

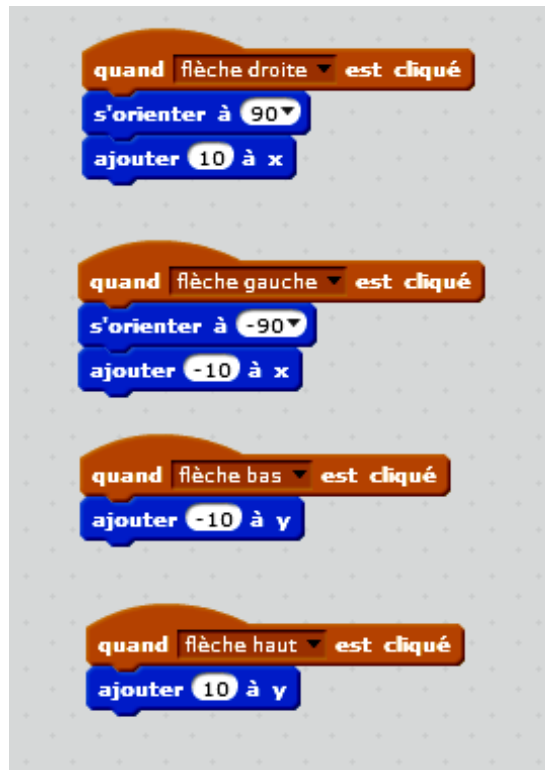
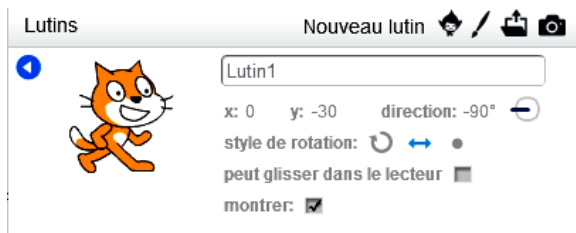
Une prise en main de SCRATCH

Niveau 1 – Éléments de réponse

Exercice 1

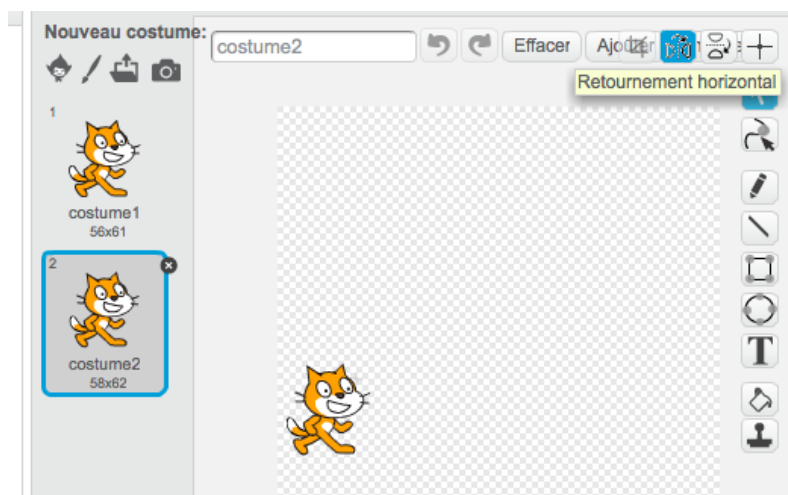
1. Écrire un script qui déplace le lutin vers la droite quand on appuie sur la touche « flèche droite » du clavier. Ajouter la possibilité d'aller vers la gauche quand on appuie sur la touche « flèche gauche ». Enfin, ajouter la possibilité d'aller vers le haut, vers le bas de la scène.

Si l'on ne veut pas que le lutin ait la tête en bas lorsqu'il se retourne, cliquer sur le « i » à côté du lutin et changer le « style de rotation » : double flèche.

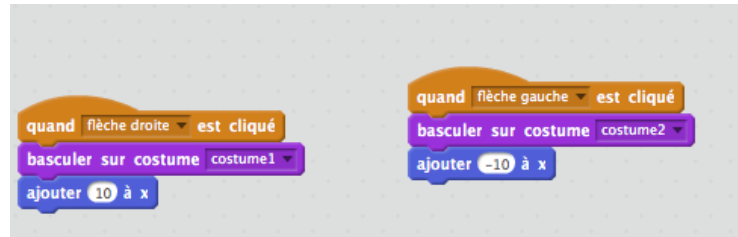
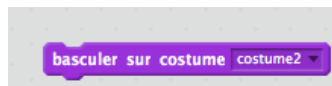


Il est possible aussi de jouer avec les costumes pour résoudre ce problème.

