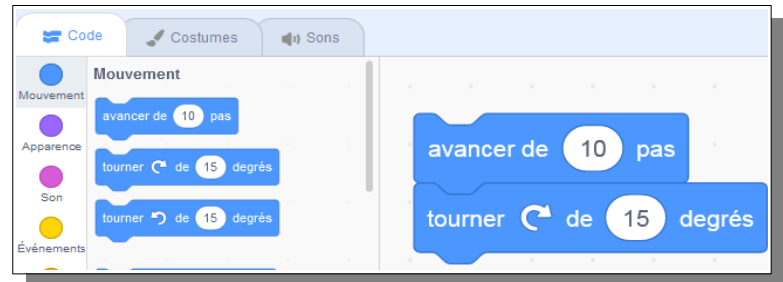


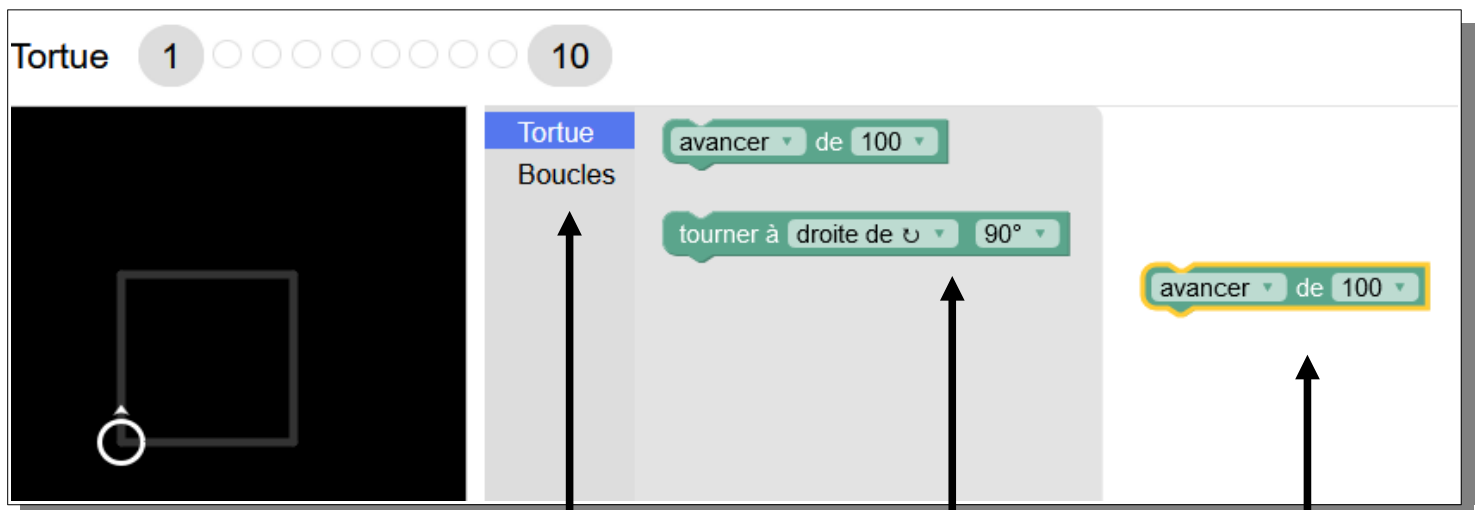
TON PREMIER DÉFI DE PROGRAMMATION

On va découvrir comment créer un programme graphique, enfin pas encore un programme mais plutôt une **suite de consignes**. Cette séquence ressemble à des phrases en français naturel, on appelle ça un **algorithme**.

