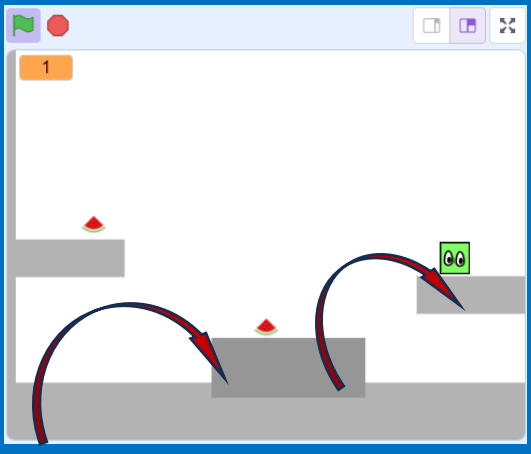
2 Jeu de plateforme



C'est parti pour un Super Mario à recréer de toutes pièces ? Un beau défi et des techniques de programmation qui se complexifient.
Avec ce jeu de cartes, tu auras une bonne base pour t'y mettre.

Vidéo

https://jeunesingenieux.be/vid-s3-2

Proposition de solution

https://scratch.mit.edu/projects/1004933752

Inspiré par https://www.youtube.com/@griffpatch

Jeu de plateforme

Tu te lances en mode Défi ?

Regarde la vidéo. Nous te proposons ces étapes.

1. Prépare ton personnage : appelons-le ***blocky*** . Un simple bloc suffit pour commencer. Prépare le sol sur lequel il se déplacera.
2. Programme la chute de blocky sur le sol de manière à ce qu'il s'arrête juste à sa hauteur.
3. Gère l'ascension : si la touche flèche-haut est appuyée.
4. Il faudra interdire à blocky de s'envoler…
5. Gère les mouvements horizontaux et les collisions.
6. Ajoute des nouveaux mondes et réfléchis à une méthode pour passer de l'un à l'autre.
7. Ajoute de l'eau. Le jeu se termine si blocky tombe dans l'eau…

Et plus encore ?
RDV sur la dernière carte pour des idées de défis supplémentaires.

2.1 LA CHUTE DE BLOCKY SUR LE SOL

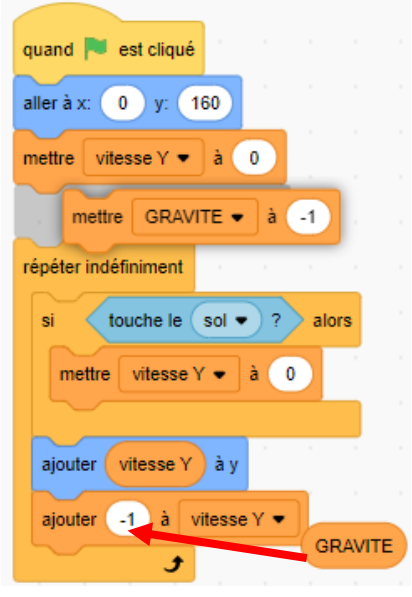
Objectifs

- Préparer le personnage (***blocky***) et ***le sol***.
- Programmer la chute de blocky sur le sol en tenant compte de la gravité.

Un nouveau sprite nommé ***blocky*** et ***le sol***

Pour ***blocky***, contente-toi d'un petit bloc. Tu pourras l'animer et l'améliorer plus tard. Crée un sprite pour ***le sol***.

La chute : ajoute ce code à ***blocky***



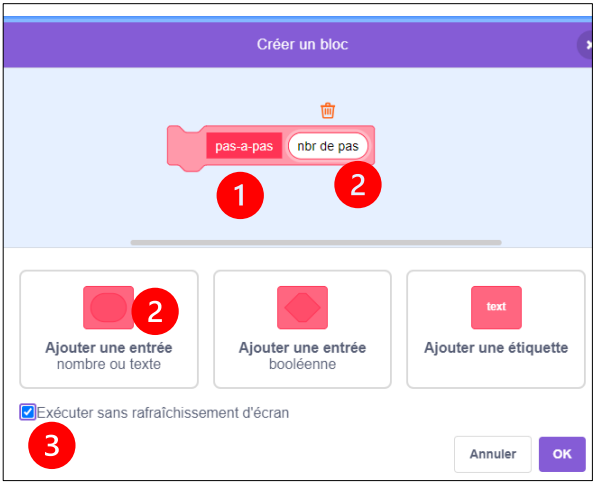
Dans un saut, il y a l'ascension et la chute. Programme d'abord la chute jusqu'au contact du sol.
La chute est un mouvement accéléré. L'accélération est provoquée dans la vraie vie par la gravité (l'attraction terrestre) et dans ce script par le bloc ***ajouter -1 à vitesse y***.