Cliquer sur l'onglet « costume ». Supprimer le second costume. Dupliquer le premier costume « bouton droit sur le premier costume ». Cliquer sur le bouton « retournement horizontal »



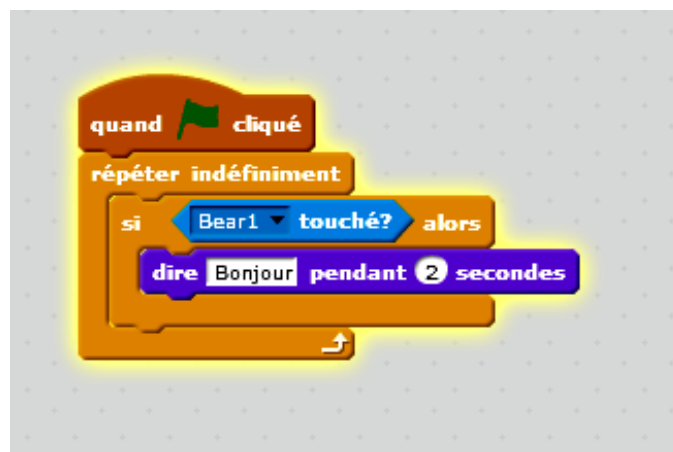
Ensuite il faut utiliser la commande



2. Compléter le script pour que le lutin dise « Bonjour » lorsque l'on appuie sur la « barre d'espace ».



3. Ajouter un second lutin de votre choix à l'exercice précédent. Écrire un script afin que le chat dise « Bonjour » quand il touche le second lutin.



4. Écrire un script permettant de faire avancer indéfiniment le second lutin et de le faire rebondir chaque fois qu'il atteint un bord.



Exercice 2

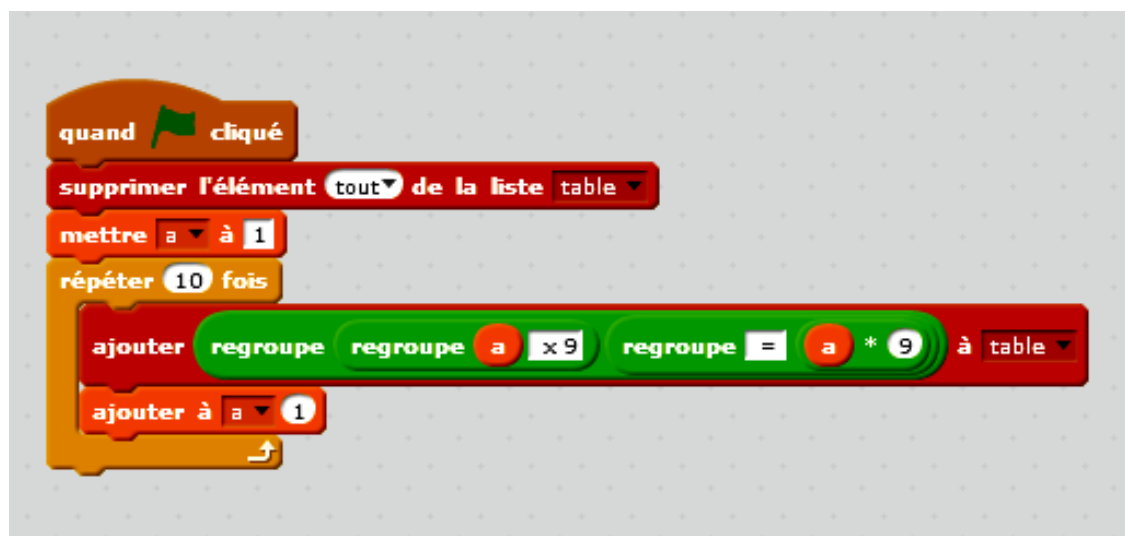
1. Écrire un programme faisant réciter le début de la table de 9 par le lutin.

Nécessité de créer une variable. Il est possible de créer aussi une liste pour afficher la table mais on peut cependant faire dire au lutin la table, c'est une autre possibilité.

