

ÉLECTRIFIONS L'AVENIR



DESCRIPTION

Dans cette leçon, les élèves explorent le concept des robots autonomes et leur rôle dans des technologies telles que les véhicules autoconduits. Ils apprendront comment les chauves-souris utilisent l'écholocation pour naviguer et comment ces principes sont appliqués à la robotique et aux systèmes autonomes. Les sujets clés comprennent la conscience spatiale, l'évitement des obstacles et la pensée logique liée à la navigation des robots.

Pour les enseignants, des infographies et des présentations sont fournies afin de guider l'enseignement et d'améliorer la compréhension. Les élèves participeront à des activités pratiques de codage de blocs pour programmer des robots afin d'éviter les obstacles, en testant et en perfectionnant leur code dans un environnement contrôlé. Les résultats du programme d'études liés au codage, à la robotique et aux applications des systèmes autonomes dans le monde réel ont été clairement indiqués afin de garantir l'alignement sur les objectifs éducatifs.



OBJECTIFS



- ☆ Comprendront le concept des robots autonomes et leur application dans les véhicules autonomes.
- ☆ Sauront comment les chauves-souris utilisent l'écholocation, c.-à-d. les ultrasons pour naviguer.
- ☆ Connaîtront l'application des principes de l'écholocation dans la robotique et les véhicules autonomes.
- ☆ Comprendront l'importance de la conscience spatiale et de l'évitement des obstacles en robotique.
- ☆ Posséderont des compétences en réflexion logique touchant l'évitement des obstacles.
- ☆ Posséderont des connaissances de base du codage de bloc et son application dans la programmation des robots.
- ☆ Posséderont de l'expérience pratique en codage pour effectuer des tâches d'évitement des obstacles.
- ☆ Seront en mesure de tester et de parfaire des programmes codés dans un environnement contrôlé.
- ☆ Lieront les concepts appris à des applications dans le monde réel en robotique et dans les systèmes autonomes

Remarque : Étant donné que cette activité utilise le mBot de MakeBlock, nous utiliserons le terme « mBot » au lieu de « robot ».

MATÉRIEL REQUIS



Trousse(s) de mBot

- 4 piles AA (pour le mBot)
- Tablette(s)/téléphone(s)/ordinateur personnel (PC) doté de l'appli mBlock
 - Appli mBlock app pour Web/PC/Mac/Chrome : <https://mblock.cc/pages/downloads>
 - mAppli mBlock pour androïde (Google Play, [mBlock apk](#))
 - Appli mBlock pour iOS ([iOS App Store](#))



mBlock mobile app
Learn coding on phones and tablets



Android
Android 6.0 +
(ARM-based devices only.
X86 Android not supported)



iOS
iOS 10.0 +

- Petits obstacles déplaçables (p. ex. : Mega Bloks, blocs en bois)
- Mur d'enceinte (4-6 morceaux de bois de 2 pouces sur 4 pouces de 2 pieds chacun)
- Jeux de cartes d'indices et Défi Solutions (disponibles à electrifionslavenir.ca)
- Ordinateur et projecteur pour la présentation
- Présentation « Codage et fonctions autonomes dans les automobiles » (disponible à electrifionslavenir.ca)
- Infographies Voir avec le son et Comprendre mBlock

ÉLECTRIFIONS
L'AVENIR

INSTALLATION

electrifiionslavenir.ca

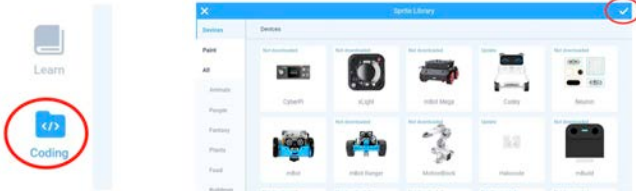
- Assemblez le mBot et les piles.
- Installez l'appli mBlock app sur les appareils.



APPUYEZ ICI POUR
LE VISIONNEMENT

Tablette/PC/téléphone :

- Installez l'appli mBlock
- Pour le codage
 - Ouvrez l'appli mBlock
 - Sélectionnez Codage
 - Sélectionnez mBot comme appareil et cliquez sur la coche dans le coin supérieur droit.



La tablette/le PC/le téléphone est prêt pour le codage.

- Mur d'enceinte** : Disposez quatre blocs pour former un rectangle (ou six pour former un hexagone) et attachez les coins pour qu'ils ne se déplacent pas quand le mBot les heurte.
- Petits obstacles déplaçables** : Ajustez la hauteur de l'obstacle afin qu'elle soit à peu près égale à celle du mBot.

