

Programmation de circuits		6 ^e année
Plan de leçon	Outil de programmation	TinkerCAD Circuits
	Durée	Deux périodes
Liens avec le curriculum des mathématiques <u>Algèbre</u> Attentes générales C3. Résoudre des problèmes et créer des représentations computationnelles de situations mathématiques au moyen de concepts et d'aptitudes de programmation. Attentes particulières C3.1. Résoudre des problèmes et créer des représentations computationnelles de situations mathématiques en écrivant et exécutant du code, y compris du code qui comporte des événements séquentiels. C3.2. Lire et modifier du code existant, y compris du code qui comporte des instructions conditionnelles et d'autres structures de contrôle, et décrire la façon dont les changements apportés au code influencent les résultats et l'efficacité du code.	Liens avec le curriculum des sciences <u>Niveau 6</u> Électricité et dispositifs électriques <ul style="list-style-type: none"> • L'énergie électrique peut être transformée en d'autres formes d'énergie. Attentes générales 2. Étudier les caractéristiques du courant électrique et construire des circuits simples. Attentes particulières 3.6 Expliquer les fonctions des composants d'un circuit électrique simple.	
Description Les étudiants apprendront les bases des nombres binaires comme fondation de l'informatique et mèneront une étude pratique sur papier des motifs dans les nombres binaires. En extrapolant à partir de ces connaissances, les étudiants construiront des circuits simples dans TinkerCAD Circuits et étudieront les algorithmes de programmation, y compris les boucles et les instructions conditionnelles, qui peuvent être utilisés pour contrôler la sortie d'un circuit.		

<p>Critères de réussite</p> <ul style="list-style-type: none"> • Les étudiants de niveau 6 seront en mesure d'identifier les composants d'un circuit simple et de déterminer comment les composants sont connectés pour leur fonctionnement. • Les étudiants de niveau 6 seront en mesure de démontrer la façon dont la sortie d'un circuit peut être contrôlée au moyen d'instructions conditionnelles avec TinkerCAD Circuits. 	<p>Matériel et médias</p> <ul style="list-style-type: none"> • Appareils qui prennent en charge TinkerCAD Circuits (application d'ordinateur de bureau ou version navigateur – voir la note ci-dessous dans Ressources supplémentaires) • Activité pratique sur les nombres binaires • Guide de programmation
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Compétences en pensée computationnelle

Cette leçon adopte deux approches à la programmation : abstraite et pratique. Dans l'activité pratique sur papier, les étudiants prendront un recul des langages de programmation et de la logique qu'ils peuvent connaître pour se pencher sur la façon dont les ordinateurs interprètent l'information à son niveau le plus fondamental : l'information binaire ou les bits (voir le document « *Activité pratique sur les nombres binaires* »).

Dans l'activité de programmation en ligne, les étudiants seront en mesure d'appliquer les concepts de programmation à des exemples du monde réel. Ils modifieront des circuits et contrôleront les sorties à l'aide de boucles et d'instructions conditionnelles. Les instructions conditionnelles sont utilisées dans la programmation pour exécuter une condition si un énoncé est vrai (cela peut être considéré comme analogue à aider un ordinateur à prendre une décision programmée en fonction de facteurs ou de renseignements donnés).

Les étudiants exploreront et modifieront les instructions conditionnelles qui contrôlent les sorties d'un circuit au moyen de TinkerCAD, un logiciel gratuit qui utilise la programmation par blocs simplifiée (semblable à Scratch). Les étudiants plus avancés (ou curieux) peuvent également utiliser TinkerCAD Circuits pour voir comment leur code par blocs se traduit en C, le langage de programmation utilisé pour le microcontrôleur Arduino.

Le « *Guide de programmation* » pour cette leçon comporte une procédure étape par étape plus détaillée pour utiliser TinkerCAD Circuits comparativement à la description ci-dessous.

Introduction

Introduction aux nombres binaires

Les ordinateurs utilisent le système binaire pour représenter l'information. Nous utilisons les nombres binaires dans les ordinateurs puisque, à la base, les ordinateurs sont formés de circuits très simples dans lesquels le courant électrique « circule » alors que les « interrupteurs » électroniques s'ouvrent et se ferment. Ces circuits sont si simples que tout ce qui permet à l'ordinateur de fonctionner est représenté par FERMÉ, ou la présence de courant (représenté par le chiffre 1) ou OUVERT, l'absence de courant (représenté par le chiffre 0). Les nombres binaires correspondent directement à la façon dont les ordinateurs stockent l'information. Chaque zéro ou un s'appelle un bit (ou un chiffre binaire).

