

Jeux de plateforme

V2 : suivi de terrain

Pierre Huguet

mail : Pierre@onvaessayer.org

site : <https://onvaessayer.org/scratch>

JEU DE PLATEFORME

Réalisation d'un algorithme de suivi de terrain :

<https://onvaessayer.org/scratch?app=platform>

V1 : [initialisation et déplacement horizontal](https://onvaessayer.org/scratch?video=platformV1)

<https://onvaessayer.org/scratch?video=platformV1>

V2 : [**déplacement avec suivi de terrain**](https://onvaessayer.org/scratch?video=platformV2)

<https://onvaessayer.org/scratch?video=platformV2>

V3 : [déplacement avec saut](https://onvaessayer.org/scratch?video=platformV3)

<https://onvaessayer.org/scratch?video=platformV3>

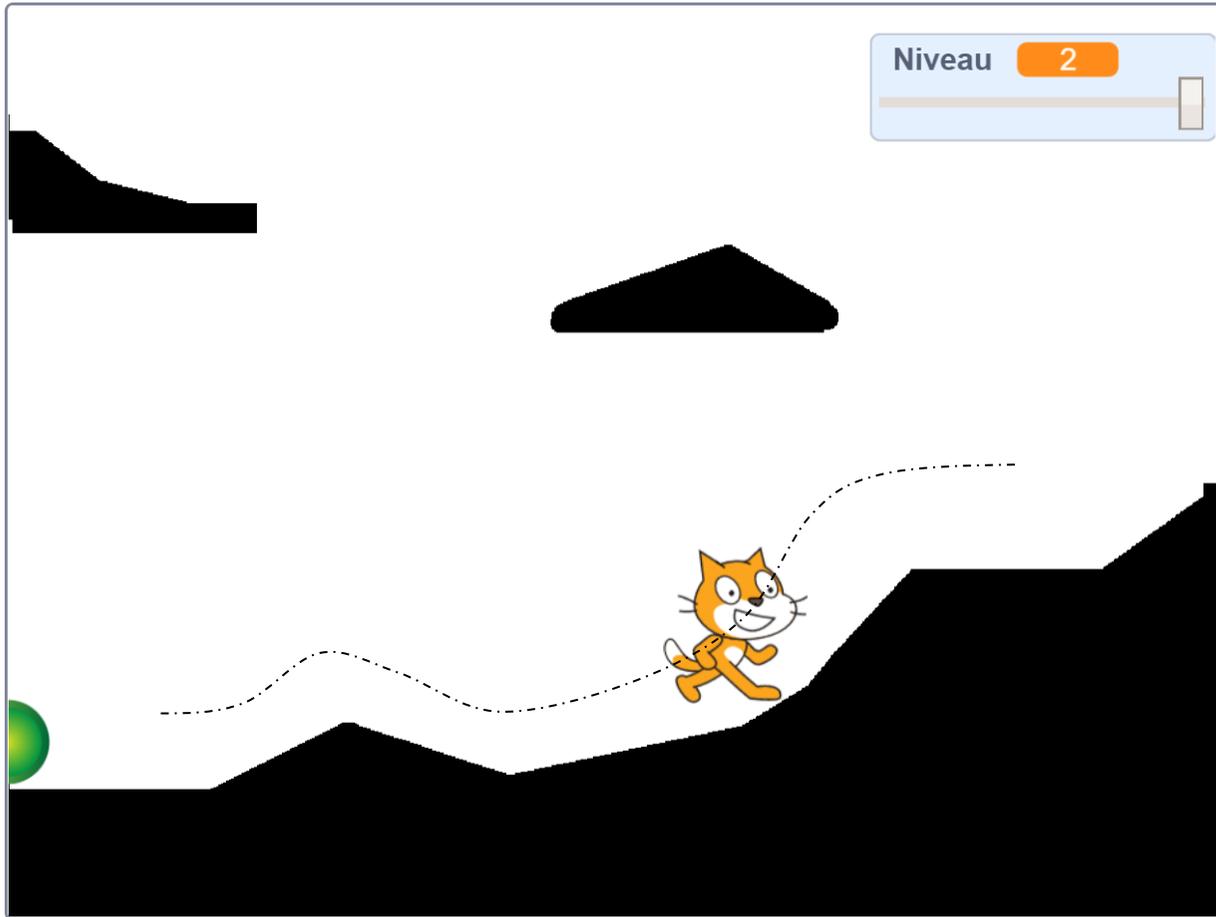
V4 : [bonhomme animé et chute ralentie](https://onvaessayer.org/scratch?video=platformV4)

