



8 Un décor qui défile



Dans un film, on appelle cela du *travelling*.

Faire défiler un décor pendant que ton personnage se déplace permet d'agrandir le monde dans lequel tu évolues et donne un caractère plus dynamique à ton jeu ou ton récit. Avec ce jeu de cartes, apprend à gérer un décor défilant !



▼ Vidéo

<https://jeunesingenieux.be/vid-s3-8>



▼ Proposition de programme

<https://scratch.mit.edu/projects/1242231407>



Un décor qui défile

► Tu te lances en mode Défi ?

1. Prépare deux sprites : un petit personnage et un décor tout simple. Code le déplacement de Chick de droite à gauche avec les touches ad-hoc.
2. Le décor doit défiler vers la droite quand le poussin se déplace à gauche et vers la gauche quand....
3. Ajoute de nouveaux éléments de décor et veille à ce que les raccords soient invisibles.
4. Veille à ce que le poussin se retourne quand il change de direction.
5. Pour pouvoir multiplier aisément les décors, passe à une gestion par clones...
6. Veille à ce que le poussin puisse grimper sur les obstacles...

Et plus encore ?

RDV sur la dernière carte pour des idées de défis supplémentaires.



8.1 UN PERSONNAGE ET UN DÉCOR

Objectifs

Prépare le poussin **Chick**.

Code un déplacement simple pour **Chick** :

- Vers la droite avec la touche flèche-droite
- Vers la gauche avec la touche flèche-gauche
- Un petit saut avec la touche flèche-haut

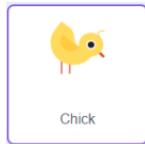
Ajoute un arrière-plan pour le ciel avec un soleil et deux nuages.

Les sprites



Charge le sprite **Chick**,
et dessine le sprite
Décor.

Le décor est un sprite
et non un arrière-plan car nous
allons le faire bouger...
Renomme ce sprite en **décor-1**.



L'arrière-plan

L'arrière-plan est composé par
tes soins et restera fixe.

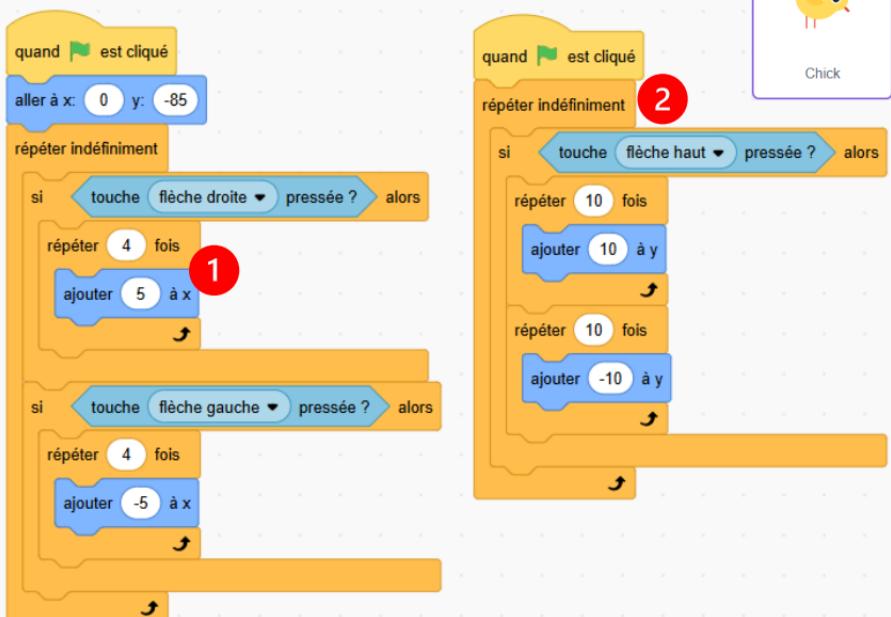




UN PERSONNAGE ET UN DÉCOR

► Anime le *Chick*

Pour un déplacement fluide, on préfère **1 répéter 4 X avancer de 5** plutôt que **avancer de 20**. La saut est codé à l'aide d'une deuxième boucle **2 répéter indéfiniment** afin de pouvoir combiner les mouvements horizontaux et verticaux. N'hésite pas à améliorer ce code.



