

# Rapport ACT :

## TP Programmation dynamique : Hexapawn

Binôme :

- Pelage François-Xavier
- Philippot Grégoire

### Question 1 :

Pour tout  $x_i$ ,  $x$  représentant un successeur de la configuration et  $i \in \mathbb{R}$ , et  $X$ , représentant la valeur de configuration où un des deux joueurs gagne.

*Cas où un des successeurs a au moins une valeur négative :*

$$X = -(\text{plus petite valeur négative de } x_i - 1)$$

*Cas où les successeurs sont tous positifs :*

$$X = -(\text{plus grande valeur positive de } x_i + 1)$$

### Question 2 :

Plateau :

- configuration : `String[hauteur][largeur]`
- largeur : `int`
- hauteur : `int`
- partie finie : `char`
- valConfig: `List<int>`
- ListConfig : `List<String[hauteur][largeur]>`

```
int CalculevaleurConfig(List<String[hauteur][largeur]> ListConfig) {
```

```
    -vérifier que partie non finie
```

```
    -vérifier que partie non bloquée
```

```
    -si toutes les parties sont finies
```

```
    // pour chaque configuration de la liste, vérifie si on peut bouger/manger avec le pion et si on  
    peut, ajoute 1 (le pas) dans la liste valconfig à la valeur du même indice que la configuration actuelle.
```

-Retourner valeur : si blanc gagnant plus petite si blanc perdant plus grande

}

### Question 3 :

Utilisation d'une hashmap pour stocker les configurations.