Par exemple avec les logiciels inspirés de Scratch, on utilise des blocs dont chacun représente une phrase :



Pour s'entraîner, on va commencer par résoudre les défis de 1 à 9 sur le site de Blockly Games : <https://blockly.games/turtle?lang=fr>



Forme à réaliser

Catégories de blocs

Blocs de la catégorie

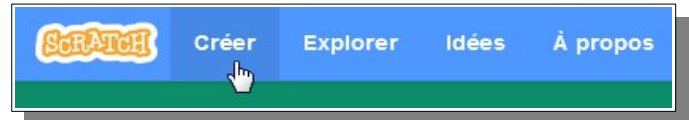
Zone de 'programme'

Il faut laisser cliqué avec le bouton gauche de la souris sur le bloc, puis le déplacer dans la zone de 'programme', et enfin relâcher la souris pour déposer le bloc

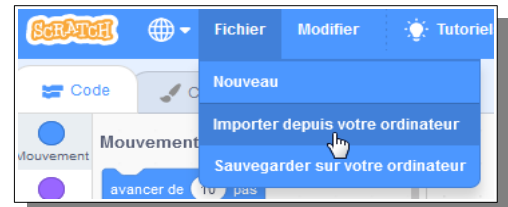
SE REPÉRER

Se rendre sur le site : <https://scratch.mit.edu>

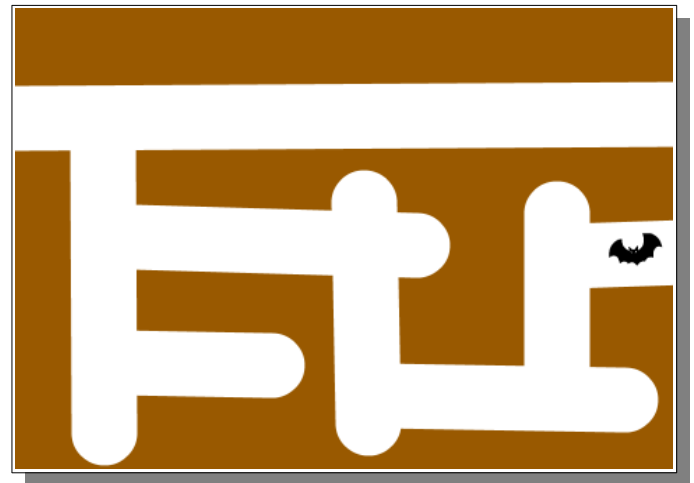
En haut de l'écran, cliquer sur 'Créer' :



Ensuite cliquer sur le menu 'Fichier' en haut de l'écran puis 'Importer depuis votre ordinateur' :



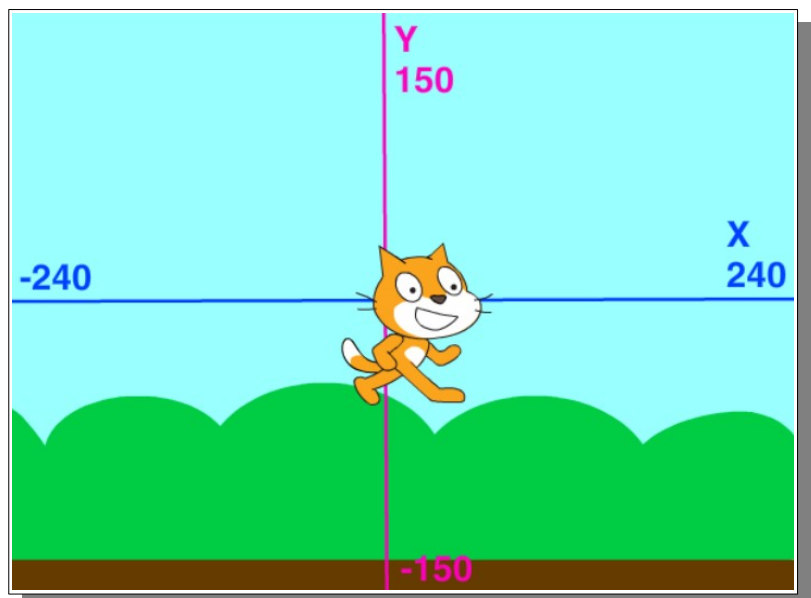
En ouvrant le fichier TP02, tu disposeras des éléments : une chauve-souris (appelons la *BATBAT*) et un labyrinthe de grottes.



