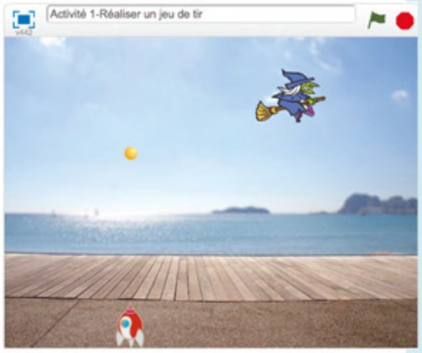
Étudier la logique algorithmique d'un programme

Enseignant :
Mahjoubi BILEL

J'apprends à ▶ **Programmer des actions déclenchées par des événements**

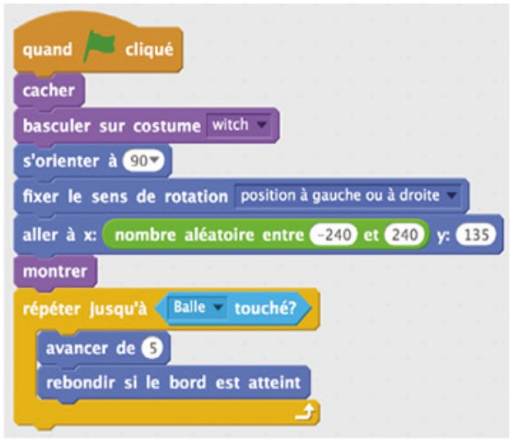
1 Réaliser un jeu de tir

Gabriel a réalisé un jeu de tir avec le logiciel Scratch.
Il a sélectionné l'arrière-plan « boardwalk » pour la scène et trois lutins qu'il a nommés « Sorcière », « Balle » et « Canon » :

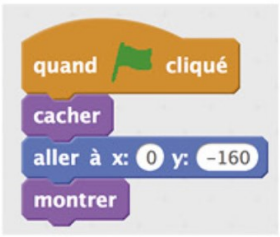


1 a. Ouvrir le logiciel Scratch et mettre en place les objets de l'animation de Gabriel.

b. Analyser, construire et tester le script de « Sorcière ».



2 a. Construire et tester les scripts de « Canon » :

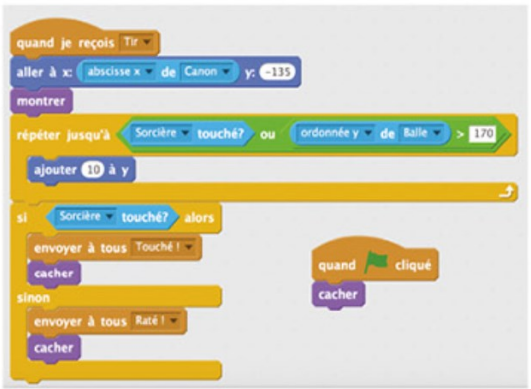


b. Gabriel a prévu d'ajouter à ce lutin le script ci-contre. Expliquer sa fonction. Construire ce script.

3 Le lutin « Balle » entre en action lorsqu'il reçoit le message « Tir ».
Voici ci-contre les deux scripts de ce lutin.

a. Quelle est sa position de départ ? Quelles sont les deux conditions qui arrêtent le mouvement du lutin ? Lorsque chacune d'elles est réalisée, quel est le message envoyé aux autres lutins ?

b. Construire ces scripts.



4 Le comportement du lutin « Sorcière » est complété par les scripts ci-contre.

a. Gabriel a prévu un deuxième costume « Wizard1 » à son lutin. Faire de même.

b. Construire ces scripts et tester l'ensemble du projet.

Étudier la logique algorithmique d'un programme

Enseignant :
Mahjoubi BILEL

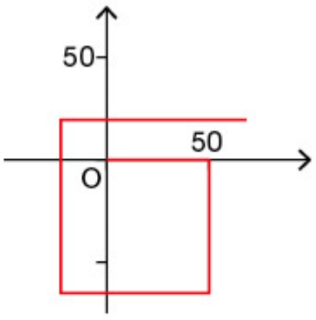
J'apprends à Décrire une démarche à l'aide d'un algorithme

2 Programmer une construction géométrique

1 Pauline souhaite dessiner une spirale à l'aide d'un programme. Elle décrit sa méthode de construction ci-contre.

a. Donner les valeurs successives prises par la variable L lors de la construction.

b. Voici le début de la construction de la spirale. Reproduire ce dessin et terminer la construction décrite par Pauline (prendre 2 cm pour représenter la longueur 50).



