

NSIT	03f	Structure de données
Programmation orientée objet (POO)		
Défi 'Space Invaders'		1/2

*Votre vaisseau spatial rencontre une flotte de vaisseaux extraterrestres qui s'avance vers lui. Il doit détruire ces vaisseaux en tirant des balles pour les faire exploser.*

### Objectifs :

- + Prendre en main la bibliothèque Pygame.
  - + Comprendre les éléments d'un code proposé.
  - + Utiliser la programmation orientée objet pour créer les éléments du jeu.
- Pour les images du jeu, on pourra : soit choisir les images proposées, soit utiliser des images libres de droits (exemples sur [flaticon.com](http://flaticon.com)), soit fabriquer ses propres images.



### Mission 1 : Comprendre le code proposé.

Le fichier "**space\_invader\_1.py**" propose quelques éléments de code que vous n'aurez pas à modifier.

Vous pouvez vous aider de la documentation du site de pygame:

<https://www.pygame.org/docs/>

### 1°) Création de la fenêtre :

Que permet de faire la fonction **pygame.display.set\_mode()** ? (ligne 10)

Que permet de faire la fonction **pygame.display.set\_caption()** ? (ligne 11)

Que permet de faire la fonction **pygame.image.load()** ? (ligne 13)

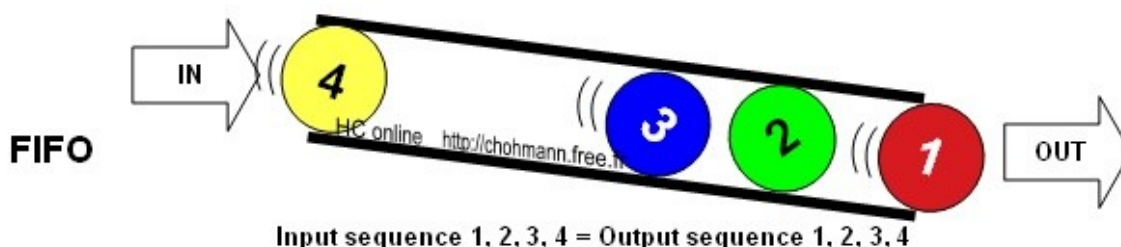
Quelle instruction permet d'afficher une image sur l'écran ? (ligne 23)

Quels sont les formats d'images supportés par pygame ?

Corriger les erreurs du fichier.

### 2°) Gestion des événements :

Un événement peut être une entrée clavier (soit l'appui, soit le relâchement d'une touche), le déplacement de votre souris, un clic... Il faut savoir que chaque événement créé est envoyé sur une **file** (ou *queue*), en attendant d'être traité, comme lorsque l'on fait la queue au supermarché !



La ligne 23 crée une boucle qui va parcourir cette file d'événements, puis tester chaque événement enregistré en interrogeant son type (ligne 24)

```
22      ### Gestion des événements ###  
23      for event in pygame.event.get(): #  
24          if event.type == pygame.QUIT :
```

A quel événement correspond le type `pygame.QUIT` ? (ligne 24) \_\_\_\_\_

A quel événement correspond le type `pygame.KEYDOWN` ? (ligne 29) \_\_\_\_\_

Quel est le type d'événement correspondant à l'appuie sur la flèche droite ? \_\_\_\_\_

Quel est le type d'événement correspondant au relâchement d'une touche ? \_\_\_\_\_

A quoi sert la variable `running` définie ligne 16 ? \_\_\_\_\_

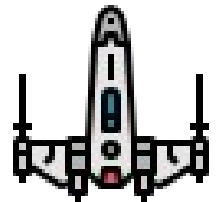
### **Mission 2 : Déplacement du vaisseau :**

Le vaisseau du joueur se déplace horizontalement, en bas de l'écran (800 x 600) à 500 pixels du haut.

Ce déplacement est réalisé par les flèches droites et gauches.

Quand la flèche droite est pressée, le vaisseau part vers la droite et ne s'arrête qu'au bord de l'écran.

Il change de sens lorsque la touche gauche est pressée. visualiser la vidéo.



Ouvrir le fichier "**space\_invader\_2.py**" (à ne pas modifier)

Il est lié au fichier "**space.py**" sur lequel vous allez travailler.

Quelle instruction permet de réaliser ce lien ?