VUE D'ENSEMBLE DES TÂCHES,
ACTIVITÉS ET DÉFIS DE LA LEÇON

#	Nom de la tâche	Matériel
1	Introduction et mot de bienvenue	Ordinateur de l'enseignant, diaporama
2	Contexte scientifique (robots et écholocation)	
3	Activité d'écholocation	
4	Contexte scientifique (capteurs ultrasoniques dans les automobiles)	
5	Présentation du logiciel mBlock	Jeu de cartes d'indices
6	Défi 1 - Discussion	Jeu de cartes d'indices, fiche de tâche 1
7	Défi 1 - Codage	mBot, appli mBlock, ordinateur portable/ tablette, fiche de tâche

#	Nom de la tâche	Matériel
8	Défi 1 -Test	mBot, appli mBlock, ordinateur portable/ tablette
9	Défi 2 - Discussion	Jeu de cartes d'indices, fiche de tâche 2
10	Défi 2 - Codage	mBot, appli mBlock, ordinateur portable/ tablette, fiche de tâche
11	Défi 2 -Test	mBot, appli mBlock, ordinateur portable/ tablette
12	Défi 3 - Discussion	Jeu de cartes d'indices, fiche de tâche 3
13	Défi 3 - Codage	mBot, appli mBlock, ordinateur portable/ tablette, fiche de tâche
14	Défi 3 - Test	mBot, appli mBlock, PC/ordinateur portable/tablette

*Adaptez la durée de la leçon en fonction de l'expérience des participants. Si vous avez besoin de plus de temps pour le codage, sautez certaines activités comme indiqué.

GUIDE DE PRÉSENTATION,
ACTIVITÉS ET DÉFIS

Ce plan de cours est conçu pour être réalisé parallèlement à la présentation « Codage et fonctions autonomes dans les automobiles » (les versions Powerpoint ou Google Docs peuvent être téléchargées à electrifiionslavenir.ca).

► Introduction (tâches 1-2)

Diapositives de la présentation : 1-4, 6

Objectifs : Initier les participants au concept de robot en leur montrant que les voitures autoconduites fonctionnent comme des systèmes robotiques, en mettant l'accent sur la technologie de détection des obstacles. Étudier la façon dont les voitures autoconduites et les chauves-souris utilisent l'écholocation, en soulignant les parallèles entre les systèmes naturels et technologiques.

Variante : Pour les participants plus âgés, il est possible d'utiliser les informations données dans le contexte scientifique afin d'offrir des renseignements approfondis.



ÉLECTRIFIONS

L'AVENIR

► Écholocalisation (tâches 3-4)

Diapositive de la présentation : 5

Matériel : Bandeaux pour les yeux (facultatifs)

Objectif : Faire comprendre comment le son peut être utilisé pour la navigation et la sensibilisation à l'environnement en l'absence de vision.

Description : Un participant a les yeux bandés (ou ferme les yeux), tandis que deux autres se tiennent au hasard dans la pièce. Ces derniers frappent à tour de rôle dans leurs mains et le participant aux yeux bandés utilise le son des battements de mains pour déterminer qui et où sont les autres. Il pointera alors du doigt la direction du son et identifiera les participants. Cette activité est conçue pour faire comprendre comment le son peut être utilisé pour la navigation et la sensibilisation à l'environnement, de la même manière que les chauves-souris utilisent l'écholocalisation lorsqu'elles ne peuvent pas voir.

Variante : Pour les participants plus âgés, l'activité peut être améliorée en incorporant un bandeau sur les yeux et quelques obstacles. Installez une simple course d'obstacles faite de petits objets tels que des chaises, des cônes ou des boîtes dans un grand espace ouvert. Jumelez des participants, l'un ayant les yeux bandés et l'autre jouant le rôle d'« écholocateur ». Le participant aux yeux bandés ne peut qu'avancer, tourner à droite ou s'arrêter en fonction des clics émis par l'écholocateur pour guider le participant autour des obstacles.

