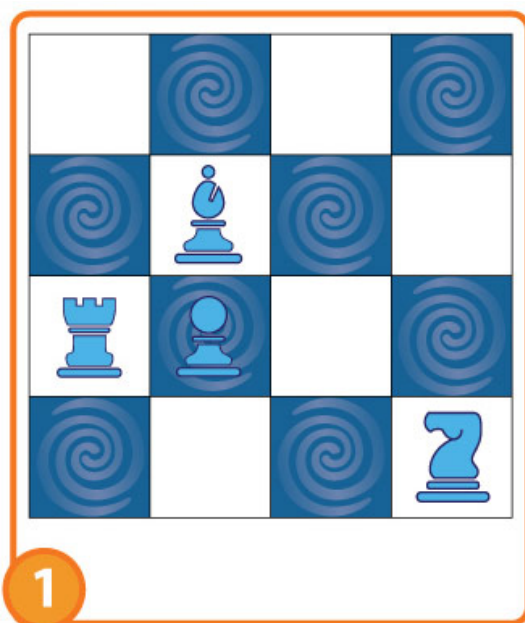


<http://www.thinkfun.com/products/solitaire-chess/>

Descripción del Juego

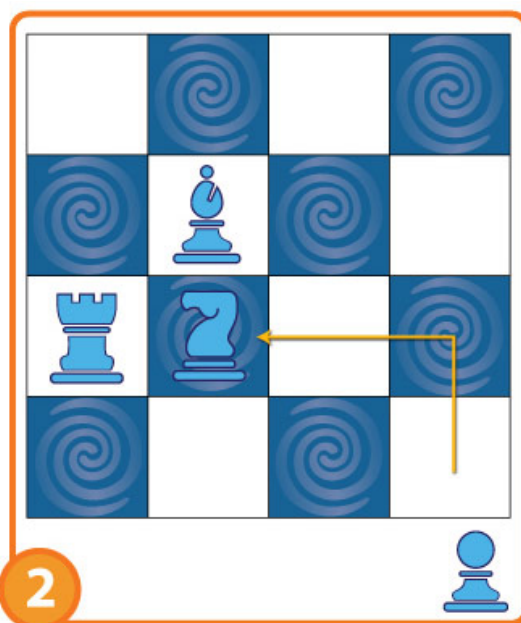
El juego se compone de un tablero de tamaño 4x4, con 60 tarjetas de instrucciones para la ubicación inicial de las piezas. Adicionalmente se incluyen 10 fichas de ajedrez: un rey, una reina, 2 alfiles, 2 caballos, 2 torres y 2 peones. Cada tarjeta de instrucciones tiene marcadas unas posiciones iniciales con las figuras de las fichas, es allí donde se deben ubicar las piezas para comenzar el juego. En cada reto dado por las tarjetas, no se utilizan necesariamente todas las piezas.

Las fichas deben moverse entonces una por vez, de acuerdo a las reglas de movimiento y captura del ajedrez, garantizando que cada movimiento resulta en la captura de una de las piezas disponibles. Cuando sólo queda una única pieza en el tablero, el juego termina y se ha podido resolver adecuadamente el reto. A continuación se presenta un ejemplo de juego:



