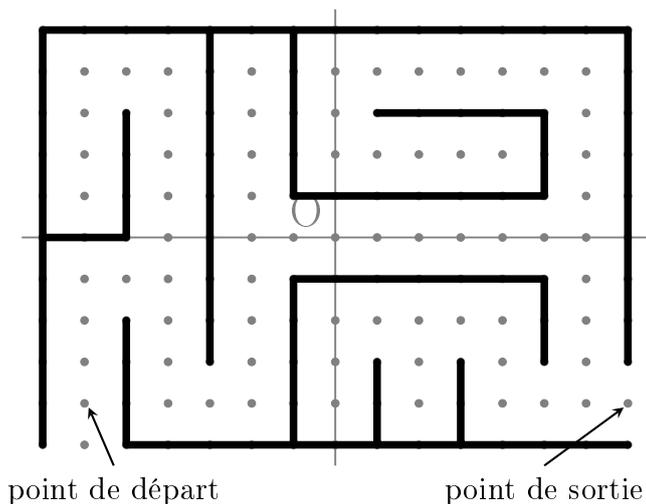


(Source : Amérique de Nord, 4 juin 2019, exercice 4, 10 points)

On a programmé un jeu. Le but du jeu est de sortir du labyrinthe. Au début du jeu, le lutin se place au point de départ. Lorsque le lutin touche un mur, représenté par un trait noir épais, il revient au point de départ. L'arrière-plan est constitué d'un repère d'origine O avec des points espacés de 30 unités verticalement et horizontalement.

Dans cet exercice, on considèrera que seuls les murs du labyrinthe sont noirs.

Le point de départ est vert et le point d'arrivée est rouge



Un premier programme

Réaliser un programme qui permet de gérer les touches « flèches » pour se déplacer dans le labyrinthe et qui renvoie le chat au départ lorsqu'il touche un mur.

On donne, au départ, le plateau de jeu constitué des murs, du repère et des points gris.

Etape 1 : gestion des déplacements avec les flèches

On utilise la brique **Quand la touche** flèche haut ▼ **est pressée**

Utiliser ensuite la brique **ajouter** ● **à y** pour se déplacer de 30 points vers le haut.

Pour que les mouvements ne soient pas trop rapides, ajouter enfin une brique **attendre** 0.1 **secondes**

Créer des scripts identiques pour les 3 autres flèches de déplacement

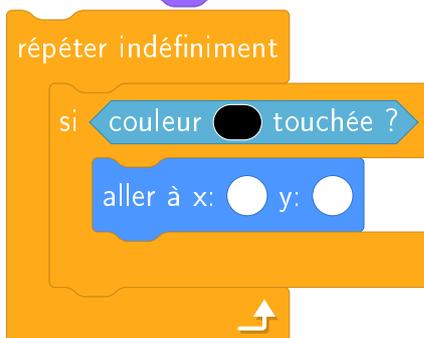
Etape 2 : gestion des murs

Utiliser la brique **quand** 🚩 **est cliqué**

Ajouter la brique **mettre la taille à** 45% **de la taille normale.**

Ajouter la brique **aller à x:** ● **y:** ● pour aller au point de départ.

Ajouter **répéter indéfiniment** pour retourner au point de départ si on touche un mur.



Amélioration 1

Ajouter un programme pour compter le nombre de pas pour atteindre la sortie

On peut :

- créer une variable **Nombre de pas** qui contient le nombre de touches appuyées
- initialiser cette variable à 0 au départ et modifier les scripts des "flèches" pour ajouter 1 à chaque appui.

Amélioration 2

Dire « bravo » lorsqu'on atteint la sortie

Refaire un nouveau programme avec une boucle répéter indéfiniment en testant si on touche la couleur rouge