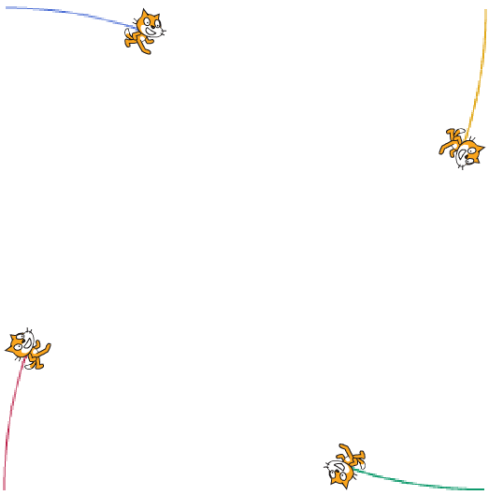
Plusieurs lutins

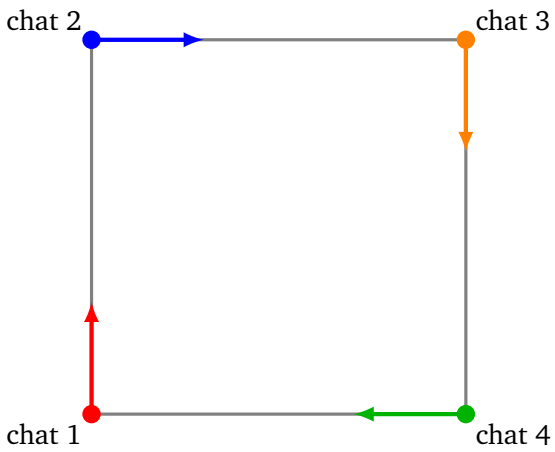
Vidéo ■ Plusieurs lutins - Activité 1
Vidéo ■ Plusieurs lutins - Activité 2
Vidéo ■ Plusieurs lutins - Activité 3

Activité 1.
Programme quatre chats qui courent les uns après les autres :



- le chat 1 court après le chat 2,
- le chat 2 court après le chat 3,
- le chat 3 court après le chat 4,
- le chat 4 court après le chat 1.

Voici les positions et orientations de départ :

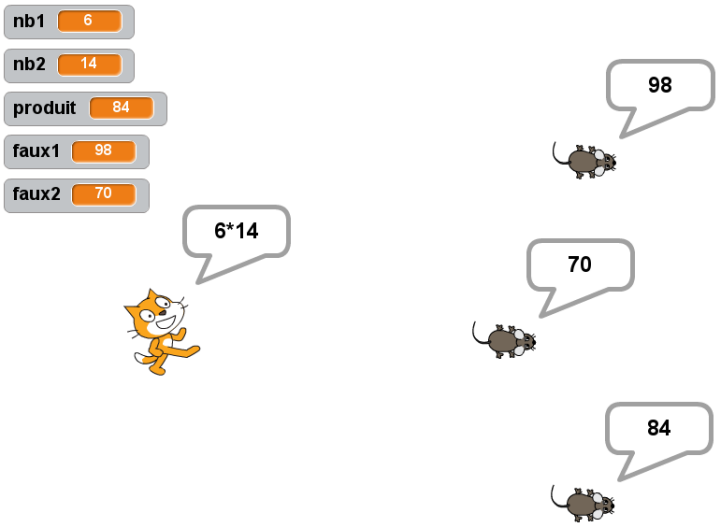


Plusieurs lutins.
Avec Scratch, tu peux contrôler plusieurs lutins en même temps. Chaque lutin aura ses propres instructions. Pour définir un nouveau lutin, clique sur l'icône « nouveau lutin » ou bien clique le bouton droit de la souris sur un lutin existant, puis sur « dupliquer ».

Blocs utiles.

Activité 2.
Programme un petit jeu de calcul mental avec un chat et trois souris.

- Le chat demande le résultat d'une multiplication.
- La souris 1 affiche le bon résultat.
- Les souris 2 et 3 affichent des résultats faux.
- Le chat doit avancer en suivant le pointeur de la souris de l'ordinateur jusqu'à toucher la souris qui affiche le bon résultat.



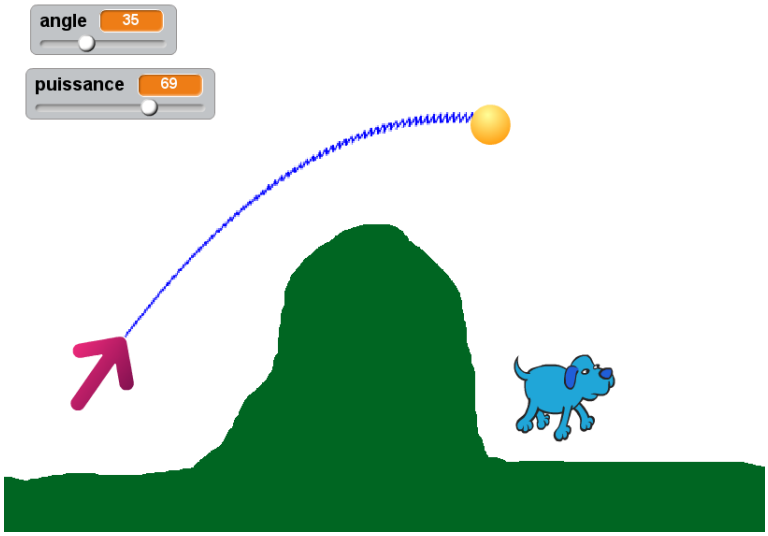
Voici comment structurer ton programme :

- **Initialisation.**
  Débute par les instructions suivantes, qui peuvent être incluses avec celles du chat.
  — Choisis deux nombres `nb1` et `nb2` au hasard entre 5 et 15.
  — La bonne réponse sera `produit` = `nb1` × `nb2`.
  — Deux mauvaises réponses seront proposées par exemple : `faux1` = (`nb1` +1)× `nb2` et `faux2` = (`nb1` −1)× `nb2`.