Notre chauve-souris pourra donc se déplacer dans les 4 directions qui sont repérées par convention :

- horizontale → abscisse → axe x
- verticale → ordonnée → axe y

En voici les valeurs maximales :





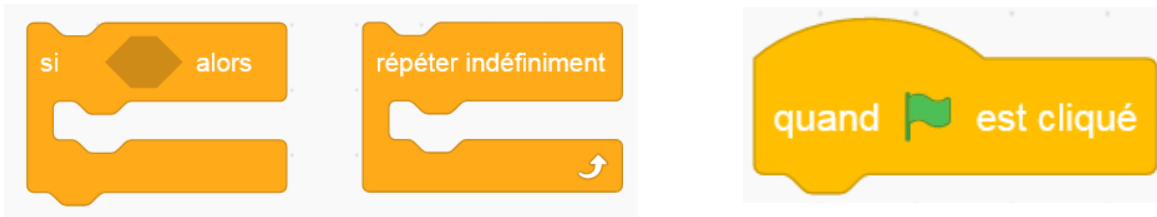
PROGRAMMONS LES DÉPLACEMENTS DE NOTRE CHAUVE SOURIS

Quels sont-ils ? Posons toutes nos idées et créons un algorithme :

- Si j'appuie sur la flèche 'haut' alors BATBAT monte ;
- Si j'appuie sur la flèche 'bas' alors BATBAT descend ;
- Si j'appuie sur la flèche 'droite' alors BATBAT va à droite ;
- Si j'appuie sur la flèche 'gauche' alors BATBAT va à gauche.



À toi de jouer ! N'oublie pas de lui faire tester les touches indéfiniment...

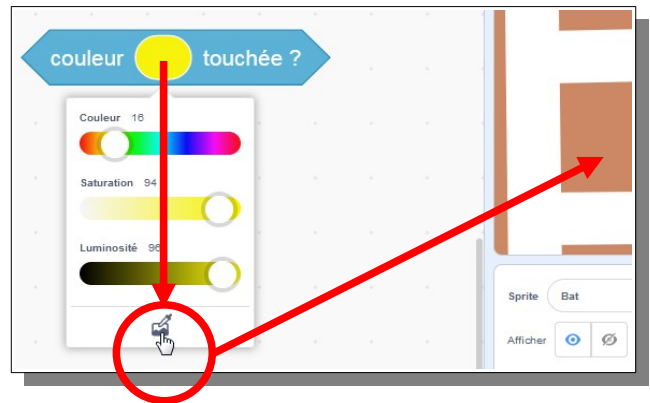


QUEL EST LE BUT DU JEU ?

Déplaçons BATBAT pour qu'elle sorte SANS toucher un mur de la grotte !

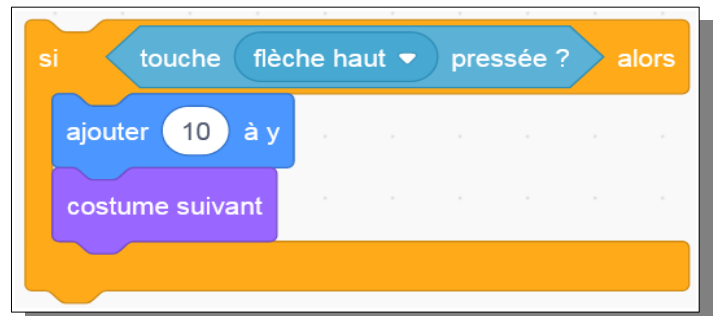
Pour cela, aidons-nous des couleurs : SI BATBAT touche la couleur de la grotte, on a perdu.

On verra plus tard comment gérer la séquence 'perdu'...



ET SI ELLE VOLAIT VRAIMENT ?

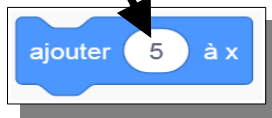
Pour créer une animation, il suffit de lui faire changer de costume pendant qu'elle effectue une action :



CE N'EST PAS TROP DIFFICILE ?

C'est ce qu'on appelle l'ergonomie d'un jeu, son gameplay, sa jouabilité. S'il est trop dur, personne ne voudra continuer ; s'il est trop facile, tout le monde s'en lassera...