Les données informatiques peuvent être considérées comme une série de « fils » connectés à une série d'interrupteurs qui laissent passer ou bloquent la transmission de courant dépendamment qu'un 1 ou 0 doit être transmis.

Introduction aux circuits

Les circuits électriques requièrent une alimentation (pour fournir un courant électrique au circuit), des conducteurs pour transmettre le courant (par exemple, des fils) et une charge (un dispositif qui transforme l'énergie du courant électrique pour générer une sortie, comme de la lumière, du son ou du mouvement). Nous pouvons contrôler la sortie d'un dispositif en contrôlant le courant qui traverse le dispositif. En général, nous faisons cela avec les interrupteurs, lesquels ouvriront un circuit pour bloquer la transmission du courant vers le dispositif et le fermeront pour permettre au courant de traverser le dispositif, arrêtant et démarrant le dispositif, respectivement.

Au cours de cette leçon, nous construirons des circuits simples et nous les contrôlerons avec des interrupteurs. Ces interrupteurs seront eux-mêmes contrôlés par le code afin que nous soyons en mesure de programmer des sorties particulières en fonction de conditions données, comme lorsqu'un bouton est enfoncé (et un interrupteur est fermé) et vice versa.

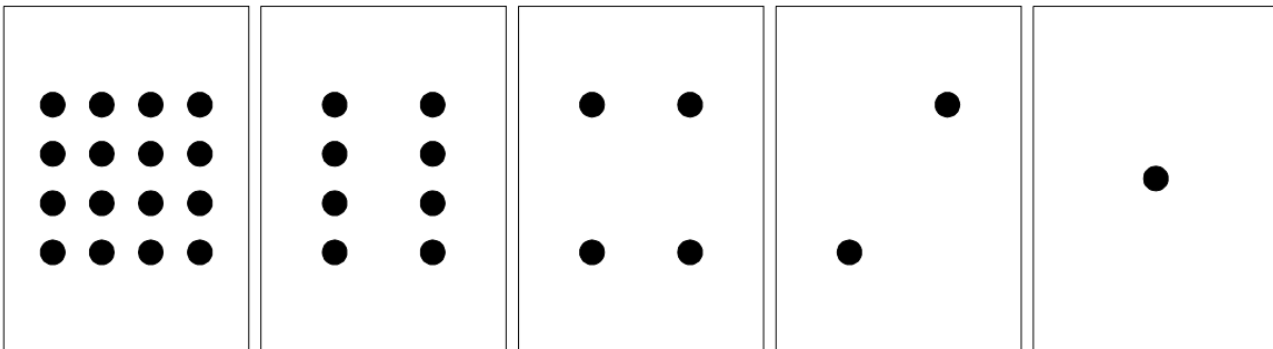
Action

Activité pratique

Pour comprendre comment l'information peut être transmise au moyen de 1 et de 0 seulement, demandez aux étudiants de pratiquer la traduction de nombres binaires en nombres entiers.

Démonstration des nombres binaires

Créez cinq cartes avec des points (illustrées ci-dessous). Chaque carte a le double du nombre de points de la carte à sa droite (demandez aux étudiants s'ils peuvent trouver ce motif). Elles doivent être disposées dans l'ordre montré ci-dessous (de la gauche vers la droite : 16 points; 8 points; 4 points; 2 points; et 1 point).



Choisissez cinq étudiants comme volontaires pour cette démonstration. Chaque étudiant reçoit l'une de ces cartes et se met en ligne dans le même ordre montré. Expliquez que lorsqu'une carte n'est **pas** montrée (c'est-à-dire lorsque la carte n'est pas retournée pour dévoiler ses points), elle représente un zéro. Lorsque la face de la carte **montre** ses points, elle représente le chiffre un. Pour convertir un nombre binaire en nombre entier, les étudiants additionnent les points montrés, c'est-à-dire seulement les points représentés par le chiffre un.

