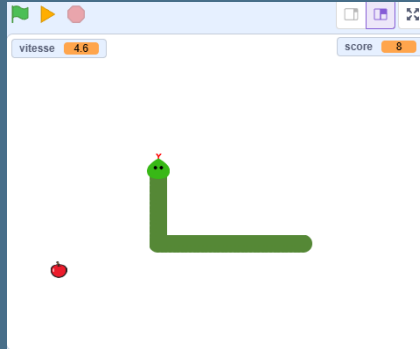




5 Snake / arcade



Un jeu de Snake à la mode Arcade, Consignes

- Snake se déplace vers l'avant, constamment,
- tourne à droite ou à gauche quand on appuie sur une touche de direction.
- Sa vitesse augmente et son corps s'allonge quand une nourriture est mangée et le score augmente de 1.
- Deux types de collisions fatales mettent fin au jeu : un contact avec le bord de la scène ou avec lui-même.
- Quand la nourriture est mangée, elle réapparaît aléatoirement sur la scène.



Video

<https://jeunesingenieurs.be/vid-s3-5>



Proposition de solution

<https://scratch.mit.edu/projects/1229228658>



▀ Tu te lances en mode Défi ?

Voici une proposition de méthode.

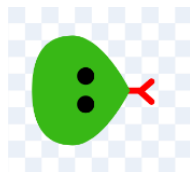
1. Prépare une tête de serpent (+/- 30x30pixels).
Code les déplacements en respectant les consignes.
2. Teste et optimise les mouvements.
3. Le corps de Snake doit donner l'impression de suivre la tête.
4. Veille à bien organiser ton code.
5. Gère les collisions fatales.
6. Ajoute de la nourriture : une pomme qui, chaque fois qu'elle est mangée par Snake, réapparaît ailleurs.
7. Snake grandit et accélère quand il mange.
Ajoute un score.
8. Game Over s'affiche quand une collision fatale a lieu.



5.1 DÉPLACEMENTS ET ROTATION

Objectifs

Créer un sprite **Snake** ; la tête uniquement. Il avance à vitesse constante. Le joueur le fait pivoter à droite ou à gauche à l'aide des touches de direction .



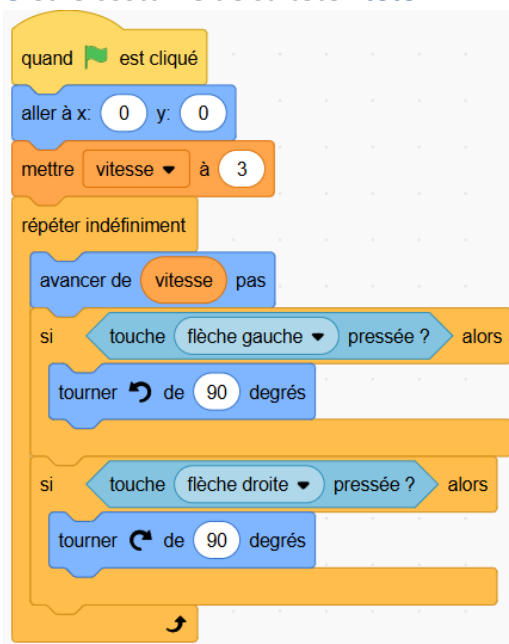
La tête de Snake

Renomme ton sprite **Snake** et le costume de sa tête : **tête**

Crée une variable vitesse et ajoute ce code

La variable vitesse n'est pas indispensable à ce stade mais nous savons que Snake va accélérer...elle sera donc utile plus loin.

Teste





5.2 OPTIMISATION, DÉBOGUAGE



Objectif : optimiser les mouvements
Il faut résoudre deux problèmes.

**Le code est à la
page suivante**

Suppression de l'effet de rebond

Tu as remarqué qu'un appui sur une touche provoque plusieurs rotations. Pourquoi ?

Parce que la boucle **répéter** tourne vite : 30 x par seconde. Si un appui sur la touche dure 0.1 seconde, la rotation s'effectue au moins trois fois...C'est l'effet de rebond.

Ajoute un bloc **attendre jusqu'à ce** ^❶ que la touche ne soit pas pressée.