Modifie la valeur du pas de déplacement et choisis ce qui te convient.



Et pour éviter de sans cesse la remettre à sa place quand on perd, créer une routine de remise à zéro :



Ou bien remet le jeu automatiquement à 0 en mixant ce qu'on connaît :



ET LES SCORES DANS TOUT ÇA ???

Il faut pouvoir compter le nombre de fois où perd/gagne, donc il faut **stocker** la mémoire du nombre précédent pour l'augmenter/diminuer. Pour cela on utilise en informatique des **variables**, c'est une sorte de 'boîte' qui stocke toute information que tu y déposes et que tu peux récupérer/utiliser quand tu veux.

Dans la catégorie des 'Données' tu vas 'créer une variable' que tu nommeras 'SCORE' :

Ta variable s'affiche alors en haut à gauche de la scène avec son contenu :



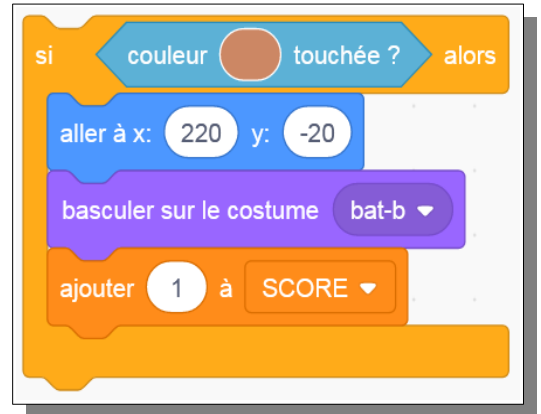
Tu peux la cacher en décochant :





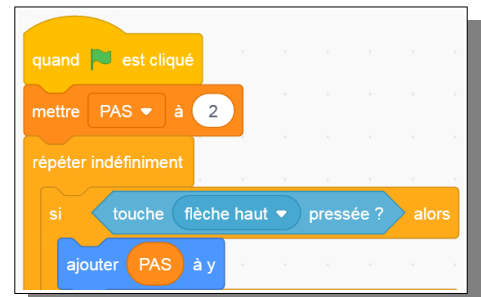
Donc au lieu de tout stopper, on va plutôt ajouter au score pour que le gagnant soit celui qui a le score (*nombre de chocs*) le plus faible :

Tu peux aussi 'mettre SCORE à 5' au début, et 'ajouter -1 à SCORE' à chaque contact, donc SI SCORE = 0 alors on stoppe tout. De cette façon tu changes le gameplay.



ALLONS PLUS LOIN...

Pour simplifier, nous pouvons utiliser une variable pour changer la vitesse de BATBAT et ainsi pouvoir la modifier pour changer la difficulté/gameplay.

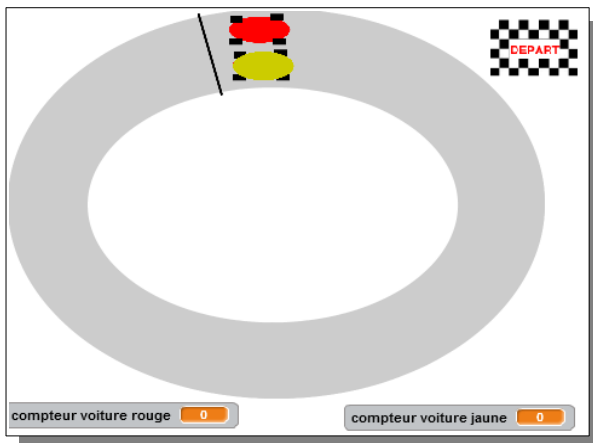


Par exemple à chaque victoire on va 'ajouter 2 à PAS' :



On a vu comment compter si on percute un mur, mais comment compter une victoire ? Et c'est quoi une victoire ? Qu'inventerais-tu pour définir la victoire ?