- **Le chat.**
  — Il démarre de la gauche.
  — Il affiche l'opération « `nb1` × `nb2` ».
  — On répète indéfiniment : s'orienter vers le pointeur de la souris de l'ordinateur et avancer de 3 pas.
  — S'il touche la souris 1, c'est gagné !

- **Les souris.**
  — Chaque souris se place au hasard, avec x entre 0 et 150 et y entre −150 et 150.
  — La souris 1 affiche `produit`, les autres souris affichent les mauvais résultats `faux1` et `faux2`.

Activité 3.
Programme un canon qui lance une balle. Si la balle touche le chien, c'est gagné ! On peut régler l'orientation du canon par un angle et aussi régler la puissance du lancer.



Le canon.
- Définis une variable `angle`.
- Répète indéfiniment : s'orienter à `angle`.
- Une fois le programme lancé, tu peux régler l'angle à l'aide d'un curseur (affiche la variable `angle` et double clique sur cet affichage, jusqu'à obtenir le potentiomètre).

L'arrière-plan et le chien

- Dessine un arrière-plan coloré (ici vert), avec une montagne au milieu afin d'éviter un tir direct.
- Place un chien à droite de la montagne. Pour compliquer la mission, le chien peut être en mouvement de gauche à droite.

La balle : tir parabolique.
C'est la partie la plus délicate. Une fois la balle lancée, elle suit une trajectoire en forme de parabole. Le principe est expliqué plus loin.
Dans la pratique :

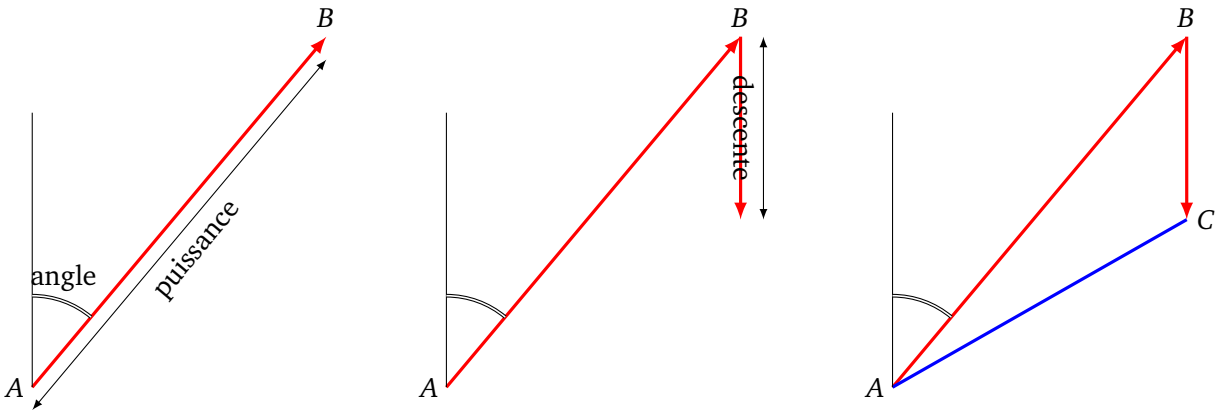
- Définir une variable `puissance` (réglable comme pour `angle` ci-dessus).
- Définir une variable `descente` et l'initialiser à 0.
- Orienter la balle à `angle`.
- Répéter :
  — avancer de 0.1×`puissance`,
  — ajouter −0.1 à `descente`,
  — ajouter `descente` à y.

La balle : gagné ou perdu ?

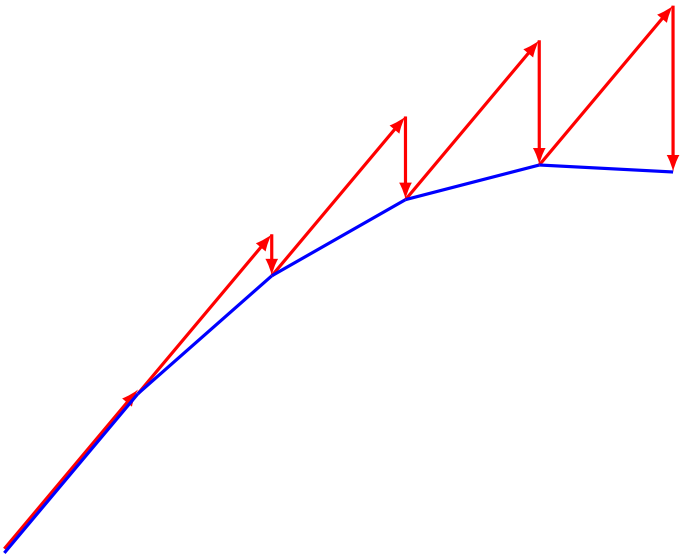
- Si la balle touche la couleur verte ou si la balle touche le bord de l'écran : c'est perdu.
- Si la balle touche le chien : c'est gagné !

Voici le principe du tracé du tir parabolique :

- La trajectoire est formée de petits segments.
- Chaque segment s'obtient en suivant deux vecteurs (des « flèches »).
- Le premier vecteur est déterminé par l'angle et la puissance : il reste tout le temps le même.
- Le second vecteur est un vecteur vertical dirigé vers le bas. Ce vecteur va être de plus en plus grand (c'est la variable `descente`).
- Le segment à tracer part au début du premier vecteur et arrive à la fin du second.
- On recommence, mais avec un vecteur vertical un peu plus grand.



(a) tir (b) descente (c) somme des vecteurs



(d) simulation du tir parabolique