

TRABAJO FINAL : Othello

Comentarios iniciales

En este documento se brindan indicaciones acerca del Trabajo Práctico Final que debe ser realizado para aprobar la materia. Tendrá información sobre el programa a implementar y la modalidad de entrega.

Condiciones generales

El trabajo debe ser entregado **al menos 2 días (hábiles) previos a la mesa examinadora.**

El trabajo puede realizarse de manera individual o en parejas de **2(dos) programadores**. En este caso, ambos participantes de la pareja deberán conocer claramente la totalidad del código y su propósito, poder explicarlo y saber cómo modificar el mismo con fines correctivos o para el agregado de nuevas funcionalidades.

Entrega del trabajo

El código entregado será el que se va a evaluar, no permitiéndose llevar el día de la mesa otro código con cambios/correcciones. Es por ello que recomendamos que sea probado intensivamente en un ambiente similar al de los Laboratorios o bien traer equipo propio donde fue desarrollado.

La entrega consiste en subir un solo archivo .c o bien .zip (con el .c u otros, como ser archivos de cabecera .h), cuyo nombre sea alguna de estas opciones:

AUS2025-FinalTaller1-<Apellidos>.c
AUS2025-FinalTaller1-<Apellidos>.zip
AUS2025-FinalTaller1-<Apellidos>.tar.gz
AUS2025-FinalTaller1-<Apellidos>.tgz

en una tarea creada para tal fin en el Campus Virtual en la sección
TRABAJO PRACTICO FINAL 2025.

Othello



Desarrollar un programa en lenguaje C que permita jugar una partida del juego conocido como **Othello o Reversi**.

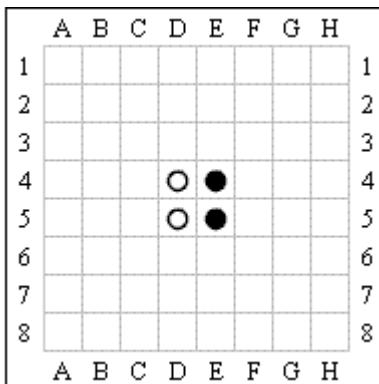
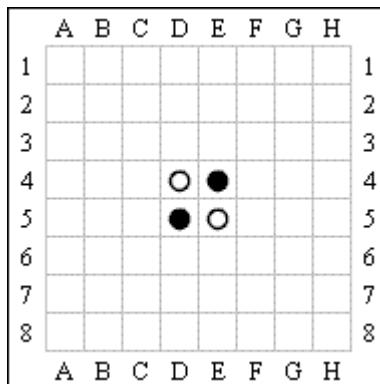
El juego de mesa llamado **Othello** tiene sus raíces en el juego llamado Reversi, inventado por los ingleses Lewis Waterman y John W. Mollett hacia 1880. El nombre "Othello" fue adoptado más tarde, probablemente en referencia a la obra de Shakespeare, Otelo, el moro de Venecia, y su conexión con el conflicto entre el personaje principal y Yago, así como el drama entre Otelo y Desdémona.

Othello es un juego de estrategia para dos jugadores que se juega sobre un tablero de dimensiones 8x8, similar al de ajedrez, pero de un solo color. El objetivo del juego es controlar la mayoría de fichas de tu color (negras o blancas) al final de la partida, que ocurre cuando el tablero está lleno o ningún jugador puede hacer más movimientos.

Recomendación:

Para practicar partidas puede dirigirse a este sitio <https://www.eothello.com/>

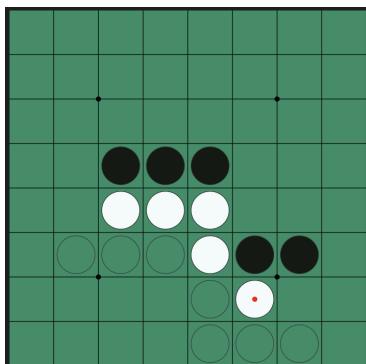
Reglas



Inicio: El tablero comienza **con 4 fichas en el centro**: dos blancas y dos negras, dispuestas **en forma de "X" o 2 columnas de igual color**. Estas son dos posibles configuraciones iniciales y pueden disponerse una u otra de forma azarosa al inicio de cada partida.

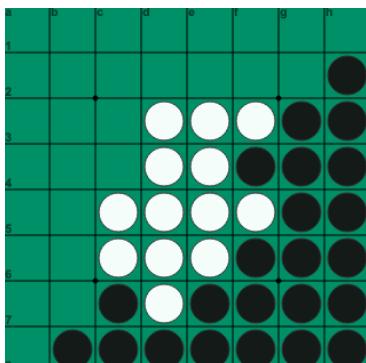
Turnos: Los jugadores se turnan para colocar una ficha de su color.