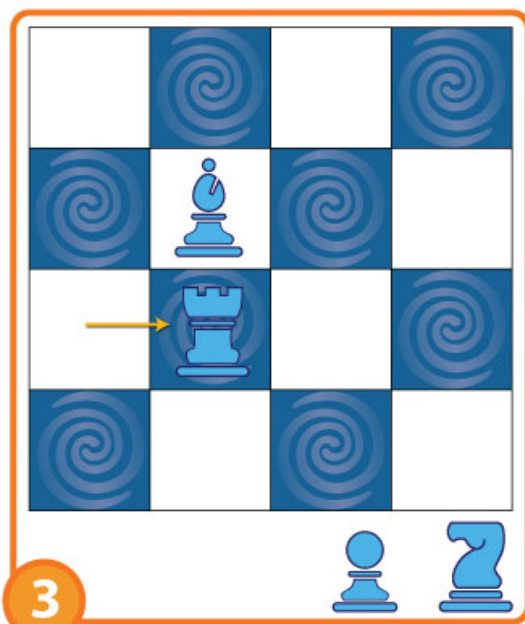
1

Set up the Challenge as shown



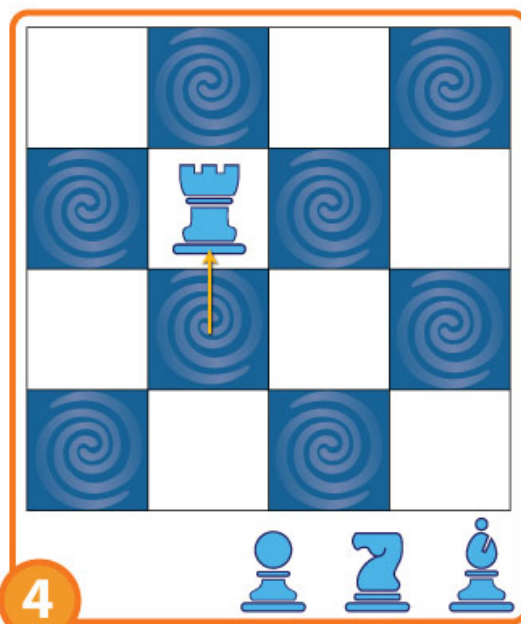
2

Knight captures Pawn



3

Rook captures Knight



4

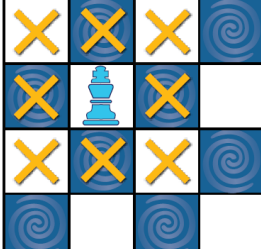
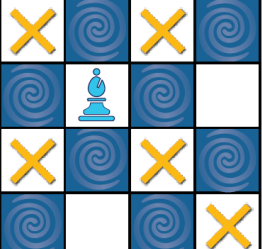

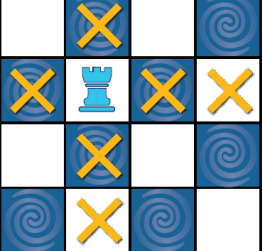
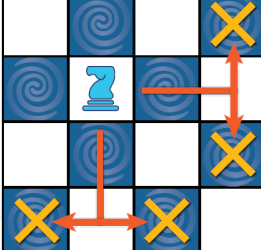
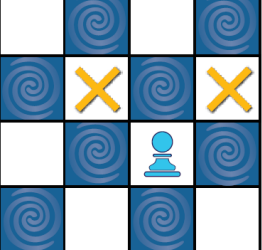
Rook captures Bishop –You Win!

<http://www.thinkfun.com/products/solitaire-chess/>

DESCRIPCIÓN DEL PROYECTO

El objetivo del proyecto del curso de Programación de Computadores será desarrollar un programa que le permita al usuario jugar *Solitaire Chess*®. Se le ofrecerá al usuario la posibilidad de indicar la jugada que desea realizar, la cual debe validarse con respecto a las reglas del juego descritas anteriormente para poder ser efectuada en pantalla. El usuario puede ver el progreso del tablero, y adicionalmente puede guardar en un archivo las jugadas realizadas para retomar posteriormente el mismo juego.

