



Universidad Simón Bolívar  
Departamento de Computación y Tecnología de la Información  
CI5437 - Inteligencia Artificial I  
Prof. Carlos Infante

## **Proyecto 2**

Javier Medina 12-10400  
Maria Gabriela Hernandez 15-10696

Caracas, Julio 2021

## **Implementación:**

Inicialmente se arregló la verificación de los diagonales en la función *outflank* y *move*, utilizando las constantes *dia1* y *dia2* ya existentes en el archivo de *othello\_cut.h*

Luego, se implementaron cuatro algoritmos: “Negamax sin poda alpha-beta”, “Negamax con poda alpha-beta”, “Scout” y “Negascout = negamax con poda alpha-beta + scout”.

En cada uno de ellos, el algoritmo fue escrito siguiendo el pseudocódigo que vimos en las clases, adaptándolo con las funciones ya existentes en *othello\_cut.h*. Adicionalmente agregamos una variable llamada *move* que indica si efectivamente el jugador pudo jugar, esto con la finalidad de conocer si fue o no generado un nuevo estado al recorrer los hijos del nodo.

## **Resultados:**

Los algoritmos fueron ejecutados en una laptop con procesador I7-9750H y 16GB de Memoria RAM. Al principio los estábamos ejecutando en Google Colabs, pero nos dimos cuenta que no ocupaba gran parte de la memoria RAM. Por lo tanto, como una de nuestras laptops tenía un mejor procesador que Google Colabs, preferimos correrla en la laptop.

Los resultados se encuentran en la carpeta *results* del proyecto. En el podemos encontrar los siguientes archivos:

- “1.txt”: Corrida del algoritmo Negamax sin poda aplha-beta.
- “2.txt”: Corrida del algoritmo Negamax con poda alpha-beta.
- “3.txt”: Corrida del algoritmo Scout.
- “4.txt”: Corrida del algoritmo Negascout.

## **Conclusión:**

Podemos observar que:

Algoritmo	Valor mínimo que logró calcular en el juego
Negamax sin poda alpha-beta	16
Negamax con poda alpha-beta	9
Scout	7
Negascout	7

Si observamos los resultados y el orden en que se ejecutaron los algoritmos, parecieran estar ordenados del peor al mejor desempeño. Siendo Negascout el del mejor rendimiento para calcular los movimientos.

Respecto a los nodos generados y expandidos, podemos notar una mejoría muy grande entre negamax cuando no realiza poda alpha-beta y cuando si la realiza, reduciendo los valores considerablemente. Ejemplo: En el valor número 16 se expandieron 2.656.726.010 nodos y se generaron 1.382.288.008 nodos sin poda. Mientras que con poda se expandieron 22.325.108 y se generaron 29.501.797 nodos.

Igualmente podemos observar que Negascout expande muchísimos más nodos a diferencia Scout, sin embargo es más rápido porque genera más nodos por segundo que Scout.