The image shows two Scratch scripts side-by-side. The left script, labeled '1', is triggered by a green flag click. It moves the chick to (-85, 0) and then enters a loop that checks for right arrow key presses. If pressed, it adds 5 to the x position. It also checks for left arrow key presses and subtracts 5 from the x position. The right script, labeled '2', is also triggered by a green flag click. It enters an infinite loop that adds 10 to the y position 10 times, then subtracts 10 from the y position 10 times. This creates a jumping effect. Both scripts have a 'Chick' sprite attached.

► Positionne le sprite *decor-1*

Au centre de la scène...

J.P. Bihin – déc. 2025.

Série 3





8.2 LE DÉCOR DÉFILE

Objectif : faire défiler le décor (et non le **Chick**)

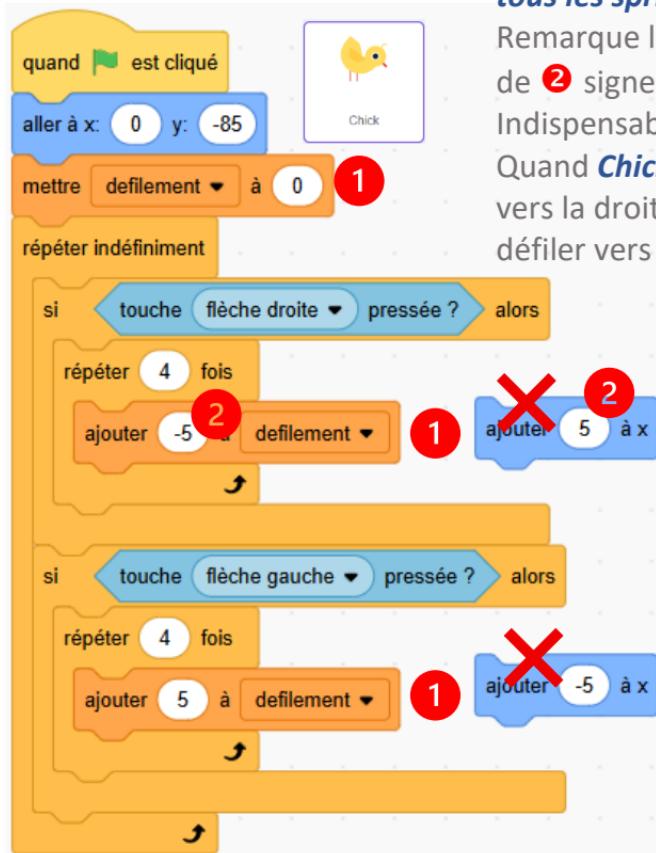
Crée une variable **défilement** et ① modifie le code de **Chick**

La variable servira **pour tous les sprites**...

Remarque le changement de ② signe !

Indispensable...

Quand **Chick** se déplace vers la droite, le décor doit défiler vers la gauche...





LE DÉCOR DÉFILE

▼ Modifie le code de **decor-1**

Pour qu'il défile....

Avec ce code et grâce à la variable **défilement**, le mouvement est appliqué au décor et non au poussin, ce qui donne cet effet de « travelling. »

Teste....

