

# Rapport de projet : Gomoku

Quentin Garrido, Antoine G  lin, K  vin Lor

22 f  vrier 2019

## **1 Introduction**

## 2 Utilisation

### 3 Repr  sentation du plateau

## 4 Choix des coups

---

**Algorithm 1** Algorithme du Negamax

---

```
1: procedure NEGAMAX(node, depth)
2:   if depth = 0 then
3:     return EVALUATE(node.move)
4:   end if
5:   max  $\leftarrow -\infty$ 
6:   for all child in node.children do
7:     score  $\leftarrow -\text{NEGAMAX}(\textit{child}, \textit{depth} - 1)$ 
8:     if score > max then
9:       max  $\leftarrow \textit{score}$ 
10:    end if
11:  end for
12:  return max
13: end procedure
```

---

## 5   valuation d'une position

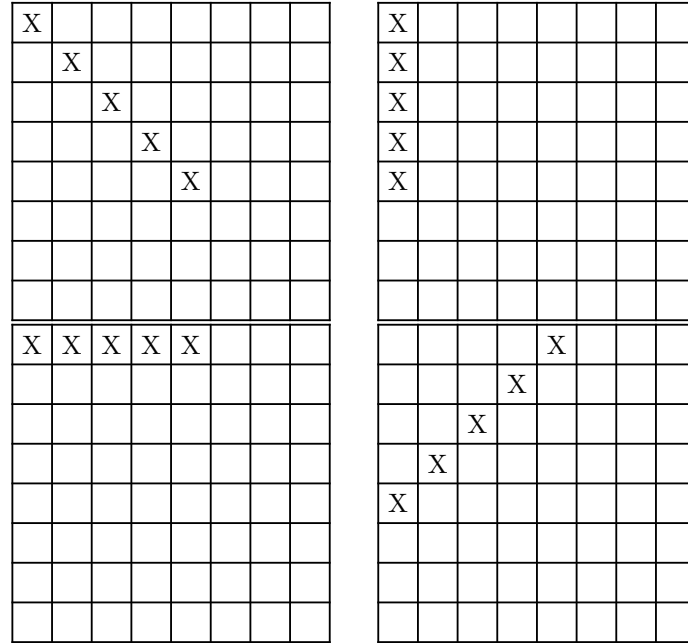


FIGURE 1 – Patterns v  rifi  s pour la victoire ou non

---

**Algorithm 2** Algorithme de pattern matching
 

---

```

1: procedure PATTERN_MATCHING(bitboard, pattern)
2:   count  $\leftarrow$  0
3:   for  $i \leftarrow 0$  to  $8 - \text{pattern.height}$  do
4:     for  $j \leftarrow 0$  to  $8 - \text{pattern.width}$  do
5:       if  $\text{pattern} \& \text{bitboard} = \text{pattern}$  then                                 $\triangleright$  & repr  sente un ET logique bit par bit
6:         count  $\leftarrow$  count + 1
7:       end if
8:        $\text{pattern} \leftarrow \text{pattern} \gg 1$                                         $\triangleright$  >> est un d  calage de  $n$  bits    droite
9:     end for
10:     $\text{pattern} \leftarrow \text{pattern} \gg \text{pattern.width} - 1$ 
11:  end for
12:  return count
13: end procedure

```

---

## 6 R  sultats