TUTORIEL - SCRATCH : LABYRINTHE AVANCÉ

CRÉATION DE CONTENU > 3.4 PROGRAMMATION

CONVIENT POUR	AGE	NIVEAU DE COMPÉTENCE	FORMAT	DROITS D'AUTEUR	LANGUE(S)
Elèves (école primaire), Elèves (école secondaire), Jeunes en décrochage scolaire	Adolescents, Enfants	Niveau 2	Fiche d'activité	Creative Commons (BY-SA)	Français

Ce tutoriel explique comment réaliser un jeu vidéo de labyrinthe avec le logiciel Scratch.

Objectif général	Compétences
Temps de préparation pour l'animateur	moins d'une 1 heure
Domaine de compétence	3 – Création de contenu
Temps requis pour compléter l'activité (pour l'apprenant)	0 – 1 heure
Nom de l'auteur	Caroline Fohrer
Matériel supplémentaire	Ordinateur avec connexion internet
Ressource originellement créée	Français

DÉROULÉ

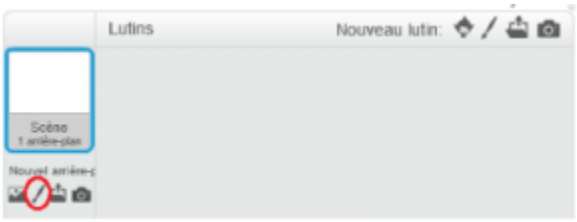
1 Introduction

Le but de ce tutoriel est de créer un labyrinthe que vous aurez dessiné. Pour réussir un niveau, il faudra récupérer une clé, qui servira à ouvrir la porte de sortie et passer au niveau suivant. Le tout en évitant un adversaire, qui tentera d'empêcher votre avatar d'ouvrir la porte.

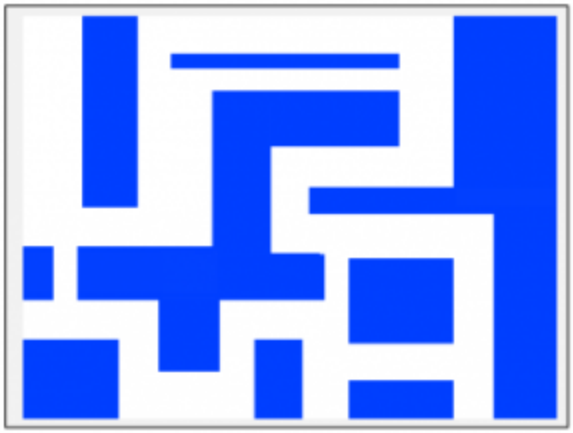
2 Dessiner les labyrinthes

Nous allons dessiner les différents niveaux de notre labyrinthe.

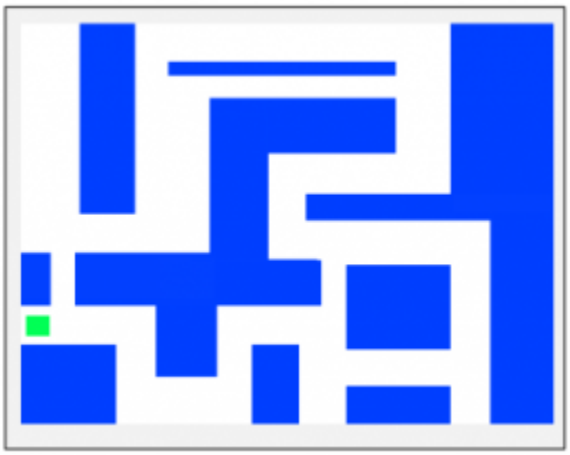
Dans la partie «scène», aller dans l'onglet «Arrière-plans» et cliquer sur «Dessiner un nouvel arrière-plan».



Choisir une seule couleur et dessiner un parcours en traçant des blocs (qui seront les murs/obstacles). Par exemple :

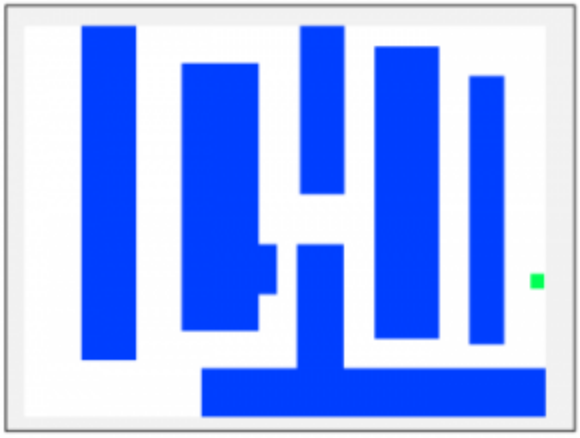


Ajouter un carré d’une autre couleur pour représenter la porte de sortie du niveau :



Le premier niveau est terminé !