Crée deux variables . : ***GRAVITE*** et ***vitesse Y*** (= vitesse verticale)

2.2 GÉRER LA COLLISION AVEC LE SOL

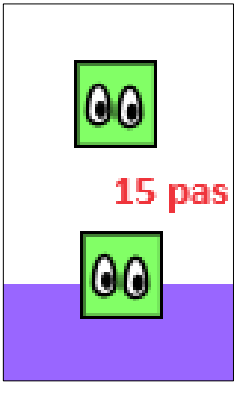
🎯 Objectif : la gestion de la collision

- Résoudre la collision de ***blocky*** avec ***le sol***. Il faut éviter qu'il s'enfonce dans le sol.



◥ Modifie le code de ***blocky***

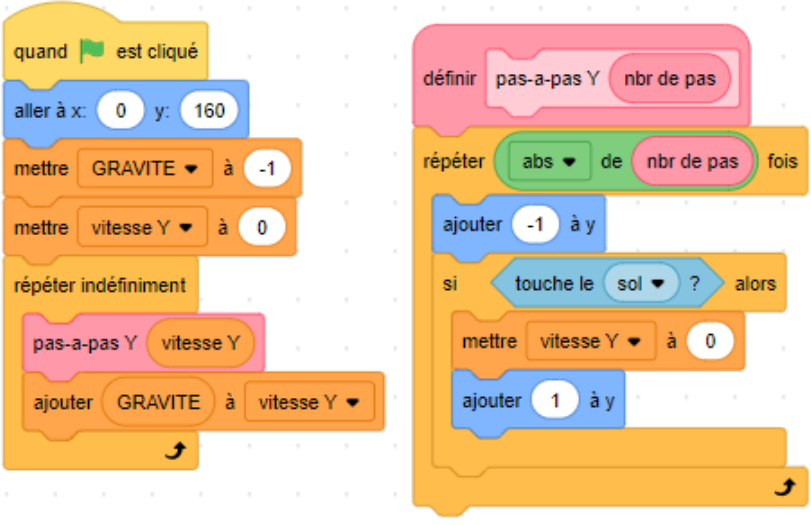
(1)Crée un bloc personnalisé ***pas-a-pas Y***, avec (2) une entrée ***nombre de pas*** et en cochant (3) ***Exécuter sans rafraîchissement d'écran***.



Avec ce bloc personnalisé, nous allons décomposer les déplacements de ***blocky*** et vérifier à chaque pas s'il touche le sol et ce « sans rafraîchissement d'écran »

Appelle ton coach à l'aide si tu ne comprends pas ce fonctionnement.

Prends le temps de comprendre ces scripts. Gérer des collisions, c'est la base !