Le participant aux yeux bandés commence par rester immobile et dit « le capteur commence à compter 1, 2...5 », en faisant signe à l'écholocateur et en commençant à compter de 1 à 5. Si le participant aux yeux bandés n'entend pas de clic durant cette période, il avance. Si un obstacle se trouve sur son chemin, l'écholocateur émet un clic avant la fin du comptage, ce qui amène le participant aux yeux bandés à s'arrêter et à tourner à droite. Après avoir tourné à droite, le participant aux yeux bandés recommence à compter de 1 à 5. Si un autre clic se fait entendre, indiquant un nouvel obstacle, le participant s'arrête et tourne à nouveau à droite. Ce processus de comptage, d'écoute des clics et de virage à droite se poursuit jusqu'à ce qu'aucun clic ne se fasse entendre au cours du comptage, ce qui indique qu'il est possible d'avancer en toute sécurité.

► Initiation au logiciel (tâche 5)

Diapositives de la présentation : 8-18

Matériel : Jeu de cartes d'indices – Distribuer aux groupes durant cette partie de la présentation.

Objectif : Initier les participants au logiciel mBlock en les guidant dans l'interface et le format de codage de blocs. Expliquer les différentes catégories de blocs qui

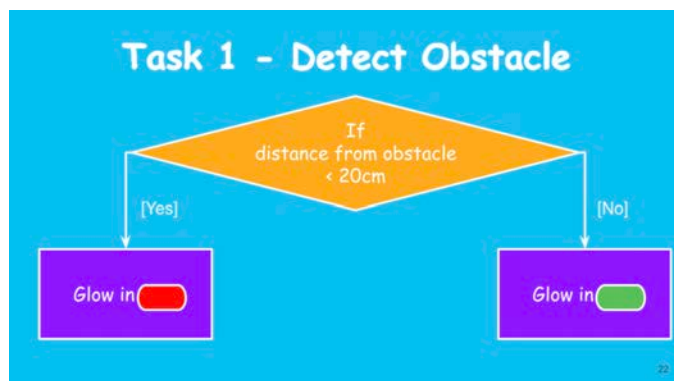
seront utilisées dans l'activité de codage d'évitement d'obstacles.

Description: L'objectif principal de l'activité de codage du robot est de concevoir un programme d'évitement d'obstacles étape par étape. Commencez par présenter le style de codage de blocs, une approche visuelle et conviviale de la programmation du mBot. Accompagnez les participants à travers l'interface du logiciel mBlock, en vous assurant qu'ils comprennent comment naviguer et utiliser les outils disponibles. Mettez en évidence les catégories de blocs qui seront essentielles pour les tâches de codage : Affichage, Actions, Détection, Événements, Contrôles et Opérateurs. Fournissez des explications et des exemples pour chaque catégorie afin de démontrer les types de blocs qu'elle contient. Ces catégories ont été sélectionnées parce qu'elles seront cruciales pour les activités de codage.

À la fin de cette séance, les participants devraient comprendre clairement l'interface du logiciel mBlock et les catégories de blocs pertinentes, ce qui constitue une base solide pour les tâches de codage.

Variante : Pour les participants qui connaissent déjà le codage de blocs, l'enseignant peut modifier l'activité en donnant un bref aperçu des catégories de blocs pertinentes au lieu d'une explication approfondie. Chaque groupe peut ensuite avoir une catégorie à explorer indépendamment. Après quelques minutes, les groupes partageront leurs résultats ou leurs connaissances préalables avec la classe. Cette approche encourage l'apprentissage par les pairs et permet aux participants les plus avancés de mener la discussion.

► Défi de codage 1 : Détection d'obstacles (tâches 6-8)



Diapositives de la présentation : 22-23

Matériel : mBot, mBlock software, Building block, Task sheet 1 (optional)

Objectif : Teach participants how to program the mBot to detect obstacles using its ultrasonic sensor and change the mBot's LED color according to the distance from the obstacle.



ÉLECTRIFIONS

L'AVENIR

Description: Commencez l'activité par une discussion sur la logique du programme, en expliquant pourquoi la couleur de la DEL change en fonction de la distance détectée par le capteur ultrasonique. Guidez les participants pas à pas tout au long du processus de codage, en veillant à ce qu'ils comprennent la fonction de chaque bloc. Ils auront le temps de coder leur programme indépendamment en se basant sur la logique discutée. Une fois le codage terminé, ils pourront tester leur programme en plaçant de petits objets (p. ex., des blocs de construction) à des distances variables devant le mBot pour observer sa réaction.

REMARQUE : Pour tester le code, allumez le mBot, connectez-le à l'application mBlock via Bluetooth et cliquez sur le drapeau vert pour lancer le programme. Les participants peuvent évaluer leur code d'évitement des obstacles en plaçant un bloc à 20 cm et au-delà de 20 cm du mBot.

