

Bougeons		1^{re} et 2^e année			
<h2 style="color: #1a3d4d;">Plan de la leçon</h2>		Outil de programmation	ScratchJr		
		Temps nécessaire	Deux périodes		
<p>Connexions du programme de mathématiques</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 50%; vertical-align: top; padding: 5px;"> <p>Algèbre C3. Programmation</p> <p>Attentes générales C3. Résoudre des problèmes et créer des représentations informatiques de situations mathématiques à l'aide de concepts et de compétences de programmation.</p> <p>Attentes particulières C3.1 Résoudre des problèmes et créer des représentations computationnelles de situations mathématiques en écrivant et exécutant une programmation, y compris une programmation qui comporte des événements séquentiels.</p> </td> <td style="width: 50%; vertical-align: top; padding: 5px;"> <p>Orientation spatiale E1. Raisonnement géométrique et spatial</p> <p>Attentes générales E1. Décrire et représenter la forme, l'emplacement et le mouvement en appliquant les propriétés géométriques et les relations spatiales afin de naviguer dans le monde qui nous entoure.</p> <p>Attentes particulières E1.1 Trier et identifier les formes bidimensionnelles en comparant le nombre de côtés, la longueur des côtés, les angles et le nombre de lignes de symétrie. E1.5 Décrire la position relative de plusieurs objets et les mouvements nécessaires pour passer d'un objet à un autre.</p> </td> </tr> </table>				<p>Algèbre C3. Programmation</p> <p>Attentes générales C3. Résoudre des problèmes et créer des représentations informatiques de situations mathématiques à l'aide de concepts et de compétences de programmation.</p> <p>Attentes particulières C3.1 Résoudre des problèmes et créer des représentations computationnelles de situations mathématiques en écrivant et exécutant une programmation, y compris une programmation qui comporte des événements séquentiels.</p>	<p>Orientation spatiale E1. Raisonnement géométrique et spatial</p> <p>Attentes générales E1. Décrire et représenter la forme, l'emplacement et le mouvement en appliquant les propriétés géométriques et les relations spatiales afin de naviguer dans le monde qui nous entoure.</p> <p>Attentes particulières E1.1 Trier et identifier les formes bidimensionnelles en comparant le nombre de côtés, la longueur des côtés, les angles et le nombre de lignes de symétrie. E1.5 Décrire la position relative de plusieurs objets et les mouvements nécessaires pour passer d'un objet à un autre.</p>
<p>Algèbre C3. Programmation</p> <p>Attentes générales C3. Résoudre des problèmes et créer des représentations informatiques de situations mathématiques à l'aide de concepts et de compétences de programmation.</p> <p>Attentes particulières C3.1 Résoudre des problèmes et créer des représentations computationnelles de situations mathématiques en écrivant et exécutant une programmation, y compris une programmation qui comporte des événements séquentiels.</p>	<p>Orientation spatiale E1. Raisonnement géométrique et spatial</p> <p>Attentes générales E1. Décrire et représenter la forme, l'emplacement et le mouvement en appliquant les propriétés géométriques et les relations spatiales afin de naviguer dans le monde qui nous entoure.</p> <p>Attentes particulières E1.1 Trier et identifier les formes bidimensionnelles en comparant le nombre de côtés, la longueur des côtés, les angles et le nombre de lignes de symétrie. E1.5 Décrire la position relative de plusieurs objets et les mouvements nécessaires pour passer d'un objet à un autre.</p>				
<p>Description</p> <p>Dans cette leçon, les élèves auront la chance d'explorer le mouvement dans des espaces 2D, à la fois sur papier et à l'écran. Ils apprendront la façon de communiquer les changements de position à l'aide de blocs de code. Cette connaissance sera appliquée dans ScratchJr pour résoudre des problèmes et s'amuser.</p>					
<p>Critères de réussite</p> <ul style="list-style-type: none"> • Les élèves de 1^{re} et 2^e année seront en mesure de créer des formes et de naviguer sur une grille à l'aide d'activités pratiques et informatiques. 		<p>Matériel et médias</p> <ul style="list-style-type: none"> • Papier • Crayons • Document sur les formes et les labyrinthes • Clé de réponse pour le document sur les formes et labyrinthes • Document sur la programmation • Appareils avec ScratchJr 			

Aptitudes en réflexion computationnelle

Cette leçon présente le concept d’algorithme. En général, dans la programmation, un algorithme est un ensemble d’étapes conçues pour accomplir une tâche. En communiquant une tâche au moyen d’une série d’étapes, les élèves mettront en œuvre leur réflexion algorithmique. Les élèves utiliseront le code de bloc comme outil pour élaborer des algorithmes qui décrivent le mouvement.

