

Martin Brönnimann

Romain Seznec

# RAPPORT NSI

Projet Binerio

## Avant le projet

Tout d'abord, avec Romain, nous avons créé un dépôt sur [Github](#) afin de nous répartir le travail bien plus facilement. Nous nous sommes ensuite répartis les fonctions afin de gagner du temps.

Romain s'est occupé des fonctions :

- Vérifie Consécutif
- Vérifie Parité
- Vérifie Identique

Et je me suis occupé du reste de "binero.py". Nous avons quand même modifié les fonctions ensemble histoire d'optimiser le tout.

## Création de la fonction "verifie\_grille"

- Premièrement, nous avons regardé si la grille donnée en argument était complète via l'utilisation de la fonction "grille\_complete". Si elle est incomplète, on retourne **False**.

```
if not grille_complete(g):  
    return False
```

- Ensuite nous avons itéré chaque élément de la grille et avons regardé si les éléments (qui sont des listes de liste) ne passaient pas les fonctions "verifie\_parite" et "verifie\_identique". Si c'est le cas on retourne **False**.

```
for i in g:  
    if not verifie_parite(i) and not verifie_identique(i):  
        return False
```

- Enfin, nous avons vérifié la consécutive de la grille via la fonction "verifie\_consecutif" et nous avons répété l'opération mais en transposant la grille via la fonction "transpose". Pour automatiser ça, nous avons utilisé une boucle for. Si l'une des grilles n'est pas consécutive, on retourne **False**.

```
for _ in range(2):  
    if not verifie_consecutif(g):  
        return False  
    g = transpose(g)
```

Si tous les “if” sont passés avec succès, la fonction retourne **True**.

## Résultat du projet “binero”

Après avoir développé les fonctions et optimisé le tout comme on le pouvait, Romain et moi avons décidé de recoder le fichier “interfaceTk.py” pour changer l’interface utilisateur.

Ainsi, avec romain avons ajouté:

- Un système de difficulté
- De nouvelles grilles

Au final, voici le résultat du projet binero:

