

Le jeu (pas de classe, ça va être dans le main) :

```
titreJeu = string : le titre
iconeJeu = string : chemin vers l'icône du jeu
tailleEcran = largeurEcran, hauteurEcran = couple d'entiers
ecran = pygame.display.set_mode(tailleEcran)

horloge = pygame.time.Clock()
FPS = int

# Définition des objets
exemple : mario = Object(['catrunx4-0.png', 'catrunx4-1.png', 'catrunx4-2.png',
'catrunx4-3.png', 'catrunx4-4.png', 'catrunx4-5.png'])

# Définition des scènes
exemple : scene1 = Scene(mario)

scenes = liste de scènes
scenecourante = index de la liste scenes

active = booléen

update():
    # Pour chaque calque de la scène courante
    # Pour chaque objet du calque
        # On augmente le compteur de temps pour chaque objet et on calcule l'animation
        # Objet.frame()
        # On imprime l'objet sur l'écran
    # On affiche les modifications

Boucle du jeu (tant que active):
    # Evenements
    # Update
    # Horloge
```

class Scene:

```
tailleScene = largeurScene, hauteurScene = couple d'entiers

calques = dictionnaire nomCalque : liste d'objets

camera = camera_x, camera_y = couple d'entiers

init (objets) :
    # Si le paramètre est une liste d'objets, définir le calque de base
    # Si le paramètre est un objet simple, le rajouter dans le calque de base
```

class Object:

```
sprite : dictionnaire nomAnimation : liste de Surface pygame
cptframe = int : compteur avant de changer l'image courante
imageCourante = int
animCourante = nom de l'animation
```

vitesseAnim = int : Vitesse en image/secondes

direction = float : en degrés

visible = booléen

opacite = float

suivreScene = booléen

parallax = p_x, p_y = couple de float (ratio)

tags = liste de string : regroupements des objets

init (image):

- # Si le paramètre est une liste de chemins vers des images

- # Pour chaque chemin vers image, crée une Surface et ajout dans une liste

- # Définition de l'animation courante

- # Création du rect selon la première image

- # Si le paramètre est un seul chemin vers une image

- # On ajoute l'image dans l'animation courante

- # Création du rect selon l'image

frame(): # Calcul de l'animation

- # Si on a atteint la vitesse de l'animation

- # Remise à zéro de cptframe

- # Changement de l'image courante