CRÉATION DU JEU

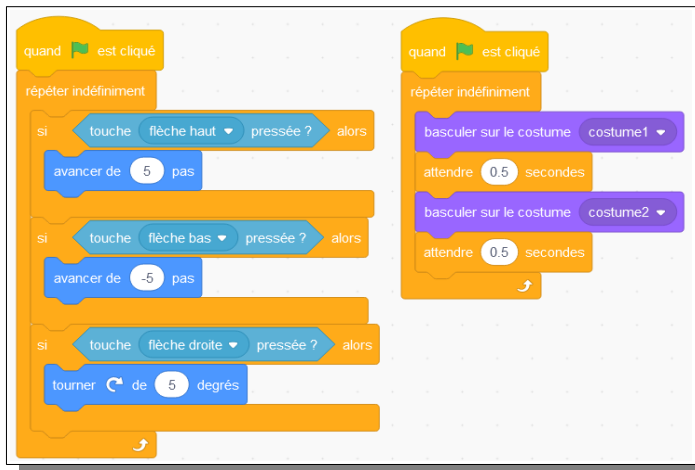


Après avoir guidé BATBAT, nous allons guider des voitures dans toutes les directions et compter les scores.

Ouvre le fichier TP03 pour avoir un point de départ à ton programme, puis nous allons utiliser tout ce que nous avons découvert.

Voici les étapes pour créer ton jeu :

- ✓ programmer la voiture pour qu'elle avance, recule, tourne à droite et à gauche ;
- ✓ rajouter un deuxième costume à la voiture pour avoir un gyrophare éteint et un autre allumé ;
- ✓ le gyrophare clignote indéfiniment, en parallèle au programme de déplacement ;

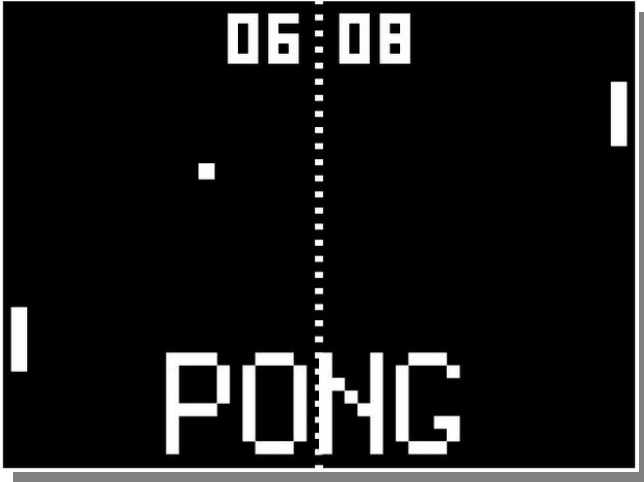


- ✓ à chaque début de programme, la voiture doit toujours partir du même point, sa position de départ ;
- ✓ quand la voiture sort de la route (*si elle touche...*), elle revient au point de départ ;
- ✓ rajoute une ligne d'arrivée et compte les tours (*si elle touche... alors ajouter...*) ;
- ✓ rajoute une deuxième voiture avec le même comportement que la première ;
- ✓ profite du jeu !

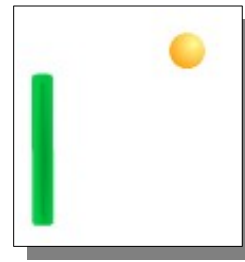


CONDITIONS & TESTS

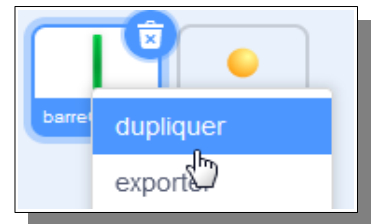
Après avoir créé des programmes pour dessiner, nous allons maintenant guider avec les touches et réaliser un jeu célèbre : PONG.



En ouvrant le fichier TP04, tu disposeras des éléments de départ : une barre et une balle.



