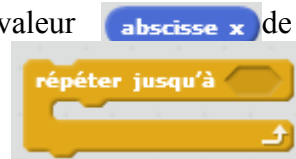


TP Rochers

Dans cette activité, on simule le déplacement aléatoire pas à pas d'un personnage dans un environnement. On commencera pour cela par programmer un pas de déplacement, puis plusieurs, et enfin on ajoutera des compteurs pour acquérir des informations sur le déplacement effectué.

1. Mise en place : ouvrez un nouveau projet et remplacez l'arrière-plan par le fichier `rochers.png` fourni. Positionnez votre lutin sur le 2^e rocher en partant de la gauche, et réduisez sa taille si nécessaire pour qu'elle corresponde à celle d'un rocher. Prévoyez un script pour réinitialiser la situation.
2. Programmez un pas de déplacement. Pour cela, on tire au sort si le lutin ira à gauche ou à droite de la manière suivante : on demande un nombre au hasard (aléatoire) à l'ordinateur (par exemple entre 1 et 2, pour avoir autant de chance d'aller à droite que d'aller à gauche) puis on effectue le déplacement en fonction du nombre obtenu. Les rochers sont distants de 50 pixels, et on considère que les deux directions sont équiprobables (c'est à dire qu'elles ont la même chance d'arriver).
3. **Enregistrez votre travail, puis appelez le professeur pour valider votre activité.**
4. Enchaînez plusieurs pas de déplacements : le lutin continue à se déplacer jusqu'à ce qu'il soit revenu au rocher le plus à gauche. On peut utiliser la valeur `abscisse x` de la catégorie mouvement pour faire ce test, ainsi que l'instruction :



Lorsque le lutin a atteint le rocher le plus à gauche, il a gagné la partie.

5. Que se passe-t-il si le lutin essaye d'aller à droite du dernier rocher ? Corriger le script si nécessaire pour qu'un lutin sur ce dernier rocher aille forcément à gauche.
6. **Enregistrez votre travail, puis appelez le professeur pour valider votre activité.**
7. Ajoutez un compteur pour déterminer automatiquement le nombre de pas effectués par le lutin au cours de son déplacement.
8. Mémorisez également l'abscisse la plus à droite atteinte par le lutin au cours de sa marche.
9. Modifiez les probabilités d'aller à gauche ou à droite et observez l'influence sur le nombre de pas dans une marche.
10. **Enregistrez votre travail, puis appelez le professeur pour valider votre activité.**

Point Info : On utilise une boucle « Répéter jusqu'à... » quand on ne sait pas à l'avance combien de fois elle devra s'exécuter ; souvent, c'est même ce nombre qui nous intéresse en sortie de boucle. La condition d'arrêt peut être complexe.

Questionnaire TP Rochers

1. De quels blocs est composé votre script pour réinitialiser la position du lutin au début de chaque partie ?
2. Lors du déplacement à droite ou à gauche, avez vous utilisé un structure du type « si ...alors...sinon »?
3. A quoi sert l'instruction « Répéter jusqu'à ... » ?
4. Quelle est la différence entre les boucles « Répéter jusqu'à ... » et « Répéter ... fois » ??
5. Que se passe-t-il si le lutin essaye d'aller à droite du dernier rocher ?
6. Comment se traduit l'influence des probabilités d'aller à droite ou à gauche sur le nombre de pas dans une marche ?

Prolongements

1. Répéter plusieurs fois la marche aléatoire et stocker dans une ou plusieurs listes les différents résultats (nombres de pas et/ou abscisse maximale atteinte)
2. Demander à l'utilisateur le nombre de fois qu'il veut que la marche aléatoire se répète.
3. Faire une moyenne des différents résultats obtenus.
4. Arrêter le lutin s'il n'a pas terminé sa marche au bout d'un nombre de pas fixé à l'avance, ou choisi par l'utilisateur.
5. Animer les déplacements du lutin (saut de rocher en rochers par exemple...)
6. Programmer une marche aléatoire dans un environnement 2D, par exemple, un robot qui se déplace dans les 4 directions et qui doit sortir d'une boîte ou d'un labyrinthe.