Algorithme :

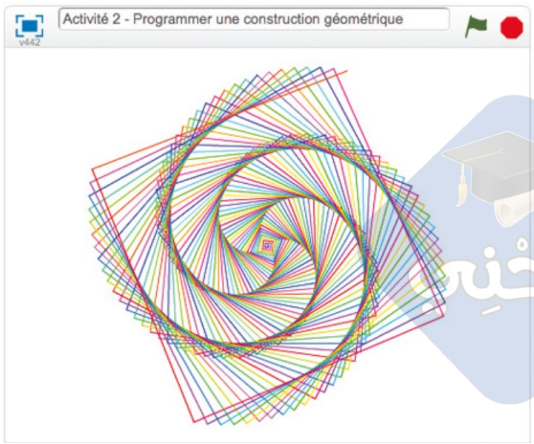
Se placer au début à l'origine du repère
S'orienter vers la droite
Initialiser une variable L à la valeur 50
Répéter 10 fois les instructions :
  Avancer d'une longueur égale à L
  Tourner de 90° vers la droite
  Augmenter la valeur de L de 10
Fin de la boucle

2 Pauline traduit sa méthode de construction dans le langage Scratch. Elle ajoute les instructions de gestion du stylo.
Voici son programme ci-contre.

a. Ouvrir le logiciel Scratch, construire et tester le programme.

b. Modifier le programme afin de construire un plus grand nombre de segments de la spirale.

3 Apporter les changements nécessaires au programme précédent afin d'obtenir un dessin comme celui ci-dessous.



Étudier la logique algorithmique d'un programme

Enseignant :
Mahjoubi BILEL

3 Calculer la somme des nombres d'une liste

1 Voici un script écrit avec le langage Scratch.
Dans ce programme, la variable T est une liste de nombres que l'on peut schématiser de la façon suivante :

	T
1	1er élément de la liste T
2	2e élément de la liste T
3	3e élément de la liste T
	…

a. Expliquer le rôle de ce programme.

b. Ouvrir le logiciel Scratch, créer la liste T, puis construire et tester ce script.

2 On s'intéresse maintenant à l'algorithme suivant :

Initialiser une variable Somme à 0 et une variable k à 1
Répéter 10 fois les instructions :
  Ajouter à Somme le k^e élément de la liste T
  Augmenter la valeur de k de 1
Fin de la boucle
Afficher le contenu de la variable Somme

a. On exécute pas à pas cet algorithme avec la liste T ci-contre. On suit l'évolution des variables dans le tableau ci-dessous.

b. Exécuter et tester le programme obtenu.

Je retiens

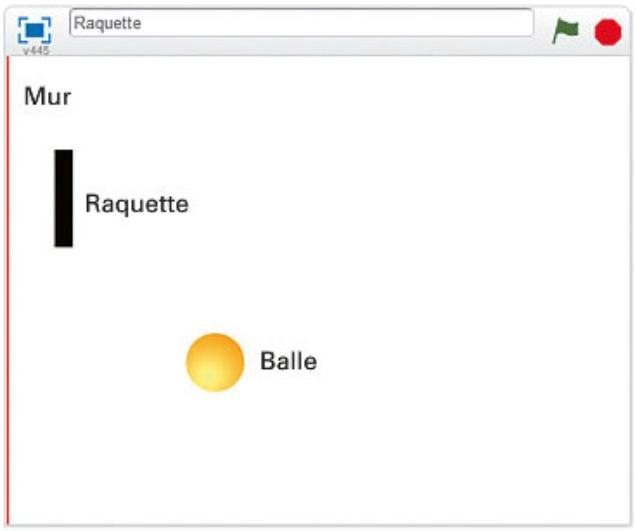
- Dans un langage informatique, il existe différents types de variables : nombres, chaînes (suite de caractères), listes…
- On travaille avec ces variables grâce à des fonctions appropriées du langage.