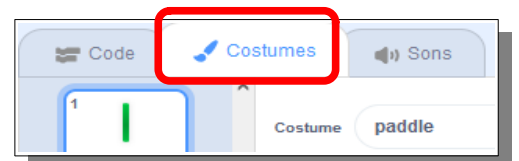
En faisant un clic droit sur la barre tu vas pouvoir la dupliquer :



Mais pour les différencier il faudra changer le nom et la couleur. Clique dans le champ 'Sprite' au-dessus du lutin et donne à chaque barre un nom :



Pour les couleurs, il faut se rendre dans l'onglet 'costumes' et utiliser des outils de dessin classiques, présents dans la barre de droite :



Pour exécuter une action plutôt qu'une autre, on fait un **test** : c'est une action sous **condition**. '*Si... alors*' est la plus classique, puis vient le '*Si... alors... sinon*'.



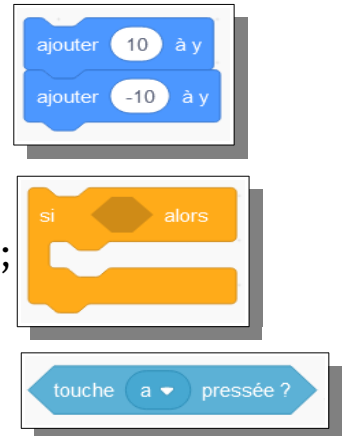
ON PROGRAMME

Le design est effectué, il nous reste à réfléchir sur le programme... il nous reste juste la partie la plus importante en fait !

Réfléchissons d'abord aux actions :

- les barres vont de haut en bas seulement ;
- SI j'appuie sur la touche A, alors la barre de gauche monte ;
- SI j'appuie sur la touche Q, alors la barre de gauche descend ;
- SI j'appuie sur la touche P, alors la barre de droite monte ;
- SI j'appuie sur la touche M, alors la barre de droite descend ;

→ n'oublie pas que tu peux dupliquer par un clic droit.



ATTENTION ! Un script, ou programme, est attribué au personnage que l'on a sélectionné, celui que l'on veut programmer, c'est SON programme. Donc pense à cliquer l'autre barre pour la programmer.

Exécute ton programme et teste !

ON RÉPARE

Ça ne fonctionne pas ! Ou plutôt cela fonctionne très bien mais cela ne se passe pas comme tu le voudrais.



En effet le logiciel teste tout de suite SI la touche 'a' est pressée, puis passe à la suite...c'est-à-dire la fin et s'arrête.

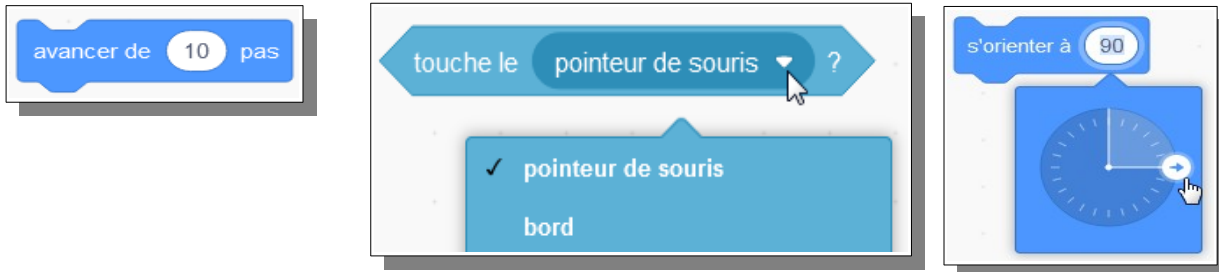
Nous voulons donc que le jeu se répète indéfiniment :



ET LA BALLE ???

Mince, on l'avait oubliée ! Bon, raisonnons sur ce que l'on attend d'elle...

Elle avance sans s'arrêter mais SI elle touche une barre elle change de sens...



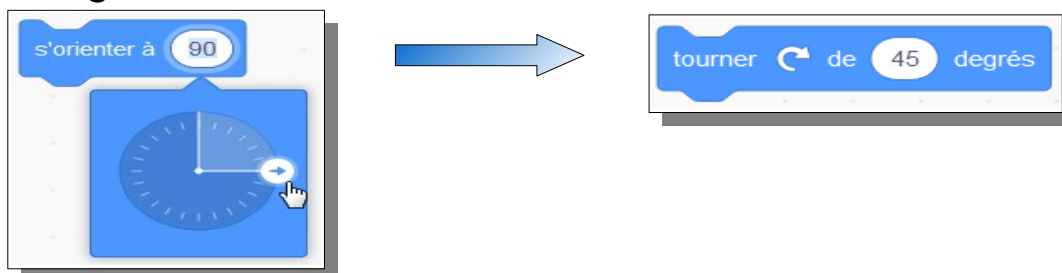
La réflexion que nous avons là, *détailler le comportement attendu*, quasiment le programme, c'est ce qu'on appelle un **algorithme**.

