

# Descripción del Algoritmo

En este documento se presenta la información necesaria para implementar el algoritmo para el controlador de la variante del ajedrez “knight chess”, mediante el método de monte carlo tree search con variante de UTC.

## Monte Carlo Tree Search

El método consiste en tres pasos:

1. **Tree policy:** Se busca al mejor candidato del árbol que maximice la función del UCT, si se llega al final del árbol y la hoja contiene movimientos que aún puede realizar, esta es expandida.
2. **Default policy:** En este paso se simula el juego entre los contrincantes, en este caso entre el caballo negro y rosado. La selección de movimientos entre ambos es totalmente aleatoria.
3. **Backup:** Cuando la simulación termina, se ve la recompensa que se obtuvo en el juego, se va desde la hoja que se seleccionó en el paso “Tree policy” hacia la raíz, actualizando la visita y la recompensa en cada nodo.

A continuación se deja un pseudocódigo para mejor comprensión de este procedimiento:

---

**Algorithm 2** The UCT algorithm.

---

```
function UCTSEARCH( $s_0$ )
  create root node  $v_0$  with state  $s_0$ 
  while within computational budget do
     $v_l \leftarrow \text{TREEPOLICY}(v_0)$ 
     $\Delta \leftarrow \text{DEFAULTPOLICY}(s(v_l))$ 
    BACKUP( $v_l, \Delta$ )
  return  $a(\text{BESTCHILD}(v_0, 0))$ 

function TREEPOLICY( $v$ )
  while  $v$  is nonterminal do
    if  $v$  not fully expanded then
      return EXPAND( $v$ )
    else
       $v \leftarrow \text{BESTCHILD}(v, Cp)$ 
  return  $v$ 

function EXPAND( $v$ )
  choose  $a \in$  untried actions from  $A(s(v))$ 
  add a new child  $v'$  to  $v$ 
    with  $s(v') = f(s(v), a)$ 
    and  $a(v') = a$ 
  return  $v'$ 

function BESTCHILD( $v, c$ )
  return  $\arg \max_{v' \in \text{children of } v} \frac{Q(v')}{N(v')} + c \sqrt{\frac{2 \ln N(v)}{N(v')}}$ 

function DEFAULTPOLICY( $s$ )
  while  $s$  is non-terminal do
    choose  $a \in A(s)$  uniformly at random
     $s \leftarrow f(s, a)$ 
  return reward for state  $s$ 

function BACKUP( $v, \Delta$ )
  while  $v$  is not null do
     $N(v) \leftarrow N(v) + 1$ 
     $Q(v) \leftarrow Q(v) + \Delta(v, p)$ 
     $v \leftarrow \text{parent of } v$ 
```

---

Figura 1: Pseudocódigo UTC Search.

En la política de selección de movimientos, se prioriza el movimiento que ataca a un caballo enemigo, si no queda opción se busca alguna casilla vacía para el movimiento de este.

# Paralelización

El árbol fue paralelizado en la raíz, esto se entiende que se deja una hebra por cada raíz, cuando se terminan todas estas, se obtiene el mejor movimiento mediante un “rate” entre todos los nodos hijos de la raíz del primer nivel, mediante la recompensa dividido en la veces visitadas del nodo por ese movimiento.