<https://onvaessayer.org/scratch?video=platformV4>

V5 : [panoramique terrain de jeu illimité](https://onvaessayer.org/scratch?video=platformV5)

<https://onvaessayer.org/scratch?video=platformV5>

V2 : SUIVI DU TERRAIN AU SOL



Le sprite doit :

- monter quand il touche (sauf si la pente est trop forte),
- rester juste au dessus du terrain et descendre si nécessaire.

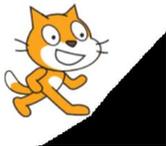
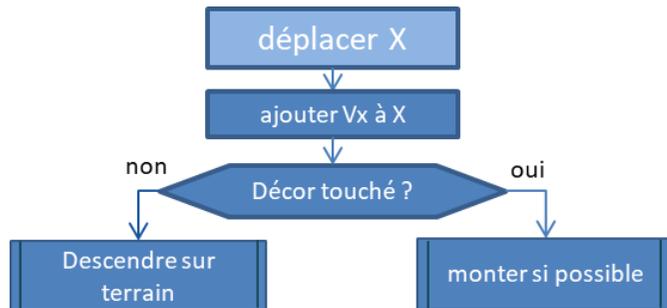
POINT DE DÉPART POUR LA V2 : DÉPLACER X

<https://onvaessayer.org/scratch?sb3=platformV1>

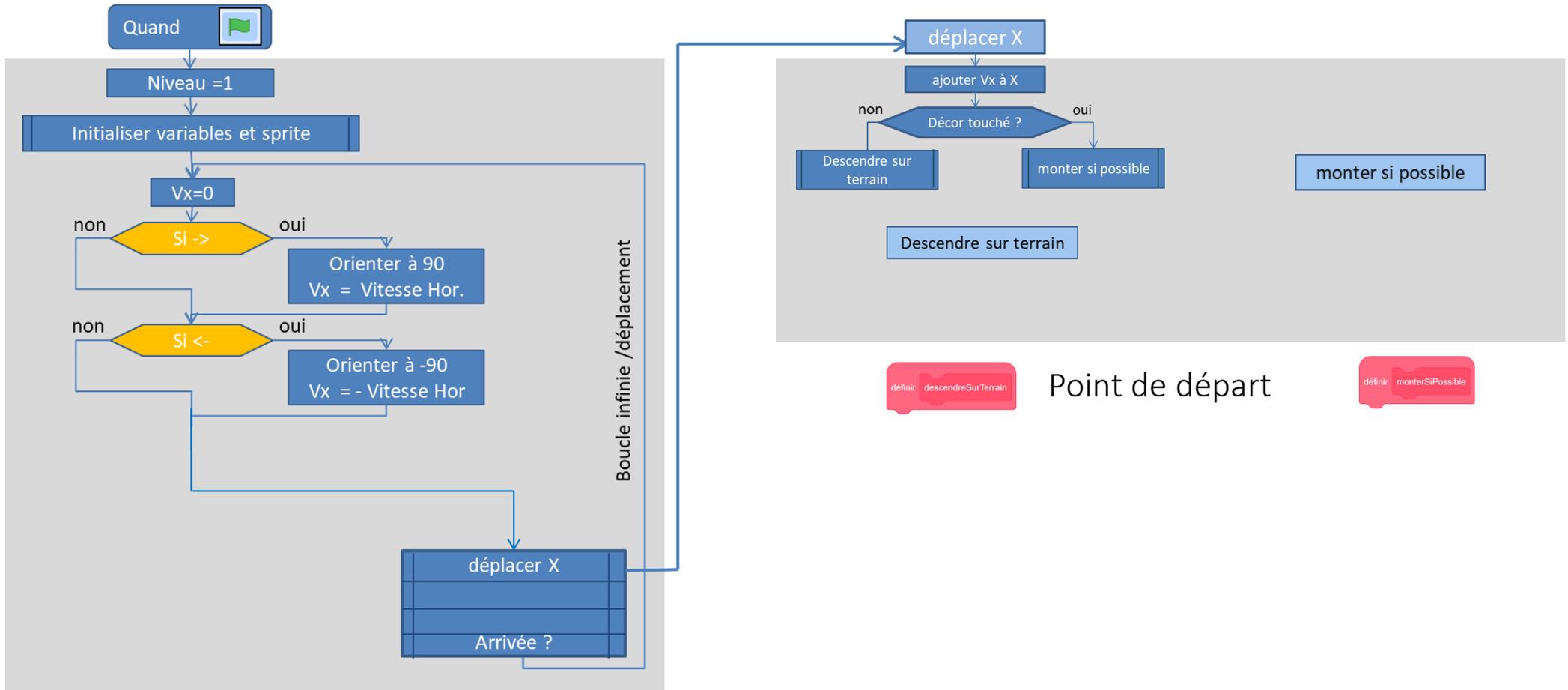
- dans la version 2, nous allons écrire le pseudo code, les diagrammes de flux, puis coder 2 procédures : **monter** (si possible) et **descendre** (jusqu'au terrain)
- le sprite doit rester au-dessus du sol, sans le toucher