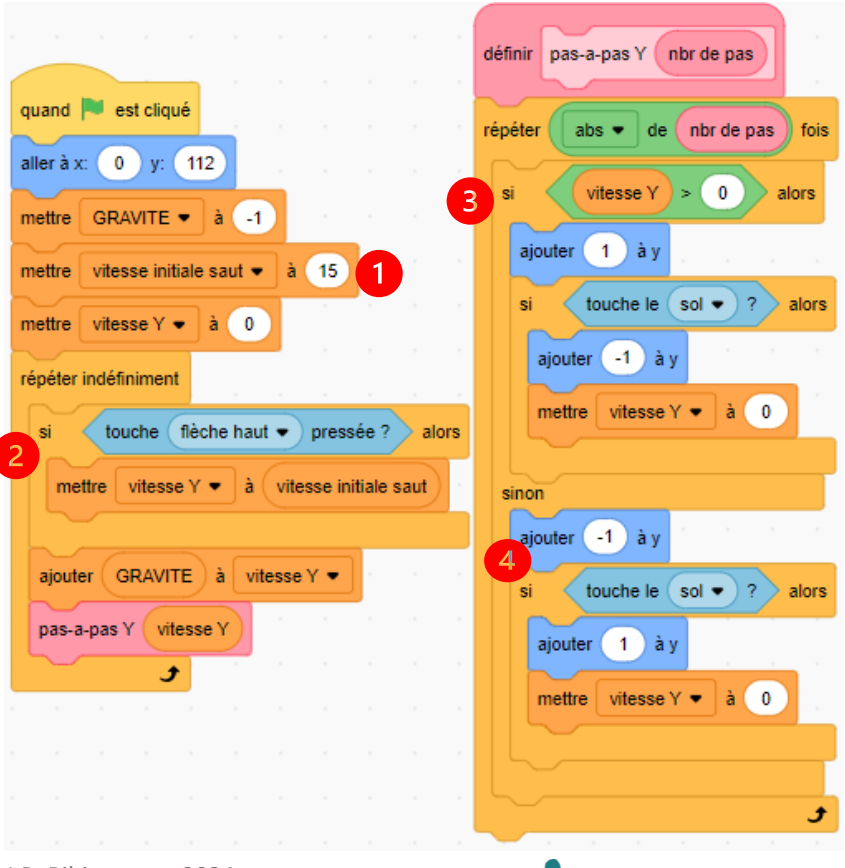
Il y a deux boucles ***répéter***. La première s'exécute à la fréquence de 30 exécutions par seconde. Parce qu'on a choisi l'option ***Sans rafraîchissement d'écran***, celle qui se trouve dans le bloc personnalisé s'exécute sans délai et n'enverra une image à l'écran que quand elle aura terminé son traitement. Cela permet de vérifier pas-à-pas si blocky touche le sol et ce, sans ralentir son déplacement apparent.

Demande à ton coach à quoi sert l'opérateur ***abs de…*** , ou retrouve-le ici : https://fr.scratch-wiki.info/wiki/Cat%C3%A9gorie:Blocs_Op%C3%A9rateur

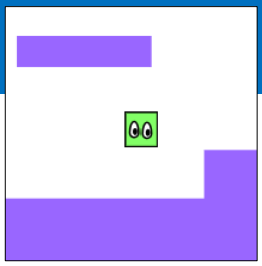
2.3 L'ASCENSION

Objectif : la gestion de l'ascension

Quand on saute, on commence avec une vitesse maximale (l'impulsion) pour atteindre une vitesse nulle au sommet et puis chuter. Modifie le code précédent comme ceci :



L'ASCENSION



Ajoute un élément flottant au sprite ***le sol*** en hauteur

Ici, nous proposons d'ajouter cet élément flottant au sprite ***le sol***.

Présentation du code (voir page précédente)

1. Crée un variable ***vitesse initiale saut***.
2. Ajoute le test ***si touche flèche-haut pressée*** pour affecter la valeur de ***vitesse initiale saut*** à ***vitesse y***.
3. Ajoute dans le bloc personnalisé le test : ***si la vitesse Y est supérieure à 0***, blocky monte : on ***ajoute 1 à Y*** et on gère les collisions avec le plafond en ajoutant -1.
4. sinon blocky descend, on ***ajoute -1 à Y*** et on gère les collisions avec le sol.

Teste ce code

Place ***blocky*** sous le plafond et observe ce qu'il se passe.
Fais-de même en plaçant ***blocky*** ailleurs.
Qu'observes-tu ?

Les sauts fonctionnent bien mais il reste possible de rebondir en l'air, ce qui n'est pas possible dans la vraie vie… Corrigeons cela.

2.4 SAUTER N'EST PAS VOLER

Objectif : éviter de s'envoler…

Tu as remarqué qu'en plein saut, si tu appuies rapidement sur la touche flèche-haut, blocky s'envole…

Ce n'est pas ce qu'on recherche.

Chouette exercice pour un programmeur en herbe ! Comment repérer le moment où ***blocky*** est sur le sol et autoriser la touche flèche-haut à produire ses effets seulement à ce moment-là ?

