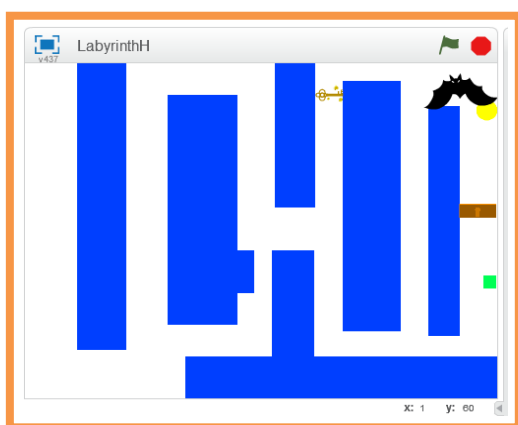


DUREE	NIVEAU	PUBLIC	ACTIVITES
60 min	Confirmé	enfant – adolescent – adulte	Programmation d'un jeu

# “Labyrinthe Compliqué”

## FICHE D'ACTIVITE SCRATCH



### ➤ Objectifs :

Le but de ce jeu est de créer, de A à Z, un labyrinthe que vous aurez dessiné et de pouvoir changer de niveau une fois l'objectif atteint. Attention à l'ennemi qui voudra votre peau, et vous empêcher d'ouvrir la porte à l'aide de la clef.

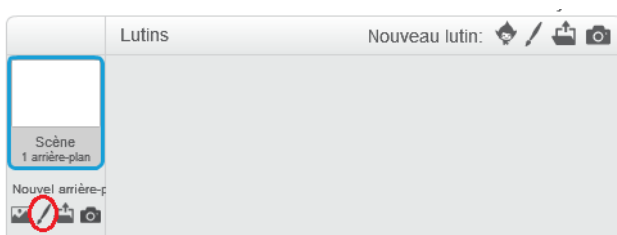
Pour programmer ce jeu, nous allons utiliser la plateforme **Scratch** qui a l'avantage d'utiliser du code visuel : pas besoin d'apprendre un langage de programmation pour construire ses propres jeux vidéo !

### ➤ Compétences travaillées :

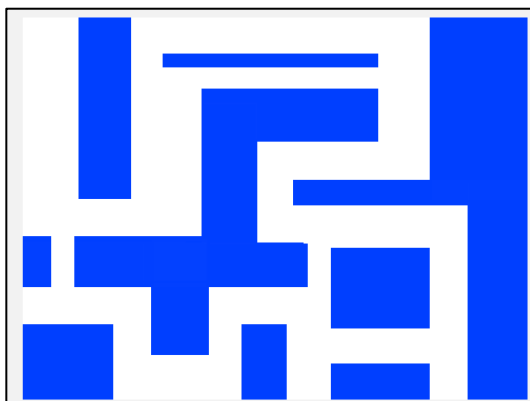
- Boucles
- Conditions
- Capteurs
- Messages

## ➤ Étape 1 : Dessiner les labyrinthes

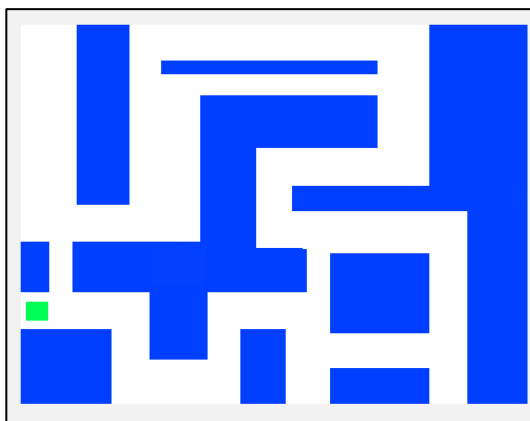
1. Nous allons **dessiner les différents niveaux** de notre labyrinthe. Dans la partie « scène », allez dans l'onglet « Arrière-plans » et cliquez sur « Dessiner un nouvel arrière-plan ».