définir monterSiPossible

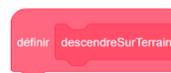
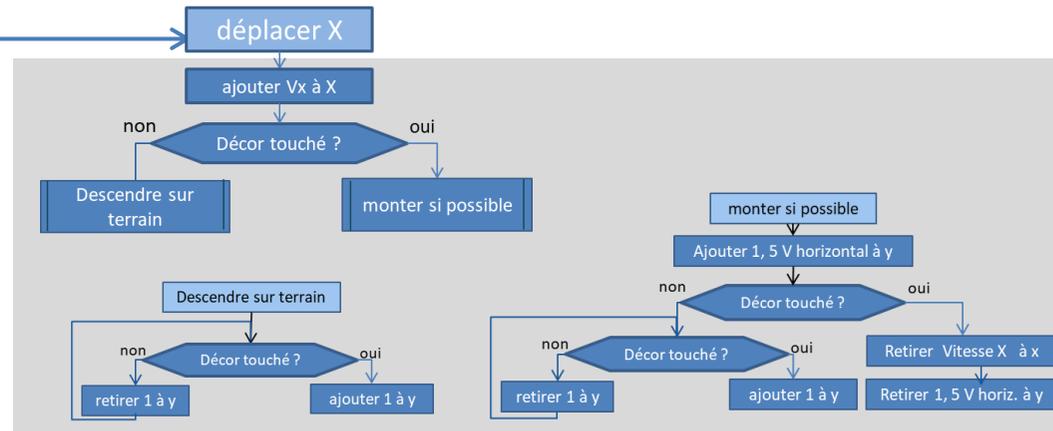
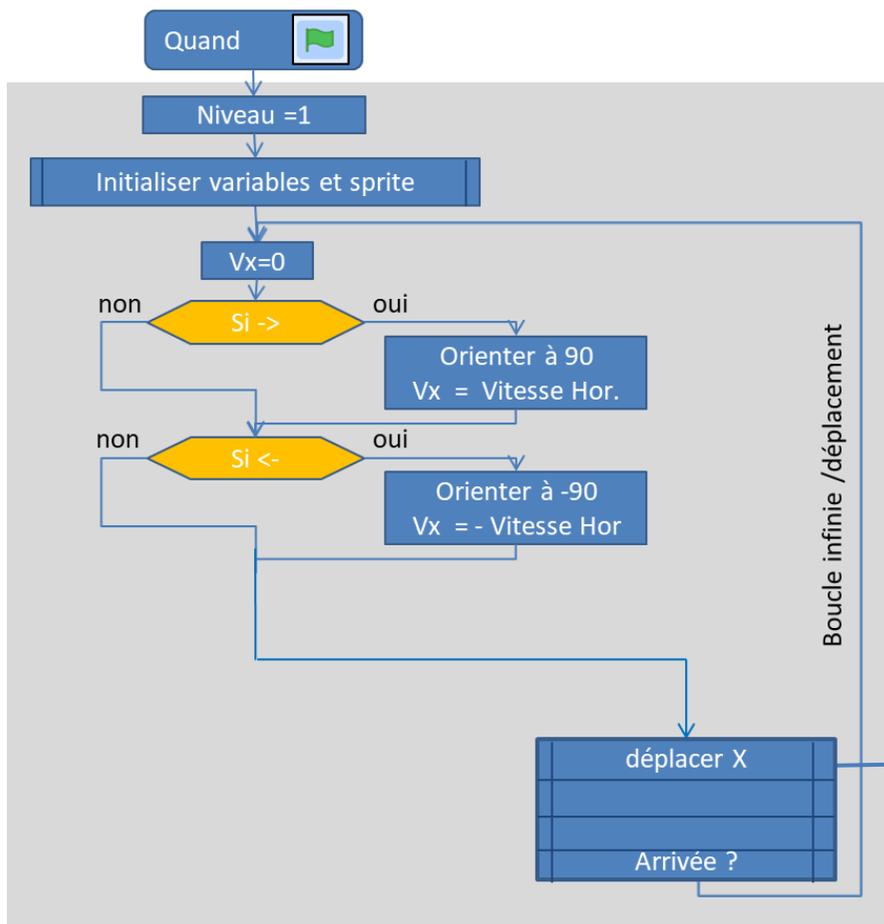
définir descendreSurTerrain