Par exemple, demandez aux étudiants de représenter 10001 avec leurs cartes. Les uns sont face vers le haut, montrant les points alors que les zéros sont face vers le bas, ce qui nous donne la séquence « haut, bas, bas, bas, haut ». En comptant le nombre de points montrés, les étudiants peuvent déterminer le nombre entier représenté :



(Réponse = 17)

Demandez aux étudiants d'utiliser leurs cartes pour faire les nombres 1 à 20 en séquence (00001, 00010, 00011, etc.). Le reste de la classe devrait observer la séquence et tenter de déterminer un motif dans la façon dont les cartes sont retournées. Chaque carte est retournée moitié moins de fois que la carte à sa droite.

*Remarque : Cette activité peut également être faite individuellement, avec chaque étudiant recevant son propre paquet de cinq cartes à manipuler à son bureau, plutôt que dans une démonstration.

Le document « *Activité pratique sur les nombres binaires* » renforce l'idée d'encoder l'information au moyen d'un système binaire en demandant aux étudiants de décoder un message binaire puis de créer leur propre message.

Activité de programmation

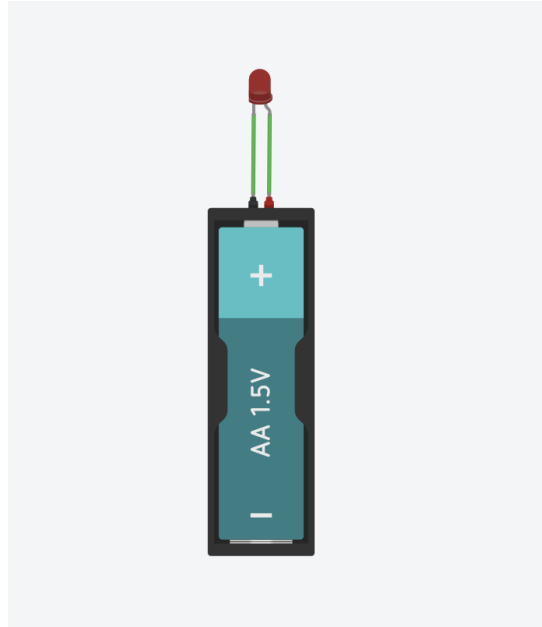
Créer un circuit simple dans TinkerCAD Circuits

Pour créer le circuit électrique le plus simple, nous avons besoin de : (1) une source d'énergie (dans ce cas-ci, une pile); (2) un conducteur (fils); et (3) une charge (une DEL). Demandez aux étudiants de décrire des appareils de tous les jours qui comportent des circuits. Comment ces appareils sont-ils contrôlés (c'est-à-dire utilisent-ils un interrupteur)?

Notez que les étudiants peuvent utiliser le « *Guide de programmation* » comme référence pour suivre les étapes.

- Trouvez la DEL et la pile de 1,5 V dans le menu des composants et glissez-les sur le canevas de programmation.
- Dessinez les fils en cliquant sur le point de départ et le point d'arrivée avec votre souris ou votre pavé tactile. Un fil sera automatiquement dessiné entre ces points.

- Mettez à l'essai votre circuit en appuyant sur le bouton « Start Simulation » (Lancer la simulation). Qu'arrive-t-il à votre DEL?



(Texte descriptif : Image montrant une DEL rouge branchée à une pile AA avec deux fils pour former un circuit simple.)

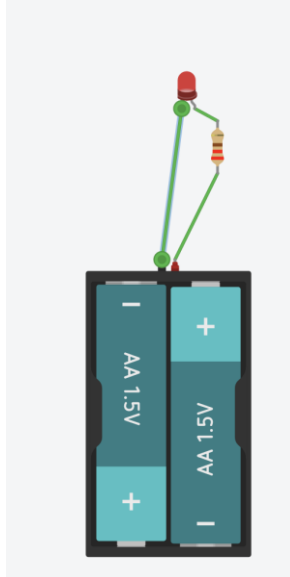
* Remarque : Comme pour les vrais circuits, la façon dont vous branchez les fils est importante! La cathode de la DEL (ou le raccord positif) doit être branchée à la terminaison négative de la pile et l'anode de la DEL (ou le raccord négatif) doit être branchée à la terminaison positive de la pile.

Ajouter des résistances

- Nous allons augmenter la quantité de courant provenant de notre source d'alimentation en ajoutant une deuxième pile. Pour y arriver, cliquez avec le bouton gauche sur la source d'alimentation et changez le nombre de pile de 1,5 V à **2 piles**. Qu'arrive-t-il à votre DEL?