Considérations importantes : Les enseignants doivent savoir que les capteurs à ultrasons du mBot ont une portée limitée et ne peuvent détecter que des surfaces planes. Il est conseillé de placer le bloc directement devant le mBot plutôt qu'en biais. Après le test, veillez à ce que tous les groupes éteignent leur mBot afin de préserver la durée de vie de la batterie.

Suppléments :

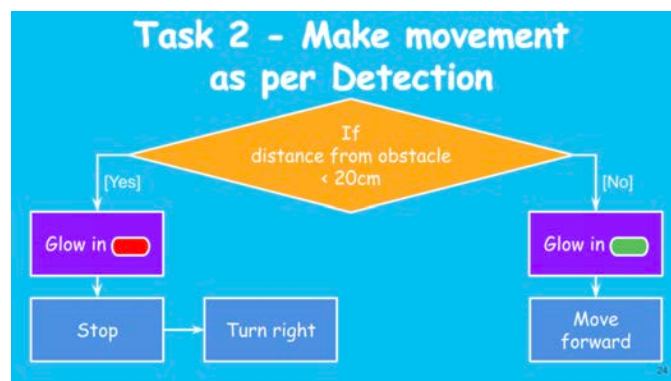
- **Comprendre la portée du capteur ultrasonique :** Si le temps le permet, vous pouvez expliquer comment les capteurs ultrasoniques détectent les distances. Après avoir testé la portée de 20 cm, les participants peuvent la réduire à 10 cm. Ils doivent déplacer le bloc pour trouver l'endroit où la LED du mBot passe du vert au rouge. Lorsqu'elle devient rouge, le déplacement du bloc vers la droite devrait faire repasser la LED au vert, marquant ainsi la fin de la portée du capteur.
- **Applications dans le monde réel :** Après cette exploration, vous pouvez discuter de la manière dont cette limitation est traitée dans les voitures. Par exemple, les véhicules utilisent souvent plusieurs capteurs ultrasoniques (généralement quatre à l'arrière) pour étendre la portée de détection et améliorer la détection des obstacles pendant la marche arrière.

Variantes :

Pour les débutants : Fournissez une fiche de tâche incluant la solution. Les participants suivront la fiche pour construire leur code en suivant l'exemple.

Pour les participants plus avancés : Si les participants ont l'expérience du codage de blocs et sont confiants dans leurs compétences de codage, concentrez-vous sur la discussion de la logique sous-jacente. Laissez-leur le temps de coder le programme indépendamment puis présentez-leur la solution pour qu'ils la comparent à leur propre travail.

► Défi de codage 2 : Évitement d'un seul obstacle (tâches 9-11)



Diapositives : 24-25

Matériel : mBot, mBlock software, Building block, Task sheet 2 (Optional)

Objectif : Enseigner aux participants à programmer le mBot pour qu'il s'arrête lorsqu'il détecte un obstacle, qu'il tourne à droite et qu'il continue à avancer lorsqu'il n'y a pas d'obstacle.

Description: Les participants amélioreront la logique de détection des obstacles du défi 1 en incorporant des commandes de mouvement dans la programmation du mBot. Lorsqu'un obstacle est détecté, le mBot s'arrêtera, puis tournera à droite à une vitesse de 30 %. S'il ne détecte aucun obstacle, il continuera d'avancer.

Commencez par discuter de la manière dont cette logique contrôle le mouvement du robot en fonction de la détection d'un obstacle. Les participants coderont ensuite le programme, soit indépendamment, soit avec des conseils, soit à l'aide d'une fiche de tâche, en fonction de leur niveau de compétence. Pour les tests, demandez aux participants d'apporter leur mBot dans une zone de test désignée, avec des murs d'enceinte pour contenir les mouvements. Le mBot doit être testé dans cette zone plutôt que sur une table.

Considérations importantes :

- Pour coder la commande de virage à droite, utilisez le même code de bloc que pour avancer, mais sélectionnez la direction dans le premier menu déroulant de ce code de bloc.
- Le mBot ne se déplacera pas si la vitesse est réglée à 20 %. Il est recommandé d'utiliser une vitesse de 30 % ou plus.
- Avant le test, assurez-vous que la distance dans le bloc de condition est réglée à 20 cm, pas moins.

