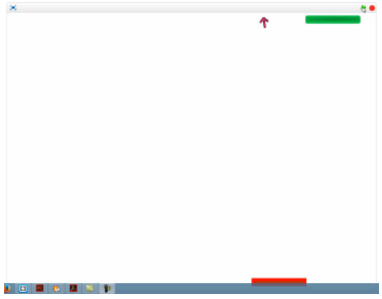
Déplacement autonome: PONG

→ Ouvrez le fichier "Pong.sb2"

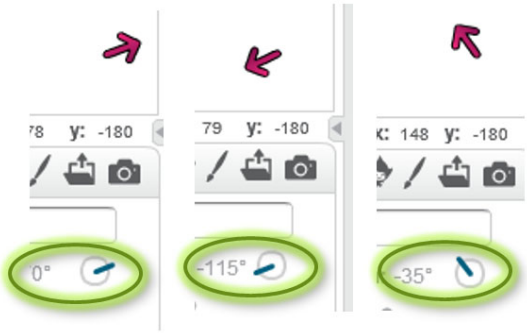
Il contient quatre lutins que nous allons programmer pour obtenir le jeu tel que décrit dans cette vidéo



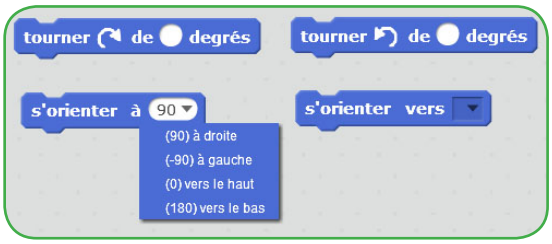
Exercice 1: déplacement relatif du lutin

« Mais comment un lutin peut-il se déplacer seul ?»

Un lutin est entre-autres caractérisé par sa position (x ; y) et par sa direction (accessible en cliquant sur l'icône de ce lutin.



Cette direction est un angle compris entre -180° et 180° qui peut se piloter grâce aux blocs :



Une fois le lutin orienté dans la direction voulue, il suffit de le faire avancer grâce au bloc :

En « bouclant » ce déplacement, on programme un déplacement autonome, indépendant des coordonnées du lutin.

→ A l'aide des blocs suivants, programmez pour le lutin « Ball » la séquence suivante et testez la.

- Au démarrage du programme, le lutin va en (0 ; 0)
- Il s'oriente dans une direction comprise entre -50° et 50°
- De manière répétée, il avance de 5, tout en rebondissant sur les parois.

Exercice 2: déplacement des raquettes

Dans ce jeu, on ne peut pas utiliser les écouteurs

En effet, un tel événement se déclenche à l'appui de la touche. Ici, s'il on veut un déplacement fluide, c'est l'état de la touche (enfoncée ou non) qui va piloter le déplacement. De plus, cela va permettre de tester l'appui de plusieurs touches simultanément (obligatoire dans un jeu à deux joueurs), ce que n'autorise pas l'écouteur qui passe par le « buffer clavier ». La différence est subtile et technique…

« Mais comment tester si une touche est pressée ou non ?»

Il suffit d'utiliser de tester le capteur de touche pressée.

Pour tester en continu, il suffit alors d'inclure le ou les tests de touche pressée au sein d'une boucle infinie: c'est ce qu'on appelle **une boucle de contrôle**.

→ En utilisant les blocs suivants (en plusieurs exemplaires), programmez le placement du lutin « Paddle » en (0 ; -170), puis son déplacement par les touches « flèche gauche » et « flèche droite ».

→ Faites de même pour le « Paddle2 », en le positionnant en (0 ;170) et en le contrôlant avec les touches « a » et « z ».

Exercice 3: Test de rebond sur les raquettes / Test de fin

On souhaite ajouter deux tests au script du lutin "Ball" :
-Si l'ordonnée du lutin est supérieure à 160 ou inférieure à -160,alors on stoppe le programme.
-Si le lutin touche le « Paddle » ou le « Paddle2 », alors il change son orientation :
le nouvel angle est égal à 180°- l'ancien angle, auquel on ajoute une variation aléatoire pour éviter que le lutin ne répète sans cesse le même parcours.

→ Complétez la boucle de contrôle du lutin « Ball » avec plusieurs des blocs suivants :

Exercice 4: Augmenter la difficulté

En utilisant une variable "Déplacement", on aimerait faire varier la vitesse du lutin "Ball".
L'avancée du lutin ne doit plus être constante mais variable:
celle-ci débute à 5, et augmente de 0,5 à chaque fois que le lutin touche une raquette.

→ Créez cette variable et modifiez le script du lutin "Ball" pour qu'il réponde à cet algorithme.