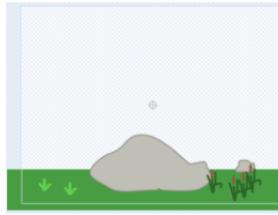
▼ Crée deux décors supplémentaires

Duplique le sprite **decor-1** à deux reprises pour obtenir trois sprites, renomme-les et modifie-les.

decor-1



decor-2



decor-3



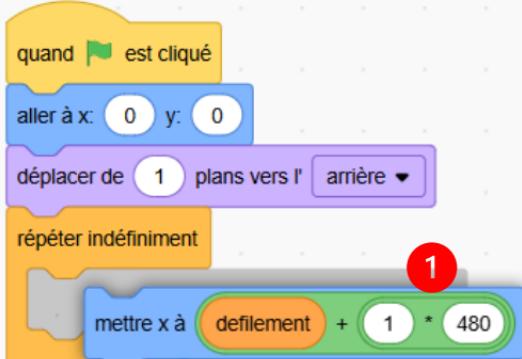
Prévois une petite zone de recouvrement à gauche ①. Il faudra aussi veiller à ajuster les raccords entre les différents décors.





8.3 LE DÉCOR S'ALLONGE

▼ Modifie le code de *decor-2*

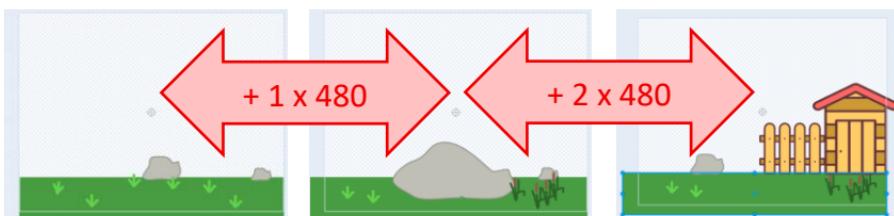


A Scratch script consisting of the following blocks:

- when green flag clicked
- go to x: 0 y: 0
- move 1 steps towards back
- repeat indefinitely [set x to (defilement) + (1 * 480)]

Ajoute ① 1×480 (= 1×480) en abscisse (X) pour que ce décor suive le premier avec 480 pas de retard.

▼ Adapte le code de chaque décor



mettre x à defilement Pour *decor-1*

mettre x à defilement + 1 * 480

On ajoute 1×480 pour *decor-2*,

mettre x à defilement + 2 * 480

2×480 pour *decor-3*, etc...

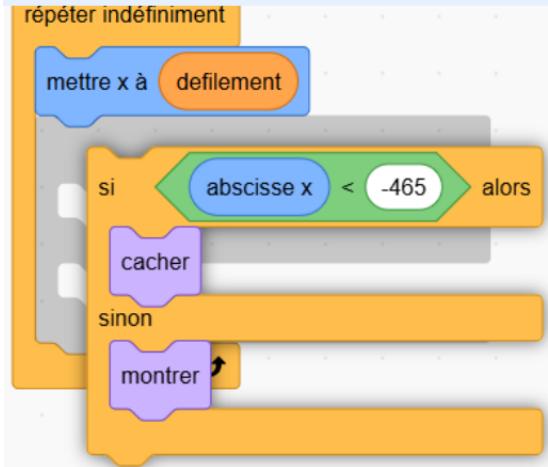
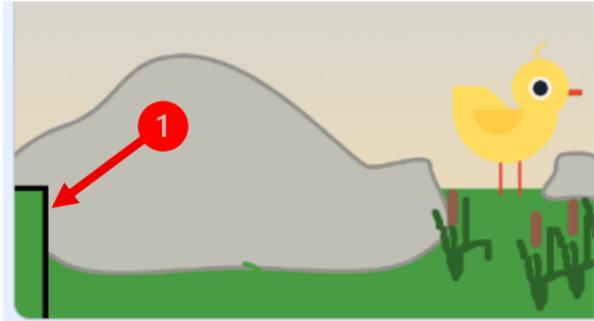


LE DÉCOR S'ALLONGE

Cache les décors inutiles

Les sprites qui sont en dehors de la scène sont bloqués. Tout se passe comme s'ils attendaient leur tour dans les coulisses.
C'est un fonctionnement propre à Scratch.

➊ Pour éviter que le décor qui doit disparaître reste apparent et bloqué, on ajoute un petit script...