El juego se basa en los movimientos permitidos del ajedrez, los cuales se presentan a continuación:

	<p>El Rey captura al moverse exactamente una casilla de forma horizontal, vertical o diagonal.</p> <p>Los retos están diseñados para que el Rey <u>nunca</u> sea capturado.</p>		<p>El Alfil captura al moverse varias casillas vacías en dirección diagonal.</p> <p>El Alfil siempre se mueve en las casillas de su color original.</p>
	<p>La Reina captura al moverse varias casillas vacías en dirección horizontal, vertical o diagonal.</p>		<p>La Torre captura al moverse varias casillas vacías en dirección horizontal o vertical.</p>
	<p>El Caballo captura al moverse en forma de "L" o "L" invertida (dos casillas en una dirección, giro de 90° y una casilla en la nueva dirección).</p> <p>El Caballo es la única pieza que puede saltar sobre otras piezas. Captura la pieza ubicada en la casilla final del movimiento.</p>		<p>El Peón captura al moverse diagonalmente hacia arriba una casilla. El Peón sólo puede moverse hacia arriba en el tablero.</p> <p>A diferencia del ajedrez normal, el Peón no se intercambia por otra figura al alcanzar el borde superior del tablero.</p>

<http://www.thinkfun.com/products/solitaire-chess/>

Representación de las piezas y el tablero de juego

La representación interna de las piezas de ajedrez se dejará a criterio del estudiante, el cual debe identificar la mejor estructura para representar la información necesaria. Esta información incluye, entre otros datos: tipo, identificador en el tablero y representación en pantalla (ver siguiente sección).

El tablero de juego se representará utilizando una matriz de 4x4 posiciones, y en ella se irán ubicando las piezas a lo largo del juego. La ubicación de cada pieza se marcará a través del identificador (número entero), de acuerdo a la siguiente tabla:

Pieza	Identificador
Rey	0
Reina	1
Alfiles	2, 3
Caballos	4, 5
Torres	6, 7
Peones	8, 9

De esta forma, el tablero corresponderá a una matriz de números. Un ejemplo de ubicación inicial de las piezas en el tablero a través del identificador se muestra a continuación:

	2		
6	8		
			4

Impresión del tablero y las piezas en pantalla

La impresión del tablero y las piezas en pantalla se realizará de forma que sea fácil para el usuario identificar cada una de las diferentes figuras, así como su ubicación actual dentro del tablero.

Para las piezas, se utilizarán tres caracteres con símbolos. En la siguiente tabla se muestran las representaciones para cada una de las piezas:

Pieza	Representación
Rey	###
Reina	***
Alfiles	@@1, @@2
Caballos	??1, ??2
Torres	\$\$1, \$\$2
Peones	&&1, &&2

Para la impresión del tablero, se utilizará la convención internacional de los tableros de ajedrez, la cual numera las filas de 1 a 4 y las columnas con las letras de la a a la d, y se utilizarán caracteres como |, - y + para marcar las subdivisiones de los cuadros del tablero. De acuerdo a esto, el estado inicial del tablero de juego, con respecto al

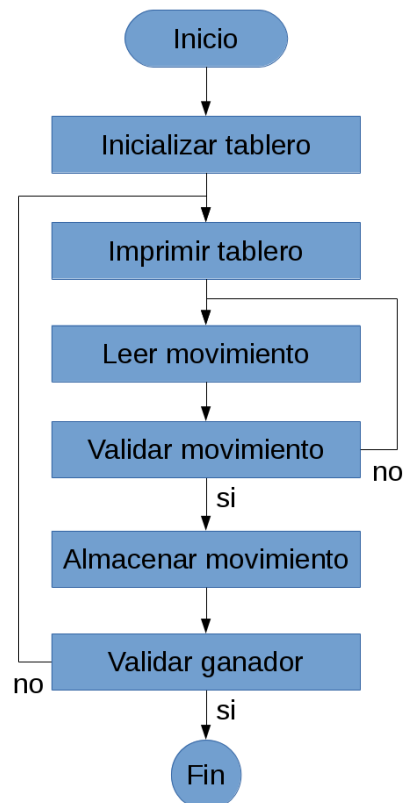
ejemplo anterior, debe verse de la siguiente forma:

	a	b	c	d
1				
2		@@1		
3	\$\$1	\$\$\$1		
4				??1

Teniendo en cuenta estos lineamientos de presentación en pantalla, es claro ver que la representación interna de la información se realiza de manera diferente a su presentación visual. Esto debe tenerse en cuenta al momento de definir la estructura para almacenar las piezas de juego.

