

# Jeu du serpent : Scratch

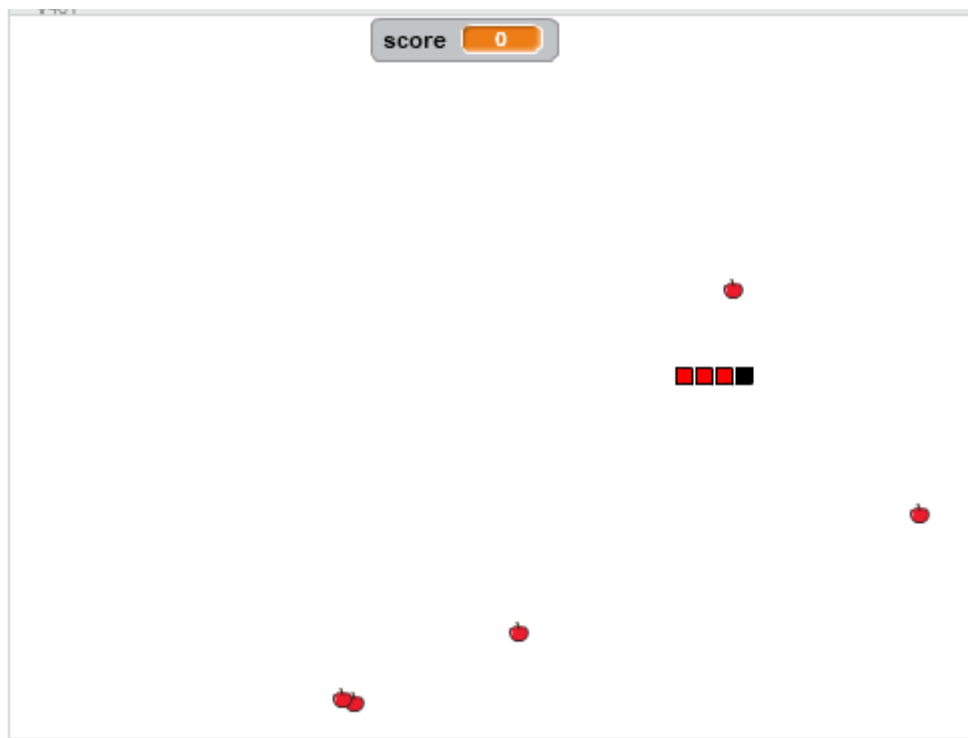
**Règles du jeu** : un serpent constitué d'une tête et d'anneaux se déplace sur la scène.

Lorsqu'il arrive sur un bord, il réapparaît de l'autre côté.

Les flèches haut, bas, à droite ou à gauche permettent de changer la direction du serpent.

Sur la scène sont placées des pommes. Lorsque le serpent mange une pomme, il marque un point et son corps s'allonge.

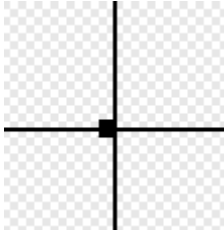
Le jeu s'arrête lorsque le serpent se mord la queue, c'est-à-dire lorsque la tête touche un des anneaux du corps.



Les objets :



La tête est un carré noir de 10 x 10 pixels, dont le point de référence est placé au milieu de son bord à droite.



L'anneau est identique à la tête, mais son centre est rouge.

La pomme est un des lutins de scratch, dont les dimensions ont été réduites à 10 x 10 pixels.

Les anneaux du serpent, seront des clones du lutin anneau.

De même les pommes placées sur la scène, seront des clones du lutin pomme.

## La mécanique du jeu

Deux listes coordX et coordY contiendront les coordonnées x et y de chaque élément du serpent : les coordonnées de la tête seront dans le premier élément des deux listes, les coordonnées du premier anneau seront dans le deuxième élément des deux listes etc.



Ici la tête du serpent a pour coordonnées  $x = -90$ ,  $y = 0$

Le premier anneau a pour coordonnées  $x = -100$ ,  $y = 0$

Le deuxième anneau a pour coordonnées  $x = -110$ ,  $y = 0$

Le troisième anneau a pour coordonnées  $x = -120$ ,  $y = 0$

A chaque top d'horloge, il faut donc calculer les nouvelles coordonnées de la tête du serpent et faire en sorte que les coordonnées de l'anneau numéro 1 soient les anciennes coordonnées de la tête, que les coordonnées de l'anneau numéro 2 soient les anciennes coordonnées de l'anneau numéro 1 etc.