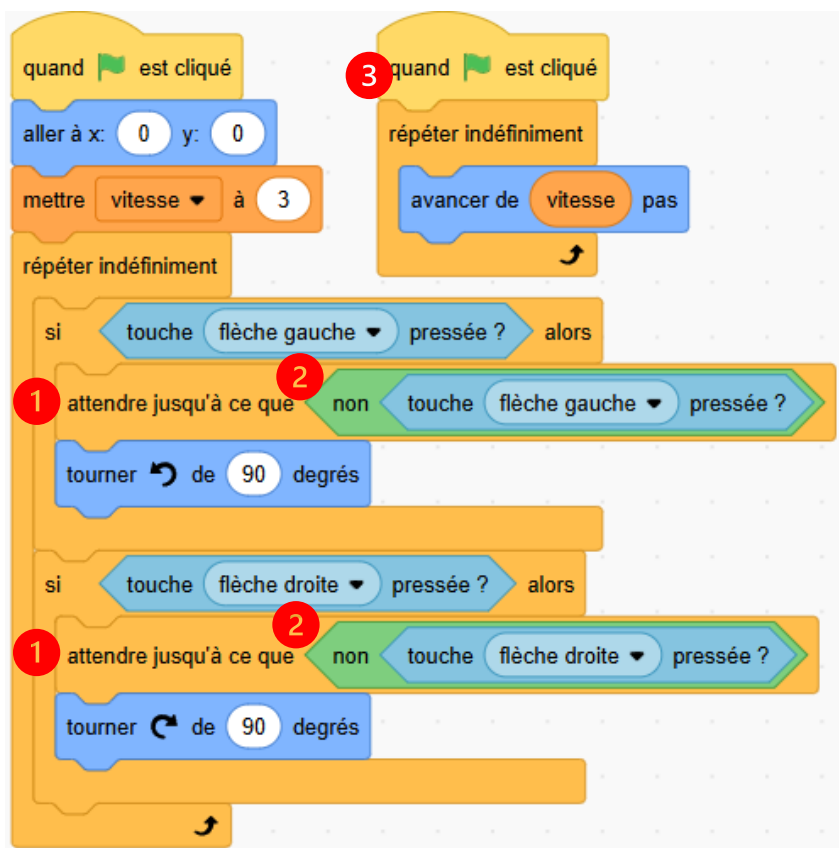
L'opérateur ^❷**non** permet d'introduire une négation.

Snake s'arrête quand on appuie sur une touche

Quand le joueur appuie quelques fractions de secondes sur une touche, le bloc **attendre jusqu'à ce que** provoque une pause. Il faudrait que Snake continue à avancer. Pour ce faire, on crée un ^❸deuxième script et une nouvelle boucle répéter indéfiniment pour le bloc **avancer de ... pas**. Ainsi, il avancera quelles que soient les actions produites par l'autre script.



▼ Modifie le code comme ceci :





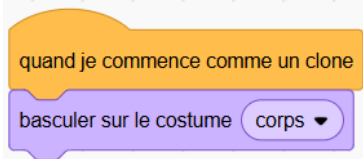
5.3 LE CORPS DE SNAKE

🎯 Objectif : créer le corps de Snake

Dans la version Arcade de Snake, le corps du serpent est composé d'éléments qui suivent la même trajectoire que la tête. On utilise le clonage pour obtenir cet effet.

📌 Crée un nouveau costume pour *Snake*

Le deuxième costume sera nommé *corps* et se compose d'un simple disque plus petit que la tête.



📌 Modifie ton programme

Les clones à créer vont revêtir le costume *corps* et ils resteront à leur place sans se déplacer alors que la *tête* elle, continuera à avancer.

Modifie le script qui contient le bloc *avancer de ... pas* et ajoute un nouveau script qui commence par *quand je commence comme un clone un clone*.

Teste et observe ce qu'il se passe.



5.3 LE CORPS DE SNAKE

🎯 Objectif : limiter la longueur du corps

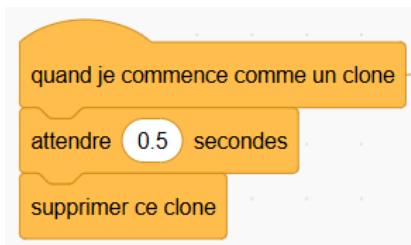
Le corps s'allonge, s'allonge puis la tête se détache.

Scratch limite à 300 le nombre de clones par sprite. Le corps de Snake va donc s'allonger pendant 10 secondes pour produire les 300 clones et puis, la tête continuera son chemin seule.



▼ Ajoute ce script

La solution : supprimer chaque clone après un certain temps, c'est-à-dire limiter sa durée de vie.