**Cette partie est libre**, vous pouvez choisir la couleur de votre choix et dessiner votre parcours mais vous devez garder la même couleur.  
Par exemple, voici mon premier niveau :

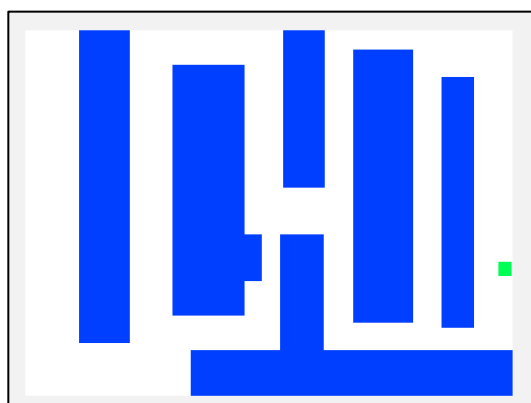


2. Une fois votre niveau terminé, **rajoutez un carré d'une autre couleur pour localiser où se trouve la fin du labyrinthe**.



Votre premier niveau est terminé !

- Maintenant, **créez-en un deuxième** en cliquant sur Dessin. N'oubliez pas de le renommer en "niveau 2".  
Voici mon **deuxième niveau** :

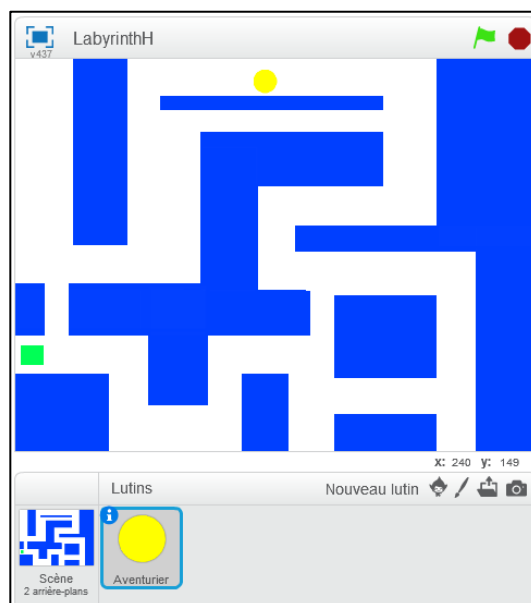


Attention, il faut que la zone de départ dans ce niveau soit la même que dans le premier.

Pour cet exemple, je m'arrête à 2 niveaux. Mais vous pouvez en rajouter autant que vous voulez. Il suffit de les dessiner. N'oubliez pas de les renommer à chaque fois !

## ➔ Étape 2 : Parcourir le labyrinthe avec un lutin

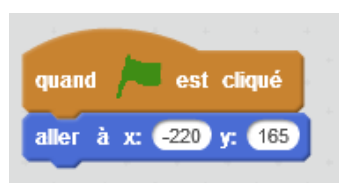
- Maintenant, nous allons **créer un petit lutin** qui aura la forme que vous voulez et **qui va parcourir le labyrinthe jusqu'à sa fin**.  
Pour cela, cliquez sur la partie « nouveau lutin » et importez ou dessinez le lutin de votre choix. Assurez-vous qu'il soit assez petit pour naviguer dans le labyrinthe!



J'ai choisi une petite boule jaune pour mon exemple.

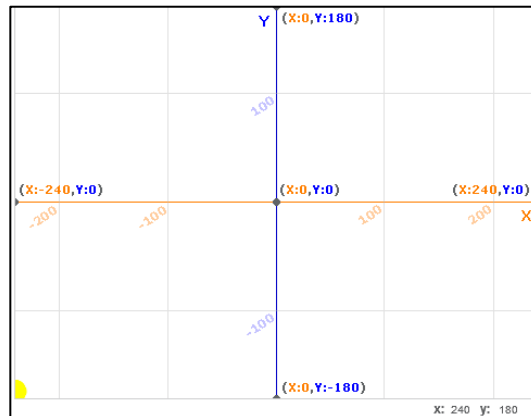
Une fois le lutin sélectionné, changez son nom en "Aventurier" par exemple (en cliquant sur le petit « i » en haut à gauche de l'icône).

2. Mince ! Notre lutin se place sur un mur, mais nous voudrions qu'il se place en haut à gauche du niveau (ou ailleurs pour vous !)  
Nous allons donc **dire au lutin de se placer à un endroit précis**.



Quand le drapeau vert est cliqué, le lutin se place en haut à gauche de la scène (X = -220 et Y = 165).

Le x et le y que vous voyez sont les coordonnées. C'est elles qui disent où vont aller les éléments. Il faut voir la fenêtre comme un quadrillage de 480 pixels de long par 360 de haut.