Pour cela, il suffit de calculer les nouvelles coordonnées de la tête, puis de les insérer comme nouvel élément en tête de liste et de supprimer l'élément en queue de liste :

Ainsi dans la situation présentée ci-dessus, si on ne change pas de direction, les nouvelles coordonnées de la tête seront  $x = -80$  et  $y = 0$

Ces valeurs sont placées en tête des deux listes : les listes ont momentanément 5 éléments.

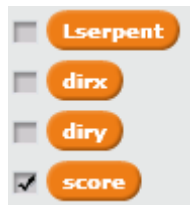
Mais on supprime le dernier élément des deux listes.

Cela donne alors :



Chaque élément de la liste fournit les nouvelles coordonnées de la tête ou de l'anneau qui lui correspond.

Nous devons également disposer de 4 variables :



Lserpent = longueur du serpent, dirx = direction du serpent en x, diry = direction du serpent en y et score = nombre de pommes mangées.

## La tête du serpent

Elle détecte l'appui sur une des flèches de direction, afin de positionner les variables dirx et diry aux valeurs permettant de faire avancer le serpent dans une direction ou une autre.

Si le serpent se déplace vers la droite dirx = 1 et diry = 0 (on avance en ligne droite vers la droite)

Si le serpent se déplace vers la gauche dirx = -1 et diry = 0 (on avance en ligne droite vers la gauche)

Si le serpent se déplace vers le haut dirx = 0 et diry = 1 (on avance en ligne droite vers le haut)

Si le serpent se déplace vers le bas dirx = 0 et diry = -1 (on avance en ligne droite vers le bas)



Le calcul des nouvelles coordonnées de la tête est alors :

Pour x :  $\text{abscisse } x + 10 * \text{dirx}$

La tête est un carré 10 x 10. On ajoute donc soit 0, soit +10, soit -10 suivant la direction mémorisée dans dirx.

Pour y :  $\text{ordonnée } y + 10 * \text{diry}$

## La boucle principale de la tête



Lorsque la tête reçoit le message « demarrer », elle regarde vers la droite et se place aux coordonnées indiquées dans le premier élément des listes coordX et coordY.

Elle entre ensuite dans une boucle infinie, dans laquelle elle :

- Elle attend x secondes : cette instruction sert uniquement à ralentir le serpent le temps des essais. On peut modifier la valeur de l'attente et on peut carrément supprimer cette instruction lorsque le jeu est au point.
- Vérifie si elle n'a pas touché un des bords de la scène.

- Calcule ses nouvelles coordonnées et les insère comme nouvel élément en tête des listes coordX et coordY
- Supprime le dernier élément des deux listes.
- Se déplace aux nouvelles coordonnées.

## Le test des bords



## La fin de la partie

Le jeu s'arrête lorsque la tête du serpent touche un des anneaux de la queue.

Nous aurions pu faire ce test ici, dans les scripts du serpent. Pour une raison expliquée plus loin (voir les anneaux), nous faisons ce test dans les scripts du lutin anneau.

La fin de la partie est signalée par la réception d'un message « fin ».




## Les pommes


Ce sont des clones du lutin pomme. Chaque clone hérite des scripts du lutin pomme.

Lorsque le serpent touche une pomme, celle-ci change ses coordonnées.

Comment savoir si le serpent touche une pomme ?

Dans les scripts du serpent, nous disposons du capteur  qui nous permet de détecter une collision entre le serpent et une pomme.

Cependant, après cette collision, il n'y a pas moyen de savoir de quelle pomme il s'agit, afin de changer ses coordonnées.

Nous détecterons donc une collision entre une pomme et le serpent dans les scripts du lutin pomme. 

Ainsi chaque clone de pomme sera capable de détecter une collision entre lui et le serpent et c'est ce clone qui changera ses propres coordonnées.

Lorsque le serpent touche une pomme, celle-ci ajoute 1 au score, change ses coordonnées et doit provoquer l'allongement du serpent, c'est-à-dire la création de 3 nouveaux anneaux.

Il envoie le message « ajouterAnneaux »

## Les scripts des pommes :



Lorsque les pommes reçoivent le message « fin », elles arrêtent tout.