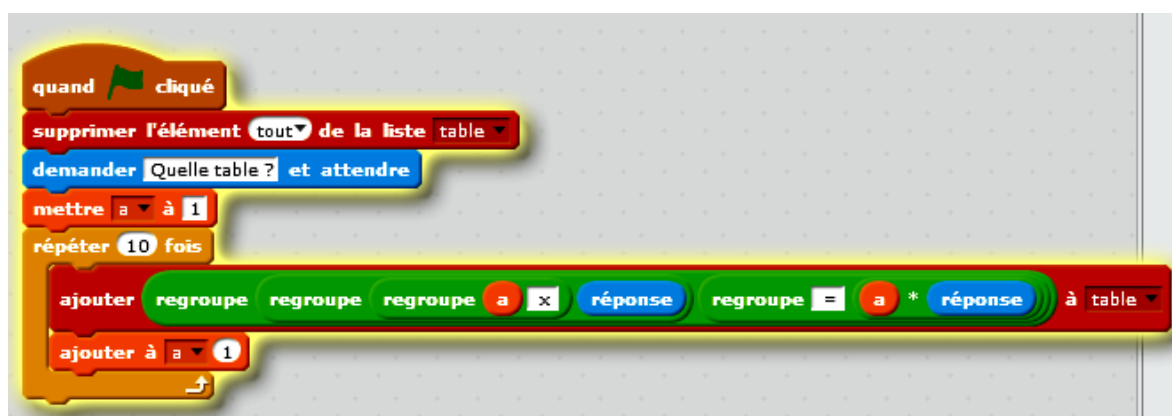
The first screenshot shows the Scratch IDE with the 'Données' (Data) category selected in the Scripts palette. A 'Nouvelle variable' (New variable) dialog box is open, with the variable name 'a' entered. The 'Pour tous les lutins' (For all sprites) radio button is selected. The second screenshot shows the same IDE with a 'Nouvelle liste' (New list) dialog box open, with the list name 'table' entered. The 'Pour tous les lutins' radio button is also selected. In the background, a script is visible: 'quand cliqué' (when clicked) followed by 'mettre a à 0' (set a to 0), 'ajouter à a 1' (add 1 to a), 'répéter 10 fois' (repeat 10 times), and 'montrer la variable a' (show variable a).

The final screenshot shows the Scratch IDE with the completed program. The variable 'a' is displayed with the value 11. A list named 'table' is shown with the following values: 9, 18, 27, 36, 45, 54, 63, 72. The script in the background is: 'quand cliqué' (when clicked), 'mettre a à 1' (set a to 1), 'répéter 10 fois' (repeat 10 times), 'ajouter a * 9 à table' (add a * 9 to table), and 'ajouter à a 1' (add 1 to a). The Scratch cat is visible on the left side of the stage.

On peut améliorer le programme (et l'affichage) :

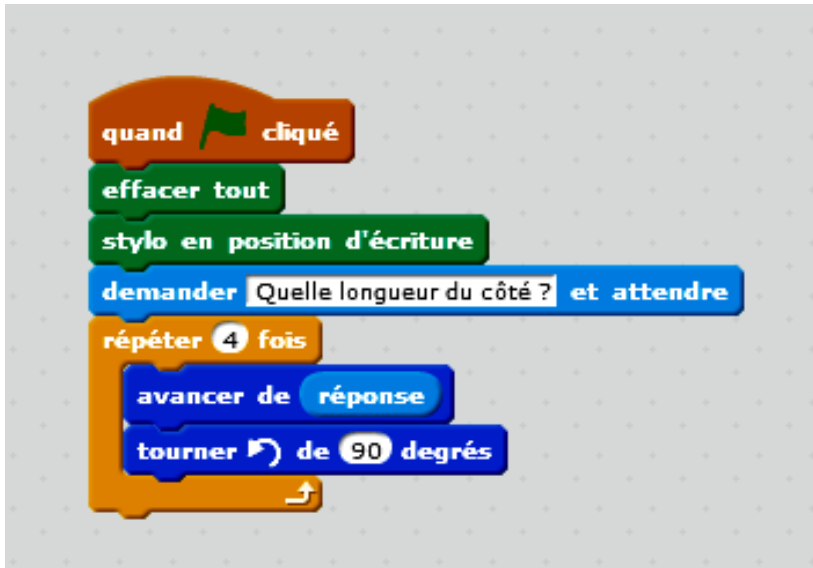


2. Demander à l'utilisateur de choisir la table qui va être récitée.
La variable « réponse » contient la chaîne de caractère entrée par l'utilisateur.



Exercice 3

Faire tracer un carré dont la longueur du côté est donnée par l'utilisateur.