PSEUDO CODE ET DIAGRAMME DES FLUX : DÉPART DE V1



PSEUDO CODE ET DIAGRAMME DES FLUX : V2



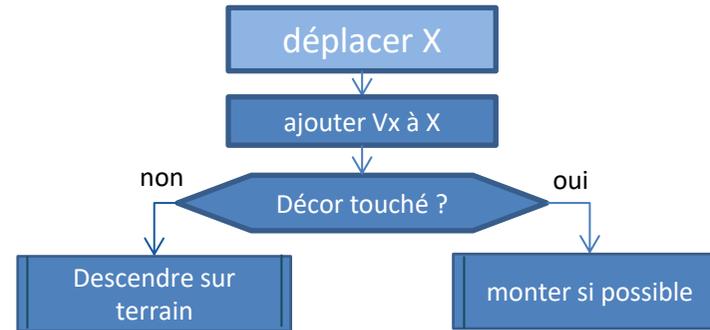
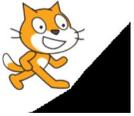
Ce que l'on va faire



V2 : DÉPLACER X EN SUIVI DE TERRAIN

Pseudo-code de la procédure **déplacer X** :

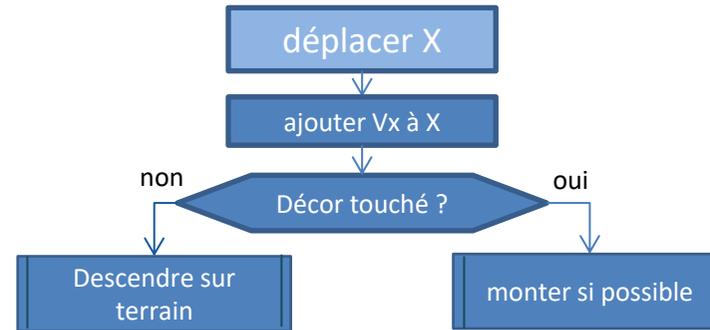
- ajouter V_x à X
- **si décor touché**
 - appeler "Monter si possible"
- **sinon:**
 - appeler "Descendre sur terrain"