Cliquer sur « dessins » et refaire ces étapes pour créer un deuxième niveau, qui devra être renommé en « niveau 2 ». Exemple :



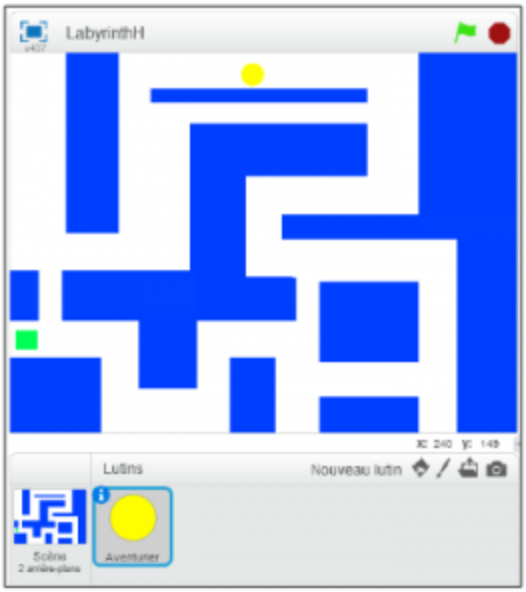
Attention, il faut que la zone de départ soit la même dans tous les niveaux.

Nous nous arrêtons à deux niveaux dans tutoriel, mais il est bien sûr possible d'en créer autant qu'on veut.

3 Parcourir le labyrinthe avec un lutin

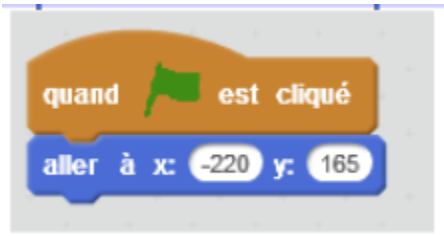
Maintenant, nous allons créer un petit lutin qui pourra se promener dans le labyrinthe.

Pour cela, cliquer sur la partie «nouveau lutin» et importer ou dessiner un lutin au choix. Attention, il doit être assez petit pour savoir passer partout !

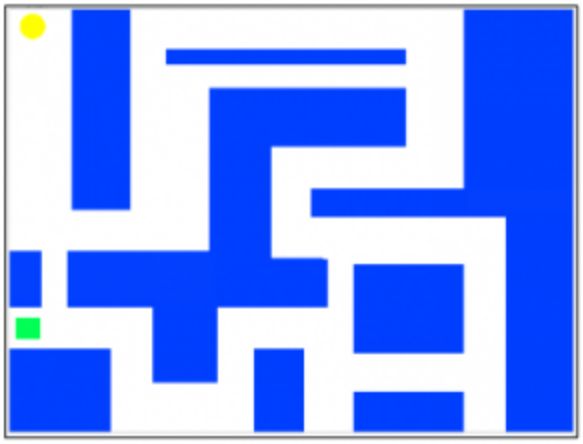


Dans l'exemple, le lutin est une boule jaune. Changer son nom en "Aventurier", en cliquant sur le petit «i» en haut à gauche de l'icône.

Il faut à présent s'assurer que le lutin apparaisse sur le point de départ et non sur un mur. Ajouter les blocs « quand drapeau est cliqué » et « aller à » au lutin, et inscrire les bonnes valeurs pour X et Y, afin que le lutin apparaisse dans le coin supérieur gauche. Pour connaître les valeurs à utiliser, placer le curseur de la souris sur le point de départ désiré. Les coordonnées s'inscrivent sous la zone de dessin. Dans notre exemple, il s'agit de -220 et 165 :



