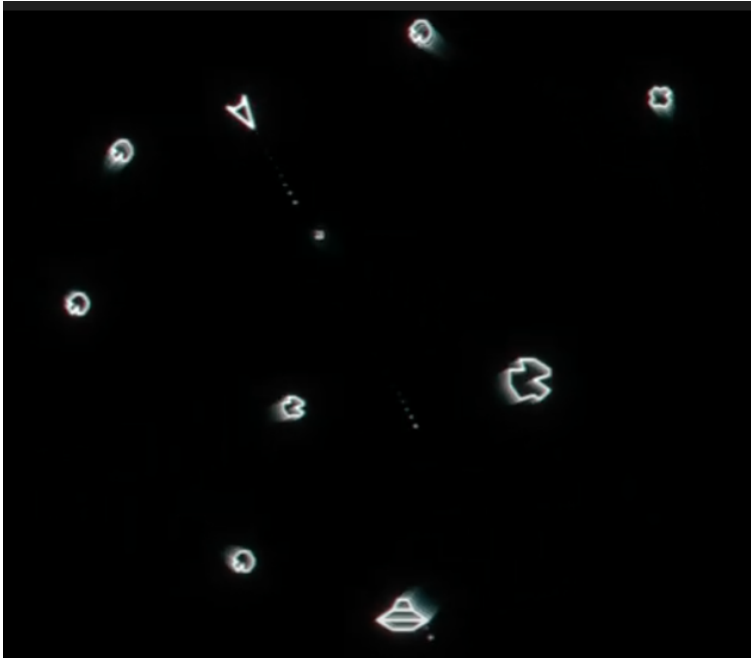


Projet 1 : Astéroïdes

Il s'agit de réaliser une implémentation de base du jeu Asteroids <https://fr.wikipedia.org/wiki/Asteroids> dans le but d'illustrer les idées de la programmation objet vues en cours et exercices

On peut voir sur Youtube une courte vidéo du jeu de l'époque (1979)

<https://www.youtube.com/watch?v=i-Gs01omJyI>



Il existe différents GUI (Graphical User Interface) permettant d'implémenter ce jeu en Python :

1. Le module `tkinter`
2. Le module `pygame`
3. Un module plus récent `arcade`

1 Règles du jeu

Au début du jeu un vaisseau spatial de forme triangulaire apparaît au centre de l'écran, immobile ainsi que quatre "gros"

astéroïdes, aux quatre coins de la fenêtre et se dirigent vers le vaisseau "lentement"

Les astéroïdes sont de trois tailles : gros, moyen et petit

Le joueur peut faire tourner le vaisseau sur lui même avec les touches 'Right' et 'Left', le faire accélérer avec la touche 'Up'

Si un astéroïde de n'importe quelle taille heurte le vaisseau le joueur perd une vie et le joueur n'a que cinq vies

Si le joueur tire des missiles sur un gros astéroïde et si ce dernier est touché alors il est cassé en deux astéroïdes moyens plus rapides qu'un gros astéroïde et le joueur augmente son score de 10 points

Si une missile touche un astéroïde moyen il est cassé en deux astéroïdes petits plus rapides qu'un astéroïde moyen et le joueur augmente son score de 20 points

Si une missile touche un petit astéroïde alors ce dernier disparaît et le joueur gagne 100 points

Le score du joueur ainsi que le nombre de vies du joueur sont affichés à l'écran

2 Matériel

Récupérer le dossier compressé `projet_asteroids` contenant :

1. un dossier `asteroides` contenant 3 fichiers python :
 - (a) un fichier `utilitaires.py`
 - (b) un fichier `elements.py`
 - (c) un fichier `jeu.py`
2. un dossier `ressources` contenant deux sous dossiers `images` et `sons`

- (a) Le sous-dossier **images** contient les images du fonds d'écran du vaisseau, d'un astéroïde et le sprite d'une explosion
- (b) Le sous-dossier **sons** contient les sons d'accélération et d'explosion

Renommer le dossier avec le nom de famille des élèves du groupe et rendre le dossier compressé via l'E.N.T du lycée la première semaine de la rentrée après les vacances de la Toussaint

3 Préparation

Plusieurs TP sont nécessaires avant le début du projet proprement dit (début Octobre) pour se familiariser avec la programmation de certains effets (langage objet de Pygame, déplacement d'une image, détection d'une collision, sprite, rotation et agrandissement/réduction d'une image)

A la fin de cette série de TP on a un **seul fichier .py** contenant le code, et ce fichier au fur et à mesure qu'on rajoute des effets devient de moins en moins lisible même s'il y a des commentaires

D'où la nécessité de découper ce fichier en des unités logiques réutilisables :

- On crée une classe **Jeu** dont les **attributs sont tous les éléments nécessaires au démarrage d'une partie** :
 1. la fenêtre graphique
 2. les images chargées en mémoire vive
 3. les sons
 4. l'horloge de Pygame
 5. une instance de la classe Vaisseau.