À toi de jouer ! Insère les éléments, torture ton programme jusqu'à ce que la balle rebondisse d'une raquette à l'autre.

BOF, ON S'ENNUIE...

C'est sûr qu'il suffit de laisser la raquette en face de la balle... mais c'est parce que nous ne faisons pas encore se déplacer la balle sur la verticale (y).

Introduisons du mouvement et changeons les orientations, *fixes*, en une rotation pour changer de direction :



À toi de jouer !



ON AMÉLIORE !

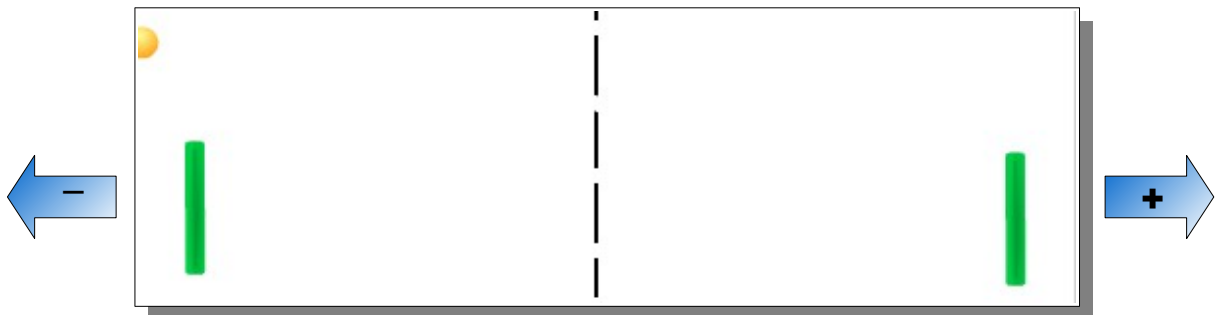
Oups, il te manque le bloc pour le rebond :

rebondir si le bord est atteint

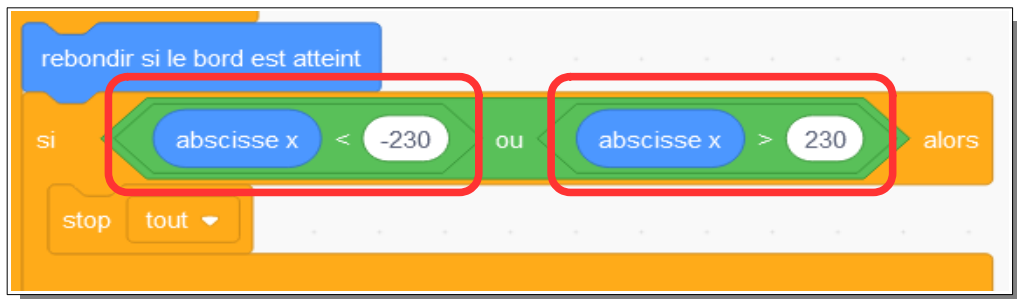
Ça y est, notre balle rebondit partout ! Même trop... Prenons le temps de formuler le comportement attendu (**algorithme**) : **SI** la balle va plus loin qu'une barre, on perd.



'Plus loin' ça n'existe pas... on veut savoir si la position horizontale (**abscisse**) de la balle est supérieure à la position de la barre de droite **OU** inférieure à la position de la barre de gauche :



Au programme de la balle, rajoute :



Oui, mais comment on compte les scores ? Et comment on affiche le gagnant ? Et comment on accélère le jeu ? Et comment on change le rebond de la balle ?

Tout ça a besoin de variables... notion à venir.