La partie sur la programmation de cette leçon nécessite ScratchJr. Il s’agit d’une application gratuite qui s’adresse aux élèves âgés de 5 à 7 ans. Elle utilise un langage d’introduction par blocs qui permet aux élèves de programmer les étapes pour leur algorithme. Le document sur la programmation pour cette leçon comprend des exemples étape par étape. ScratchJr est également disponible en version d’ordinateur de bureau que l’enseignant peut utiliser pour projeter le programme sur un tableau intelligent. Vous pouvez la trouver en cliquant sur le lien suivant (en anglais seulement) :

<https://jfo8000.github.io/ScratchJr-Desktop/>

Introduction

Donner et suivre des directives et être en mesure d’interpréter ou de visualiser des directives à l’aide d’outils comme des cartes et des grilles sont des compétences utiles que nous utilisons tous les jours. Ces compétences nous permettent de nous rendre à de nouveaux endroits (p. ex., en suivant une route tracée sur une carte ou en suivant les indications d’un système GPS), de communiquer des endroits à d’autres (p. ex., donner des directives à un ami pour visiter votre maison ou un endroit préféré, en utilisant des repères), et de trouver ou de replacer des objets dans leur emplacement approprié (p. ex., les céréales vont sur l’étagère supérieure, à côté des craquelins). Comprendre les directives en ce qui concerne l’emplacement et le mouvement améliore notre connaissance de l’espace et nous permet de mieux naviguer dans le monde qui nous entoure.

Lorsque nous donnons des indications pour aller d’un endroit à un autre, nous décomposons ces dernières en un ensemble de directives ou d’étapes. Dans la programmation, ces directives sont appelées un algorithme. Elles doivent tenir compte de la direction (p. ex., avant, arrière, gauche, droite), de la distance (p. ex., pas ou centimètres) et de tout obstacle qui pourrait se trouver sur le chemin du mouvement. Une personne pourrait prendre une décision si elle se heurte à un obstacle, mais un programme informatique ne peut qu’exécuter avec exactitude les directives reçues.

Dans cette leçon, les élèves dirigeront le mouvement dans plusieurs contextes, en suivant des chemins, en naviguant dans des labyrinthes et dessinant des formes géométriques sur une grille. Les élèves apprendront la façon de diriger le mouvement, de vive voix et à l’aide d’algorithmes de programmation, au moyen de directives précises exécutées étape par étape.

Exécution

Faire marcher le robot

Lorsque tu donnes des directives pour les mouvements, il est important d’être aussi précis que possible. Les humains peuvent prendre des décisions et interpréter tes directives, mais comme

nous le verrons plus tard, quand nous programmons des directives, un robot ou un ordinateur ne fera que ce que tu lui dis de faire.

Pour cette activité hors ligne, les élèves s'entraîneront à décomposer les mouvements en étapes et à communiquer ces étapes à un partenaire.

Un élève peut prétendre être un robot. Le robot ne peut prendre aucune décision par lui-même à propos de ses mouvements. Il ne peut que suivre les directives et il doit les suivre exactement comme elles sont écrites.

L'autre élève sera le programmeur qui crée les directives (aussi connues sous le nom d'algorithme) que l'élève robot suivra. Pour mieux modéliser la programmation, le programmeur devrait composer le code à l'avance avant que son partenaire exécute le programme. Les élèves peuvent aussi donner ces directives en utilisant des flèches directionnelles et des chiffres, ou ils peuvent les donner verbalement.

Parmi les activités que le programmeur peut programmer pour son robot, mentionnons :

- Pour une activité en classe, le programmeur peut créer un ensemble de directives qui permettront à son robot de se déplacer vers un objet placé à un endroit précis sur le sol tout en évitant les obstacles, comme les bureaux d'autres élèves et les murs.
- Autrement, le programmeur peut créer un ensemble de directives dans lesquelles son robot tracera un chemin qui crée une forme géométrique, comme un carré, un rectangle ou un triangle. Cela peut se faire en marchant ou en le dessinant sur du papier quadrillé.