6. une liste asteroïdes contenant trois instances de la classe Asteroïdes

- Les **méthodes** de la classe Jeu sont toutes les "parties logiques" dans la boucle infinie, la boucle infinie y compris

1. la méthode "publique" `boucle_infinie()` contenant les trois méthodes suivantes

2. la méthode "privée" `_capturer_evt()` qui "guettent" les événements et au cas où ils ont lieu il y aura une modification d'un des attributs de l'objet Vaisseau ou d'un des objets Asteroïde ou d'un des objets Missile :

- (a) clic de souris pour quitter le jeu à tout moment
- (b) appuyer et relâcher la touche UP pour accélérer et décélérer le vaisseau
- (c) appuyer sur les touches RIGHT et LEFT pour faire tourner le vaisseau sur lui même à droite ou à gauche
- (d) appuyer sur la barre ESPACE pour tirer des missiles

3. la méthode "privée" `_mettre_a_jour()` qui va mettre à jour tous les attributs concernés par la modification d'un des attributs.

Exemple :

si on appuie sur UP l'attribut `vitesse` est modifié dans la capture ainsi :

la vitesse est augmentée d'une petite quantité d'accélération

```
self.vaisseau.vitesse += self.vaisseau.acc
```

Il faut mettre à jour ensuite les attributs qui influencent directement la position de l'image dans la fenêtre

```
self.vaisseau.position += self.vaisseau.vitesse
```

4. la méthode "privée" `_dessiner()` prenant en compte toutes les modifications précédentes "blittent" les images aux endroits adéquats

- Les classes `Vaisseau`, `Asteroide`, `Missile`, héritent de la classe `Animation` et sont dans un module `elements` à part, qui est importé (il y a eu un TP là dessus)
- Un module `utilitaires` contient des fonctions pour charger les images , charger les sons et afficher du texte dans la fenêtre graphique

4 A vous de faire

1. Créer une classe `Missile` qui hérite de la classe `Animation`. Chercher sur le web une image de missile .png (ou la faire vous même) de taille 20 x 20

Chercher un son de tir .wav ou .mp3

Cette classe en dehors du constructeur n'aura qu'une méthode la méthode `deplacer()`.

C'est l'intérêt de **l'héritage en programmation objet** : pour "dessiner" une missile sur la fenêtre graphique on utilise la méthode `dessiner()` de la classe `Animation`

2. Par contre particulariser (on dit surcharger) la méthode `deplacer()` de la classe `Missile` de telle sorte que les missiles ne se déplacent pas sur un tore comme le vaisseau et les astéroïdes

3. Ajouter un attribut `qte_missiles` le nombre maximal de missiles que le vaisseau peut transporter par exemple 100

4. Ajouter un attribut `missiles` de type list à la classe `Vaisseau`. Cette liste vide au départ contiendra toutes les missiles tirées par le vaisseau.
5. Ajouter une méthode `tirer(image_missile,son_tir)` à la classe `Vaisseau` qui à condition que la quantité de missiles ne soit pas nulle, crée un objet de type `Missile`, l'ajoute à la liste `missiles`, joue un son de tir, règle la position et la vitesse du missile créé de telle sorte qu'il parte vers l'avant du vaisseau avec une vitesse supérieure à celle du vaisseau
6. Dans le module `jeu` ajouter à la classe `Jeu` un attribut `missile` qui référence l'image chargée en mémoire d'une missile, puis un attribut `tir_son` qui référence le son d'un tir
7. Dans le module `jeu` ajouter dans la méthode `_capturer_evt()` de la classe `Jeu` la capture de l'évènement "appui sur la barre espace" qui déclenche la méthode `tirer()` de l'attribut `vaisseau` de la classe `Jeu`
8. Dans le module `jeu` ajouter dans la méthode `_mettre_a_jour()` de la classe `Jeu` le déplacement des missiles puis supprimer de l'attribut `missiles` de l'objet `vaisseau` les missiles qui sortent de la fenêtre
9. Toujours dans la méthode `_mettre_a_jour()` de la classe `Jeu` gérer la collision missiles - astéroïdes à la manière de la collision vaisseau - astéroïdes
10. Dans le module `jeu` ajouter dans la méthode `_dessiner()` de la classe `Jeu` faire dessiner les missiles
11. Vous observerez que la collision entre un astéroïde et le vaisseau est compté plusieurs fois du fait de la vitesse de

rafraîchissement de la page. Pour corriger ce problème introduire une déviation de la trajectoire du vaisseau résultant du choc avec l'astéroïde. Ainsi une collision n'enlèvera désormais qu'une seule vie au vaisseau

12. Lorsque le vaisseau n'a plus de points de vie il disparaît de l'écran et l'appui sur la touche UP ne doit plus faire jouer un son d'accélération
13. Ajouter une méthode `scission()` à la classe `Asteroide` de telle sorte qu'en cas de collision par une missile un "gros astéroïde" est transformé en "moyen", un "moyen" en "petit" et un "petit" "disparaît" (il est enlevé de la liste)

Augmenter "suffisamment" la vitesse des asteroides au cours de la scission de telle sorte que les petits astéroïdes sont "dangereux" pour le vaisseau (pour la difficulté du jeu)

5 Créativité

Créer une classe `Soucoupe` (attributs + méthodes + images + sons) (aller voir sur le Web pour vous inspirer du rôle des soucoupes volantes dans le jeu)