



Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación  
Introducción a la Programación – Primer Semestre 2015

## Tarea 2: El Juego Othello (Reversi)

### Objetivo General

En esta tarea podrás aplicar los conceptos de Funciones, Strings, Listas y Archivos, para implementar un juego de estrategia llamado Othello (una versión simplificada del juego Reversi) (<http://es.wikipedia.org/wiki/Reversi>). Si quieres ver cómo funciona el juego online, puedes probarlo en <http://www.othelloonline.org/>.

### Descripción del Juego

Othello se juega usando un tablero de 8x8 y 64 fichas, las que son negras por un lado y blancas por el otro. Cada jugador se identifica con un color (blanco o negro). Se juega en turnos, primero el jugador negro y luego el jugador blanco.

Las columnas del tablero se identifican con letras de la A hasta la H, y las filas con números del 1 al 8. La Figura 1 muestra la configuración inicial con la que parte un juego de Othello: dos fichas pre-posicionadas de cada jugador en el centro del tablero. La primera jugada le corresponderá al jugador con fichas negras, luego al de fichas blancas y así sucesivamente.

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								

Figura 1: Tablero del Othello con las fichas iniciales pre-llenadas

El objetivo de cada jugador es lograr que su color sea mayoría en el tablero. Para hacerlo, debe tratar de flanquear las fichas de su oponente rodeándolas con sus fichas. Las fichas flanqueadas cambian de color. Para ver un ejemplo de cómo funciona el flanqueo, observa la Figura 2. Los números (en las casillas blancas y negras) indican el orden de las movidas. Nota: En tu tablero no necesitas mostrar los números de las jugadas; aquí los utilizamos solo para clarificar las reglas del juego.

	A	B	C	D	E	F	G	H
1								
2								
3			2					
4								
5								
6								
7								
8								

Fig. 2. Revisión desde posición 6

	A	B	C	D	E	F	G	H
1								
2								
3			2					
4								
5								
6								
7								
8								

Fig. 3. Cómo queda el tablero después

En la Figura 2, la primera jugada es una ficha negra en C4, que produjo el flanqueo horizontal de la ficha blanca en D4, cambiando esta ficha al color negro. Ahora, en la segunda jugada, le toca mover a las blancas y el jugador decide poner su ficha en la posición C3. Cuando se realiza una jugada (en este caso se puso una ficha blanca en posición C3), tu programa debe revisar en las ocho direcciones que tienen como centro la posición de la ficha recién jugada para revisar la existencia de flanqueos. Esto se muestra con las flechas en la Figura 2.

En este ejemplo, las fichas de color negro quedan flanqueadas por la ficha blanca recién jugada, cuando al recorrer secuencialmente en alguna de las 8 direcciones, saltando solamente sobre fichas de color negro, nos encontramos con una ficha blanca. En el ejemplo, queda flanqueada la ficha negra en D4 por las fichas blancas en C3 y E5. Entonces, la ficha negra en D4 debe cambiar a color blanco como resultado de la jugada, quedando el tablero como en la Figura 3.

Notar que en este ejemplo solo se flanqueó una ficha negra (la cual cambia a blanco); en otras ocasiones se podrían flanquear varias fichas contrarias simultáneamente (tanto en una misma dirección como en otras). Nótese que la ficha negra en C4 no queda flanqueada puesto que al verificar la flecha hacia abajo, se encuentra un espacio vacío después de la ficha negra. **Importante:** en cada jugada, el jugador está obligado a flanquear al menos una ficha contraria. En caso de que no se pueda, obligadamente debe pasar su turno (abstenerse de jugar).

El juego termina cuando ya no se pueden poner más fichas en el tablero. Esto puede suceder si el tablero se llena, o si ninguno de los jugadores puede flanquear, o si a un jugador no le quedan fichas de su color en el tablero (porque su rival las hizo desaparecer todas con sus flanqueos). El ganador es aquel jugador que tiene más fichas de su color en el tablero.