V2 : DÉPLACER X EN SUIVI DE TERRAIN

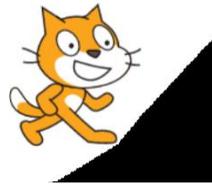
Pseudo-code de la procédure **déplacer X** :

- ajouter V_x à X
- **si décor touché**
 - appeler "Monter si possible"
- **sinon:**
 - appeler "Descendre sur terrain"



Règle retenue pour que le sprite suive le terrain :

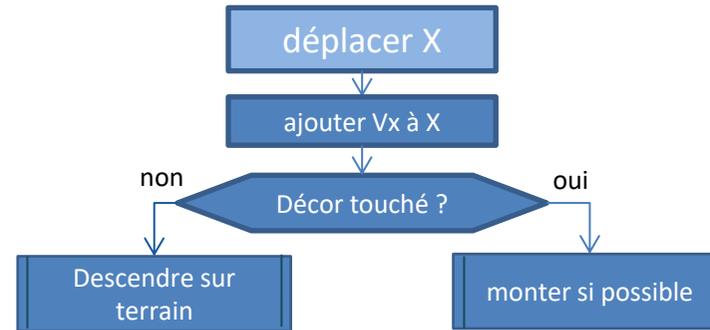
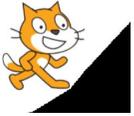
- il doit être 1 pixel au dessus du terrain,
- sans le toucher



V2 : DÉPLACER X EN SUIVI DE TERRAIN

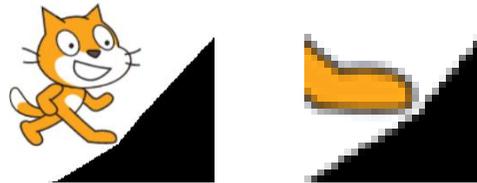
Pseudo-code de la procédure **déplacer X** :

- ajouter V_x à X
- **si décor touché**
 - appeler "**Monter si possible**"
- **sinon:**
 - appeler "**Descendre sur terrain**"



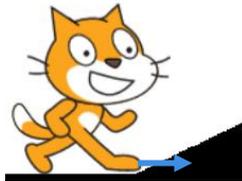
Règle retenue pour que le sprite suive le terrain :

- il doit être 1 pixel au dessus du terrain,
- sans le toucher



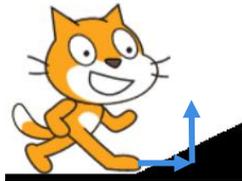
V2 : MONTER SI POSSIBLE

Pseudo-code de "**monter si possible**" :



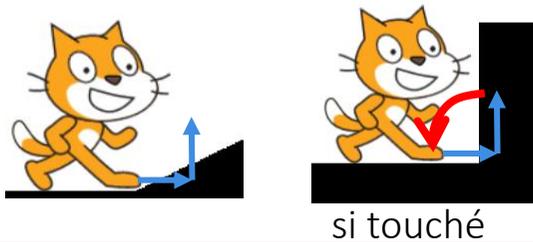
V2 : MONTER SI POSSIBLE

Pseudo-code de "**monter si possible**" :



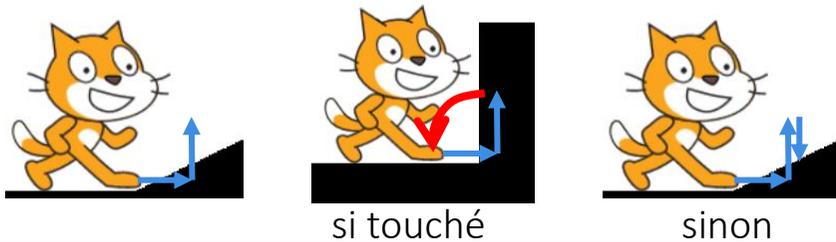
V2 : MONTER SI POSSIBLE

Pseudo-code de "monter si possible" :



V2 : MONTER SI POSSIBLE

Pseudo-code de "monter si possible" :