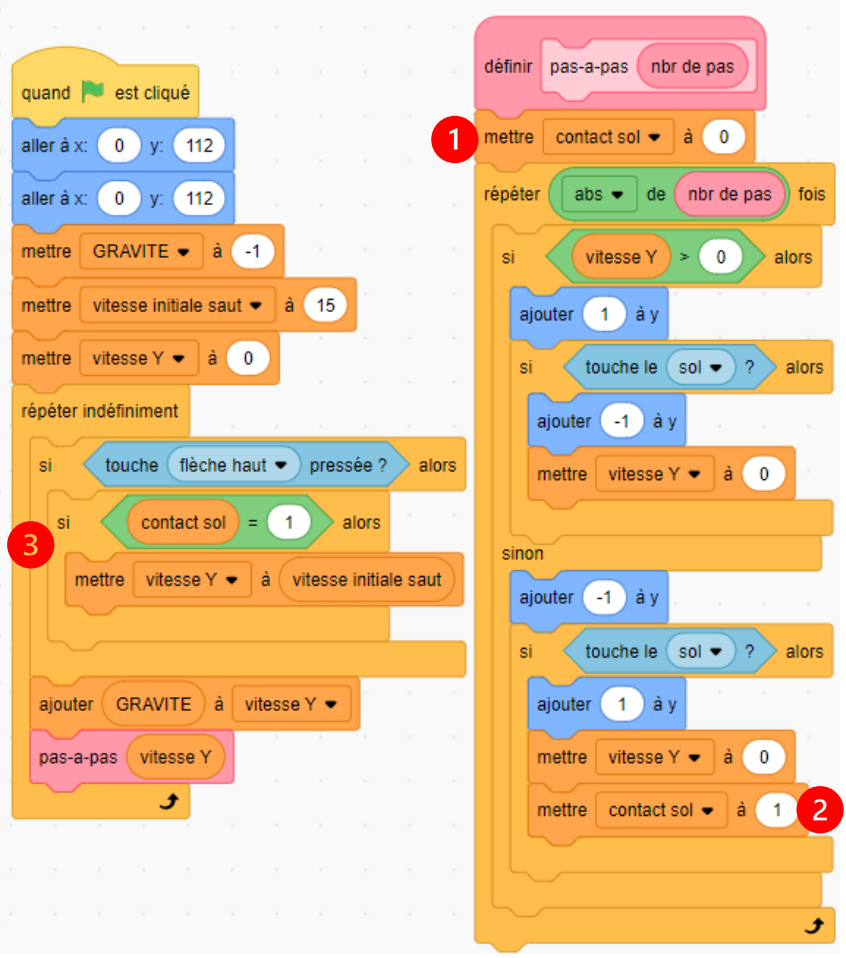
Une solution est proposée à la page suivante

Modifie le code de blocky

Créer une variable ***contact sol*** qui contiendra la valeur 1 quand ***blocky*** touchera le sol et zéro dans tous les autres cas.

Ajouter ces blocs : (1), (2) et le test (3).

SAUTER N'EST PAS VOLER

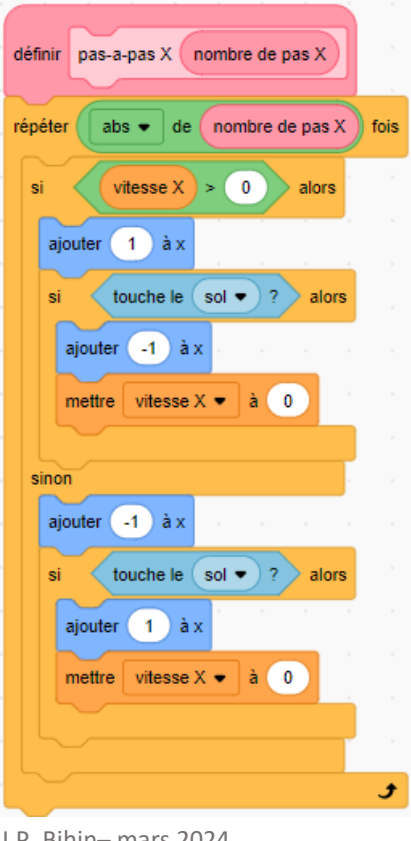


2.5 LES DÉPLACEMENTS HORIZONTAUX

🎯Objectif : gérer les déplacements horizontaux

Tu vas t'inspirer de la méthode utilisée pour les sauts.

