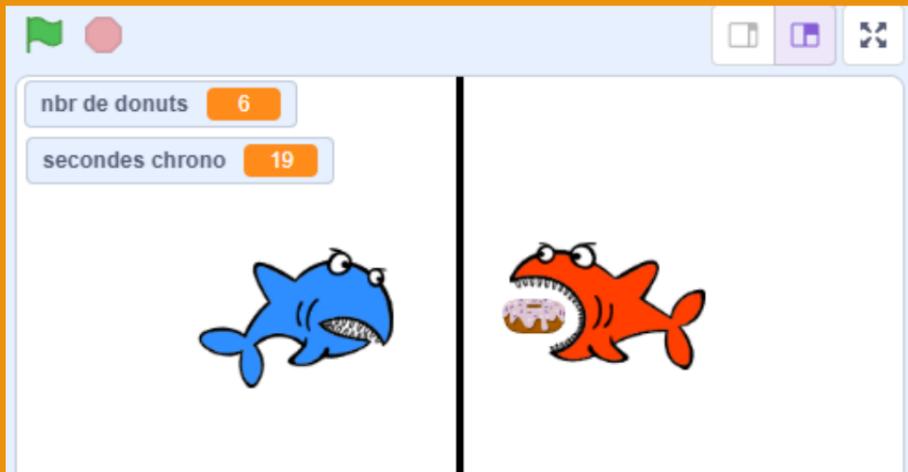


8 Joue avec ton reflet



Le requin de gauche est dirigé avec la souris. En face, son reflet suit ses déplacements... Ton job est de l'aider à dévorer 10 donuts qui apparaissent de son côté et cela, le plus vite possible.



▼ Vidéo

<https://jeunesingenieurs.be/vid-s2-8>



▼ Proposition de solution

<https://scratch.mit.edu/projects/1116627008>



Fortement inspiré par : <https://www.ac-polynesie.pf/article/defis-scratch-defi-codage-et-robotique-2023-122066>

Tu te lances en mode Défi ?

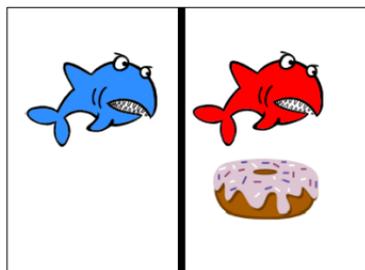
Visionne la vidéo et essaie d'atteindre le résultat proposé. Travaille étape par étape. Le requin dirigé par le joueur est le bleu, le reflet est le rouge

1. Prépare tes sprites et programme le déplacement du requin bleu de façon à ce qu'il suive le curseur de la souris.
2. Programme le déplacement « en miroir » du reflet.
3. Place le donut du côté reflet et programme cette action : quand le requin reflet touche le donut, le nombre de donuts est réduit d'une unité.
4. Language des donuts. Quand il est dévoré, le donut disparaît puis réapparaît aléatoirement dans la zone reflet jusqu'à épuisement du stock.
5. Prépare et programme une hitbox pour augmenter le réalisme lors du contact entre un donut et le requin rouge.
6. Améliore ce programme de façon à ce que le requin rouge ne puisse pas franchir le miroir et ajoute un chronomètre qui s'arrête quand tous les donuts ont été dévorés.

8.1 LES SPRITES, LE DÉPLACEMENT DU REQUIN

🎯 Objectif : prépare les sprites et l'arrière-plan

Charge le requin *Shark* et duplique-le. Change la couleur du deuxième requin. Charge le donut et modifie l'arrière-plan en ajoutant une ligne noire en plein centre de la scène. Réduis la taille des requins à 75% et le donut à 30%. Change le nom des deux requins en *requin* et *reflet*.



🎯 Piloter le *Requin* à l'aide de la souris et fixer la condition de fin de jeu

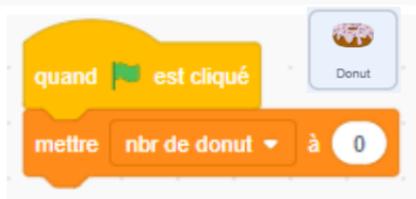
Crée une variable *nbre de donuts* et ajoute ce code à *requin*.

Le jeu se terminera quand tous les donuts seront dévorés (nbr de donuts = 0).



▾ Ajoute ce code à Donut

Initialise la variable *nbr de Donut*



8.2 LE REFLET DANS LE MIROIR

🎯 Objectif

Positionner le **requin** et son **reflet** face à face et faire en sorte que le **reflet** suive les mouvements du **requin**.