(Texte descriptif : Image montrant une DEL rouge branchée à deux piles AA avec deux fils pour former un circuit simple. La DEL est couverte d'un symbole d'explosion rouge.)

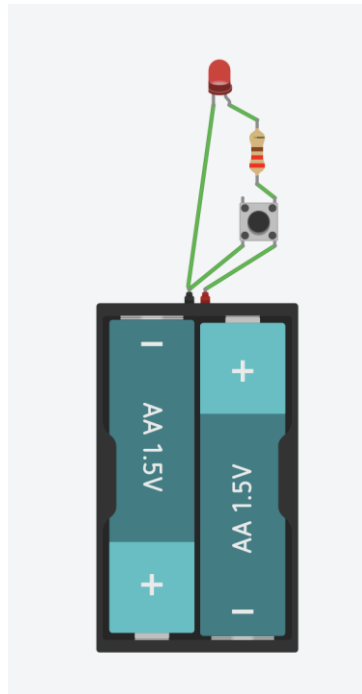
- Le courant qui traverse la DEL dépasse le maximum que notre charge peut supporter. Surcharger un circuit peut entraîner la surchauffe des fils, ruiner le dispositif que vous tentez d'alimenter ou allumer un feu. Pour réduire le courant de notre charge, nous pouvons ajouter une résistance. Prenez une résistance dans le panneau des composants et glissez-la dans le canevas de programmation.
- Rebranchez le circuit en redessinant les fils afin qu'un fil connecte la terminaison négative du groupe de piles à la terminaison 1 de la résistance et qu'un fil connecte la terminaison 2 de la résistance à la DEL.
- Lancez la simulation. Qu'arrive-t-il à votre circuit?



(Texte descriptif : Image montrant un circuit simple avec une DEL rouge, deux fils verts, un groupe de piles et une résistance. La DEL est allumée.)

Ajouter un interrupteur

- Ajoutons un interrupteur qui nous permettra de contrôler notre DEL. À partir du menu des composants, choisissez le bouton-poussoir et glissez-le dans le canevas de programmation. Nous voulons intégrer notre bouton au circuit en le branchant aux deux terminaisons de notre source d'alimentation, ainsi qu'à l'anode de notre DEL (en passant par une résistance). Consultez l'image ci-dessous qui illustre l'une des façons possibles de connecter l'interrupteur.



(Texte descriptif : Image montrant un circuit simple avec une DEL rouge, deux fils verts, un groupe de piles, une résistance et un interrupteur. La DEL est allumée.)