Le joueur est initialisé par l'instruction `player = space.Joueur()` ,

Expliquer la présence du mot `space` : \_\_\_\_\_

Quelle est la classe de l'objet `player` ? \_\_\_\_\_

Quels sont les attributs de l'objet `player` utilisés dans le fichier "**space\_invader\_2.py**" ?

Quelle méthode de l'objet `player` est utilisée dans le fichier "**space\_invader\_2.py**" ?

Coder maintenant cette classe d'objet dans le fichier "**space.py**" pour que tout fonctionne.

### **Mission 3 : Tir de balles :**

Le joueur dispose d'une balle attachée à son vaisseau, quand il tire avec la touche Espace, la balle monte verticalement. Quand elle atteint le haut de l'écran, elle réapparaît dans le vaisseau. Elle n'est pas disponible pendant qu'elle monte. (Visionnez la vidéo).



Le fichier "**space\_invader\_3.py**" contient le moteur de jeu permettant au joueur de tirer.

Il reste à **implémenter** une classe d'objets `balle()` à ajouter au fichier "**space.py**".

NSIT	03f	Structure de données
Programmation orientée objet (POO)		
Défi 'Space Invaders'		2/2

Son **interface** est la suivante :

La variable de type Joueur correspondant au tireur est transmise lors de l'initialisation de la Balle.

Elle possède 5 attributs :

tireur : la variable Joueur qui a tiré la Balle  
 depart : La position horizontale où la Balle a été tirée  
 hauteur : La position verticale de la Balle  
 image : L'image créée par pygame à partir du fichier **"balle.png"**  
 etat : "chargee" la Balle est prête à être tirée  
 "tirez" la Balle a été tirée et se déplace verticalement.

Balle
tireur : Joueur depart : int hauteur : int image : pygame.image etat : string "chargee", "tirez"
bouger()

Elle dispose d'une méthode :

bouger() pour se déplacer quand elle a été tirée

On ajoutera aussi une méthode tirer() à l'objet Joueur.

#### **Mission 4 : Apparition des ennemis :**

Les ennemis apparaissent en haut de l'écran, dans une position aléatoire horizontalement. Puis ils descendent verticalement.

Il y a deux types d'ennemis, qui ont des images et des vitesses différentes.



Comme dans la mission précédente, il s'agit de partir du fichier **"space\_invader\_4.py"** dans lequel le moteur du jeu est programmé.

Les modifications ont été effectuées ligne 20 à 23 (création des ennemis) et lignes 56 à 59 (déplacement et dessin des ennemis)

Il reste à ajouter une classe Ennemi au fichier **"space.py"** dont on précise l'interface.

Elle dispose d'une variable de classe NbEnnemis dénombrant le nombre d'ennemis affichés.

Ennemi
depart : int hauteur : int type : int image : pygame.image vitesse : int
avancer()

### **Mission 5 : Destruction des ennemis :**

Quand un ennemi est touché par la balle, il disparaît et un nouveau vaisseau apparaît en haut de l'écran.

La balle retourne dans le vaisseau du joueur. Le score du joueur est augmenté d'un point.

Seules les lignes 50 à 54 ont été ajoutées dans le fichier **"space\_invader\_5.py"** (remarquez à quel point la programmation est proche du langage naturel).

Il vous reste à ajouter, dans le fichier **"space.py"**,  
à la classe Balle une méthode `toucher(ennemi)`  
à la classe Ennemi une méthode `disparaître()`  
à la classe Joueur un attribut `score` et une méthode `marquer()`

Les trois classes ont les propriétés suivantes (éventuellement d'autres que vous avez ajoutés).

Joueur
<code>position : int</code> <code>image : pygame.image</code> <code>sens : string "droite", "gauche", "O"</code> <code>score : int</code>
<code>deplacer()</code> <code>tirer()</code> <code>marquer()</code>

Balle
<code>tireur : Joueur</code> <code>depart : int</code> <code>hauteur : int</code> <code>image : pygame.image</code> <code>etat : string "chargee", "tiree"</code>
<code>bouger()</code> <code>toucher(Ennemi)</code>

Ennemi
<code>depart : int</code> <code>hauteur : int</code> <code>type : int</code> <code>image : pygame.image</code> <code>vitesse : int</code>
<code>avancer()</code> <code>disparaître()</code>

### **Mission 5 : A vous de jouer !**

Proposez et réalisez des améliorations au jeu :

- + Diversifiez les types de balles et d'ennemis
- + Système de points de vie
- + Jouez sur la vitesse de chute des ennemis
- + Des niveaux, des paliers ...
- + Affichage du score

...