▾ Ajoute ce code au reflet

```
quand le drapeau est cliqué
fixer le sens de rotation gauche-droite
s'orienter à -90
répéter jusqu'à ce que nbr de donuts = 0
aller à x: 0 - abscisse x de Requin y: ordonnée y de Requin
```

1 Essaie les **options gauche-droite** et **à 360°** pour comprendre l'effet de ce bloc en faisant tourner ton sprite avec le cadran de la propriété **Direction**.

```
fixer le sens de rotation gauche-droite
```

- ✓ gauche-droite
- ne tourne pas
- à 360°

Le bloc **2 aller à ...** et les références à abscisse x de ... - **ordonnée y de...** sont au cœur du mouvement en miroir des deux requins. Prends le temps de comprendre son fonctionnement. Remarque que les mouvements en X (horizontaux) sont opposés alors que le mouvement en Y est identique pour les deux requins.

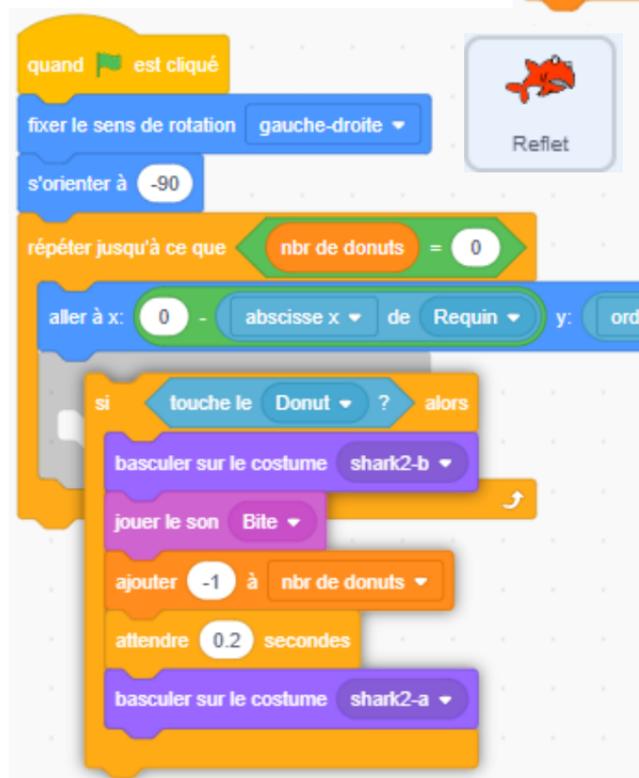
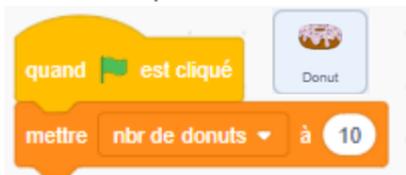
8.3 MANGER UN DONUT

🎯 Objectif : manger les donuts

Initialisation : prévoir un stock de donuts.

Le **requin rouge** (reflet) doit manger le donut quand il le touche.

▾ Ajoute ce code au **donut**



▾ Modifie le code du requin **reflet**

Place le **donut** manuellement du côté du **reflet**.

Prends le temps d'identifier et comprendre chaque action de ce script. Elles sont simples.

Teste ton code !

8.4 LARGUEZ DES DONUTS

🎯 Objectif : larguer les donuts

Faire apparaître les donuts en les positionnant de façon aléatoire dans la zone du requin *reflet*.

▾ Ajoute ce code au *donut*

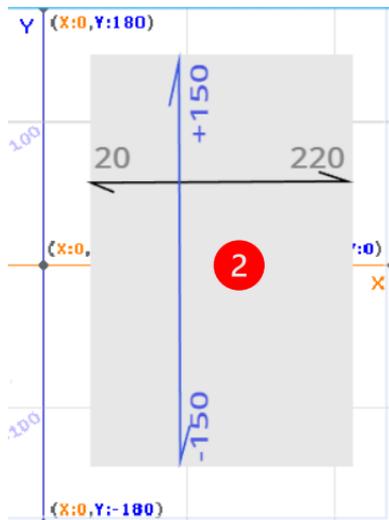


Créer un nouveau message *quand je reçois...larguer donut*

1 Pour positionner le nouveau donut à l'intérieur d'une zone, on définit aléatoirement une position comprise à l'intérieur d'une 2 zone rectangulaire.

▾ Teste ce code

... 3 en cliquant sur ce nouveau script .



