

## TAREA N°1

### *Búsqueda adversarial en Juegos.*

**Fecha de envío:** miércoles 07 de septiembre, 23:59 hrs. (vía blackboard)

**Modalidad:** Trabajo en grupo (máximo 3 personas)

### I. Objetivo.

El objetivo de este proyecto es evaluar su capacidad para:

- Aplicar técnicas de búsqueda y razonamiento en juegos.
- Diseñar sistemas computacionales basados en agentes inteligentes.
- Implementar sistemas inteligentes orientados a resolver problemas reales.

### II. Enunciado.

En este proyecto deberás implementar un agente para el juego Reversi 6x6 (<https://en.wikipedia.org/wiki/Reversi>). Este juego se realiza sobre un tablero de 6 filas y 6 columnas con fichas de dos colores (blanco y negro, generalmente). Un ejemplo de posición de inicio se puede ver en la figura 1:

	A	B	C	D	E	F
1						
2			X			
3		X				
4					X	
5				X		
6						

Figura 1: Una posición de inicio de Reversi 6x6,  
con las posibles jugadas del jugador negro marcadas con una X

En la figura 1, aparecen marcadas con una x las posiciones en donde puede jugar el jugador negro. Una jugada válida es aquella que “salta” sobre fichas de color contrario al del jugador y cae en un espacio vacío adyacente a la última ficha saltada, como en el ejemplo mostrado en la figura 2. Todas las fichas que se han saltado cambian de color.

	A	B	C	D	E	F
1						
2		X		X		
3						
4		X				
5						
6						

Figura 2: Juega el jugador blanco

Para jugar una versión en línea de este juego, pueden consultar el siguiente link:  
<https://www.mathsisfun.com/games/reversi-small.html>.

Tu juego debe:

1. Permitir escoger el tamaño del tablero (6x6 u 8x8).
2. Permitir escoger el nivel de dificultad, entre al menos tres niveles.
3. Registrar e imprimir el número de nodos explorados y el tiempo utilizado durante la selección de la jugada que realizará la computadora.
4. Poseer una interfaz que sea de fácil uso, que permita ver el estado del juego en forma clara y seleccionar una jugada sin inducir a errores.
5. Debes agregar a tu juego una característica adicional, que puedes escoger entre:
  - a. Sugerir una jugada al jugador humano si es que lo pide.
  - b. Ajustar el nivel de dificultad de acuerdo a la habilidad del jugador, recordando partidas anteriores.

### III. Consideraciones en la revisión.

1. Debes entregar un informe que explique:
  - a. Introducción al contexto del problema, explicando técnicas que se han usado para abordar el desarrollo de agentes para este juego (incluir referencias a libros o papers).
  - b. El diseño general de tu agente, las técnicas aplicadas y la justificación de su elección.
  - c. La representación escogida para el estado del juego, justificando sus ventajas por sobre otras opciones.
  - d. La estrategia de generación de jugadas.
  - e. La función de utilidad o evaluación, según sea el caso, justificando sus ventajas por sobre otras opciones.
  - f. El diseño de la estrategia utilizada para controlar la dificultad del juego y su justificación.
  - g. El diseño de la estrategia utilizada para implementar la característica especial de tu juego.
  - h. Conclusiones respecto del desempeño del agente, con casos de ejemplo, tablas con resultados numéricos y gráficos.
2. El grupo deberá hacer una demostración de su proyecto en clases. Esta demostración debe durar un máximo de 5 minutos y enfocarse en el aspecto novedoso que escogieron incorporar.
3. Puedes implementar este trabajo en lenguaje C/C++ o lenguaje Python.
4. Debes comentar cada una de las funciones, estructuras o clases que definas, indicando una descripción de la labor que lleva a cabo cada una.
5. Puedes trabajar con el IDE o lenguaje que más te acomode. No obstante lo anterior, tu programa debiera poder ser ejecutado sin problemas en Linux o Windows.
6. El sistema debe ser **robusto**, se penalizarán los errores no manejados, de cualquier tipo.

### IV. Sobre la entrega, atrasos y faltas a la ética.

1. Debes subir tu trabajo a la plataforma de <http://unab.blackboard.cl>, en una casilla que se habilitará especialmente para esto.
2. Debes subir todos los códigos fuentes y el archivo ejecutable (para C/C++). Todos estos archivos deben ser comprimidos en un solo archivo llamado **“rutcompleto1\_rutcompleto2\_rutcompleto3.zip”**, indicando el rut de

cada integrante del grupo. Por ejemplo, si tu rut es 19.000.111-3, debieras ingresarlo sin puntos ni guión, o sea 190001113.

3. Si existe sospecha de copia (con otros compañeros, o desde internet), serás interrogado acerca de su trabajo, para aclarar dudas de su entendimiento y autoría. Si se confirma la copia, el trabajo será evaluado con nota **1.0**.
4. Las consultas las puedes realizar en un foro que se habilitará para la tarea en blackboard.