Lorsque le robot exécute le programme, le programmeur devrait s'abstenir de faire des corrections ou d'émettre des directives verbales supplémentaires, tout comme le robot devrait s'abstenir de faire des choix de mouvement qui ne sont pas conformes aux directives du programmeur. Une partie importante de cet exercice est de reconnaître l'importance de la précision et de l'exactitude dans la communication des directives. Donc, s'il manque une étape à l'algorithme, ou si les directives sont inexactes (p. ex., les directives conduisent le robot sur un obstacle comme un mur, ou qu'elles ne conduisent pas le robot jusqu'à l'objectif établi), cela exigera du programmeur qu'il débogue son programme. Cela signifie qu'il trouvera son erreur, la corrigera et essaiera de nouveau le programme en demandant à son partenaire de répéter l'exercice avec l'algorithme révisé.

Pour la première tentative, une approche plus simple serait que le programmeur lise une étape de ses directives à la fois et le robot exécutera cette étape avant de passer à la directive suivante. Comme prolongement, le programmeur peut créer un nouveau chemin, mais cette fois il doit lire le code complet (toutes les étapes en ordre) à haute voix avant que le robot ne

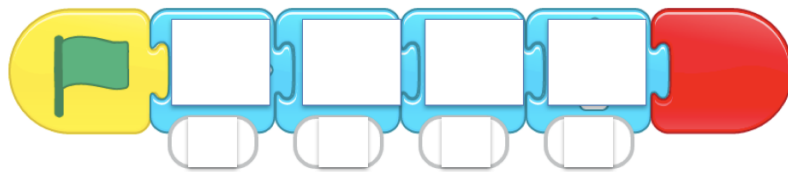
commence à exécuter les directives. Si le temps le permet, les deux partenaires devraient avoir l'occasion d'essayer d'être le programmeur et le robot.

Activité pratique

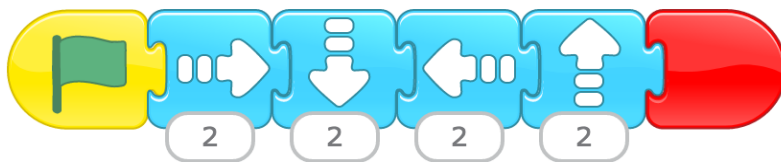
L'activité pratique mettra en œuvre les concepts exécutés pendant l'activité Faire marcher le robot dans un format écrit.

Dans le document fourni, les élèves utiliseront des blocs fléchés pour créer des algorithmes qui communiquent le mouvement sur une page. Ils seront chargés de créer un algorithme pour deux formes différentes ainsi que pour naviguer dans deux labyrinthes. La tâche finale permettra aux élèves de suivre un code donné pour parcourir un labyrinthe.

Les élèves utiliseront une grille où chaque bloc de code de mouvement indique une étape vers la case suivante dans la direction de la flèche. Les élèves devront également indiquer le nombre d'étapes (cases) dans la direction donnée. Vous trouverez ci-dessous un exemple de code incomplet. Le bloc de départ indique que le robot se déplacera lorsqu'on cliquera sur le drapeau vert. Le robot doit connaître le moment où commencer le mouvement. Le prochain bloc aura besoin d'une flèche de direction ainsi que d'un chiffre pour indiquer le nombre de cases du déplacement. Le dernier bloc rouge indique que le code est complet.



Ci-dessous, nous voyons un algorithme complet. Pour faciliter l'exécution, les élèves peuvent dessiner une simple flèche directionnelle sans les lignes pointillées.



Il y a cinq tâches qui vont avec le document sur les formes et les labyrinthes. On fournit également une clé de réponse aux enseignants.

Tâche 1 : Formes géométriques

Pour la première tâche, les élèves apprendront au robot de marcher le long du plus grand carré possible dans la grille. Le plus grand carré possible suivra le périmètre de la grille.

Tâche 2 : Rectangle

Pour la deuxième tâche, les élèves feront déplacer le robot dans un rectangle plus large que haut. Dans le cas présent, il peut y avoir beaucoup de bonnes réponses. Le nombre de cases pour la largeur (gauche, droite) doit être supérieur au nombre de cases pour la hauteur (haut, bas).

