

# TP Space Invaders

## Espace de jeu

Définir en Python une classe `Space` dont le constructeur fournit 2 attributs aux instances de cette classe : `nb_l`, et `nb_c` de type entiers, et signifiant les nombres de lignes et de colonnes de l'espace de jeu.

## Affichage de la grille

Ajouter à cette définition de classe un attribut d'objet `grille`: une liste initialement vide. Écrire, dans le constructeur de la classe, à l'aide de deux boucles `for ... in ...` imbriquées, un algorithme permettant d'initialiser la grille comme un tableau de `nb_l` lignes et `nb_c` colonnes. Les cellules du tableau contiennent initialement un unique caractère " " (espace).

Par exemple, à l'issue de l'instanciation d'un espace de 4 lignes et 10 colonnes :

```
>>> s = Space(4, 10)
```

sa grille vaudra :

```
>>> s.grille
[[' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ']]
```

## Invaders

Définir en Python une classe `Invader` dont le constructeur déclare l'attribut `c` permettant de spécifier le type d'envahisseur. Ce sera un simple caractère, qui vaudra `#` par défaut.

## Affichage d'un invader

Ajouter à cette classe une méthode `__str__` (méthode automatiquement appelée par la fonction `str`), et qui renvoie une chaîne de caractères représentant l'envahisseur. Dans un premier temps, ce sera simplement l'unique caractère défini à la question précédente.

Par exemple, à l'issue de l'instanciation d'un envahisseur, on pourra l'afficher ainsi :

```
>>> i = Invader()
>>> str(i)
#
```

## Envahisseurs dans la grille

On souhaite faire apparaître en haut de l'espace une ligne d'envahisseurs !

Modifier la construction de la grille de l'espace de jeu en remplaçant les espaces de la première ligne par des envahisseurs (ces cellules de la grille ne contiennent alors plus un caractère, mais un objet de type `Invader`).

## Afficher l'espace de jeu

Il est temps d'afficher l'espace de jeu :

Ajouter à la classe `Space` une méthode `__repr__` (cette méthode, automatiquement appelée par la fonction `print`, n'attend aucun autre argument que l'objet lui-même, et doit renvoyer une chaîne de caractères).

À présent si l'on crée un espace de jeu de 6 lignes et 10 colonnes :

```
>>> s = Space(6, 10)
```

On peut l'afficher avec un simple `print` :

```
>>> print(s)
#####

>>>
```

## Cadre de jeu

Rajouter un « cadre » autour le l'espace pour qu'il soit mieux visible à l'écran (utiliser les caractères `┌` `┐` `└` `┘` `─` `│`). L'espace de jeu doit à présent avoir cette apparence :

```
┌#####┐
│       │
│       │
│       │
│       │
└#####┘
```

## Tir

Ajouter à la classe `Space` une méthode `tirer` qui permet de lancer un missile depuis le sol, à partir d'une position donnée. Le missile traverse alors l'espace de jeu jusqu'à toucher un envahisseur (On utilisera la fonction `time.sleep` pour ralentir l'affichage au fur et à mesure que le missile monte).

Par exemple, si on déclenche un tir depuis la position 2 :

```
s.tirer(2)
```

L'espace de jeu prend les apparences successives suivantes :



# Objet canon

1. Créer une classe `Canon`, représentant le canon qui tire les missiles depuis le sol. Le canon possède un attribut `pos`, qui mémorise sa position dans la grille. Ne pas oublier de lui donner une méthode `__str__` !
2. Instancier un unique canon dans le constructeur de l'espace de jeu.

# Canon dans la grille

1. Modifier la construction de la grille pour y faire figurer le canon.
2. Modifier la méthode tirer pour que le tir parte depuis la position du canon.

## Utilisation du clavier

Utiliser le module `keyboard` pour déplacer le canon de gauche à droite et déclencher les tirs.

Il faudra auparavant installer ce module :

```
pip install keyboard
```

Exemple d'utilisation (<https://newbedev.com/python-keyboard-on-press-python-code-example>)

Documentation complète (<https://github.com/boppreh/keyboard#api>)

Nous n'utiliserons que la fonction `keyboard.read_key()` de la manière suivante :

```
keyboard.read_key(suppress=True)
time.sleep(0.2) # permet d'éviter de détecter plusieurs appuis sur la touche
```

Grâce à cette fonction, le programme s'interrompt jusqu'à ce que l'utilisateur appuie sur une touche, puis la valeur de la touche est renvoyée.

## Score et animation

Ajouter un attribut `score` à la classe `Space`, l'incrémenter à chaque destruction d'envahisseur et l'afficher en dessous de l'espace de jeu. Faire descendre les envahisseurs progressivement au cours de la partie (un à un ou en bloc) : le jeu doit rester jouable !!