- Appuyer sur la flèche-droite pour un déplacement à droite. Prévoir une phase d'accélération et de ralentissement quand la touche est lâchée. Prévoir aussi les collisions avec des obstacles.



- Idem à gauche…

Utilise la même méthode que pour les déplacements verticaux.

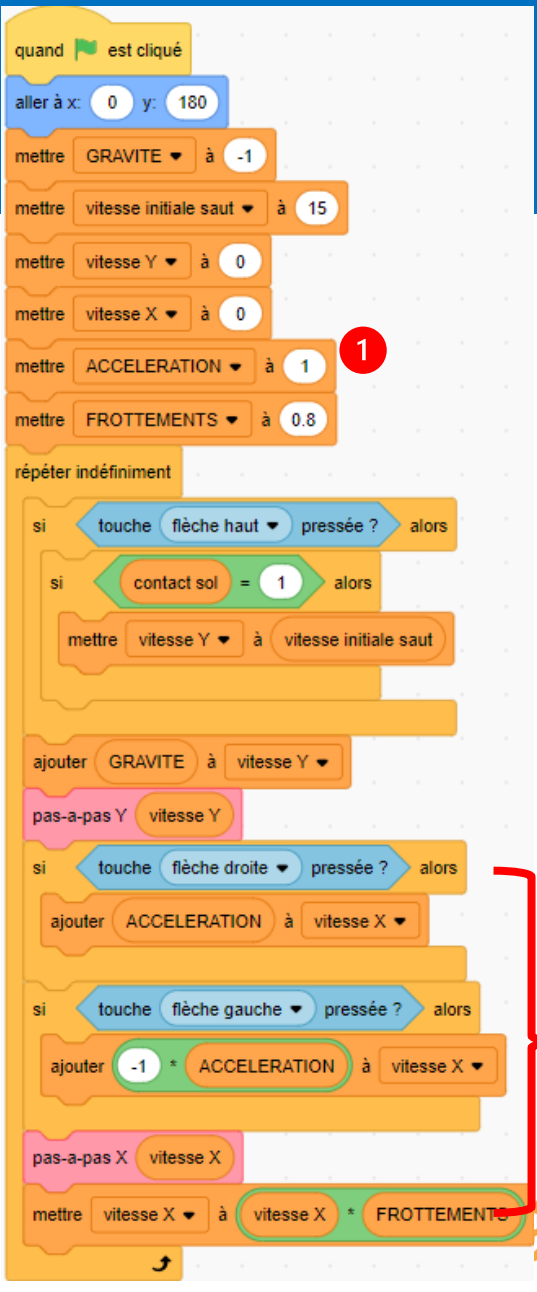
◥ Un nouveau bloc personnalisé

Crée une variable ***vitesse X*** et ajoute ce code pour ***blocky***.

Crée un bloc personnalisé ***pas-a-pas X*** et une entrée ***nombre de pas X***.

Précise bien le ***X*** pour éviter toute confusion avec les déplacements en Y du saut.

LES DÉPLACEMENTS HORIZONTAUX



Modifie le script principal de blocky

Crée une variable ***ACCELERATION*** et une variable ***FROTTEMENTS***.
(1) Initialise-les
(2) ajoute les blocs de déplacement

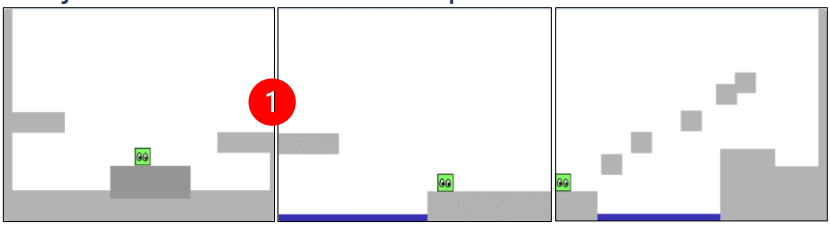
Fais appel à ton coach pour t'expliquer les notions d'accélération et de frottement.
Il pourra aussi t'expliquer pourquoi certaines variables sont en majuscules…
Le code de blocky : https://scratch.mit.edu/projects/1004856070

2.6 On change de monde !

Objectif : ajouter de nouveaux mondes

1. Crée des nouveaux mondes en ajoutant de costumes supplémentaires au sprite ***le sol***.
2. Gère le passage d'un monde à l'autre en testant la position de ***blocky***.