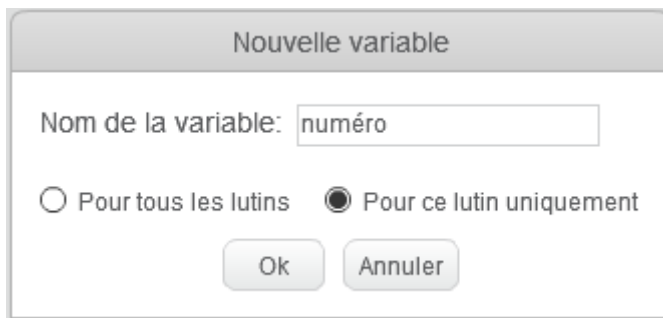
## Les anneaux

Les anneaux sont des clones du lutin anneau.

Pour pouvoir se déplacer, chaque anneau doit connaître sa position dans le serpent.

Cette position lui permet de trouver ses coordonnées dans les listes coordX et coordY.

Dans le lutin anneau, nous devons donc définir une variable numéro, **locale** à ce lutin.

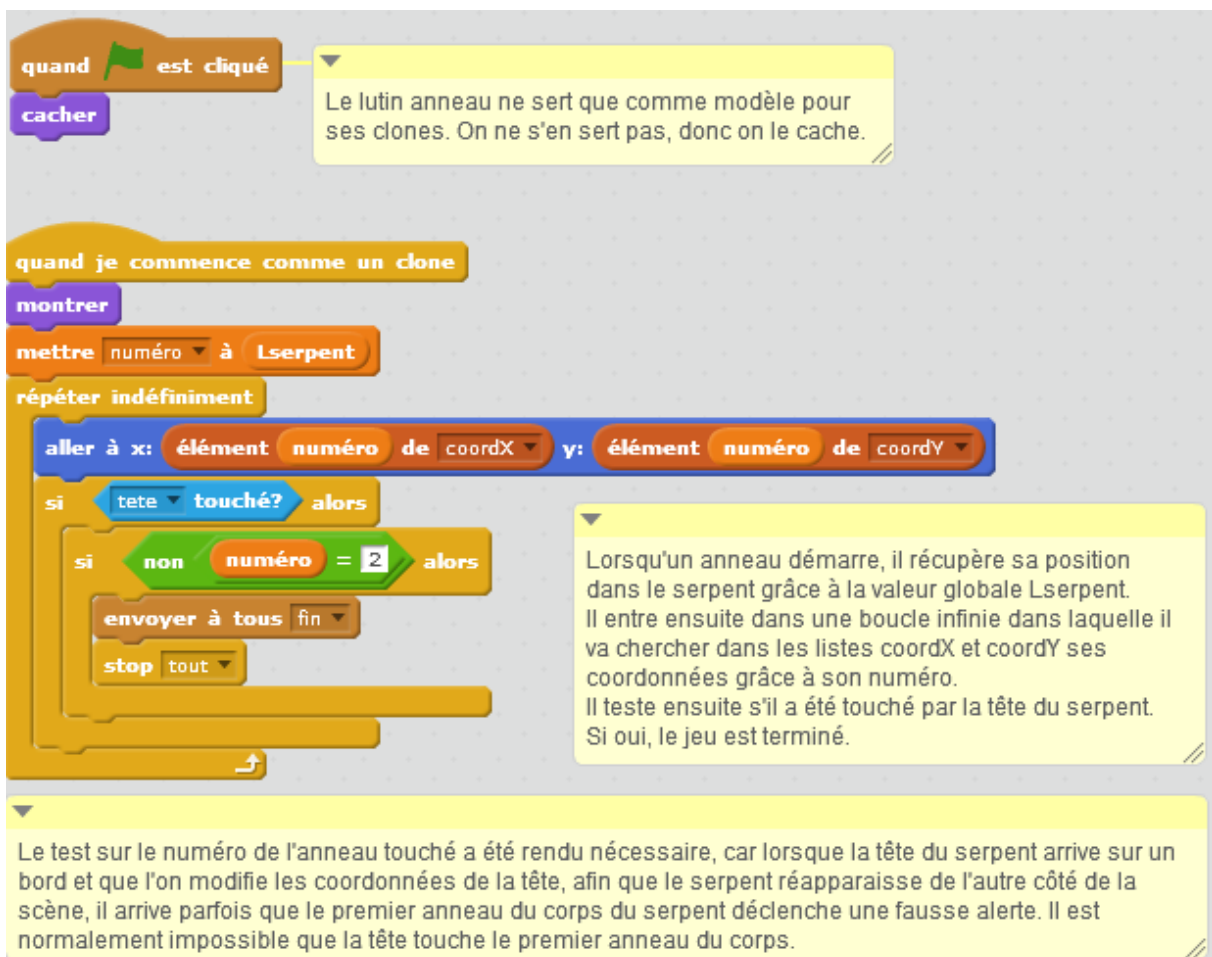


Chaque clone d'anneau, va hériter de cette variable et y mettre au moment de sa création, sa position dans le serpent.

