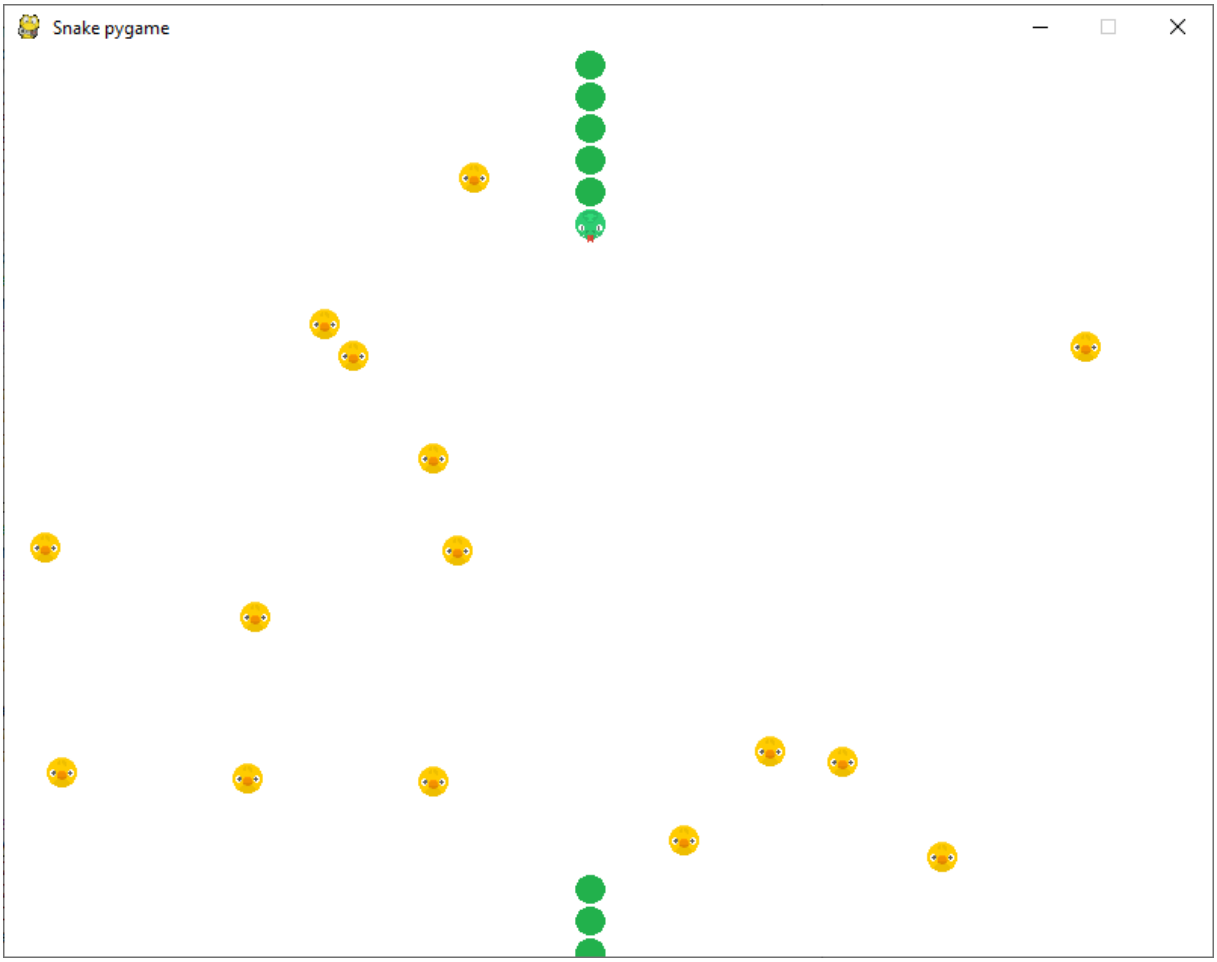
Pygame Snake

Le jeu



Le serpent sera constitué d'une tête ayant pour image serpent_tete.png, et d'un corps composé d'anneaux ayant pour image serpent_anneau.

Une seule classe Bloc permettra de créer la tête et les anneaux du serpent. Au départ le serpent possède x anneaux.

On modifie le sens de déplacement de la tête du serpent à l'aide des touches fléchées.

Ici le serpent doit manger des poussins. A chaque collision avec un poussin, la queue du serpent s'agrandit de x anneaux.

Il y a 15 poussins sur la scène, placés de façon aléatoire. Lors d'une collision avec la tête du serpent, le poussin change de place.

Un poussin a pour image poussin.png.

Analyse du programme à réaliser

Nous repartons du programme précédant Lutin1 (voir document Pygame Sprite)

La tête et les anneaux du serpent

La tête et les anneaux seront créés à partir de la classe Bloc.

Les paramètres à passer au constructeur de cette classe Bloc, chargée de créer un morceau du serpent, sont :

- Les coordonnées d'affichage de l'image du bloc créé : tête ou anneau
- L'image à afficher. Celle-ci n'est pas la même pour la tête et pour les anneaux.

