

Programmer un jeu vidéo avec Pyxel : 6/6

Ajouter des graphismes

Nous repartirons dans ce tutoriel du script réalisé précédemment : *tuto_pyxel_5.py*

Faire Enregistrer Sous pour renommer à présent ce script en *tuto_pyxel_6.py*

1. Fichier ressource...

Dans pyxel, les ressources graphiques et son sont regroupées dans un fichier pyxres. Celui-ci contient 4 rubriques : graphiques, tilemaps, sons et musiques.

Pour que les ressources soient disponibles pour notre programme, il faut d'abord les charger avec l'instruction ***pyxel.load*** :

```
# chargement des images
pyxel.load("shooter_1NSI.pyxres")
```

Il faudra bien sûr prendre soin que le fichier *shooter_1NSI.pyxres* soit dans le même répertoire que le fichier source (sinon il faut indiquer son chemin d'accès).

Jalon 1 : Ajouter la ligne pour charger le fichier pyxres et vérifier que son exécution ne provoque pas d'erreur (le fichier est donc bien présent et chargé en mémoire).

2. Afficher les images

Au lieu de dessiner des rectangles avec `pyxel.rect`, on va maintenant afficher les images en recopiant les graphismes du fichier ressource à l'écran. On dit qu'on « blitte » (c'est-à-dire copie les données graphiques) du fichier ressource vers l'écran. Cela se fait avec la fonction ***pyxel.blit(x, y, img, u, v, w, h, [colkey], [rotate], [scale])*** où ***x*** et ***y*** sont les coordonnées sur l'écran, ***img*** est le numéro de la banque d'image du fichier ressource (de 0 à 2), ***u*** et ***v*** les coordonnées du point en haut à gauche dans les ressources graphiques (se repérer à ce qu'indique la barre de titre lorsqu'on déplace la souris dans l'éditeur pyxres), ***w*** et ***h*** la largeur et la hauteur (ces deux paramètres peuvent être négatifs si on veut représenter l'image en miroir (utile quand l'objet représenté change d'orientation)) et ***colkey*** (paramètre optionnel) indique quelle couleur est considérée comme transparente (les pixels de cette couleur ne seront pas copiés).



Dans le fichier *shooter_1NSI.pyxres*, repérer les coordonnées (*u* et *v*) des images correspondant au vaisseau, au tir et aux ennemis.

Modifier ensuite les lignes de la fonction ***draw*** pour remplacer le tracé des rectangles par un blitt de l'image correspondante :

- Pour le vaisseau, remplacer ***pyxel.rect(vaisseau_x, vaisseau_y, 8, 8, 1)*** par ***pyxel.blit(...***
- Pour les tirs, remplacer ***pyxel.rect(tir[0], tir[1], 1, 4, 10)*** par ***pyxel.blit(...***
- Pour les ennemis, remplacer ***pyxel.rect(ennemi[0], ennemi[1], 8, 8, 8)*** par ***pyxel.blit(...***

Pour les explosions, on laissera nos cercles, mais il aurait été possible d'utiliser des images d'explosion à la place.

Jalon final : Vérifier qu'on a bien l'affichage des graphismes à la place des rectangles

Le jeu est terminé. On pourrait ajouter la possibilité de recommencer après un game over, d'autres ennemis, le calcul et l'affichage d'un score, ... Maintenant c'est à vous de jouer !