Chaque clone d'anneau, hérite également des scripts de l'anneau.

Au démarrage du jeu, 3 anneaux sont créés. Par la suite, 3 nouveaux anneaux sont ajoutés au serpent, lorsque le serpent touche une pomme.

## Les scripts des anneaux



La détection de la collision entre la tête du serpent et d'un anneau de la queue, dans le script des anneaux, permet à chaque anneau de savoir s'il a été touché et permet ainsi de détecter

la fausse détection du premier anneau (position 2 dans le serpent), lors de la rencontre avec un bord de la scène.

Cette détection ne serait pas nécessaire, si une des règles du jeu était l'arrêt du jeu si la tête du serpent touche un bord.

## L'arrière plan

Nous regroupons dans les scripts de l'arrière plan, toute la partie initialisation du jeu, création des clones de pomme et création des clones d'anneaux.

Au démarrage, il faut nettoyer les listes coordX et coordY, initialiser toutes les variables, mettre dans les listes coordX et coordY les coordonnées de départ de la tête du serpent, créer les pommes, créer les trois premiers anneaux et lancer le message « démarrer », pour indiquer à la tête que tout est prêt.



La tête va se diriger au départ vers la droite ( $dirx=1$  et  $diry=0$ )

### La création des anneaux

Pour chacun des nouveaux anneaux, il faut calculer ses coordonnées et les placer en queue de listes.

Ce calcul s'effectue à partir des coordonnées indiquées dans les derniers éléments des deux listes coordX et coordY et de la direction dans laquelle la tête se déplace.

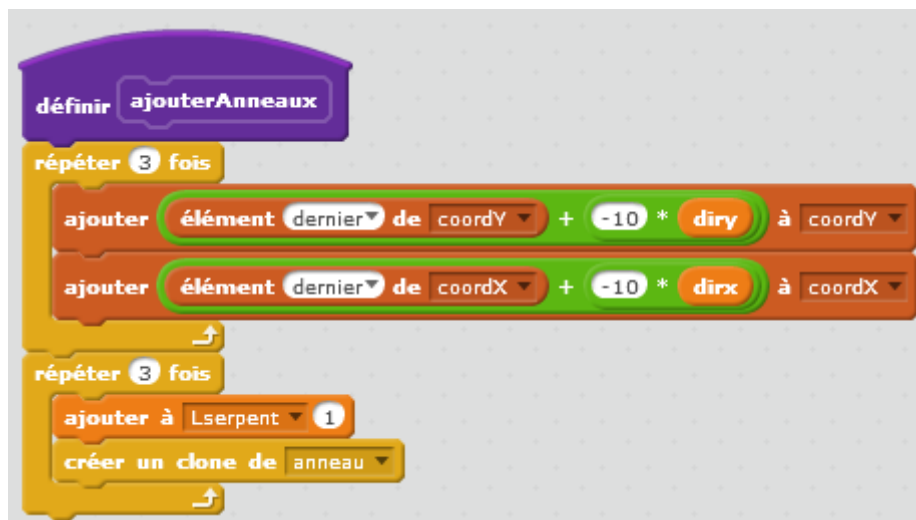
Les anneaux doivent se placer derrière la tête.



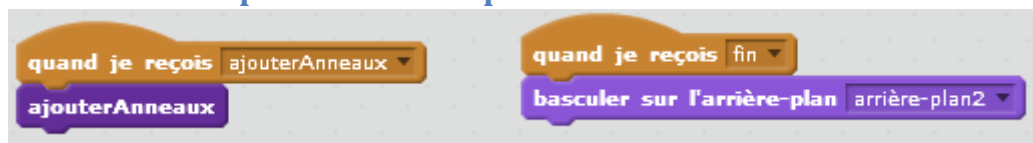
Pour x : 

Pour y : 

Ensuite, on clone le lutin anneau après avoir mis à jour la variable Lserpent, qui permet à chaque clone de connaître sa position dans le serpent.



## Les autres scripts de l'arrière plan



Le message « ajouterAnneaux » est envoyé par une pomme, lorsqu'elle est touchée par la tête du serpent.

Le message « fin » est envoyé par un anneau (autre que le premier), lorsqu'il est touché par la tête du serpent.

Dans ce cas, l'arrière plan bascule sur :