V2 : MONTER SI POSSIBLE

Pseudo-code de "monter si possible" :

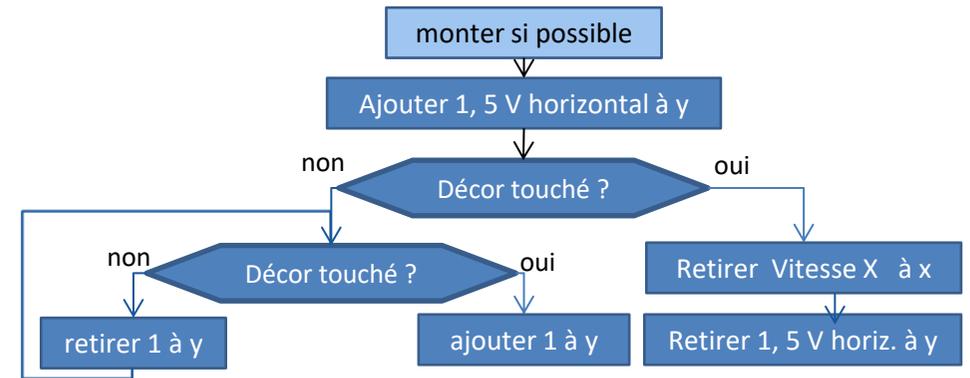
- ajouter $1,5 * V$ horizontal à Y (monter)
- si décor touché (revenir en arrière)
 - retirer $1,5 * V$ horizontal à Y
 - retirer Vitesse X à X
- Sinon (descendre jusqu'au terrain)
 - tant que décor non touché :
 - retirer 1 à Y
 - ajouter 1 à Y



V2 : MONTER SI POSSIBLE

Pseudo-code de "monter si possible" :

- ajouter $1,5 * V$ horizontal à Y (monter)
- si décor touché (revenir en arrière)
 - retirer $1,5 * V$ horizontal à Y
 - retirer Vitesse X à X
- Sinon (descendre jusqu'au terrain)
 - tant que décor non touché :
 - retirer 1 à Y
 - ajouter 1 à Y



V2 : MONTER SI POSSIBLE

Pseudo-code de "monter si possible" :

- ajouter $1,5 * V$ horizontal à Y (monter)
- si décor touché (revenir en arrière)
 - retirer $1,5 * V$ horizontal à Y
 - retirer Vitesse X à X
- Sinon (descendre jusqu'au terrain)
 - tant que décor non touché :
 - retirer 1 à Y
 - ajouter 1 à Y

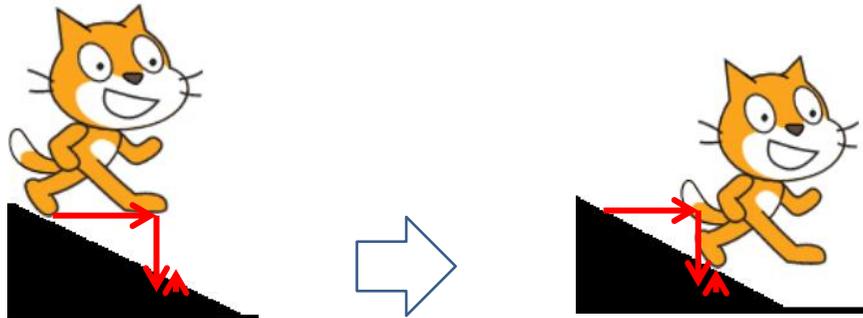


```
définir monterSiPossible
ajouter vitesse horizontale * 1.5 à y
si couleur [noir] touchée ? alors
  ajouter vitesse X * -1 à x
  ajouter vitesse horizontale * -1.5 à y
sinon
  répéter jusqu'à ce que couleur [noir] touchée ?
    ajouter -1 à y
  ajouter 1 à y
```

V2 : DESCENDRE SUR TERRAIN

Pseudo-code de "descendre sur terrain" :