À l'aide de quelques tests, on a pu évaluer qu'il fallait faire disparaître le décor quand son abscisse était inférieure à 465.

N'hésite pas à faire quelques tests en changeant la couleur de l'herbe du deuxième décor pour observer les raccords.



8.4 LE POUSSIN SE RETOURNE

Objectif : améliorer le comportement du poussin
Il doit se retourner quand il change de direction.

Modifie le ① code de *Chick*

when green flag clicked

- aller à l' avant plan
- aller à x: 0 y: -85
- fixer le sens de rotation gauche-droite
- répéter indéfiniment
 - si touche flèche haut pressée ? alors
 - répéter 10 fois
 - ajouter 10 à y
 - alors
 - répéter 10 fois
 - ajouter -10 à y
 - si touche flèche droite pressée ? alors
 - s'orienter à 90
 - répéter 4 fois
 - ajouter -5 à defilement
 - si touche flèche gauche pressée ? alors
 - s'orienter à -90
 - répéter 4 fois
 - ajouter 5 à defilement

when green flag clicked

- répéter indéfiniment
 - si touche flèche haut pressée ? alors
 - répéter 10 fois
 - ajouter 10 à y
 - alors
 - répéter 10 fois
 - ajouter -10 à y



8.5 GÉRER LE DÉCOR PAR CLONAGE

Objectif : un programme performant

À ce stade, il y a trois éléments de décor et un code simple. Pour pouvoir anticiper un programme qui va grossir et des décors qui vont se multiplier, il faut passer à une gestion par clonage.



1 : un sprite **decor** et trois costumes

Garde le sprite **decor-1** et renomme-le **decor**. À l'aide du sac à dos, récupère les deux costumes de **decor-2** et **decor-3** et ajoute les au sprite **decor** qui a désormais 3 costumes.

Renomme tes trois costumes **decor-1**, **decor-2** et **decor-3**.

2 : modifie le code de **Chick**

Ajoute le bloc **envoyer à tous... et attendre** juste avant la boucle répéter indéfiniment.

```
fixer le sens de rotation [gauche-droite v-]
mettre [défilement v-] à [0]
répéter [touche [flèche droite v-] pressée? alors
  envoyer à tous [préparer décor v-] et attendre
  si [touche [flèche droite v-] pressée? alors
    ...
  end
end]
```



GÉRER LE DÉCOR PAR CLONAGE

▼ 3°: Supprime les sprites **decor-2** et **decor-3**

▼ 4° : code un script qui génère 3 clones pour **decor**.

Chaque clone sera identifié par son costume.

Tous les clones exécuteront le même programme.

Rappel : un costume peut être identifié
par son ① nom ou par son ② numéro
d'ordre (index)

Avec le script ci-dessous, trois clones sont
générés.





GÉRER LE DÉCOR PAR CLONAGE

▼ 5 : Crée un deuxième script pour **decor**

The Scratch script starts with a "when I start as a clone" hat. It moves to position (0, 0) and then enters a "repeat forever" loop. Inside the loop, it changes the x coordinate by 480 units (defilement + (numéro du costume - 1) * 480). It then checks if the x coordinate is less than -465. If yes, it hides ("cacher"), otherwise, it shows ("montrer").

```
when I start as a clone
  go to [0 v] [0 v]
  repeat (forever)
    move (480 v) [defilement]
    change costume by (-1) [du costume]
    if < (x) < (-465) then
      hide
    else
      show
    end
  end
end
```

Chaque clone est positionné en fonction du numéro d'ordre de son costume.
Clone 1 , **costume 1** : $(1-1) \times 480 = 0$

Clone 2 , **costume 2** : $(2-1) \times 480 = 480$

Clone 3 , **costume 3** : $(3-1) \times 480 = 960$

Etc...