## Tu programa

Tu programa debe hacer lo siguiente:

- Al comenzar, debe crear un tablero con 4 fichas ya posicionadas en el centro: dos blancas en D4 y E5, y dos negras en E4 y D5; estas posiciones están indicadas en el tablero de la Figura 1.
- El jugador negro debe comenzar.
- Debe pasar alternadamente el turno a cada jugador (después de negro juega blanco y así sucesivamente hasta que termine el juego).
- Debe pedir al jugador una identificación de fila y columna, en el formato del juego (A1, por ejemplo) y realizar la movida, revisando todos los posibles flanqueos (8 direcciones) y modificando el tablero (cambiando los colores de las fichas flanqueadas) de acuerdo a las reglas antes descritas.
- Debe dar la opción de pasar el turno (es decir, el jugador se abstiene de poner una ficha en el tablero).
- Debe revisar si ya está completo el tablero. En este caso, se debe terminar el juego, indicar el ganador y el puntaje alcanzado por cada uno (número de fichas de cada color). Si ambos jugadores pasan su turno, se asume que ya no se pueden poner fichas y también se termina el juego. Si es el turno de un jugador y no hay fichas de su color en el tablero, también se termina el juego.
- **OPCIONAL:** Debes dar la opción de guardar el estado del juego, en un archivo definido por el usuario. Si creas esta opción, cuando tu programa comience debe preguntar al jugador si quiere crear una nueva partida o cargar una existente. **Ojo:** esto te sirve para guardar configuraciones avanzadas del juego y poder luego probar tu programa de manera más cómoda.

Tú puedes escoger la forma de mostrar el tablero y los mensajes que utilizarás en tu programa.

Para implementar lo anterior y poder sacarte una buena nota en el control correspondiente a esta tarea, debes cumplir correctamente lo siguiente:

1. Representar el tablero, indicando con una letra **B** una ficha Blanca, con una **N** una ficha negra y con una **V** una celda vacía. Para lograr esto debes definir una lista de listas, en donde cada celda sea un string con los valores "B", "N" o "V".
2. Crear un tablero inicial con 4 fichas ya posicionadas en el centro, dos de cada color, tal como se indicó en la Figura 1.
3. Debes mostrar (imprimir en pantalla) el tablero al inicio y después de cada jugada (con el resultado del flanqueo).
4. Debes crear un programa principal que:
  - Pida la jugada que desea hacer cada uno en su turno (primero negro, luego blanco).
  - Identifique si la jugada es válida en una casilla vacía, dentro de los límites del tablero. También identifique que la jugada haga algún flanqueo para que sea válida. En caso de no serlo debe pedir que el mismo jugador ingrese otra jugada.

- Permitir que el jugador se abstenga de jugar, suponiendo que él decide que no puede flanquear. (No es necesario que tu programa detecte si hay alguna posible jugada con flanqueo.)
  - Detecte las condiciones de término del juego.
5. Debes calcular el número de fichas de color 'c' en el tablero indicado (donde c puede ser "N", "B", o "V").
  6. Debes crear una función llamada **jugar(tablero, posicion, valor)**, que reciba como parámetro un tablero, una posición en el formato del juego y un string con "B" o "N", según quien esté realizando la jugada. La función debe modificar el tablero **entregado como parámetro** según la jugada, retornando True si la jugada era válida (y se modificó el tablero) y False si no.
  7. **Opcionalmente**, y para que practiques el uso de archivos, implementa en tu programa una función llamada **guardarPartida(tablero, nombre\_archivo)** que permita almacenar el estado del tablero para continuar el juego en otro momento. Para continuar una partida, debes implementar también una función llamada **cargarPartida(nombre\_archivo)** que debe retornar un tablero con el estado que está almacenado en el archivo. El archivo debe contener una línea por cada fila, y separar el valor de cada celda por una coma, siguiendo el siguiente formato:

```
A1, B1, C1, D1, E1, F1, G1, H1
A2, B2, ...
...
A8, B8, C8, D8, E8, F8, G8, H8
```

Es importante que en tu solución implementes las funciones que te solicitamos, y que además crees todas las funciones que consideres necesarias para evitar la repetición de código y organizar adecuadamente tu programa, dividiéndolo en varias partes que en conjunto implementan el juego.

## Entrega

Esta tarea no tiene nota, pero será evaluada a través de un control el día **sábado 9 de Mayo**. El control evaluará todas las habilidades y conocimientos que esperamos adquieran mediante la realización de esta tarea.

Al igual que en la tarea anterior, deberás entregar semanalmente un avance a través del SIDING.