Descripción de la lógica del programa

El programa debe permitir al usuario jugar una partida de *Solitaire Chess*®. Esto implica presentar al usuario un tablero en su configuración inicial (seleccionado de forma aleatoria o escogido por el usuario de una lista), solicitar el movimiento a realizar, validar el movimiento, y presentar al usuario el tablero actualizado, en un proceso iterativo hasta que haya ganador o hasta que el usuario decida terminar. De esta forma, los procesos principales a realizar por el programa pueden verse en el siguiente diagrama de flujo:



- **Inicializar tablero:** implica acomodar las piezas en la representación matricial del tablero para que coincidan con el estado inicial del juego, de acuerdo a una de las tarjetas de reto que se han escogido.
- **Imprimir tablero:** implica realizar la impresión en pantalla del tablero y las piezas en su posición actual, siguiendo los lineamientos de impresión en pantalla ya especificados.
- **Leer movimiento:** implica solicitar al usuario el movimiento a realizar. Este movimiento se expresará como una cadena de 5 caracteres: los dos primeros identifican la casilla de inicio o salida (que representa la ficha con la que se va a jugar), el siguiente será siempre el carácter '-', y los dos últimos identifican la casilla de llegada (a donde se quiere mover la ficha). Por ejemplo: la jugada d2-b3 representa mover la pieza que se encuentra en la columna d y la fila 2 hacia la posición en la columna b y la fila 3. Nótese que para la identificación de la casilla siempre va primero la letra de la columna y luego el número de la fila.
- **Validar movimiento:** implica verificar si la pieza puede realizar el movimiento descrito de acuerdo a las reglas del juego. Es necesario verificar la posición con respecto al tablero de juego (para evitar que la pieza se salga de los límites del cuadrado de 4x4). También debe verificarse que la casilla de llegada debe contener otra pieza (dado que cada movimiento debe resultar en la captura de una ficha). Finalmente, debe verificarse que el movimiento es permitido para el tipo de ficha (cada una tiene reglas de movimiento diferentes). Si es una jugada válida, se actualiza el tablero, si no, se debe solicitar nuevamente una jugada hasta que ésta sea válida.
- **Almacenar movimiento:** implica, por un lado, actualizar la representación matricial del tablero para reflejar el efecto del nuevo movimiento, así como ir almacenando el movimiento realizado en una estructura en memoria o en un archivo (para su recuperación posterior).
- **Validar ganador:** una vez se ha actualizado la jugada, es necesario verificar si el jugador ha logrado cumplir con el objetivo de dejar una única pieza en el tablero. Si es así, se informa en pantalla que el juego termina.

Descripción de la interfaz de usuario

La interfaz de usuario del programa se basará en menús, en los que se presentarán al usuario las opciones disponibles y se le permitirá escoger aquella que desee ejecutar. El menú principal presentará al usuario las siguientes opciones: nuevo juego, cargar juego guardado y salir del programa.

Al iniciar un nuevo juego, se le presentará al usuario la posibilidad de escoger una tarjeta de reto, o permitir que aleatoriamente se seleccione una de forma automática. Si el usuario decide escoger una tarjeta, deben presentársele en una lista para

permitirle la selección. Una vez se tiene la tarjeta de reto, se presenta en pantalla el tablero de juego con las posiciones iniciales (de acuerdo a la tarjeta), junto con un menú con tres posibilidades: introducir una jugada, guardar el juego y terminar el juego. La opción de introducir una jugada debe preguntar el movimiento de acuerdo a la convención ya establecida, y luego debe validarlo para actualizar el tablero y verificar si hay ganador. Este proceso se repite hasta que hay ganador, o hasta que se selecciona la opción de terminar el juego. En este caso, debe remitirse al usuario al menú inicial para comenzar otra partida, cargar una previa, o terminar el programa. La opción de guardar el juego debe solicitar al usuario un nombre de archivo para guardar el progreso actual, y luego debe presentar nuevamente el menú (es decir, debe permitir seguir jugando).