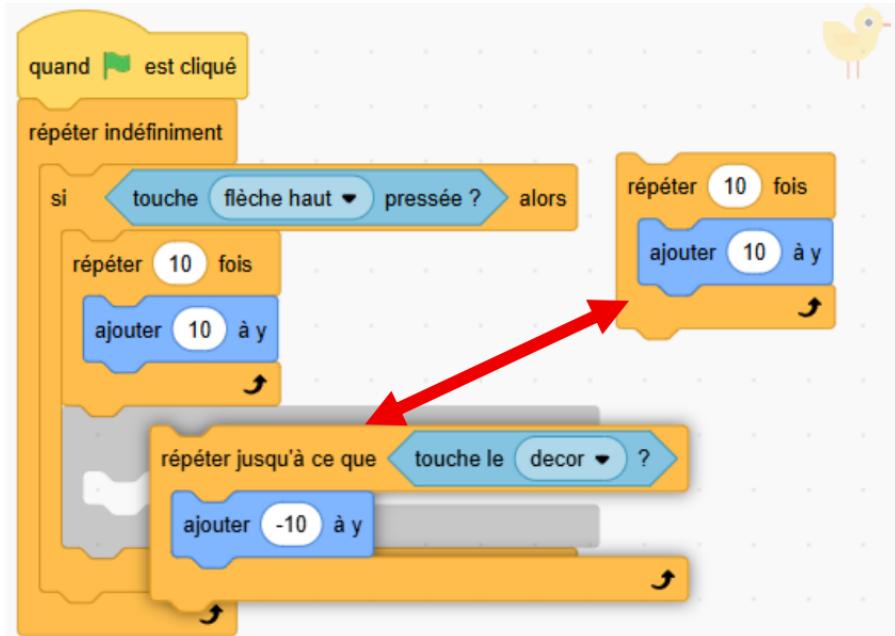
Ces deux scripts sont applicables quel que soit le nombre de clones et on peut donc agrandir le décor autant que nécessaire



8.6 LE POUSSIN GRIMPE

Objectif : faire en sorte que le poussin puisse grimper sur les objets qu'il rencontre.

Modifie le deuxième script de **Chick**

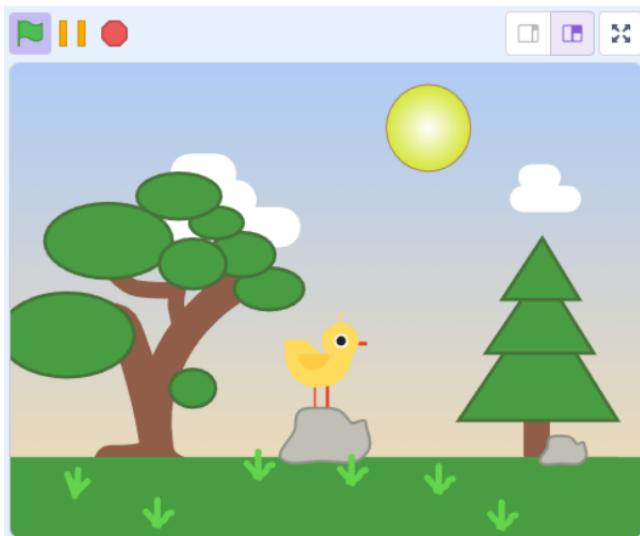


Grâce à la gestion par clonage, ce genre d'amélioration est facilitée. La condition **touche le sprite decor** va s'appliquer à tous les clones... Ceci aurait été plus compliqué dans la configuration précédente, sans les clones.



8.7 DÉFIS

L'objectif principal est atteint : gérer efficacement un décor défilant. À toi de jouer maintenant.



En t'inspirant de ce programme et d'autres, tu peux

- Créer un jeu original avec des objets à récolter, des ennemis qui tentent de te faire du mal, des énigmes à résoudre.
- Améliorer l'animation du poussin et la gestion des collisions.
- Ajouter un décor plus éloigné avec un effet de parallaxe
<https://scratch.mit.edu/projects/1254757278>

Have fun !

J.P. Bihin – déc. 2025.

Série 3