8.5 HITBOX

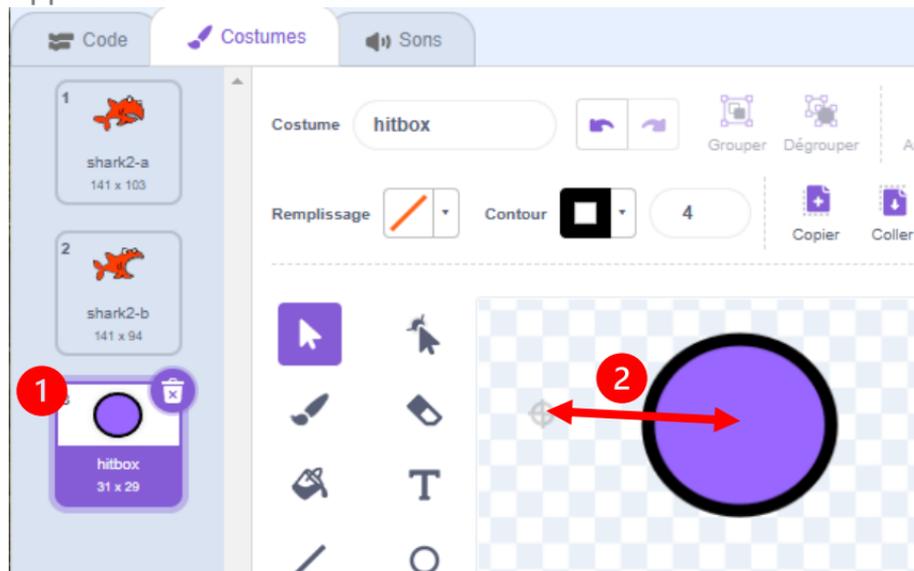
🎯 Objectif : améliorer le rendu à l'aide d'une **hitbox**

Tu constates que le donut est dévoré dès qu'il touche une extrémité du **reflet**. Cela n'est pas très réaliste. Tu peux améliorer cela grâce à la technique de la **hitbox**.

Le principe est de créer un costume qui sera invisible et qui sera la zone de contact entre le requin **reflet** et le **donut**.

📌 Crée un nouveau costume pour le requin **reflet**

❶ Dessine un simple cercle d'une couleur quelconque de +/- 30 pixels de diamètre. ❷ Décentre-le légèrement vers la droite. Appelle ce costume **hitbox**.



8.5 HITBOX

▶ Ajoute deux blocs au code du *reflet* ① et ②

The code block is a loop that repeats until the 'nbr de donuts' variable equals 0. The blocks inside the loop are:

- aller à x: 0 - abscisse x de Requin
- basculer sur le costume hitbox ①
- si touche le Donut ? alors
 - basculer sur le costume shark2-b
 - jouer le son Bite
 - ajouter -1 à nbr de donuts
 - attendre 0.2 secondes
 - basculer sur le costume shark2-a
 - envoyer à tous donut et attendre
- basculer sur le costume shark2-a ②

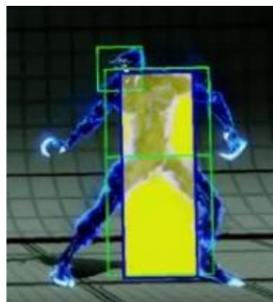


▶ Teste ce code.

Tu constateras que le passage par le costume *hitbox* n'est pas perceptible. En effet, dans ce script les costume est remplacé immédiatement par un autre costume. Le costume *hitbox* n'existe que le temps de faire le test *si touche le donut*.

▶ La *hitbox* est souvent utilisée dans les jeux de combat.

C'est cette zone que l'ennemi devra atteindre pour toucher son adversaire.

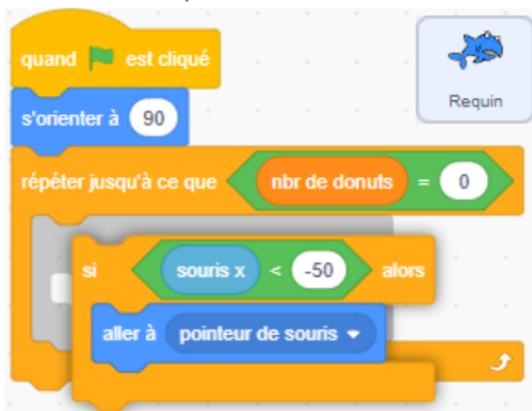


8.6 UN MIROIR INFRANCHISSABLE

🎯 Objectif : éviter que **requin** franchisse le miroir

Tu as certainement remarqué que le requin bleu peut traverser le miroir ... ce n'est pas très réaliste. Tu peux éviter cela à l'aide d'un simple test.

▾ Ajoute ce code au **requin** bleu



8.7 AFFICHER UN CHRONO

▾ Ajoute ce code au **donut**

Crée une nouvelle variable **secondes chrono**.

Ajoute ce code au **donut** :