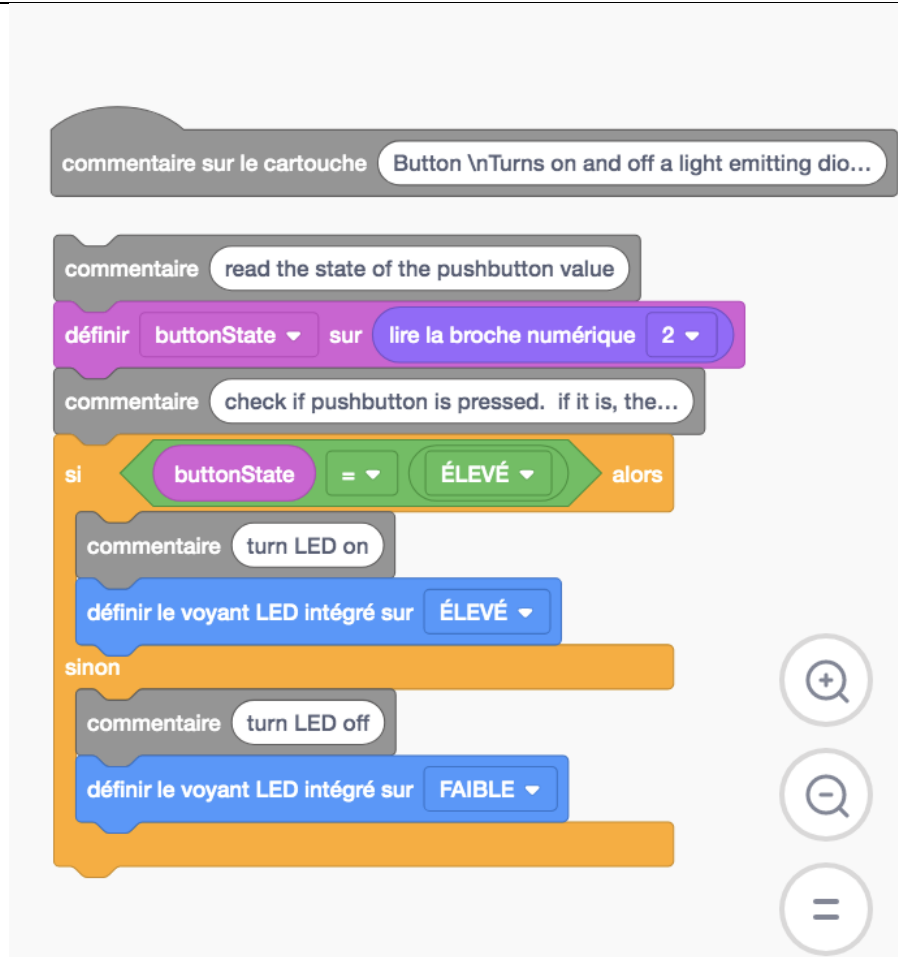
Avec la façon dont ce circuit est configuré, la DEL s'allume lorsque le bouton N'EST PAS enfoncé et s'éteint lorsque le bouton est enfoncé. Cela est configuré comme ce qui est appelé un interrupteur normalement fermé, signifiant que le circuit est interrompu lorsque vous appuyez sur le bouton. Vous connaissez peut-être mieux les interrupteurs normalement ouverts, où il n'y a aucune connexion jusqu'à ce que le bouton soit enfoncé (par exemple, sonnettes de porte, boutons de clé de voiture électronique).

Ajouter du code

À côté du bouton de lancement et d'arrêt de la simulation, vous remarquerez qu'il y a un bouton **Code**. Cliquer sur ce bouton dévoilera le code pour contrôler la sortie de notre circuit (dans ce cas-ci, comment notre DEL s'allume). Avec notre circuit actuel, il n'y a aucun code puisque notre circuit n'est pas programmable présentement; le seul contrôle que nous avons est si le circuit est ouvert (éteint ou 0) ou fermé (allumé ou 1).

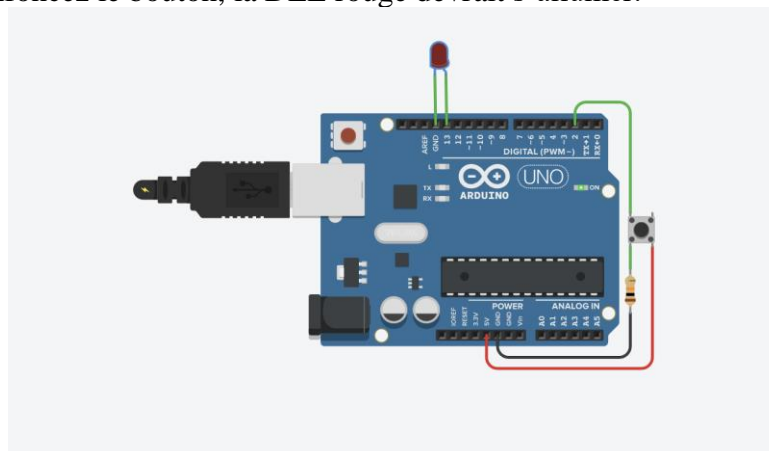
- Dans le menu déroulant **Components (Composants)**, choisissez **Arduino**. Glissez l'exemple **Button (Bouton)** dans le canevas de programmation.
- Ce circuit est très semblable au circuit que nous avons fait précédemment. Notre Arduino est un microcontrôleur programmable alimenté par USB. Le courant passe par la fiche 13, est contrôlé par une résistance jusqu'à la charge (la DEL) qui, dans ce cas, est une toute petite lumière intégrée à l'Arduino.
- Si vous appuyez sur le bouton Code, vous verrez le code par blocs pour programmer le circuit (si vous avez utilisé Scratch ou Blockly auparavant, vous remarquerez que ce format de programmation est très semblable).