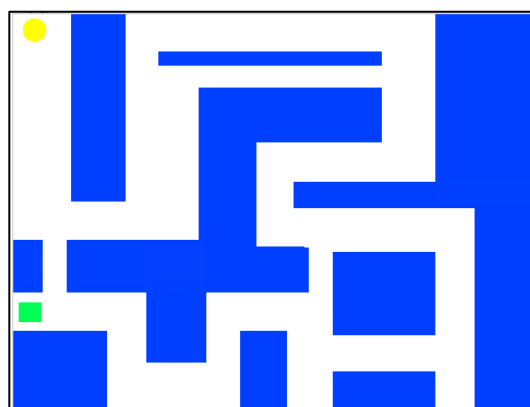
Pour le X, plus on va vers la gauche, plus les chiffres sont petits et vont dans les négatifs. Si l'on va en dessous de -240, on sort de l'écran. Si on va à droite, les chiffres deviennent grands. Si l'on dépasse 240 l'objet, sort de l'écran.

Pour le Y, plus on va en bas, plus les chiffres sont petits et vont dans le négatif. Si l'on dépasse -180, l'objet sort de l'écran. Si l'on va vers le haut, les chiffres deviennent plus grands. Si l'on dépasse 180, on sort de l'écran. Le milieu est donc de coordonnées (0,0).

Dans notre animation pour voir le x et le y que l'on veut avoir, il suffit de passer sa souris sur l'écran et de regarder les coordonnées changer.

Pour moi, le X est de -200 et mon Y à 120 correspondent au coin supérieur gauche, et c'est là que je veux mettre mon lutin.

Voici ce que ça donne :



Essayez de changer les valeurs de votre X et Y pour placer votre lutin où vous le voulez !

Et voilà nous sommes maintenant prêts à commencer à faire le premier script.

## ➤ Étape 3 : On monte, on descend, on va à gauche, puis à droite

1. Désormais, il est temps **d'animer notre lutin**. Rendez-vous dans l'onglet script une fois que vous avez sélectionné le lutin.

Le principe est simple : **dès que l'on appuie sur l'une des touches fléchées, le lutin part dans cette direction.**

Voici comment faire :



Quand le drapeau vert est cliqué, le lutin se place en haut à gauche de la scène (X = -220 et Y = 165). Le programme vérifie ensuite en permanence si une des touches directionnelles est pressées. Si la flèche du haut est pressée, le lutin reste sur son axe horizontal (X) et avance de 5 sur l'axe vertical (Y). Si la flèche du bas est pressée, le lutin reste sur son axe horizontal (X) et recule de 5 sur l'axe vertical (Y). Si la flèche de droite est pressée, le lutin avance de 5 sur son axe horizontal (X) et reste sur son axe vertical (Y). Si la flèche de gauche est pressée, le lutin recule de 5 sur son axe horizontal (X) et reste sur son axe vertical (Y). Dans tous les cas, si le lutin rencontre le bord, il est repoussé.

Quand on lance l'application, on place le lutin au début du labyrinthe, puis on attend que l'une des touches soit pressée et on change la direction.

Vous vous souvenez de X et Y ? Et bien ici aussi on les utilise ! Quand on veut aller en haut, on ajoute 5 à la valeur de y, si on veut descendre, on enlève 5 au y. Pendant ce temps, on ne touche pas à la position de x sinon on risque de le déplacer à gauche ou à droite, alors qu'on ne le veut pas.

Tada ! Notre lutin bouge maintenant correctement quand on appuie sur les touches fléchées !

## ➔ Étape 4 : Et les murs alors ?

1. Le lutin est capable de se déplacer, mais quand on passe sur un mur, il passe à travers ! Il ne devrait pas.

**Nous allons donc ajouter une condition qui va dire que si le mur est touché, on revient à notre position précédente.**

Comment faire ? Comme ci-dessus (là ce n'est que pour la touche du haut) :



Si le lutin touche la couleur bleu, alors il reste sur son axe horizontal (X) et recule de 5 sur l'axe vertical (Y), ce qui annule, en quelque sorte, le mouvement créé par l'action de presser sur la flèche du haut : le lutin est repoussé.

Ici la condition est simple : si la couleur du mur ou le bord est touché alors on change de direction. La position est l'inverse de la précédente. Effectivement si l'on avançait dans un mur, on reculerait (mouvement dans le sens inverse). Donc si le mouvement précédent était  $Y + 5$ , le mouvement inverse est donc  $Y - 5$ .