Tâches 3 et 4 : Labyrinthes

Les deux prochaines tâches dans le document nécessitent que les élèves communiquent les étapes à l'aide de blocs de code pour naviguer dans les grilles fournies. Chaque grille aura un caractère pour débiter ainsi qu'un but à atteindre. Les élèves peuvent commencer par tracer le chemin sur leurs feuilles, puis déterminer les blocs de code qui communiqueront ce chemin. Il y aura de nombreuses voies possibles. Les élèves n'auront pas toujours besoin d'utiliser tous les espaces fournis.

Encouragez les élèves à utiliser le chemin le plus court possible. Souvent, les algorithmes cartographiques déterminent le chemin le plus court vers une destination donnée. Il y a aussi des cas où nous pouvons changer les paramètres de recherche pour obtenir d'autres itinéraires. Par exemple, si nous voulons éviter les embouteillages ou les péages. Si nous changeons les règles de l'algorithme, il se peut que nous obtenions un résultat différent.

Tâche 5 : Chasse au trésor

Dans la cinquième et dernière partie du document, les élèves reçoivent une grille et un algorithme rempli. Ils devront exécuter le code donné afin de déterminer où placer l'objectif, un X pour marquer l'endroit.

Activité mathématique de programmation

Au cours de l'activité mathématique culminante, les élèves appliqueront ce qu'ils ont appris pour glisser et pour déposer des blocs de code dans ScratchJr afin de formuler des algorithmes qui décrivent le mouvement.

À noter que le document de programmation illustre les directives étape par étape pour la programmation des formes géométriques dans l'application.

Après avoir présenté ScratchJr et la façon dont les blocs de déplacement fonctionnent, les élèves devront effectuer les trois tâches qui composent les activités pratiques. Comme il y a trois tâches, les enseignants peuvent choisir d'effectuer les différentes tâches en groupe soit à l'aide d'un tableau intelligent, soit indépendamment sur iPads, soit avec une combinaison des deux méthodes selon le niveau d'aise des élèves. De notre point de vue, cette activité est probablement mieux faite en groupe, en projetant la version d'ordinateur de bureau de ScratchJr sur le tableau intelligent et en guidant les élèves, mais n'hésitez pas à ajuster l'activité en fonction du niveau de compétence de vos élèves.

Tâche 1 : Grand rectangle

La première tâche sur ScratchJr exigera que les élèves déplacent leur lutin pour former le plus grand rectangle possible dans la grille de ScratchJr.

Tâche 2 : Grand carré

Lorsque les élèves auront tracé le plus grand rectangle possible, ils appliqueront les connaissances du premier défi pour également élaborer le code du plus grand carré possible sur la grille du projet.

Tâche 3 : Rectangle visible

La dernière tâche des élèves sera de programmer le plus grand rectangle possible tout en gardant leur lutin visible tout au long du mouvement. Au fur et à mesure que les élèves accomplissent leurs tâches, faites-leur tester leur code souvent et présentez ensuite leur travail pour évaluer leur apprentissage.

Conclusion et évaluation

- À la fin de la leçon, les élèves devraient être en mesure de donner des directives pour se déplacer d'un endroit à un autre et les suivre, y compris en effectuant les mouvements nécessaires pour passer de la position d'un objet à un autre. Ils pourront également décrire et modifier le mouvement d'un lutin à l'aide d'une programmation par blocs.
- Le document sur les formes et labyrinthes peut être récupéré par les enseignants et utilisé pour évaluer l'apprentissage. Recherchez l'exactitude de la programmation pour évaluer si les élèves peuvent appliquer les concepts et les compétences clés liés à l'activité.

Adaptations

- L'activité Faire marcher le robot peut se faire en demandant aux élèves de tracer des chemins plutôt que de marcher dans ces derniers.
- L'activité pratique et de programmation est divisée en plus petites tâches. Les enseignants peuvent choisir de faire des tâches ensemble ou en groupe avant que les élèves ne tentent de se familiariser avec elles.

Prolongements

- Les élèves peuvent ajouter des obstacles dans les chemins de leurs labyrinthes. Ils peuvent ajuster leur programmation pour contourner les objets afin d'atteindre l'objectif.
- Ensemble, les élèves peuvent essayer de créer des rectangles avec différentes caractéristiques dans ScratchJr, ou des formes plus difficiles.

Ressources supplémentaires

- Téléchargement de la version d'ordinateur de bureau de ScratchJr (en anglais seulement) : <https://jfo8000.github.io/ScratchJr-Desktop/>