

## Séance 8: LES ENVAHISSEURS DE L'ESPACE

L1 – Université Nice Sophia Antipolis

Dans ce TP, vous allez manipuler des images aux formats PBM, PGM et PPM. Ces formats permettent très facilement de lire et modifier une (petite) image avec un éditeur de texte. Pour commencer, récupérez l'archive `tp8.zip` sur la page du cours, décompressez-la dans le répertoire qui contient tous vos TP Python, puis allez dans le répertoire `tp8`. Vous devez y trouver les fichiers

```
exemple.pbm
space-invader-pieuvre.pbm      tp8_utils.py
space-invader-soucoupe.pbm    viewer.py
```

### Le format PBM

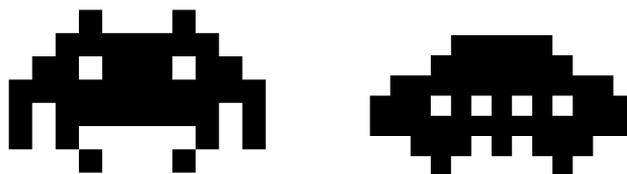
Un fichier PBM (portable bitmap) est un fichier qui contient une image noir et blanc pixelisée.

```
# fichier exemple.pbm
P1
# indique le format pbm
5 # indique la largeur de l'image
3 # indique la hauteur de l'image
1 1 1 1 1 # une premiere ligne noire
0 0 0 0 0 # une deuxieme ligne blanche
1 1 1     # debut troisieme ligne
1 1      # fin troisieme ligne
```

En plus d'une extension `.pbm`, un fichier PBM doit commencer par `P1`. Ensuite, le fichier contient seulement des nombres et des commentaires. Comme en Python, tout ce qui suit un `#` sur une ligne est un commentaire. Les nombres sont séparés par des espaces ou des passages à la ligne. Les deux premiers nombres sont la largeur et la hauteur de l'image. Les nombres suivants codent les couleurs de chaque pixel, ligne par ligne et de gauche à droite, et prennent la valeur 0 pour un pixel blanc et 1 pour un pixel noir.

### Exercice 1 (Lire et modifier un fichier PBM, ☆)

1. Depuis votre navigateur de fichier, à l'aide d'une application de visualisation d'image de votre système, ouvrez les fichiers `space-invader-pieuvre.pbm` et `space-invader-soucoupe.pbm` (double-clic pour l'application par défaut, ou clic droit pour choisir l'application parmi celles suggérées). Sur les machines de la salle TP, vous devriez trouver l'application `lrfanview`, qui gère ce format correctement. Vous devriez voir les images suivantes :



Si votre système ne possède pas une application par défaut pour ouvrir les fichiers PBM, vous devrez utiliser la fonction `voir_fichier` du module `viewer.py` fourni dans l'archive.

```
1 from viewer import voir_fichier
2 voir_fichier('space-invader-pieuvre.pbm')
3 voir_fichier('space-invader-pieuvre.pbm' , quadrillage=False)
```

- À l'aide d'un éditeur de texte (sous Windows, préférez WordPad à NotePad), ouvrez le fichier `space-invader-pieuvre.pbm`. Quelles sont les dimensions de l'image ? Si vous remplacez le 9 par un 8, comment est modifiée l'image ? Et le 13 par un 12 ? Comment faut-il modifier le fichier pour obtenir l'image ci-dessous ?



- Créez un fichier `tp8.py` dans le répertoire contenant tous les fichiers extraits de l'archive. Ajoutez-y le code ci-dessous

```
1 from tp8_utils import *
2 M = pbm_vers_matrice('space-invader-pieuvre.pbm')
3 print(M)
```

Exécutez votre code. À quoi correspond le contenu de la matrice `M` ?

- Complétez les `...` dans le code ci-dessous de sorte que quand votre programme s'exécute il crée un fichier `space-invader-pieuvre2.pbm` qui contiendra une pieuvre qui ferme son oeil gauche comme vu plus haut.

```
1 from tp8_utils import *
2
3 M = pbm_vers_matrice('space-invader-pieuvre.pbm')
4 M[ ... ][ ... ] = ....
5 matrice_vers_pbm(M, 'space-invader-pieuvre2.pbm')
```

Attention, il y a un bord blanc dans l'image `space-invader-pieuvre.pbm`.

□

## Exercice 2 (Afficher une image PBM en mode texte, \* →\*\*)

Dans le fichier `tp8.py`, écrivez une fonction `affiche_matrice_booleens(M,plein,vide)` qui prend en argument une matrice `M`, un caractère plein et un caractère vide et qui affiche la matrice sous forme rectangulaire en remplaçant `True` par `False` par `vide`. Par exemple, si `M` est la matrice obtenue en lisant `space-invader-pieuvre.pbm`, `affiche_matrice(M,'#','.')` affichera le texte ci-dessous.

```
.....
...#...#...
..#####...
.###.###.##..
.#####.
.#.#####.#.
.#.#....#.#.
...#...#...
.....
```

□

### Exercice 3 (Une armée tentaculaire, \*\*)

1. On veut maintenant construire l'image suivante



On va pour cela utiliser un accumulateur pour construire chaque ligne puis faire leur somme par concaténation. Cependant, l'opération de concaténation de matrices n'est pas prédéfinie en Python, et il va falloir commencer par la définir.

Écrivez une fonction `ajoute_horizontal(M1,M2)` qui prend en arguments deux matrices  $M_1$ ,  $M_2$  de dimensions respectives  $l_1$  colonnes  $\times$   $h$  lignes et  $l_2$  colonnes  $\times$   $h$  lignes et qui renvoie la matrice de dimension  $l_1 + l_2$  colonnes  $\times$   $h$  lignes obtenue en concaténant  $M_2$  à droite de  $M_1$ . Testez votre fonction en exécutant le code ci-dessous puis en ouvrant le fichier `space-invader-3-pieuvres.pbm`.

```
1 from tp8_utils import *
2
3 def ajoute_horizontal(M1,M2) :
4     # votre code ici
5
6 M = pbm_vers_matrice('space-invader-pieuvre.pbm')
7 M2 = ajoute_horizontal(M , M)
8 M3 = ajoute_horizontal(M2 , M)
9 matrice_vers_pbm(M3 , 'space-invader-3-pieuvres.pbm')
```

2. Écrivez une fonction `repete_horizontal(M,n)` qui prend en argument une matrice  $M$  et un entier  $n > 0$  et qui renvoie la matrice  $M$  répétée  $n$  fois horizontalement. En utilisant cette fonction, créez un fichier `space-invader-20-pieuvres.pbm` qui contient une rangée de 20 pieuvres.

3. Écrivez une fonction `ajoute_vertical(M1,M2)` qui prend en arguments une matrice  $M_1$  de dimension  $l_1 \times h_1$  et une matrice  $M_2$  de dimension  $l_2 \times h_2$  et qui renvoie une matrice  $M$  de dimension  $\max(l_1, l_2) \times (h_1 + h_2)$  obtenue en concaténant  $M_2$  en dessous de  $M_1$  en les alignant sur la gauche.

Optionnel : au lieu d'aligner à gauche, on pourra centrer les matrices l'une sous l'autre.

Testez votre fonction avec le code suivant.

```
1 M1 = pbm_vers_matrice('space-invader-pieuvre.pbm')
2 M2 = pbm_vers_matrice('space-invader-soucoupe.pbm')
3 M3 = ajoute_vertical(M1 , M2)
4 matrice_vers_pbm(M3 , 'space-invader-vertical.pbm')
```

4. Créez le fichier `space-invader-army.pbm` qui contient l'image représentée plus haut.

□