Suppléments : Les participants peuvent ajuster la distance pour observer les effets. Vous devez expliquer que la distance est fixée à 20 cm parce que le mBot a besoin de suffisamment d'espace pour s'arrêter complètement avant de heurter un obstacle. Tout comme les véhicules du monde réel doivent maintenir une distance de sécurité avec la voiture qui les précède



ÉLECTRIFIONS

L'AVENIR

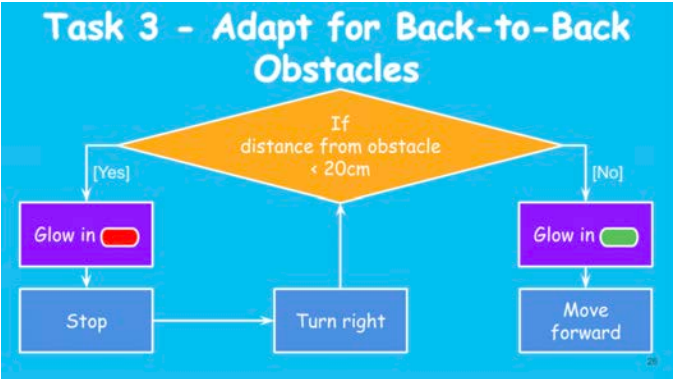
pour éviter les arrêts brusques, le mBot a besoin d’une distance suffisante pour s’arrêter en toute sécurité.

Variantes :

Pour les débutants : Fournissez des fiches de travail supplémentaires ou des instructions détaillées pour les aider à compléter le codage. Guidez-les attentivement à chaque étape pour vous assurer qu’ils comprennent les principes de base du contrôle des mouvements dans le codage par blocs.

Pour les participants plus avancés : Encouragez-les à expérimenter différents types de mouvements, par exemple en faisant tourner le mBot dans différentes directions ou en ajustant la vitesse en fonction de différents scénarios. Laissez-les travailler indépendamment en les guidant au besoin.

► Défi de codage 3 : Évitement de plusieurs obstacles (tâches 12-14)



Diapositives : 8-18

Matériel : Clue Card Decks

Objectif : Améliorez la capacité de navigation du mBot en le programmant pour qu’il détecte et évite plusieurs obstacles à la suite. Il doit continuellement tourner à droite jusqu’à ce qu’il trouve un chemin libre.

Description: Les participants peuvent améliorer les capacités d’évitement d’obstacles du mBot en le programmant pour qu’il contourne plusieurs obstacles. Si le mBot détecte un obstacle (obstacle 1) et tourne à droite pour l’éviter, mais qu’il rencontre ensuite un autre obstacle (obstacle 2) dans la nouvelle direction, il doit effectuer un nouveau virage à droite. Le mBot continuera de tourner à droite jusqu’à ce qu’il trouve une voie libre pour avancer.

Commencez par discuter de cette logique améliorée, en expliquant comment programmer le mBot pour qu’il puisse franchir plusieurs obstacles sans se bloquer. Le processus de codage consistera à ajouter une boucle à la logique existante du défi 2, qui vérifie la présence d’obstacles après chaque tour et effectue des tours supplémentaires si nécessaire jusqu’à ce qu’une voie libre soit trouvée.

Les participants utiliseront des codes de blocs de contrôle, des codes de blocs d’opération (tels que des boucles et des conditionnels), et le code de bloc du capteur ultrasonique pour mettre en œuvre cette amélioration de l’évitement d’obstacles. L’approche sera cohérente avec les défis précédents, ce qui permettra aux participants de coder indépendamment, de suivre des instructions ou d’utiliser une feuille de travail, en fonction de leur niveau de compétence.

Pour les tests, les participants doivent amener leur mBot dans la zone de test désignée qui sera dotée de murs d’enceinte. Vous devrez aménager la zone avec plusieurs obstacles pour simuler un environnement complexe.

Suppléments :

Tourner à droite puis à gauche : Instead of always turning right, participants can program the mBot to first turn right, then check for obstacles again. If an obstacle is detected after the right turn, the mBot should make a 180-degree turn to the left (relative to the initial direction) and then proceed.

Virages à angles précis : Les participants doivent programmer le mBot pour qu’il tourne à des angles précis plutôt que simplement à droite. Ils peuvent y parvenir en programmant une roue pour qu’elle se déplace tandis que l’autre reste immobile, en utilisant un bloc d’attente pour contrôler la durée du virage, puis en ajoutant un bloc de déplacement vers l’avant une fois le virage terminé. Cette amélioration permettra de contrôler les mouvements du mBot avec une précision et d’explorer des techniques de codage avancées.