Étudier la logique algorithmique d'un programme

Enseignant :
Mahjoubi BILEL

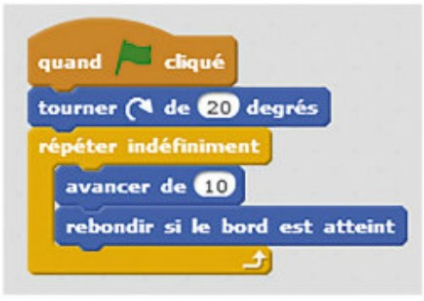
4 Instructions conditionnelles : *Jeu de Raquette*

On se propose de réaliser un jeu de raquette avec le logiciel Scratch.



1. a. Ouvrir le logiciel Scratch, supprimer le lutin « Sprite1 » et choisir le lutin « Ball » dans la bibliothèque ( ).

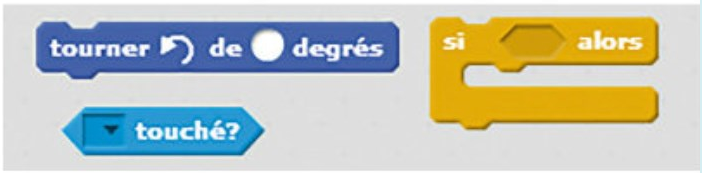
b. Voici le script qui permet à la balle de rebondir sur les côtés du terrain.
Saisir ce script et le tester.



2. a. Dessiner un nouveau lutin « Raquette » en forme de rectangle (cliquer sur).

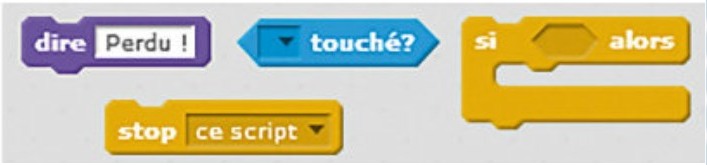
b. Voici le début du script de ce lutin.

3. Lorsque la balle touche la raquette, elle doit rebondir de 45°.
Ajouter dans le bon ordre les briques ci-dessous au script du lutin « Ball ». Les compléter.



4. a. Dessiner un lutin « Mur » (utiliser) et le positionner le long du bord gauche.

b. Si la balle touche le mur, alors la partie est perdue. Utiliser les blocs ci-dessous pour programmer cette éventualité. Les compléter.

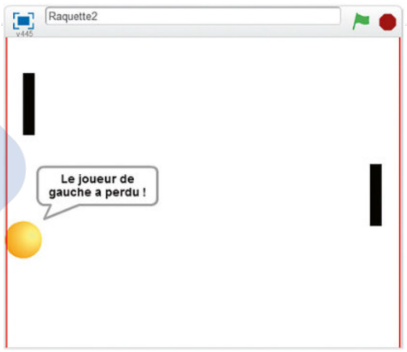


5. a. Utiliser l'instruction pour positionner le lutin « Raquette » au même endroit au début de chaque partie.

b. Utiliser l'instruction pour placer, au début de chaque partie, le lutin « Ball » au point d'abscisse 200 et d'ordonnée un nombre aléatoire compris entre –80 et 80.

c. Tester le jeu.

