

## Consignes.

- Comment livrer mon code ?  
Si mon nom de famille est **yunès** et que je souhaite rendre le travail du tp 3 alors je dois construire une archive au format ZIP de nom **yunes-tp3.zip**, le nom ayant été converti en minuscules, sans accent et sans espace (*ie* : **de Gaulle** doit être traduit en **degaulle**). Si vous avez un homonyme rajoutez des initiales, par exemple **yunesjb-tp1**. Respectez votre nommage pour tout le semestre.
- Que mettre dans l'archive ?  
Le code source (et uniquement le code source) correspondant à l'exercice.

## Exercice 1

Créer une classe Python de nom `Column` permettant de représenter une colonne d'objets (le tout dans un fichier de nom `column.py`, c'est donc un module python). La hauteur de la colonne étant fixée à la construction de l'instance et chaque case ne doit contenir que des `None`. De plus il faut que cette colonne s'affiche «normalement» :

```
+---+  
|   |  
+---+  
|   |  
+---+
```

Ainsi elle doit pouvoir fonctionner avec le code suivant (`main.py`) :

```
import column  
c = column.Column(2)  
print(c)
```

## Exercice 2

On réutilisera la classe `Column` de l'exercice précédent. Créer une classe `Pawn` (module `pawn`) permettant de représenter un pion (un `X` un `O`) de sorte que lorsqu'on le dépose sur la colonne, il tombe jusqu'à la dernière case ou avant une case déjà occupée :

```
import column  
import pawn  
c = column.Column(2)  
c.drop(pawn.Pawn('X'))  
c.drop(pawn.Pawn('X'))
```

qui produira :

```
+---+  
| X |  
+---+  
|   |  
+---+  
  
+---+  
|   |  
+---+  
| X |  
+---+  
  
+---+  
| X |  
+---+  
| X |  
+---+
```

### Exercice 3

On réutilisera les codes précédents. Créer une classe `Grid` (module `grid`) contenant  $n$  colonnes de hauteur  $m$  de sorte que l'on puisse poser un pion sur une colonne choisie et que celui-ci glisse en tombant d'une case à la fois :

```
import grid
import pawn
g = grid.Grid(2,3)
p1 = pawn.Pawn('X')
g.drop(0,p1)
```

qui produira :

```
+---+---+---+
| X ||  ||  |
+---+---+---+
|  ||  ||  |
+---+---+---+

+---+---+---+
|  ||  ||  |
+---+---+---+
| X ||  ||  |
+---+---+---+
```

### Exercice 4

Compléter les classes de sorte que l'on puisse jouer au jeu puissance 4 (vous trouverez facilement les règles de ce jeu).  
Livrables : l'archive contenant les codes `grid.py`, `column.py`, `pawn.py`, `main.py`, `p4.py`.