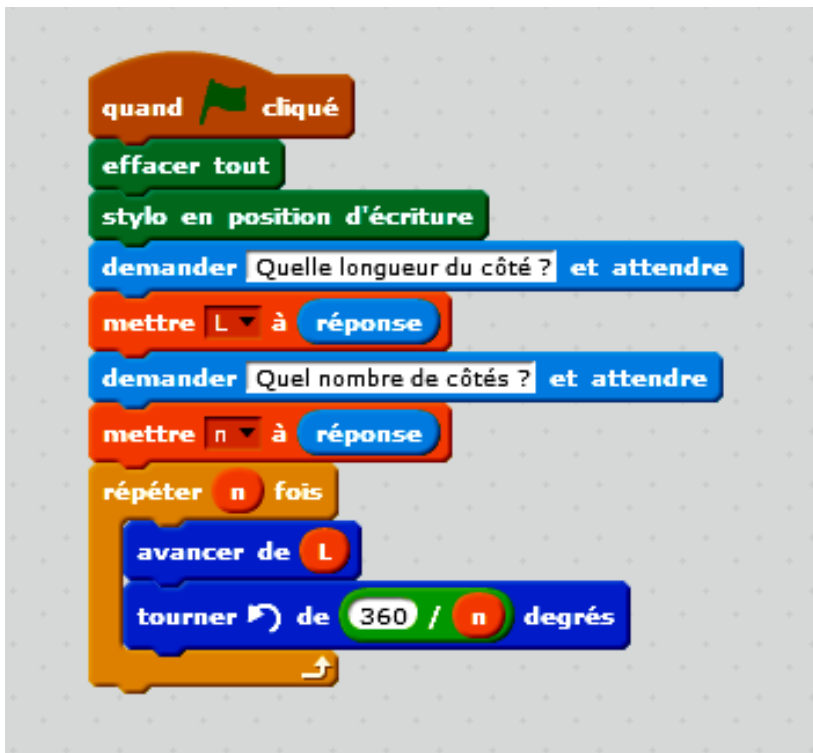
Niveau 2 – Éléments de réponse

Exercice 1

On peut reprendre l'exercice précédent.

Dans Scratch, les valeurs entrées sont stockées dans « réponse ». Il faut donc affecter immédiatement « réponse » à la variable correspondante.

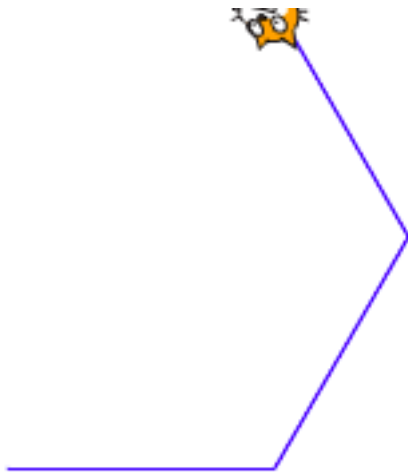
Remarque : l'angle de rotation correspond à l'angle au centre.



Remarque : L'utilisation des angles sous Scratch demande un peu de la réflexion. En effet, si nous souhaitons par exemple, tracer un triangle équilatéral, ce script devrait répondre au problème.



Or voici ce qui est construit avec ce script.



Tourner de 60° sous Scratch, correspond à ce tracé.



En fait Scratch construit un angle de $180^\circ - 60^\circ$.

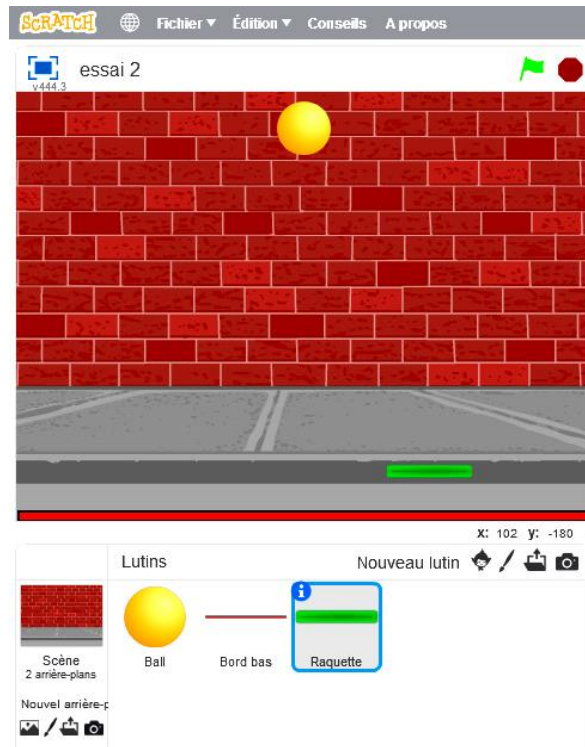
Pour le tracé d'un polygone régulier à n côtés, il suffit donc de demander à tracer un angle de $360^\circ/n$.