6. Modifier le jeu afin d'ajouter un second joueur (il peut être utile de ralentir la balle).

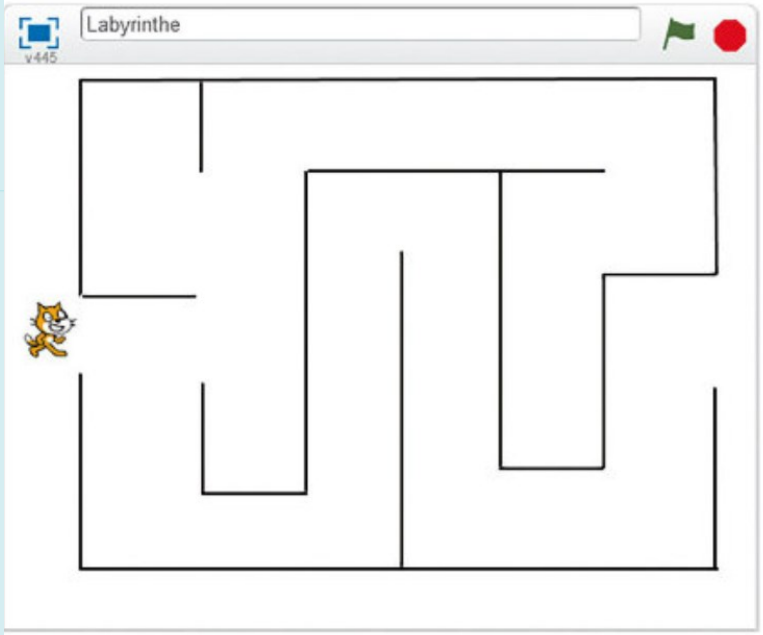


Étudier la logique algorithmique d'un programme

Enseignant :
Mahjoubi BILEL

5 Programmer un jeu de labyrinthe

1. a. Ouvrir le logiciel Scratch et dessiner un nouvel arrière-plan de la scène, de façon à obtenir un labyrinthe (utiliser ⟍).
Appeler cet arrière-plan « Labyrinthe ».

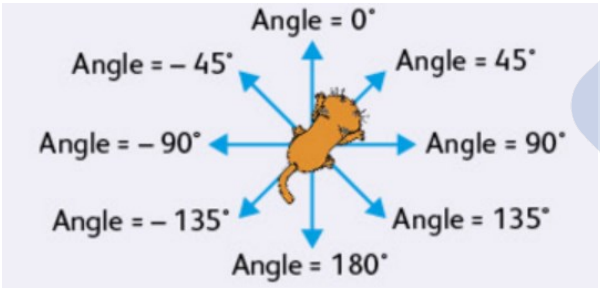


b. Diminuer la taille du lutin ( ) pour rendre possibles ses déplacements dans le labyrinthe.

2. a. Placer le lutin à l'entrée du labyrinthe.
Repérer les coordonnées de son centre.

b. Voici le début du script du lutin.

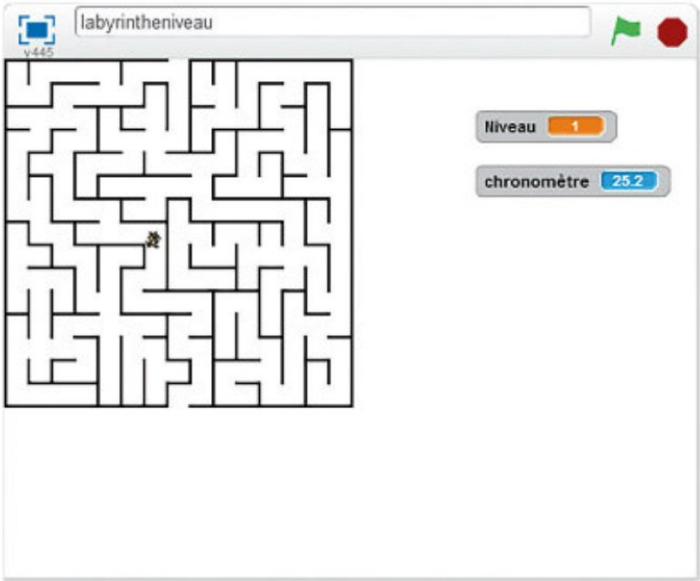
L'événement « Appui sur la flèche droite » du clavier doit provoquer un déplacement du lutin de 1 vers la droite.



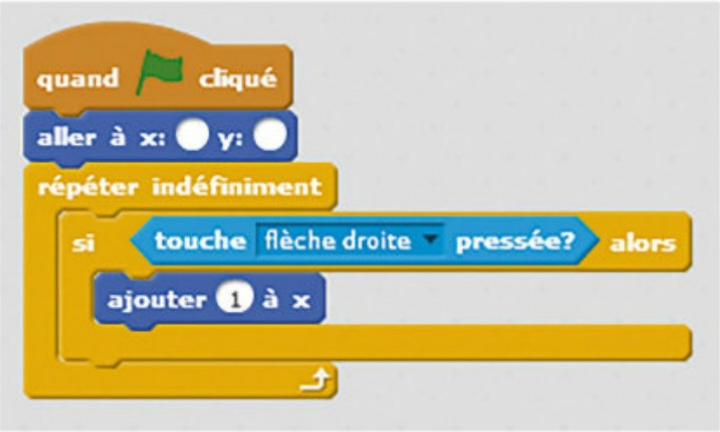
e. Le jeu se termine lorsque le lutin atteint la sortie. Terminer le programme avec ces blocs.