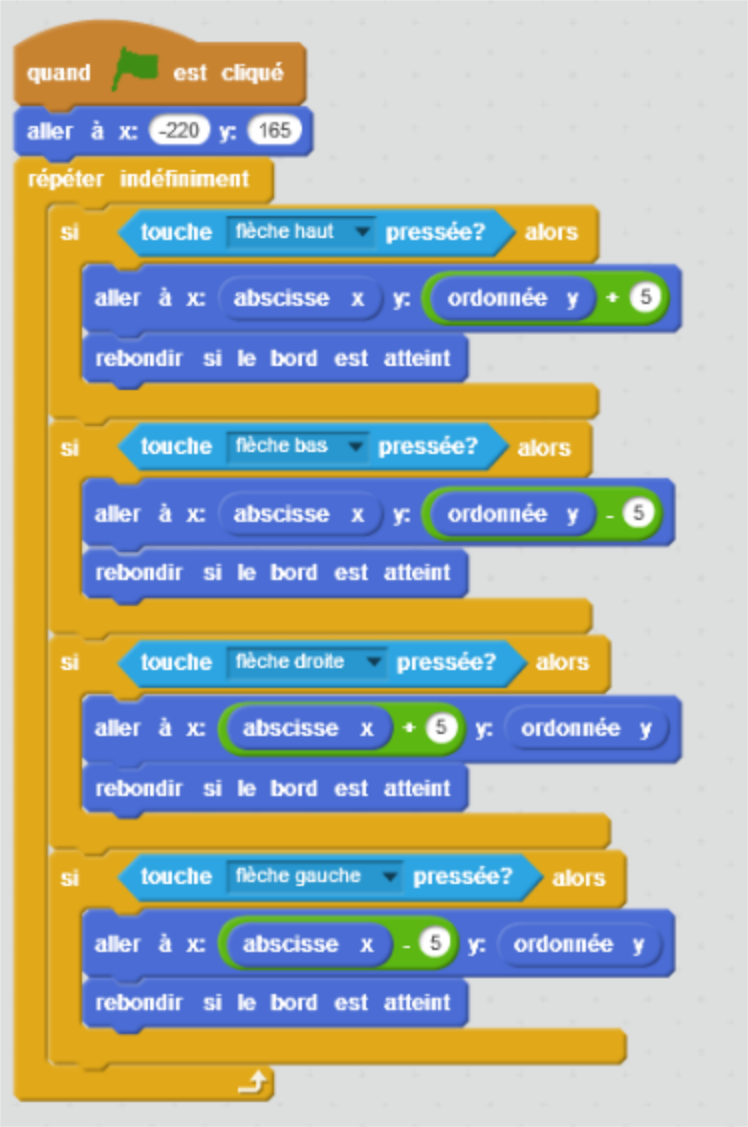
Ce qui donne :



4 Piloter le lutin

Il est temps d'animer notre lutin, pour qu'il puisse être piloté grâce aux touches fléchées.

Ajouter les blocs suivant au script du lutin :



Explication du code : quand le drapeau vert est cliqué, le lutin se place en haut à gauche de la scène. L'ordinateur vérifie ensuite en permanence si une des touches « flèche … » est pressée. 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.

Cliquer sur le drapeau et appuyer sur les touches pour vérifier le résultat obtenu.

5 Se cogner aux murs

Le lutin est capable de se déplacer, mais il traverse les murs ! 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.

Voici les blocs à ajouter pour la flèche du haut :

Explication du code : si le lutin touche la couleur choisie pour dessiner le labyrinthe (ici le bleu), alors il reste sur son axe horizontal (X) et recule de 5 sur l'axe vertical (Y). Ceci va donc faire l'effet inverse du déplacement qu'on vient de réaliser suite à l'appui sur la flèche haut. On crée ainsi l'illusion que le lutin n'a pas bougé, comme s'il était bloqué par le mur.

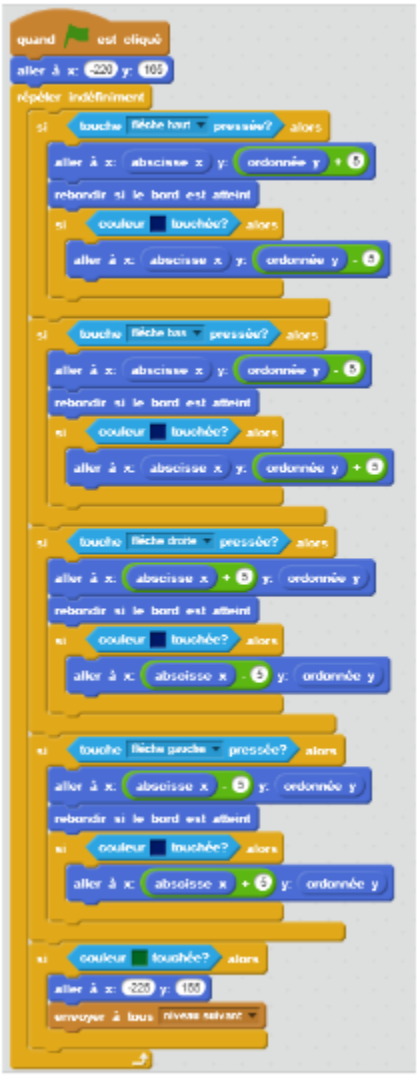
Ajouter les mêmes blocs pour les autres flèches, mais en ajustant les X et Y pour chaque fois inverser le déplacement prévu. Le résultat devrait être :

