



# Le 2048

## I- Présentation

Définition wikipédia :

[https://fr.wikipedia.org/wiki/2048\\_\(jeu\\_vid%C3%A9o\)](https://fr.wikipedia.org/wiki/2048_(jeu_vid%C3%A9o))

2048 est un jeu vidéo de type puzzle conçu en mars 2014 par le développeur indépendant italien Gabriele Cirulli et publié en ligne sous licence libre via Github le 9 mars 2014.

Le but du jeu est de faire glisser des tuiles sur une grille, pour combiner les tuiles de mêmes valeurs et créer ainsi une tuile portant le nombre 2048. Le joueur peut toutefois continuer à jouer après cet objectif atteint pour faire le meilleur score possible.

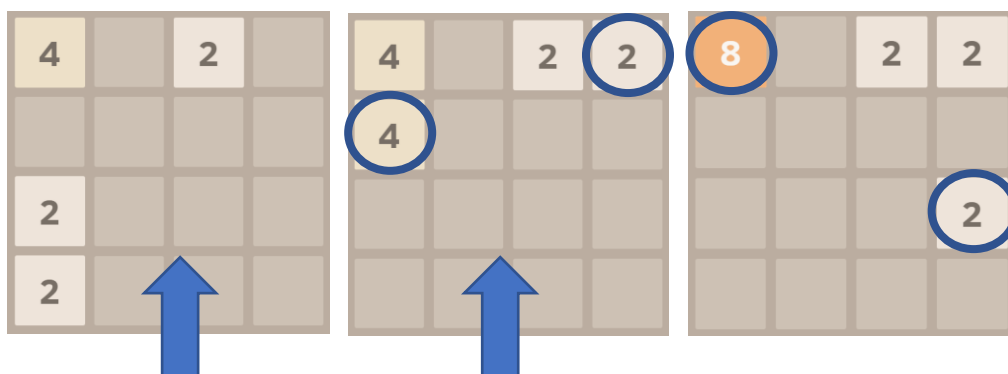
Vous pouvez tester le fonctionnement du jeu ici :

<https://2048game.com/>

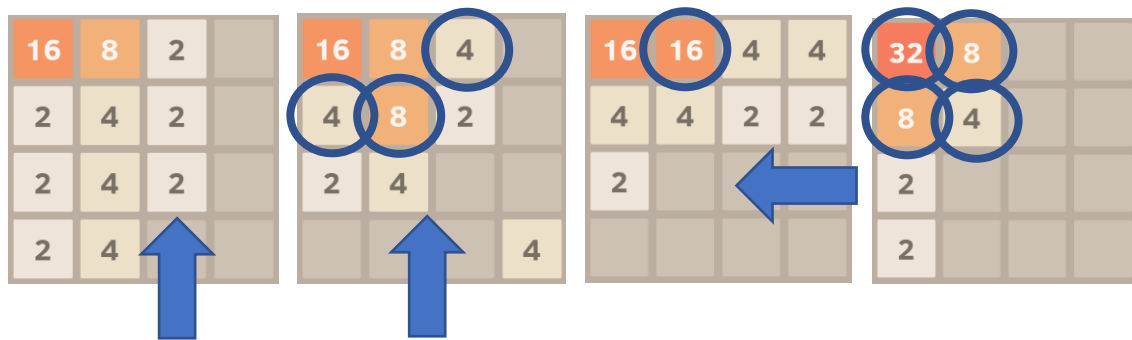
### Quelques exemples de combinaison de tuiles :

A chaque vous pouvez déplacer les tuiles dans les quatre directions : Si leurs valeurs sont identiques, elles vont se combiner en une tuile de valeur double et une nouvelle tuile de valeur 2 ou 4 va apparaître dans la grille.

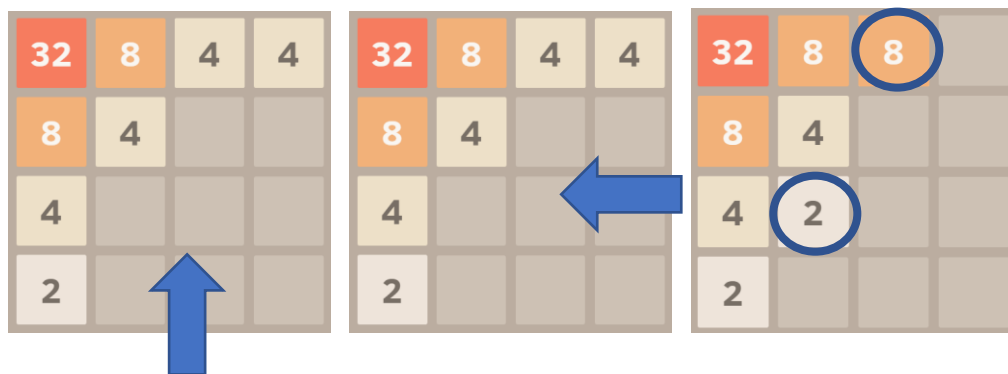
Une tuile ne peut se combiner qu'une seule fois par tour.