Avant de créer un anneau, nous devons calculer ses coordonnées en fonction des coordonnées de la tête ou des coordonnées du dernier anneau créé et en fonction de son rang lors de cette création.

Chacune des images occupent un carré de 20 pixels de côté. Les éléments du serpent sont espacés de 1 pixel.

Nous définissons 5 variables :

anneau_L = 20 #largeur d'un anneau

anneau_H = 20 #hauteur d'un anneau

anneau_E = 1 #ecart entre 2 anneaux

dep_x = anneau_L+anneau_E #déplacement en x

dep_y = 0 #déplacement en y

Au départ le serpent va se déplacer de la gauche vers la droite. A chaque déplacement l'abscisse x de la tête est augmentée de dep_x et son ordonnée y est augmentée de dep_y.

D'où les valeurs de départ de dep_x et dep_y. Ces valeurs seront modifiées si le joueur appuie sur une touche fléchée.

Les anneaux se placent derrière la tête ou derrière le dernier anneau créé. Nous utiliserons une fonction pour créer x anneaux d'un seul coup.

Les coordonnées des anneaux à créer sont calculées par la formule :

x = xdepart - dep_x*i

y = ydepart - dep_y*i

xdepart et ydepart sont les coordonnées du dernier élément créé pour le serpent.

Au démarrage se seront les coordonnées de la tête, puis lorsqu'on rajoutera des anneaux après avoir mangé un poussin, ce sera les coordonnées du dernier anneau de la queue.

i est le numéro du nouvel élément ajouté. Si j'ajoute 3 anneaux, i devra varier de 1 à 3.

Pour déplacer les anneaux de la queue, après un déplacement de la tête, nous avons vu dans les autres programmes implémentant ce jeu, qu'il suffisait de disposer d'une liste contenant la référence à tous les anneaux crée, de suprimer le dernier élément de la liste, d'en créer un nouveau aux coordonnées précédentes de la tête et de placer ce nouvel élément en tête de liste.

Nous aurons donc une liste queue, contenant une référence à tous les anneaux créés.

Les poussins

Les poussins seront des objets de type Poussin.

La classe Poussin calculera la position de départ de chaque poussin créé et lui affectera l'image d'un poussin.

Nous devons détecter une collision entre la tête du serpent et les poussins.

Nous crérons donc un groupe lesPoussins, auquel nous ajouterons tous les poussins créés et nous utiliserons ce groupe pour détecter une collision avec la tête du serpent.

Si un poussin est touché, il doit changer de place.

La classe Poussin doit contenir une méthode reset() qui réinitialise les coordonnées du poussin touché.

La fin de partie

Si le serpent se mord la queue, la partie s'arrête.

Nous devons donc détecter une collision entre la tête du serpent et ses anneaux. Nous créerons un groupe lesAnneaux auquel nous ajouterons tous les anneaux créés et nous utiliserons se groupe pour détecter une collision entre la tête et les anneaux.