6 Changer de niveau

Il ne nous reste plus qu'à changer de niveau. Il faut d'abord détecter que le niveau actuel est terminé.

Ajouter les blocs ci-dessous à la fin du programme pour détecter quand le lutin a touché la couleur de la porte de sortie (ici le vert), replacer le lutin dans sa position initiale, et envoyer un message « niveau suivant ». Pour créer le message, cliquer sur la flèche puis sur « nouveau message ». Ce message sera envoyé à tous les éléments du jeu, nous l'utiliserons plus tard.

Le code devrait ressembler à ceci :



Aller dans Scène puis dans le script de la scène, et écrire le script suivant pour gérer le changement de niveau :

Explication du code : quand le drapeau vert est cliqué, l'arrière-plan affiche le premier niveau. Quand la scène reçoit le message «niveau suivant», l'arrière-plan change pour afficher le niveau suivant (pris dans la liste des costumes, donc il faut que les niveaux soient dans l'ordre de la liste).

7 Ajouter une clé, une porte et un adversaire

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

Ajouter trois lutins et leur donner le nom et l'apparence respectivement d'une clé, une porte et une chauve-souris. Il est possible de les dessiner, ou de choisir parmi les costumes existants de Scratch.

Nous voulons que la clé apparaisse seulement au second niveau. Ajouter les blocs suivants au script du lutin de la clé :

Explication du code : quand le jeu commence (le drapeau est cliqué), la clé se cache. Quand on passe au second niveau (l'arrière-plan change), la clé 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». Le script est alors le suivant:

Explication du code : quand le drapeau vert est cliqué, le lutin disparaît. Puis l'ordinateur vérifie en permanence si le lutin « clé » touche le lutin «Aventurier». Si cette condition est vérifiée, le message «ouvrir porte» est envoyé à tous les éléments, et le lutin « clé » disparaît. On donne ainsi l'illusion que la clé a été ramassée par l'aventurier.

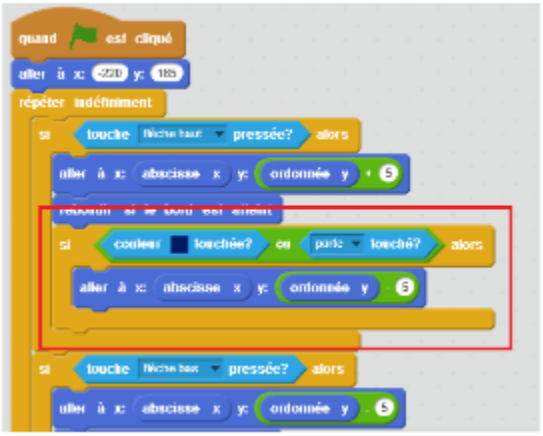
8 Animer la porte

La porte ne doit apparaître qu'au deuxième niveau, et doit s'ouvrir quand l'aventurier ramasse la clé.

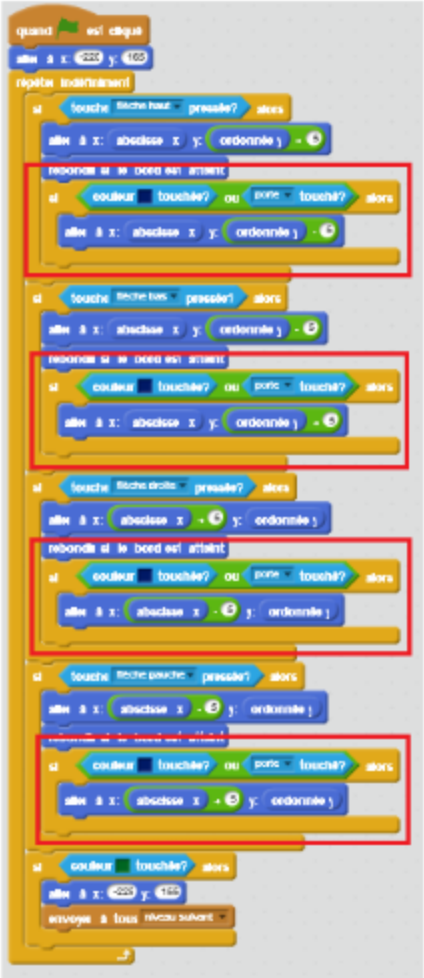
Aller dans le script de la porte. Ajouter les mêmes blocs que pour la clé, pour cacher le lutin quand le jeu démarre et le montrer lorsqu'on passe au deuxième niveau. Ensuite, ajouter un script pour cacher la porte quand le message «ouvrir_porte» est reçu :