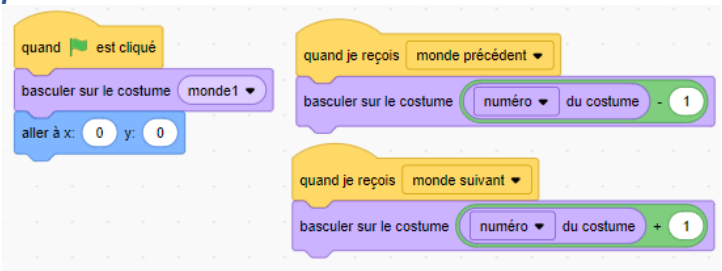
Ajoute deux costumes au sprite ***le sol***.



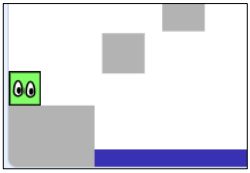
On ajoutera la mer (la ligne bleue) à l'étape suivante.
(1) Attention, les jonctions doivent être au même niveau.
Renomme les costumes ***monde1***, ***monde2*** et ***monde3***.

Ajoute ce code au sprite ***le sol***

Crée deux nouveaux messages : mode suivant et ***monde précédent***.

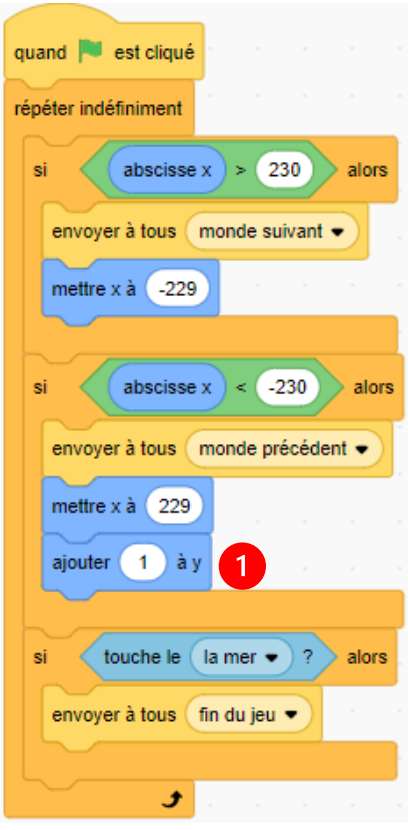


Crée un sprite pour l'eau (***la mer***)



Une simple ligne bleue placée en bas de la scène.

Ajoute ce code au sprite ***blocky***



Comme chaque fois, prends le temps de comprendre comment fonctionne ce code.

J'ai constaté en testant que ***blocky*** refusait dans certains cas à passer dans le monde précédent… comme s'il touchait le sol.

Pour résoudre ce problème, on ajoute (1) un petit saut de un pas.

2.7 À TOI DE JOUER !

Bien sûr, ce jeu peut encore être amélioré et enrichi !.

Voici quelques idées…

- Ajouter des objets à récolter sur le parcours et qui rapportent des points.
- Ajouter un Score, un stock de points de vie.
- Ajouter des ennemis.
- Ajouter une ou plusieurs conditions de fin de jeu. Par exemple quand un certain temps s'est écoulé, quand blocky chute trop lourdement sur le sol, quand le stock de points de vie est épuisé…
- Améliorer l'animation de blocky en lui ajoutant des costumes et des expressions qui changent en fonction des circonstances.
- Ajouter un costume hitbox qui permet d'améliorer le réalisme des contacts.

Etc…

Tu trouveras quelques-unes de ces idées réalisées ici :

https://scratch.mit.edu/projects/1005139414