CONTEXTE SCIENTIFIQUE

L'échololocation chez les chauve-souris

Les chauves-souris utilisent une méthode fascinante appelée écholocation pour naviguer et chasser dans l’obscurité. En émettant des ondes sonores ultrasoniques qui dépassent la portée de l’ouïe humaine, elles peuvent créer une carte mentale de leur environnement. Lorsque ces ondes sonores rebondissent sur des objets et reviennent sous forme d’échos, elles analysent le moment et les caractéristiques des échos pour déterminer l’emplacement, la taille et la forme des objets qui les entourent. Cette capacité leur permet de voler rapidement et avec précision, même dans l’obscurité totale.

Les ultrasons et l’ouïe humaine

Les ultrasons sont des ondes sonores dont la fréquence est supérieure à la limite supérieure de l’ouïe humaine, qui est d’environ 20 000 Hz. L’être humain ne peut pas entendre les ultrasons, mais de nombreux animaux,



ÉLECTRIFIONS

L'AVENIR

notamment les chauves-souris, les dauphins et certains insectes, utilisent ces sons à haute fréquence pour communiquer et naviguer. En technologie, les ultrasons sont utilisés dans diverses applications, telles que l'imagerie médicale et la détection d'obstacles, en raison de leur capacité de fournir des informations détaillées sans être intrusives.

Influence de la nature sur les systèmes d'IA

Les scientifiques et les ingénieurs s'inspirent souvent de la nature pour concevoir des technologies de pointe. Le concept d'écholocation, par exemple, a été adapté pour créer des systèmes artificiels qui imitent cette capacité naturelle. En étudiant la manière dont les chauves-souris utilisent le son pour se diriger, les chercheurs ont mis au point des capteurs ultrasoniques qui permettent aux machines de percevoir leur environnement de la même manière. Cette approche bio-inspirée permet d'améliorer l'efficacité et la fonctionnalité des systèmes d'intelligence artificielle (IA).

Voitures autonomes

Les voitures autonomes, ou voitures autoconduites, sont un excellent exemple de la façon dont l'IA et la technologie des capteurs s'associent pour créer des systèmes intelligents. Ces véhicules sont équipés de divers capteurs qui leur permettent de fonctionner sans intervention humaine. Les principaux capteurs sont des caméras, des radars, des lidars et des capteurs ultrasoniques qui fournissent chacun différents types de données pour aider la voiture à comprendre son environnement.

Les capteurs ultrasoniques dans les voitures autonomes

Les capteurs ultrasoniques jouent un rôle crucial dans la sécurité des voitures autonomes. Ils émettent des ondes ultrasoniques qui se réfléchissent sur les objets proches, et le système de la voiture analyse les échos renvoyés pour détecter les obstacles et mesurer les distances. Cette technologie est particulièrement utile pour la détection à courte distance, notamment lors du stationnement et des manœuvres à faible vitesse. Grâce aux capteurs ultrasoniques, les voitures autonomes peuvent naviguer avec précision dans des environnements complexes, éviter les collisions et assurer la sécurité des passagers.



ÉLECTRIFIONS

L'AVENIR

CURRICULUM CONNEXE DE L'ONTARIO

(niveau intermédiaire) Les résultats d'apprentissage suivants sont directement liés à la leçon suivante. Des détails peuvent être ajoutés à la leçon pour correspondre à un plan d'unité spécifique ou pour étendre l'apprentissage de l'expérience. Veuillez noter que cette liste n'inclut pas nécessairement tous les résultats pertinents.

1re – 8e années

Codage et technologies émergentes

- A2.1 écrire et exécuter des codes lors de l'exploration et de la modélisation de

G1 pour créer des instructions claires et précises pour des algorithmes simples.

G2 pour décomposer des problèmes en petites étapes.

G3 pour tester, déboguer et améliorer des programmes.

G4 pour produire différents types de données de sortie à des fins diverses.

G5 pour utiliser différentes méthodes de stockage et de traitement de données à des fins diverses.

G6 pour obtenir des données d'entrée de différentes façons à des fins diverses.

G7 pour planifier et concevoir des programmes.

G8 pour automatiser de grands systèmes en action.
- A2.2 déterminer et décrire l'incidence du codage et des technologies

G1-3 sur des situations de la vie quotidienne.

G4-8 sur des situations de la vie quotidienne et sur des métiers spécialisés.