Le code existant est une boucle (bien qu'aucun bloc de boucle n'existe dans ce cas; si vous choisissez de changer l'affichage à « Blocks + Text » (Blocs + Texte), vous serez en mesure de voir la boucle en C (le langage de programmation utilisé pour Arduino) qui vérifie la fiche connectée au bouton-poussoir (fiche 2) pour voir si le bouton est enfoncé (« HIGH » [ÉLEVÉ] ou 1) ou pas (« LOW » [FAIBLE] ou 0). Nous voyons ensuite une instruction conditionnelle pour les deux états du bouton-poussoir. Si la variable `buttonState = HIGH`, alors le contrôleur fixe la fiche qui contrôle la DEL (fiche 13) à HIGH et la DEL s'allume; sinon, la fiche qui contrôle la DEL est fixée à LOW et la DEL s'éteint.



(Texte descriptif : Image montrant le code par blocs qui décrit le programme qui utilise Arduino et un bouton-poussoir pour contrôler une DEL intégrée.)

- Ajoutons une DEL externe pour voir les effets plus clairement. Nous pouvons glisser une DEL dans le canevas de programmation et brancher la cathode à la fiche de mise à la terre (GND) et l'anode à la fiche 13 (la fiche de la DEL sur la carte Arduino). Maintenant, lorsque vous enfoncez le bouton, la DEL rouge devrait s'allumer.



(Texte descriptif : Image montrant une carte Arduino où des fils connectent un bouton-poussoir et une DEL rouge.)

Aller plus loin

- Changeons la sortie de notre instruction conditionnelle afin que la DEL clignote toute seule lorsque le bouton-poussoir est enfoncé. Nous ferons cela en ajoutant une boucle à l'intérieur de l'instruction Si de notre instruction conditionnelle. La boucle fixera la DEL à HIGH, attendra 500 millisecondes (ou 0,5 seconde), fixera la DEL à LOW, attendra 500 millisecondes et répétera ces étapes tant que le bouton est enfoncé.



(Texte descriptif : Image montrant le code par blocs qui décrit le programme qui utilise Arduino et un bouton-poussoir pour contrôler une DEL et la faire clignoter.)

- Essayez d'expérimenter avec les paramètres de votre code. Pouvez-vous créer votre propre motif de clignotement en ajustant les blocs d'attente?

Conclusion et évaluation

À la fin de la leçon, les étudiants devraient être en mesure de reconnaître les composants requis pour créer un circuit simple et être en mesure de décrire les ressemblances entre un circuit électrique et le flux de l'information à l'intérieur d'un ordinateur comme étant contrôlé par des interrupteurs OUVERT/FERMÉ ou des chiffres binaires. Les étudiants devraient être en mesure de modifier de simples instructions conditionnelles pour changer la sortie d'un circuit créé de manière numérique, de mettre à l'essai leur code et d'itérer.

Pour l'évaluation, recueillez les documents de l'activité pratique sur les nombres binaires des étudiants. Examinez leur travail afin de vous assurer qu'ils comprennent le concept du système binaire en décodant correctement le message binaire et qu'ils ont créé leur propre message qui suivait les règles de logique du système.

Adaptations

- La démonstration des nombres binaires peut se faire individuellement, avec chaque étudiant recevant son propre paquet de cinq cartes qu'il manipulera à son propre bureau, plutôt que de faire une démonstration.
- La démonstration des nombres binaires peut également être faite avec les étudiants assis dans une rangée de cinq.
- La couleur des fils et des composants DEL dans ce programme peut facilement être changée en cliquant avec le bouton gauche sur le composant et en sélectionnant une couleur différente dans les cas où la distinction des couleurs peut être difficile.

Extensions

- Les étudiants qui terminent rapidement peuvent tenter de créer des circuits plus créatifs. Peuvent-ils réussir à programmer une DEL RVB?
Remarque : la cathode doit être branchée à la fiche de mise à la terre (GND) et chacun des raccords R, V et B doivent avoir leurs propres fiche et résistance.

Ressources supplémentaires

- Tinkercad.com (compte gratuit requis) *Remarque : en date de novembre 2020, Circuits est seulement disponible dans la version pour ordinateur de bureau et navigateur Internet de TinkerCAD. L'application TinkerCAD ne vous permet pas de créer ou simuler des circuits pour l'instant. Le compte en ligne est gratuit et un seul compte devrait être requis (tous les étudiants peuvent se connecter sous le même compte), toutefois vous pouvez choisir de créer des comptes pour chaque étudiant.