3. Tester le programme obtenu. Pour jouer, mettre la scène en mode plein écran ( ).

4. On peut créer des niveaux de jeu.



a. Se rendre sur le site *www.mazegenerator.net*.
- Générer un labyrinthe carré de taille 15.
- Télécharger ce labyrinthe au format PNG dans



Saisir ce script et le compléter avec les coordonnées de la question précédente.

c. Programmer de même les déplacements du lutin vers la gauche, vers le haut et vers le bas.

d. Lorsque le lutin touche un mur du « Labyrinthe », on le replace à sa position de départ. Utiliser 

pour programmer cet événement.

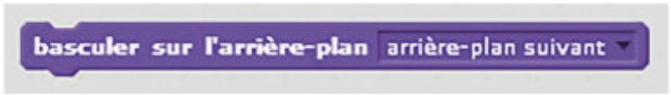
un dossier et le nommer « Niveau1 ».

• Importer ( ) ce labyrinthe comme arrière-plan de la scène.

b. Adapter le jeu à cette nouvelle scène en modifiant le script du lutin et sa taille si besoin.

c. Télécharger d'autres labyrinthes, les nommer « Niveau2 », « Niveau3 » puis importer ces labyrinthes comme arrière-plans.

d. Utiliser l'instruction ci-dessous pour qu'un joueur puisse accéder aux différents niveaux.



e. Tester le jeu.

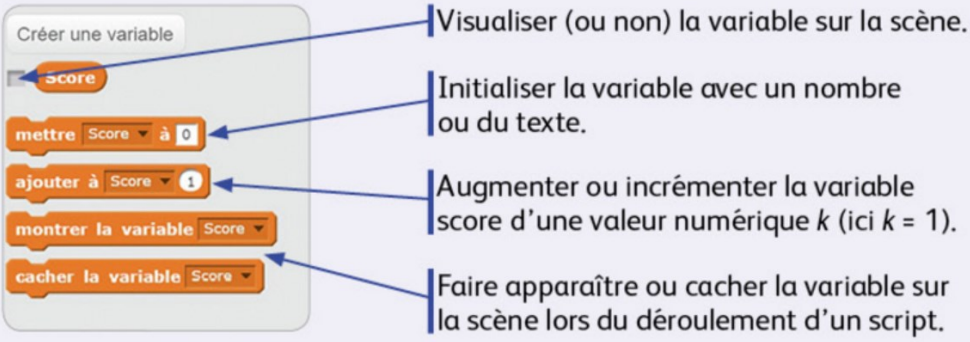
f. On peut améliorer le jeu en créant une variable « Niveau » qui affiche le niveau dans lequel le joueur se trouve.
On peut aussi afficher le chronomètre qui s'initialise en début de partie.

Je découvre

Les variables sont très utiles en programmation. On peut considérer une **variable** comme une « boîte » contenant une valeur. Chaque boîte a une étiquette : c'est le nom de la variable.

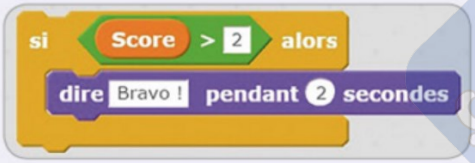
On parle de « variable » car elle peut varier au cours du temps. Au début d'un jeu, une variable « score » peut valoir 0, puis être augmentée par la suite. On dit qu'elle est **incrémentée**.

Pour créer une variable, clique sur Créer une variable dans la catégorie **Données** et choisis Pour tous les lutins.



Pour certains projets, on peut utiliser  pour comparer des variable ou  pour effectuer des opérations. Ces briques sont proposées dans la catégorie **Opérateurs**.

Par exemple, on peut ainsi tester si une variable dépasse une certaine valeur pour féliciter le joueur.



À toi de jouer !