

## Outils Info. - TD : Tetris

### Exercice 1 – Grille et Niveau

Un niveau est une grille de hauteur  $H$  et de largeur  $W$  qui va contenir des valeurs entières :  $0$  représente l'absence de bloc, et les nombres de  $1$  à  $7$  représentent des blocs de 7 couleurs différentes. En python on peut représenter une grille comme une liste de lignes ou comme une liste de colonnes.

**Q1)** Pourquoi la représentation sous forme d'une liste de ligne est-elle plus adaptée pour Tetris ?

Dans tout le TD on considère que le coin inférieur gauche est la case de coordonnées  $(y, x) = (0, 0)$  :

$(H-1,0)$	$(H-1,1)$	...	$(H-1,W-1)$
...	...	...	...
$(1,0)$	$(1,1)$	...	$(1,W-1)$
$(0,0)$	$(0,1)$	...	$(0,W-1)$

Ainsi,  $y$  représentera le numéro de la ligne et  $x$  le numéro de la colonne et on considèrera un niveau comme une liste de lignes.

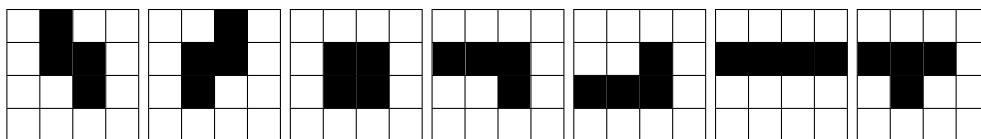
**Q2)** En supposant qu'il existe une fonction `grille_vide(n_lignes, n_colonnes)` qui crée une grille remplie de  $0$ , écrivez une fonction `creer_niveau()` qui reçoit en paramètres, les deux dimensions du niveau de Tetris ( $H$  et  $W$ ) et renvoie un niveau vide.

### Exercice 2 – Pièce

Une pièce est une minigrille de taille  $4 \times 4$ . On utilisera la même représentation et la même façon d'indicer les cases que pour un niveau.

**Q3)** Écrivez une fonction `creer_piece_vide()` qui renvoie une minigrille de taille  $4 \times 4$  dont tous les éléments valent  $0$ .

Il existe 7 types de pièces dans Tetris:



Pour créer une pièce, on procède comme suit :

1. Créer une pièce vide.
2. Mettre à  $1$  (ou  $2$ , ou  $3$ , ... ou  $7$ ) toutes les cases noircies de la pièce.

**Q4)** Écrivez une fonction `creer_pieces()` qui renvoie une liste contenant une fois chaque pièce possible.

**Q5)** Écrivez une fonction `tourner_piece_droite(piece)` qui renvoie une nouvelle pièce correspondant à la pièce initiale à laquelle on a appliqué une rotation de  $90^\circ$ . Attention, la pièce initiale ne doit pas être modifiée.

$$\text{tourner\_piece\_droite} \left( \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \hline \square & \square & \square & \square \\ \hline \square & \square & \square & \square \\ \hline \end{array} \right) = \begin{array}{|c|c|c|c|} \hline \square & \square & \blacksquare & \square \\ \hline \square & \square & \blacksquare & \square \\ \hline \square & \square & \blacksquare & \square \\ \hline \square & \square & \blacksquare & \square \\ \hline \end{array}$$

**Q6)** Proposez deux façons (dont une fainéante) d'écrire la fonction `tourner_piece_gauche(piece)`.

### Exercice 3 – Problème et fusion

Pour les questions suivantes, on considère que  $p_y$  et  $p_x$  indiquent la position du coin inférieur gauche d'une pièce par rapport à la grille.

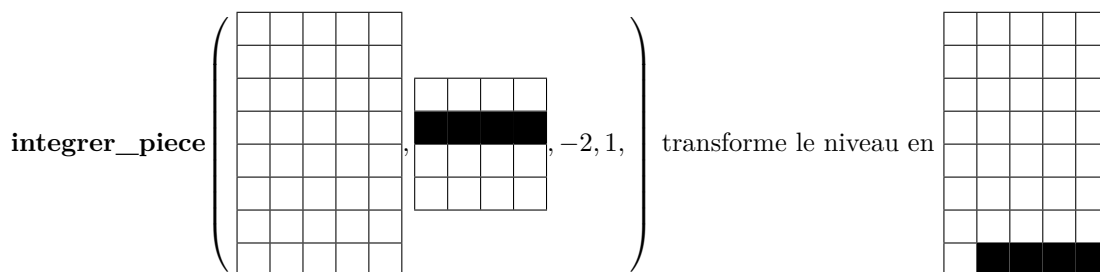
Le pseudo-code de la fonction problème est le suivant:

- Pour chaque case de la minigrille:
  - Si la case vaut plus de 0:
    - \* Si la case est en dehors de la grille ou bien si elle correspond à une case valant déjà plus de 0 dans la grille alors renvoyer « vrai ».
- Si aucun élément de la pièce ne pose problème alors renvoyer « faux ».

**Q7)** Écrivez la fonction `probleme(niv, piece, p_y, p_x)` qui implémente le pseudo-code précédent.

Intégrer une pièce dans un niveau revient à passer à 1 (ou 2, ou 3...) les cases du niveau qui sont superposées avec les cases de la pièce qui valent 1 (ou 2, ou 3, ...).

**Q8)** En vous inspirant de la fonction `probleme(niv, piece, p_y, p_x)`, écrivez la fonction `integrer_piece(niv, piece, p_y, p_x)` qui intègre une pièce dans un niveau.



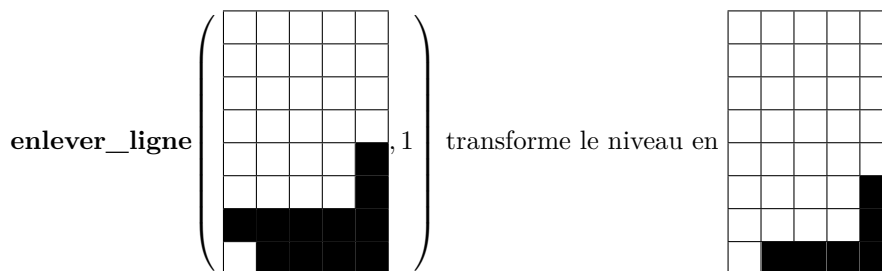
### Exercice 4 – Suppression des lignes pleines

Une case est vide si elle contient 0 et est pleine sinon.

**Q9)** Définissez avec des mots ce qu'est une ligne pleine.

**Q10)** Écrivez une fonction `est_pleine(ligne)` qui renvoie « vrai » si une ligne est pleine et « faux » sinon.

**Q11)** Écrivez la fonction `enlever_ligne(niv, y)` qui enlève la ligne d'indice  $y$  dans le niveau. N'oubliez pas qu'il est nécessaire d'ajouter une nouvelle ligne vide en haut du niveau après avoir supprimé une ligne.



**Q12)** En supposant qu'il existe une fonction `enlever_ligne(gr, y)` qui enlève la ligne d'indice  $y$  dans la grille  $gr$ , écrivez une fonction `nettoyez_lignes_pleines(niv)` qui supprime toutes les lignes pleines d'un niveau et renvoie le nombre de lignes supprimées. Vérifiez que votre algorithme marche dans le cas où il y a deux lignes pleines consécutives.