Le reste du temps, la porte doit empêcher l'aventurier de passer. Pour cela il faut gérer la collision via un capteur dans le script de l'aventurier.

Remplacer toutes les conditions «si couleur mur touchée» par «si couleur mur touchée ou si Porte touchée», de cette façon :



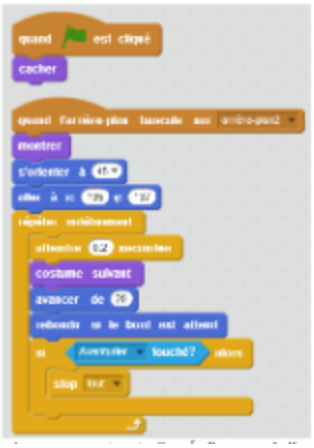
Ainsi, chaque fois que l'aventurier touche la porte, on va annuler son mouvement, comme on le faisait pour les murs, sans devoir refaire tout le code. A présent, le script de l'aventurier doit ressembler à :



9 Animer l'adversaire

La chauve-souris doit voler au-dessus du labyrinthe et essayer de mordre le personnage. Elle ne doit agir que dans le deuxième niveau.

Ajouter les blocs suivants au script de la chauve-souris :



Explication du code : quand le drapeau vert est cliqué, la chauve-souris disparait. Quand l'arrière-plan change pour «arrière-plan2», elle apparait, change son orientation, pour parcourir toute la scène, et va se placer dessus. Ensuite, elle attend 0,2 secondes, change de costume (pour donner l'illusion qu'elle vole), et bouge dans tout l'espace en repartant de l'autre côté si elle atteint le bord. Si jamais elle touche l'Aventurier, le jeu 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.

10 Bonus

Quelques pistes pour améliorer encore le jeu :

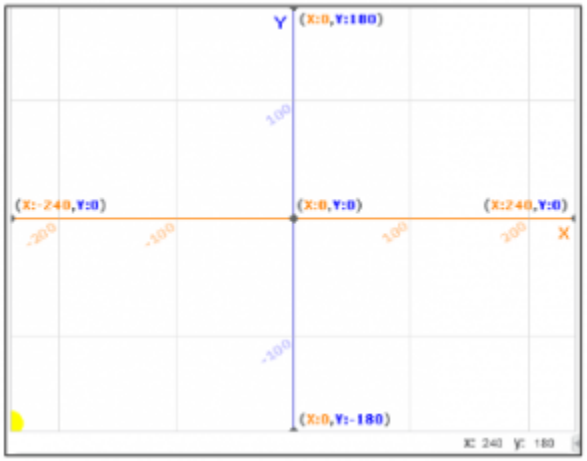
- Ajouter des niveaux
- Créer d'autres adversaires
- Créer des potions qui font rétrécir le héros pour qu'il puisse passer par des chemins auxquels il n'avait pas accès précédemment
- Créer des portes de couleurs différentes qui réagissent chacune uniquement à la clé de même couleur
- Ajouter des bruitages lors des différents évènements (ouverture de porte, défaite, changement de niveau, ...)

Variantes pour faire l'activité autrement :

- Dessiner les niveaux sur papier, puis les scanner et les utiliser comme arrière-plan.

11 Annexe : les coordonnées dans Scratch

Le x et le y que vous voyez sous la zone d'animation sont les coordonnées. Cette zone est en fait un quadrillage de 480 pixels de long par 360 de haut.



Le centre a les coordonnées (0,0).

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 la zone d'animation. 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.

12 Pour aller plus loin

Conseil médiation

Pour aller plus plus loin sur le sujet, nous vous conseillons de vous référer à la fiche outil « Guide de présentation de Scratch«