

Jeu 2048

Le jeu fonctionne très bien, sans erreur apparente.

L'affichage en console est un peu surprenant au début, mais on s'y fait. Le rendu graphique est un peu décevant à cause des longueurs différentes des valeurs affichées qui créent des décalage sur les colonnes. Ce point aurait pu être facilement amélioré en affichant des cases de longueurs constantes, par exemple de 4 caractères. La fonction `.format()` de la classe `texte` réalise notamment cela, et on en avait parlé. Voir à droite ce qu'on obtient en remplaçant l'instruction `print(grille[i])` de la fonction `printGrid(grille)` par

		2			

				8	

		2		4	

```
ligne=""
for j in range(4):
    ligne+=" {:4} |".format(grille[i][j])
print(ligne)
print("-"*29)
```

Mais cette question de l'affichage me paraît secondaire, comparée à celle du dynamisme du jeu qui est très bien rendu, de façon simple et compréhensible avec des outils simples qui ont assez bien été assimilés (fonctions, listes, ...).

Les fonctions sont assez bien et systématiquement documentées, avec un titre, les données en entrée et les résultats en sortie : ces commentaires sont assez clairs même si, parfois, il faut comprendre entre les lignes. Par exemple la fonction `newGrid(grille)` crée une nouvelle grille qui, si j'ai bien compris, inverse les lignes et les colonnes. Mais le commentaire indique : « résultat : Une grille composée de 4 listes contenant des entiers, de puissance de 2 et de position grille[x][y] rangées par ordre d'index ou: index 1 = liste1 ; index2= liste 2... » L'expression rangées par ordre d'index n'est pas forcément très parlante.

Des commentaires supplémentaires viennent renforcer les explications systématiques, ce qui accentue la compréhension du programme.

Le vocabulaire n'est pas toujours explicite : dans l'instruction

```
grille[t].insert(w,"") # insert la valeur 0 avec comme index w
```

la valeur 0 est en fait "" (une chaîne vide). Cela témoigne sans doute d'une modification (la valeur 0 qui ne rend pas bien à l'affichage a été remplacée par "" qui est mieux) qui n'a pas été reportée dans les commentaires. Le terme `index` est, je trouve, mal adapté ici car il y a deux index. Il vaudrait mieux parler de ligne et colonne puisque dans `grille[t][w]`, `t` est l'index de ligne et `w` l'index de colonne.

Certaines actions sur les listes ne me paraissent pas utiles, mais je peux me tromper :

- Dans la fonction `newGrid(grille)`, pourquoi ne pas écrire `return(liste)` à la place de

```
grille=[[[],[],[],[]]]
grille=liste
return(grille)
```
- La fonction `rebuildGrid(grille)` apparaît comme identique à `newGrid(grille)`. J'ai remplacé les deux appels à la fonction `newGrid` qui sont dans `deplacement(direction,grille)` par des appels à `rebuildGrid(grille)` et le programme fonctionne toujours bien.

Projet de note : 18/20

Détail de la note :

- préprojet : 2/2
- compte-rendu : 3.5/4 (bonne narration de la difficulté rencontrée pour l'affichage graphique et de la solution adoptée, on pourrait adjoindre une illustration au moins du rendu obtenu)

- Programme : 12.5/14 (très fonctionnel, simple, bonne utilisation des fonctions, affichage à améliorer même en console)