La opción de cargar un juego guardado debe solicitar al usuario el nombre del archivo donde se guardó previamente la partida, para luego, de forma secuencial, leer uno a uno los movimientos del archivo y presentar el tablero actualizado paso a paso con cada jugada, para ver el progreso de la partida. Una vez agotados todos los movimientos del archivo, debe permitir continuar el juego, o si ya hay un ganador, debe informarlo para luego llevar al usuario al menú principal.

Atendiendo a criterios de portabilidad entre diferentes máquinas y sistemas operativos, y también a criterios de usabilidad, la interfaz de usuario debe ser lo más sencilla y funcional posible. Esto implica que no se aceptará el uso de librerías especializadas para modificar la apariencia de la interfaz (colores, posiciones), como la librería `windows.h` y similares.

Evaluación del proyecto

El proyecto se desarrollará a lo largo del semestre, y su evolución se evaluará en dos entregas. Cada una de las entregas se enviará a través de la correspondiente actividad en Uvirtual, hasta la medianoche del día indicado para cada entrega. La entrega consistirá de un único archivo comprimido (únicos formatos aceptados: .zip, .tar, .tar.gz, .tar.bz2, .tgz) que contendrá en un único nivel (sin estructura interna de carpetas) documentos (único formato aceptado: .pdf) y código fuente (únicos formatos aceptados: .h, .hxx, .hpp, .c, .cxx, .cpp). La inclusión de archivos en cualquier otro tipo de formato diferente a los especificados, o el envío del archivo comprimido en otro formato diferente a los especificados, resultará en una calificación de 0 (cero) sobre 5 (cinco) para la entrega respectiva.

Entrega parcial (miércoles 28 de septiembre de 2016)

La entrega parcial consistirá en un programa base que imprima el tablero y algunas piezas en una posible configuración inicial en pantalla. Debe permitir al usuario introducir un único movimiento para una torre (es decir, sólo debe permitir mover una torre) y debe imprimir en pantalla el tablero actualizado con la torre en su nueva posición. Esta entrega debe incluir:

- **Documento de diseño (40%):** debe seguir las pautas de ingeniería dadas en

clase para el diseño (identificación de entradas, salidas y condiciones para el programa principal y para las operaciones auxiliares), junto con algunos esquemas que resuman el problema y/o solución gráficamente.

- **Código fuente (60%):** compilable en gcc v. 4.7.2 (Linux) o Dev-Cpp – CodeBlocks (Windows)

Entrega final (martes 22 de noviembre de 2016)

La entrega final consistirá en el programa completo descrito anteriormente. Esta entrega debe incluir:

- **Acta de evaluación (10%):** documento de evaluación de la entrega anterior, debe incluir cada una de las observaciones realizadas junto con la forma en que se corrigieron o arreglaron los problemas indicados.
- **Corrección de errores (15%):** corrección efectiva (en el documento y/o en el código fuente) de los errores o problemas detectados en la entrega anterior.
- **Documento de diseño (25%):** debe seguir las pautas de ingeniería dadas en clase para el diseño (identificación de entradas, salidas y condiciones para el programa principal y para las operaciones auxiliares), junto con algunos esquemas que resuman el problema y/o solución gráficamente. Este documento es incremental, es decir, debe incluir lo definido en la primera entrega, junto con los complementos realizados para completar la segunda entrega.
- **Código fuente (50%):** compilable en gcc v. 4.7.2 (Linux) o Dev-Cpp – CodeBlocks (Windows).

Información adicional:

- Instrucciones de juego *Solitaire Chess*®:
<http://www.thinkfun.com/wp-content/uploads/2015/09/SolChess-3400-IN03.pdf>
- *Solitaire Chess*® – juego en línea:
<http://www.thinkfun.com/play-online/solitaire-chess/>