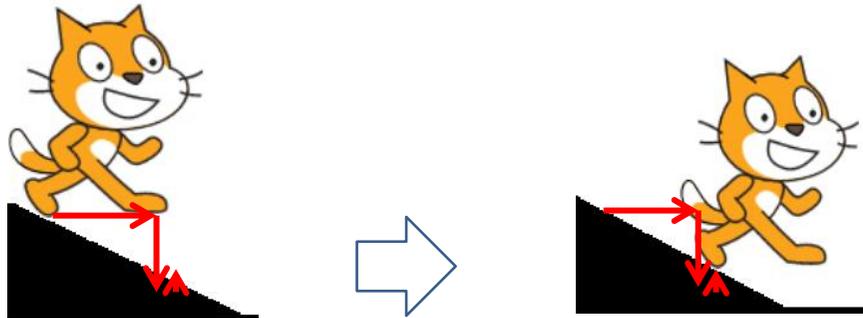
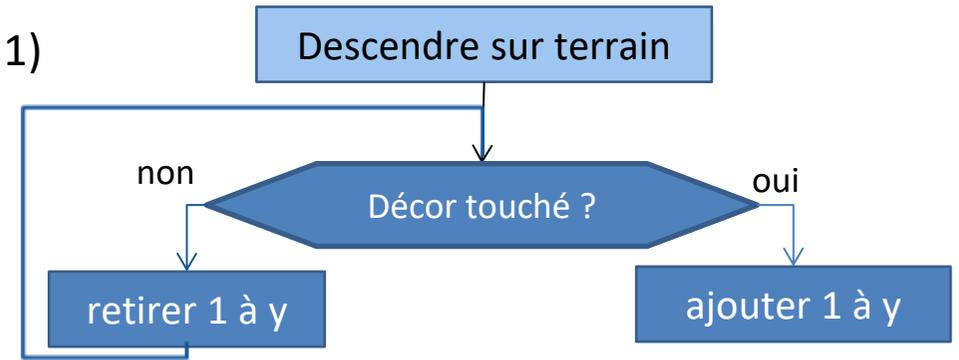
- tant que décor NON touché (descendre de 1)
 - retirer 1 à Y
- Ajouter 1 à Y
(remonter juste au-dessus du terrain)



V2 : DESCENDRE SUR TERRAIN

Pseudo-code de "descendre sur terrain" :

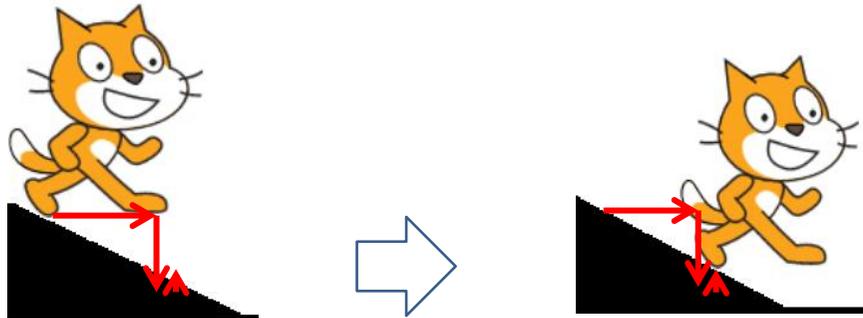
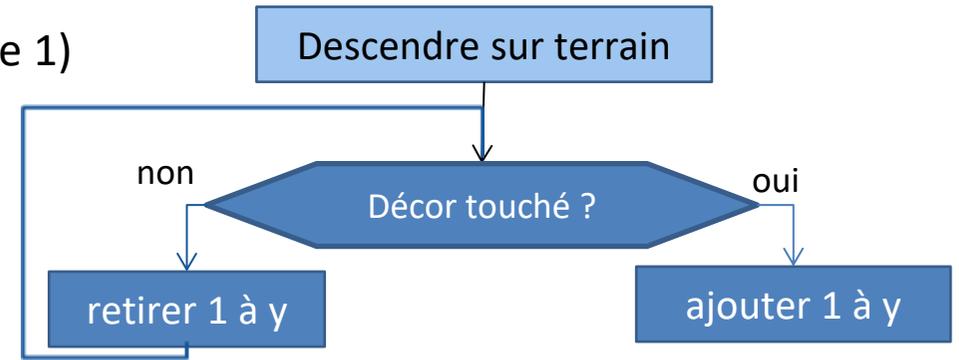
- tant que décor NON touché (descendre de 1)
 - retirer 1 à Y
- Ajouter 1 à Y
(remonter juste au-dessus du terrain)