9e année

Recherches et expériences liées aux STIM (SNC1W)

- A1.3 appliquer un processus de design en ingénierie et les habiletés connexes pour concevoir, construire et tester des dispositifs, des modèles, des structures et/ou des systèmes.
- A1.4 appliquer des habiletés en codage pour examiner et modéliser des concepts scientifiques et des relations connexes.

Applications, carrières et liens connexes (SNC1W)

- A2.1 concevoir une expérience ou un prototype pour explorer un problème authentique à un secteur d'activité lié aux STIM, tel qu'un métier spécialisé, à partir de résultats de recherches.

Résolution de problèmes et gestion de projet (TIJ10)

- B1.1 Appliquer les étapes d'un processus de conception ou d'un autre processus de résolution de

- problèmes pour planifier et développer des produits et des services (par exemple, définir le problème ou le défi, en tenant compte des informations contextuelles ou de fond pertinent; recueillir des informations [sur les critères, les matériaux, les contraintes]; générer des solutions possibles, en utilisant des techniques telles que le remue-méninge; choisir la meilleure solution; développer et produire un modèle ou un prototype; incorporer des améliorations ou revoir la conception et refaire les tests; rendre compte des résultats).
- B1.5 démontrer la capacité de travailler en coopération dans un environnement de groupe pour résoudre des problèmes (p. ex., partager des outils, des tâches, des matériaux et des ressources).

10e année (ICD20)

Applications, carrières, et liens connexes (ICD20)

- A3.1 examiner des façons dont des habiletés liées aux technologies numériques et à la programmation peuvent être utilisées au sein de diverses disciplines dans des situations authentiques.
- A3.2 examiner des changements au sein de divers secteurs d'activité, y compris les métiers spécialisés, engendrés par des innovations en matière de technologies numériques et de programmation.

Compréhension du matériel informatique et des logiciels (ICD20)

- B1.1 décrire les fonctions et les caractéristiques de divers composants de matériel informatique de base associé aux technologies numériques utilisées quotidiennement
- B1.2 décrire les fonctions et les caractéristiques de divers appareils connectés associés aux technologies numériques utilisées quotidiennement

Innovations en matière de technologies numériques (ICD20)

- B4.1 examiner des innovations récentes, y compris l'automatisation et les systèmes d'intelligence artificielle, et évaluer leur incidence sur des situations de la vie quotidienne
- B4.3 examiner des innovations émergentes en matière de matériel informatique et de logiciels, ainsi que leurs avantages et leurs limites dans des situations de la vie quotidienne de demain

Concepts de programmation et algorithmes (ICS20), ICD20

- (B1.5) employer la terminologie appropriée pour décrire les concepts de programmation et les algorithmes
- C1.1 employer la terminologie appropriée pour décrire les concepts de programmation et les algorithmes
- C1.2 décrire des algorithmes simples utilisés dans des situations de la vie quotidienne



ÉLECTRIFIONS

L'AVENIR

C1.3 déterminer divers types de données et expliquer la façon dont elles sont utilisées dans des programmes

C1.4 déterminer les expressions et les instructions appropriées d'un programme, en tenant compte de la priorité des opérations

C1.5 déterminer et expliquer des situations requérant des structures conditionnelles et répétitive

Développement de programmes (ICD20)

C2.2 développer des programmes qui utilisent et génèrent des données de divers formats et provenant de diverses sources

C2.4 développer des programmes comportant des événements séquentiels, sélectionnés et répétitifs de manière efficace

C2.5 développer des programmes comprenant l'utilisation d'opérateurs booléens, d'opérateurs de comparaison, d'opérateurs de texte et d'opérateurs arithmétiques

C2.6 interpréter des erreurs de programmation, et mettre en œuvre des stratégies pour les corriger

Modularisation et modification (ICD20)

C3.1 analyser des codes existants pour en comprendre les composants et les résultats

C3.2 modifier un programme existant, ou des composants de programme, pour lui permettre d'atteindre un résultat différent

Programming Concepts and Algorithms (ICS20)

B1.5 use appropriate terminology to describe programming concepts and algorithms

B1.6 describe the function of Boolean operators (e.g., AND, OR, NOT), comparison operators (i.e., equal to, not equal to, greater than, less than, greater than or equal to, less than or equal to), and arithmetic operators (e.g., addition, subtraction, multiplication, division, exponentiation, parentheses), and use them correctly in programming

Développement de programmes (ICS20)

B2.1 utiliser un modèle visuel de résolution de problèmes (p. ex., un tableau IPO [Input, Process, Output], un tableau et un diagramme HIPO [Hierarchy plus Input, Process, Output], un organigramme, un storyboard) pour planifier le contenu d'un programme.