Le programme

```
import pygame

import random

#----------Nos constantes----------

BLANC = (255, 255, 255)

ECRAN_L = 800

ECRAN_H = 600
```

```
#-------La classe Bloc: anneau et tête serpent ------------
class Bloc(pygame.sprite.Sprite):
    """cette classe représente un morceau du serpent """
    def __init__(self,x,y,monimage):
        super().__init__()
        #Création de l'image représentant le lutin.
        self.image=monimage
        self.rect = self.image.get_rect()
        self.rect.x = x
        self.rect.y = y
#-------La classe Poussin ---------------------------------
class Poussin(pygame.sprite.Sprite):
    """cette classe représente un poussin """
    def __init__(self):
        super().__init__()
        self.image = pygame.image.load("images/poussin.png").convert_alpha()
        self.rect = self.image.get_rect()
        self.rect.y = random.randrange(0, ECRAN_H)
        self.rect.x = random.randrange(0, ECRAN_L)
     #-----------------------------------
    def reset(self):
        """Changement de coordonnées lors d'une collision """
        self.rect.y = random.randrange(0, ECRAN_H)
        self.rect.x = random.randrange(0, ECRAN_L)
#-------Création d'un ou plusieurs anneaux ----------------------------
def creerAnneau(nb,xdepart,ydepart):
    for i in range(1,nb+1):
        x = xdepart - dep_x*i
        y = ydepart - dep_y*i
        anneau=Bloc(x,y,anneau_image)
```

```
    #Ajouter le bloc à nos trois listes de lutins
    queue.append(anneau) #les anneaux du serpent
    lesAnneaux.add(anneau) #les anneaux pour la détection d'une collision avec la tête
    tout_lutins.add(anneau) #tous les lutins pour l'affichage

#-------Nos variables -------------
arret = False
horloge = pygame.time.Clock()
anneau_L = 20
anneau_H = 20
anneau_E = 1
dep_x = anneau_L+anneau_E
dep_y = 0
#-------Créer une fenêtre ---------
pygame.init()
ecran = pygame.display.set_mode([ECRAN_L, ECRAN_H])
pygame.display.set_caption(" Snake pygame ")
#Chargement des images.
#Le chargement des images ne peut se faire que lorsque la fenêtre est créée
tete_image = pygame.image.load("images/serpent_tete.png").convert_alpha()
anneau_image = pygame.image.load("images/serpent_anneau.png").convert_alpha()
#-------Les listes de lutins -----------
#groupe de poussins pour la détection des collisions avec la tête
lesPoussins = pygame.sprite.Group()
#groupe des anneaux pour la détection des collisions avec la tête
lesAnneaux = pygame.sprite.Group()
#groupe de tous les lutins pour l'affichage
tout_lutins=pygame.sprite.Group()
```

```
#Création de la tête n'importe où sur la scène
tete = Bloc(random.randrange(0, ECRAN_H),random.randrange(0,
ECRAN_L),tete_image)
tout_lutins.add(tete)
#liste des anneaux constituant la queue du serpent
queue=[]
#création des anneaux derrière la tête du serpent : ici 1 seul
creerAnneau(1,tete.rect.x,tete.rect.y)
#création des poussins
for i in range(15):
    poussin = Poussin()
    lesPoussins.add(poussin)
    tout_lutins.add(poussin)
#----Boucle principale ---------
while not arret:
    #boucle des évènements
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            arret = True
        if event.type == pygame.KEYDOWN:
            #le déplacement en x doit faire diminuer la valeur de x
            if event.key == pygame.K_LEFT:
                dep_x = (anneau_L + anneau_E) * -1
                dep_y = 0
            #le déplacement en x doit faire augmenter la valeur de x
            if event.key == pygame.K_RIGHT:
                dep_x = (anneau_L + anneau_E)
                dep_y = 0
```

```
        #le déplacement en y doit faire diminuer la valeur de y
        if event.key == pygame.K_UP:
            dep_x = 0
            dep_y = (anneau_H + anneau_E) * -1
        #le déplacement en y doit faire diminuer la valeur de y
        if event.key == pygame.K_DOWN:
            dep_x = 0
            dep_y = (anneau_H + anneau_E)

    #logique du jeu : calcul des nouvelles coordonnées tête
    #ancienne coordonnée de la tête
    x=tete.rect.x
    y = tete.rect.y
    #nouvelle coordonnée tete
    tete.rect.x+=dep_x
    tete.rect.y+=dep_y
    #faire réapparaitre le serpent de l'autre coté s'il atteint un bord
    if tete.rect.x > ECRAN_L:
        tete.rect.x = 0
    if tete.rect.x < 0:
        tete.rect.x = ECRAN_L
    if tete.rect.y > ECRAN_H:
        tete.rect.y = 0
    if tete.rect.y < 0:
        tete.rect.y = ECRAN_H

     #Supprimer le dernier anneau du serpent
    dernier_anneau = queue.pop()
```

```
 #Enlever la référence de l'anneau supprimé dans les deux groupes où elle est
# mémorisée
lesAnneaux.remove(dernier_anneau)
tout_lutins.remove(dernier_anneau)
#La tête est-elle entrée en collision avec un anneau.
#nous plaçons ce teste ici, car la tête est pour l'instant bien séparée
#de la queue, et nous ne risquons pas de fausse détection.
collision_liste = pygame.sprite.spritecollide(tete,lesAnneaux,False)
if len(collision_liste)> 0:
    print("Le serpent s'est mordu la queue")
    arret = True

 #créer un nouvel anneau et le mettre en tête de liste
anneau=Bloc(x,y,anneau_image)
queue.insert(0,anneau)
#ajouter ce nouvel anneau dans les deux groupes qui le concerne
lesAnneaux.add(anneau)
tout_lutins.add(anneau)

#La tête est t'elle entré en collision avec un poussin
collision_liste = pygame.sprite.spritecollide(tete,lesPoussins,False)
#Tester la liste des collisions
for poussin in collision_liste:
    #réinitialiser les positions des poussins touchés
    poussin.reset()
    #créer de nouveaux anneaux : ici 1 seul
    #queue[-1] donne la référence du dernier anneau de la liste
    creerAnneau(1,queue[-1].rect.x,queue[-1].rect.y)
```

```
    #Code traçant les objets du jeu
    #commencer par effacer l'écran
    ecran.fill(BLANC)
    #afficher tous les lutins
    tout_lutins.draw(ecran)
    #Mettre l'écran à jour
    pygame.display.flip()
    #Mise à jour de l'écran à raison de 10 images/seconde
    #60 images par seconde est beaucoup trop rapide.
    horloge.tick(10)

pygame.quit()
```

Tester le programme en ajoutant plus qu'un seul anneau au départ ou lorsque le serpent à manger un poussin, afin de vérifier le bon placement des anneaux.