V2 : DESCENDRE SUR TERRAIN

Pseudo-code de "descendre sur terrain" :

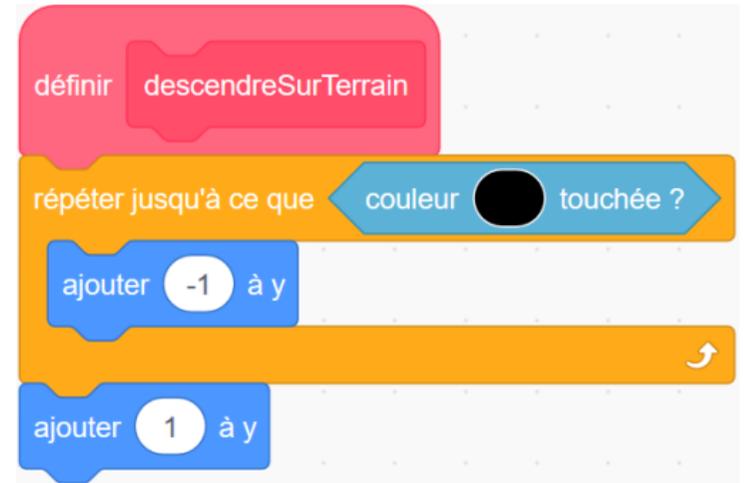
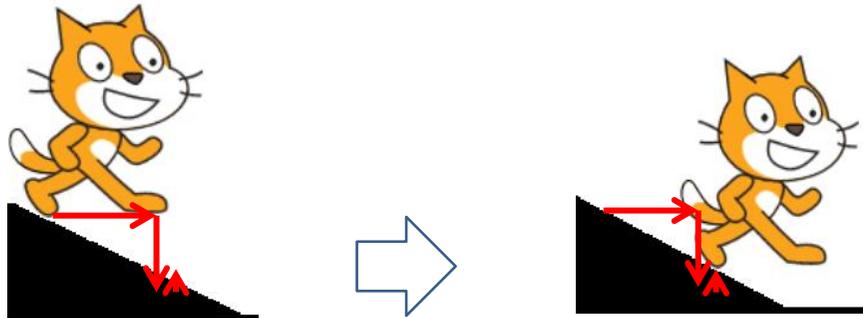
- Répéter jusqu'à décor touché (descendre de 1)
 - retirer 1 à Y
- Ajouter 1 à Y
(remonter juste au-dessus du terrain)



V2 : DESCENDRE SUR TERRAIN

Pseudo-code de "descendre sur terrain" :

- Répéter jusqu'à décor touché
 - retirer 1 à Y (descendre de 1)
- Ajouter 1 à Y (remonter juste au-dessus du terrain)



V2 : MISE À JOUR DE LA PROCÉDURE DÉPLACER X

V1

```
définir déplacer X
ajouter vitesse X à x
```

V2

```
définir déplacer X
ajouter vitesse X à x
si couleur [noir] touchée ? alors
  monterSiPossible
sinon
  descendreSurTerrain

définir monterSiPossible
ajouter vitesse horizontale * 1.5 à y
si couleur [noir] touchée ? alors
  ajouter vitesse X * -1 à x
  ajouter vitesse horizontale * -1.5 à y
sinon
  répéter jusqu'à ce que couleur [noir] touchée ?
  ajouter -1 à y
  ajouter 1 à y

définir descendreSurTerrain
répéter jusqu'à ce que couleur [noir] touchée ?
  ajouter -1 à y
  ajouter 1 à y
```