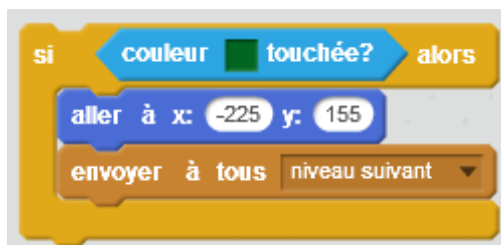
**Répétez la même opération pour les 3 autres touches fléchées.** Combinez ensuite avec les blocs créés précédemment, ainsi :



Le même principe est appliqué à tous les mouvements (haut, bas, droite et gauche). Ces nouvelles conditions viennent se placer sous les conditions précédentes, car elles ne s'appliquent que dans le cas où une touche est pressée, et sont vérifiées indéfiniment par la boucle.



2. **Ajoutez un dernier bloc pour que votre lutin revienne à sa position initiale quand vous atteignez la sortie du labyrinthe.** Le message que vous envoyez (qu'il faut créer en cliquant sur la flèche et « nouveau message ») sera reçu par l'arrière plan pour changer de niveau.



Si le lutin touche la couleur verte, alors il va se replacer en haut à gauche, et le message « niveau suivant » est envoyé à tous les éléments programmables.

Maintenant votre code devrait ressembler à ça :



## ➤ Étape 5 : Changement de niveau

### 1. Il ne nous reste plus qu'à changer de niveau.

Si on a touché la couleur du bloc de fin, le lutin revient à sa position initiale.

Allez dans Scène puis dans le script de la scène et écrivez le script suivant. Quand vous aurez fini un niveau, le niveau suivant commencera automatiquement.



Quand le drapeau vert est cliqué, l'arrière plan affiche « arrière-plan1 ».

Quand l'élément reçoit le message « niveau suivant », l'arrière plan bascule sur l'arrière plan suivant dans la liste des costumes.

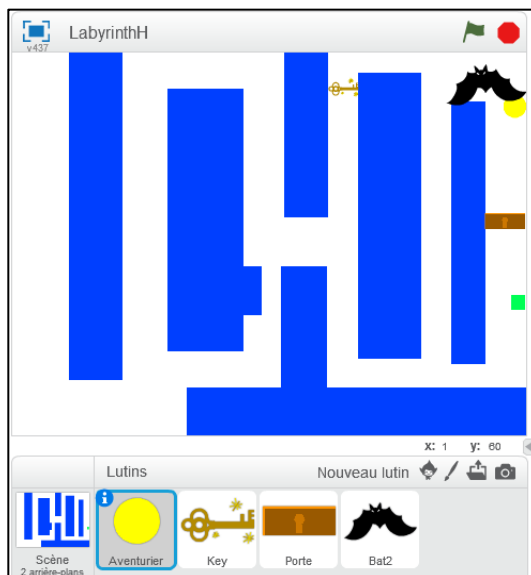
Et voilà on peut lancer l'application et jouer. À vous de créer vos propres labyrinthes, le code marchera automatiquement !

N'oubliez pas de faire des murs de même couleur, un bloc final d'une autre couleur et de faire votre lutin petit pour qu'il puisse passer entre vos murs !

## ➤ Étape 6 : Créer une porte et un ennemi

Nous avons à présent un jeu fonctionnel mais pas forcément passionnant. **Afin d'améliorer un peu** ce qui existe déjà, nous allons rajouter une porte qui s'ouvrira grâce à une clé et donnera accès à la fin du niveau 2 et une ennemie chauve-souris. Le deuxième niveau ressemblera ainsi à cela :

1. **Créez les trois lutins**, une clé, une porte (que vous pouvez dessiner) et une chauve-souris.



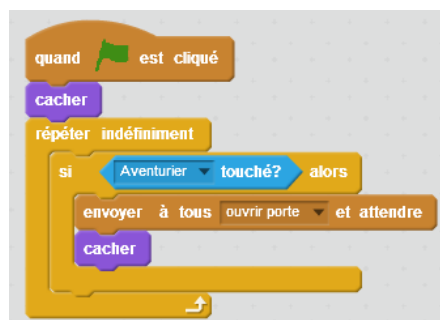
2. **Création de la clé.** La clé permet d'ouvrir la porte, quand l'aventurier passe dessus, elle disparaît et la porte s'ouvre. Nous ne voulons pas que la clé apparaisse dès le premier niveau. Pour cela, nous allons entrer le script suivant :



Quand le drapeau vert est cliqué, le lutin disparaît.