Mais plusieurs tuiles différentes peuvent se recombinaison dans le même tour :



Si une demande de déplacement ne conduit ni à un déplacement, ni à une combinaison : rien ne se passe et aucune nouvelle tuile n'apparaît. C'est le cas de la première figure ci-dessous après un clic vers le haut. Le second clic vers la gauche entraîne la combinaison des deux tuiles 4 donc une nouvelle tuile apparaît alors.



## II- L'interface graphique

On vous fournit une interface graphique à travers la classe GUI2048 que vous pouvez importer depuis le module fourni. Si le répertoire GUI\_2048 est dans le même répertoire que votre fichier :

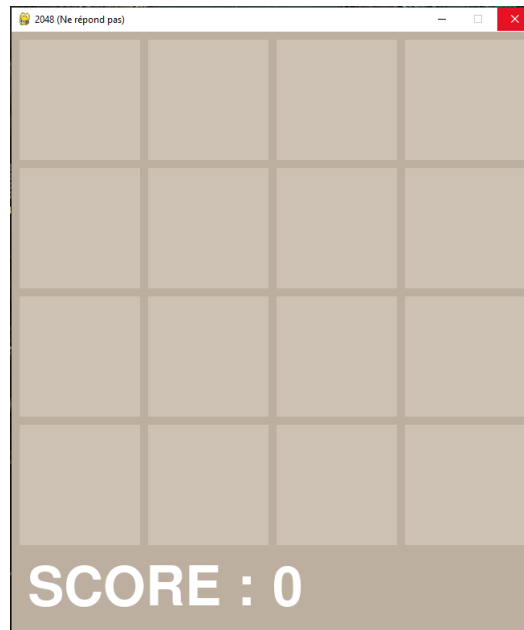
```
from GUI_2048.gui2048 import GUI2048
```

Constructeur de la classe :

- Attribut public : **aucun**.
- Le constructeur prend en paramètre le nombre de case sur un côté de la grille. Il permet de créer un objet graphique vierge.

Par exemple pour une grille de 4 par 4 :

*GUI = GUI2048(4)*



- Les méthodes :
  - **refresh(g, score , txt)** : Cette méthode rafraichie l'affichage du jeu 2048 conformément à la grille passée en argument.
    - **g** est une liste de nb listes (ou n'importe quel type indexable contenant des éléments itérables). Par exemple, dans une grille de 4\*4 :
      - g[0][0]** est la case en haut à gauche ;
      - g[3][3]** est la case en bas à droite.
    - Attention** : la valeur affichée est la puissance de 2 du contenu de g.
      - 0 -> case vide
      - 1 -> affiche 2
      - 2 -> affiche 4
      - 3 -> affiche 8
      - 4 -> affiche 16
      - 8 -> affiche 256
      - 11 -> affiche 2048
      - ...
    - **score** est un entier qui s'affichera en bas de l'écran.
    - **txt** est un texte affiché à destination du joueur, sous le score.

Par exemple :

```
grille = [[i + j for i in range(4)] for j in range(4)]  
GUI.refresh(grille, 2048, 'à vous de jouer !!')
```



- **gameOver()** : Cette méthode permet d'afficher GAME OVER plein écran jaune sur fond rouge.

Par exemple :

`GUI.gameOver()`



- **waitClick()** : Cette méthode attend l'action d'un joueur. Elle gère trois types d'actions :
  - **demande fermeture de la fenêtre** : fermeture propre de la fenêtre pygame et fin du programme python.
  - **click sur la fenêtre** : retourne un tuple contenant les numéros (x, y) de la case choisie.
  - **appui sur des touches spéciales** :
    - fleche RIGHT : retourne '\_R';
    - fleche LEFT : retourne '\_L';
    - fleche DOWN : retourne '\_D';
    - fleche UP : retourne '\_U';
    - touche BACKSPACE : retourne '\_B';
    - touche RETURN : retourne '\_E';
    - touche ESCAPE : retourne '\_S';
  - **appui sur une autre touche du clavier** : retourne le caractère unicode correspondant.

**Attention**, une fois exécutée, on ne peut sortir de cette méthode que par l'une de ces quatre actions.