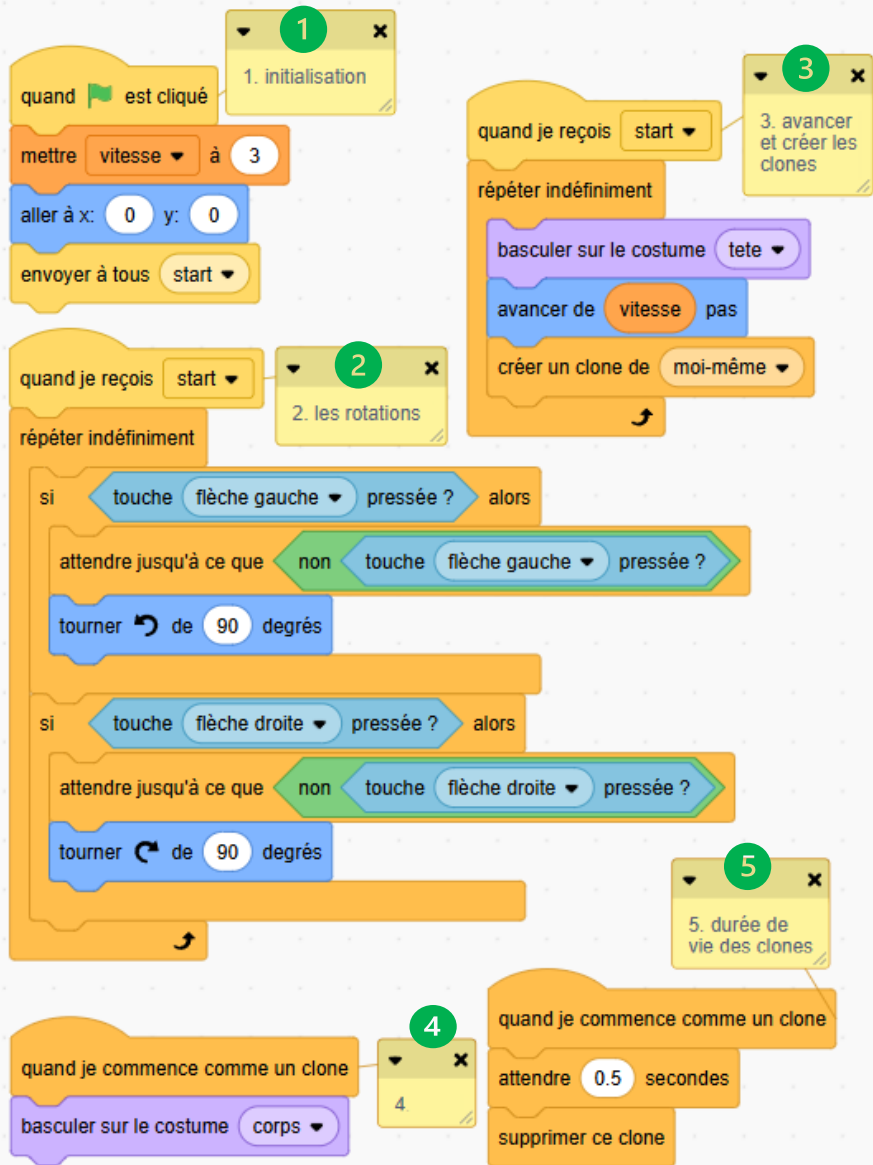


▼ Organiser le code

Quand on programme, il est important de bien séparer les tâches et organiser son code. Il faut pouvoir identifier le rôle de chaque script. Tu peux utiliser les commentaires si nécessaire.

Modifie le programme comme proposé à la page suivante et note bien les numéros attribués à chaque script car nous les utiliserons plus loin.

5.4 ORGANISER LE CODE





5.5 GÉRER LES COLLISIONS FATALES



Objectifs : **Game over** si **Snake** touche un bord ou se touche lui-même

Modifie le script 4

Dans une boucle **répéter indéfiniment**, on teste :

1 si un des clones touche la langue rouge de Snake

2 si un des clones touche le bord de la scène

Créer un nouveau message **game over**

Ajoute un nouveau script 6

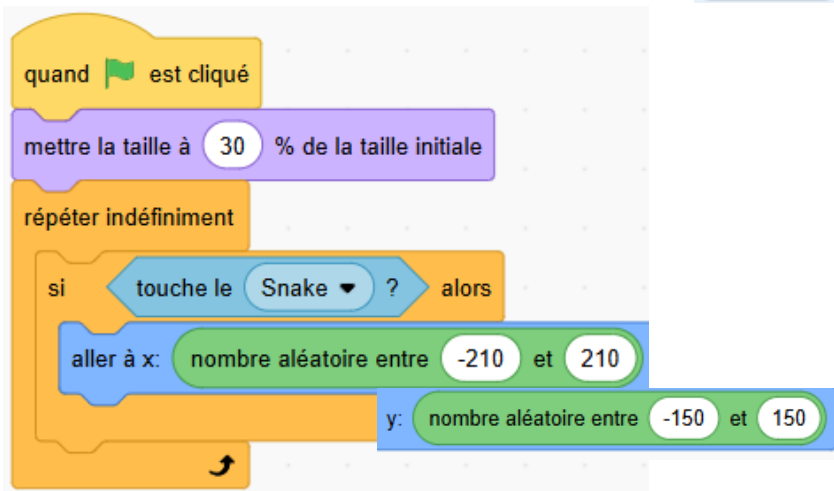




5.6 NOURRIR SNAKE

🎯 Objectif : faire apparaître une pomme et la faire réapparaître ailleurs quand elle est mangée

📌 Ajoute un nouveau sprite Apple et son code



Le positionnement aléatoire de la pomme est géré de façon à éviter qu'elle aille se coller au bord.



5.7 GRANDIR, ACCÉLÉRER ET SCORER

Objectifs

Chaque fois qu'une pomme est mangée, Snake s'allonge, accélère et un score s'incrémente.



Crée deux nouvelles variables

Crée : **score** et **durée de vie** (nous avons déjà créé **vitesse**)

Script 1 de Snake : initialisation

Script de Apple



5.8 GAME OVER

Objectif

Afficher Game Over quand le jeu se termine.

Crée un arrière-plan « Game Over »

Modifie le code des scripts 1 et 6

Ajoute les blocs *basculer sur l'arrière-plan...*