Captura: Al colocar una ficha, debe encerrar una o más fichas del oponente en línea recta (horizontal, vertical o diagonal) entre la ficha colocada y otra del mismo color. Las fichas atrapadas se voltean al color del jugador.



Movimiento válido: Solo se puede colocar una ficha si al hacerlo se captura al menos una ficha del oponente. Todas las fichas capturadas deben cambiar de color siguiendo las reglas indicadas.

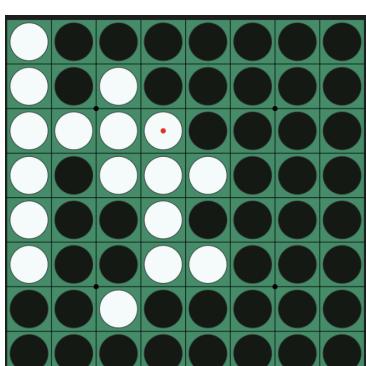
En esta situación, siendo el turno del jugador con fichas negras se observan señaladas 7 posibles jugadas.



Pasar turno: Si un jugador no puede hacer un movimiento válido, debe pasar el turno.

En este caso el jugador con fichas blancas debe pasar ya que no dispone de movimientos que permitan atrapar fichas negras.

Debe quedar bien claro cuando sucede esto.



Fin del juego: El juego termina cuando el tablero está lleno o ninguno de los jugadores puede mover. Gana quien tenga más fichas de su color.

En este caso con 47 fichas negras y 17 fichas blancas, el ganador es "negras".

Comentarios Relevantes

- El programa debe simular una partida entre dos oponentes humanos y al finalizar debe determinar el ganador de la misma.
- Como se mencionó, se restringe a un máximo de 2(dos) jugadores, permitiéndose tambien la modalidad *partida vs el sistema*.
NOTA: la implementación de juego contra el sistema aplica solo a equipos de 2(dos) programadores. Ver detalles en Tareas de acuerdo a la composición del equipo de desarrollo.
- Inicialmente se debe solicitar el nombre de los jugadores y que color desee emplear cada uno.
- Se debe sortear cual de los dos colores inicia.
- A continuación debe comenzar la ronda de turnos.
- Mientras se desarrolla la partida, se debe mostrar claramente a que jugador le toca y cuantas casillas

Tareas de acuerdo a la composición del equipo de desarrollo

En caso de realizar el TP sólo (un programador)

- Crear un programa que cumpla con todas las consignas indicadas anteriormente.
- **Reset:** Permitir el reseteo del juego, es decir, iniciar la partida nuevamente para los mismos jugadores, no pedir nuevamente los nombres.

En caso de realizar el TP en pareja (dos programadores)

- Mismas tareas que el caso anterior de único programador.
- **Estadística.** Apenas se haga un movimiento que diga cuantas fichas ganó temporalmente en esa jugada, esto se debe mostrar antes de que continúe el otro jugador.
- **Partida contra el sistema.** Dotar la posibilidad de que el juego pueda ser humano vs *algoritmo* y que la partida se realice de acuerdo a las reglas pautadas. Para esta modalidad de juego no se exige ningún grado de “inteligencia” o “estrategia inteligente” de parte del sistema para ganar la partida, sino que solo haga movimientos válidos o advierta que no puede hacer movimientos para ganar fichas del oponente.

RECOMENDACIONES FINALES

- Analizar el enunciado atentamente y prestar atención a los detalles, requerimientos y valores. En caso de dudas puede consultar al docente de su comisión.
- Definir el conjunto de variables y funciones que empleará en su algoritmo y evitar variables declaradas sin utilizar. El programa debe ser compilado sin warnings. Evitar también el uso innecesario de variables globales.
- Diseñar el algoritmo antes de programar, verificando que todos los requerimientos puedan satisfacerse.
- Establecer previamente el set de funciones que deberán desarrollarse, para ello diagramar antes de programar el algoritmo que se prentede implementar. *Es obligatorio el uso de funciones.*
- Programar escribiendo código prolíjo, es decir empleando construcciones adecuadas, identación de código, agregado de comentarios pertinentes, interlineados, etc.
- Considerar fuertemente la parametrización mediante macros para facilitar cambios futuros.
- Realizar pruebas parciales del programa para depurarlo.
- Exponer a diferentes pruebas al programa final y revisar que no se reciban errores ni warnings durante la compilación antes de ser entregado y que se cumplan todos los requerimientos.
- **Para favorecer la experiencia del usuario, trate de ser “amable” con los jugadores** permitiendo una interacción no intrincada o compleja, es decir desarrollar una interfaz sencilla e intuitiva.
- Asegurarse que al momento de enviar el programa, se trate de su versión final.
- Al trabajar en parejas, comunicarse activamente y no desentenderse de los avances del otro. Considerar el uso de un sistema de control de versiones. Ambos programadores deben conocer cada detalle del código que entregan, será evaluado de este modo.