B2.4 développer des programmes utilisant une structure de décision incluant au moins deux choix possibles pour répondre à un problème donné (p. ex., jeu de devinettes, roche-papierciseaux, question à choix multiples).

B2.5 écrire des programmes qui utilisent efficacement des structures en boucle (par exemple, animation simple, jeux de société simples, jeu de pile ou face).

Programmation informatique (TEJ20)

B5.3 utiliser une structure de décision et une structure de répétition dans un programme (p. ex., déterminer si un utilisateur est assez âgé pour conduire, lancer un jeu de devinettes, compter d'une valeur de départ à une valeur d'arrivée).

B5.4 utiliser un processus de conception (voir p. 18-19) pour planifier, écrire et tester un programme informatique destiné à contrôler un robot simple ou un périphérique (p. ex., un servomoteur, un écran LED)

Technologie et société (TEG20)

C2.2 décrire comment les ordinateurs sont utilisés dans diverses professions (p. ex., calculs d'ingénierie, dessins d'architecture, suivi des clients et collecte de données commerciales, navigation aérienne et maritime), et ce que serait le travail dans ces professions sans les ordinateurs.

11e année

Types de données et expressions (ICS3U)

A1.4 démontrer la capacité à utiliser correctement les opérateurs booléens (p. ex., ET, OU, PAS), les opérateurs de comparaison (p. ex., égal à, non égal à, supérieur à, inférieur à, supérieur ou égal à, inférieur ou égal à), les opérateurs arithmétiques (p. ex., addition, soustraction, multiplication, division, exponentiation, parenthèses) et l'ordre des opérations dans les programmes d'ordinateur.

Structures de contrôle et algorithmes simples (ICS3U)

A2.2 utiliser les structures de contrôle de séquence, de sélection et de répétition pour créer des solutions de programmation

A2.3 écrire des algorithmes avec des structures imbriquées (p. ex., pour compter les éléments d'un tableau, calculer un total, trouver la valeur la plus élevée ou la plus basse, ou effectuer une recherche linéaire).

Stratégies de résolution de problèmes (ICS3U)

B1.1 utiliser diverses stratégies de résolution de problèmes (p. ex., affiner par étapes, diviser pour régner, travailler à l'envers, exemples, cas extrêmes, tableaux et graphiques, essais et erreurs) pour résoudre différents types de problèmes

B1.2 démontrer la capacité à résoudre des problèmes de manière indépendante et en tant que membre d'une équipe.

Techniques de développement de logiciels (ICS3U)

B2.1 concevoir des programmes à partir d'un modèle ou d'un squelette de programme (p. ex., un squelette fourni par l'enseignant, un extrait de code de la fonction d'aide).



ÉLECTRIFIONS

L'AVENIR

B2.2 utiliser le vocabulaire et le mode d'expression appropriés (c'est-à-dire écrit, oral, diagramme) pour décrire des conceptions de programme alternatives et pour expliquer la structure d'un programme.

B2.4 représenter la structure et les composants d'un programme à l'aide d'outils de programmation standard (p. ex., diagramme de structure, diagramme de flux, UML [Unified Modeling Language], diagramme de flux de données, pseudocode).

Exploration de l'informatique (ICS3U)

D2.1 démontrer une compréhension des nouveaux domaines de recherche en informatique (p. ex., la cryptographie, le traitement parallèle, l'informatique distribuée, l'exploration de données, l'intelligence artificielle, la robotique, la vision par ordinateur, le traitement d'images, l'interaction homme-ordinateur, la sécurité, les systèmes d'information géographique [SIG]).

D2.2 démontrer une compréhension d'un domaine de recherche collaborative entre l'informatique et un autre domaine (p. ex., la bio-informatique, la géologie, l'économie, la linguistique, l'informatique de santé, la climatologie, la sociologie, l'art)

12e année

Structures de données

A1.2 utiliser correctement les opérateurs booléens (par exemple, ET, OU, PAS), les opérateurs de comparaison (p. ex., égal à, non égal à, supérieur à, inférieur à, supérieur ou égal à, inférieur ou égal à), les opérateurs arithmétiques (p. ex., addition, soustraction, multiplication, division, exponentiation, parenthèses) et l'ordre des opérations dans la programmation.