Exercice 2

On choisit dans la bibliothèque un arrière-plan et deux lutins (ball et paddle par exemple, que l'on peut ensuite renommer en français en cliquant sur « i »).

Pour distinguer le bord bas des autres bords, il a été ici choisi de dessiner un troisième lutin (un long rectangle rouge) placé en bas de la scène (on peut déplacer les lutins à la souris sur la scène).

Il est possible aussi de procéder autrement en utilisant l'ordonnée de la balle et de tester cette valeur.



On peut décomposer le problème en quatre ou cinq sous-problèmes. On gèrera la question du score à la fin.

On aura, dans un premier temps et par ordre de difficulté :

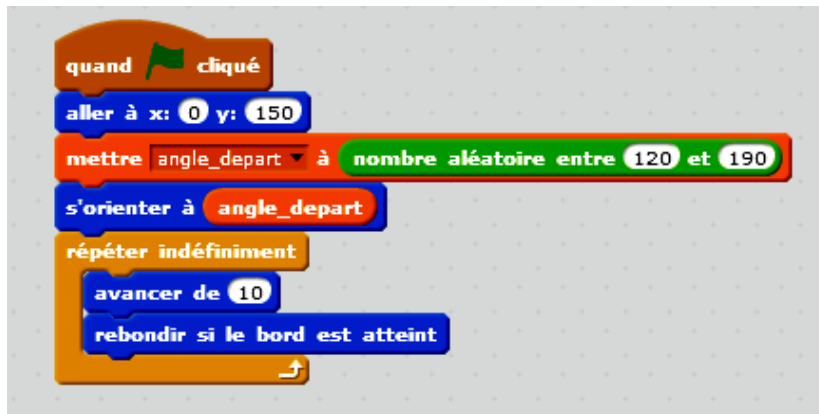
1. Un script pour la raquette permettant son déplacement horizontal à la souris ;
2. Un premier programme pour la balle permettant son départ dans une direction aléatoire et le rebond sur les bords ;
3. Un deuxième programme pour la balle permettant d'afficher « perdu » et de stopper le jeu si la balle touche le bord inférieur ;
4. Un troisième programme pour la balle gérant son comportement lorsqu'elle touche la raquette.

Le point 5. Consistera à créer et gérer la variable « score ».

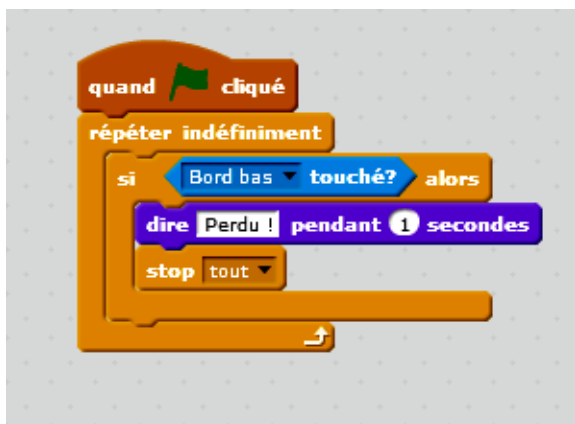
1. Un script pour la raquette.



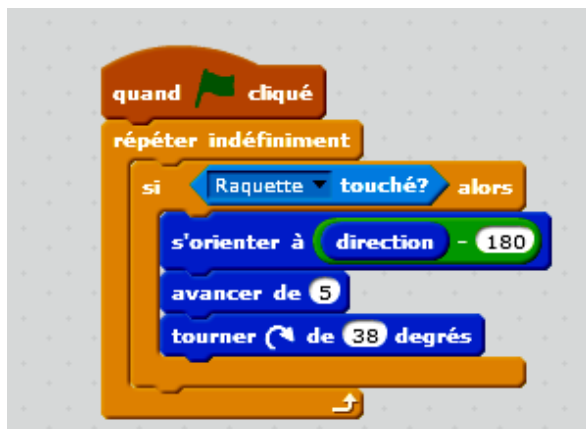
2. Départ et rebond de la balle.



3. Fin de la partie



4. Rencontre balle – raquette : une solution.



5. Compteur de score.