Quand l'arrière-plan bascule sur « arrière-plan2 », le lutin apparaît.

Ensuite, faisons en sorte que lorsque l'aventurier touche la clé celle-ci disparaisse et envoie un message à la porte pour lui dire de s'ouvrir. Il faut créer le message « ouvrir porte » comme avec « niveau suivant tout à l'heure ». Le script est alors le suivant :



Quand le drapeau vert est cliqué, le lutin disparaît. Puis le programme vérifie en permanence si le lutin touche le lutin « Aventurier ». Si cette condition est vérifiée, le message « ouvrir porte » s'envoie à tous les éléments et le lutin disparaît.

3. **Création de la porte.** La porte empêche d'accéder directement à la sortie. Allez dans le script de la porte. Pour commencer, nous allons faire en sorte que la porte n'apparaisse qu'au niveau 2. Pour cela, nous allons utiliser les mêmes scripts que tout à l'heure. Nous allons aussi ajouter un script pour cacher la porte quand le message « ouvrir\_porte » est reçu, cela donne le code suivant :

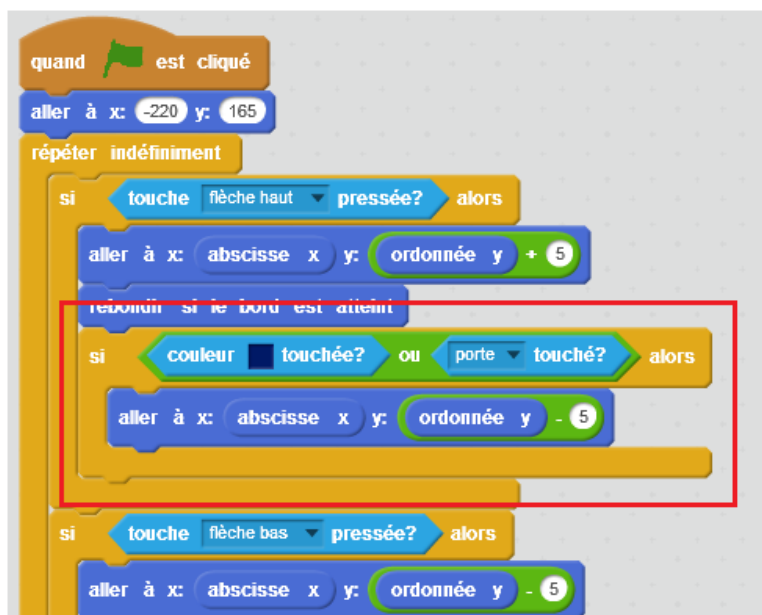


Quand L'arrière plan affiche « arriere-plan2 », la porte apparait.

Quand le message « ouvrir porte » est reçu, la porte disparaît.

Quand le drapeau vert est cliqué, la porte disparaît.

**Collision de l'aventurier contre la porte.** Pour cela il faut gérer la collision via un capteur dans le script de l'aventurier, remplacez partout où vous trouverez « si couleur touchée » par « si couleur touchée ou « si Porte touchée » ainsi :



Si l'Aventurier touche la couleur bleue ou la porte, il est repoussé.



4. **Création d'un ennemi.** Nous allons maintenant créer un ennemi. Pour ne pas compliquer le problème, il ne sera pas gêné par les murs (pensez à un fantôme ou à une chauve-souris). Si l'aventurier est touché, il meurt. Entrez le code suivant :



Quand le drapeau vert est cliqué, l'ennemi disparaît.

Quand l'arrière-plan change pour « arriere-plan2 », l'ennemie apparaît, change son orientation, pour parcourir toute la scène, et va se placer dessus. Ensuite, il attend 0.2 seconde, change de costume (pour donner l'illusion qu'il vole) et bouge dans tous l'espace en repartant de l'autre côté s'il atteint le bord. Si jamais il touche l'Aventurier, le programme s'arrête. Toutes ces actions se répètent indéfiniment.

L'instruction « costume suivant » sert si par exemple vous avez choisi la chauve souris qui bat des ailes quand elle change de costume. Sinon vous pouvez ne pas la mettre.

## ➔ Bonus

Voilà votre jeu est terminé ! Si vous souhaitez encore l'améliorer vous pouvez :

- Ajouter des niveaux
- Créer d'autres monstres
- Créer des potions qui font rétrécir le héros pour qu'il puisse passer par des chemins auxquels il n'avait pas accès précédemment.