A picture containing toy

Description automatically generated

**PLAN DE LA LEÇON**

**Titre :** Heure de Code 2021 (TimeCraft)

**Âges :** 7 ans et +

**Matières :** informatique, mathématiques, arts du langage, histoire, arts visuels, sciences, ingénierie

**Compétences du 21e siècle :**créativité, pensée critique, résolution de problèmes

**Langues prises en charge :** anglais (États-Unis), anglais (Royaume-Uni), allemand, espagnol (Espagne), espagnol (Mexique), français (France), français (Canada), italien, japonais, coréen, portugais (Brésil), portugais (Portugal), russe, chinois (simplifié), chinois (traditionnel), turc, polonais, indonésien, ukrainien, grec, bulgare, hongrois, finnois, danois, suédois, norvégien, slovaque, tchèque, néerlandais.

**Mode solo ou multijoueur :** mode solo

**Niveau d'expérience de Minecraft requis (enseignant) :** novice

**Description de la leçon :** une introduction créative et amusante à l'informatique. Voyagez dans le temps et utilisez un langage de programmation par blocs ou Python afin de résoudre de mystérieux incidents sur la ligne du temps de la Terre. Recueillez des indices et utilisez votre jugement pour trouver le malfaiteur dans cette aventure à travers le temps. Les élèves découvriront aussi comment l'informatique s'intègre dans les carrières et les champs d'intérêts personnels.

**Temps total requis :** 1 heure

**Normes pédagogiques :**

|  |  |  |
| --- | --- | --- |
| Normes | | |
| États-Unis | Australie | Royaume-Uni – Angleterre |
| Je peux modéliser des processus quotidiens en créant et en suivant des algorithmes pour accomplir les tâches.  **1A-AP-08** | Je peux suivre, décrire et représenter une séquence d'étapes et de décisions (algorithmes) nécessaires à la résolution de problèmes simples.  **ACTDIP004** | Je peux créer des programmes simples et les déboguer.  **Key Stage 1 (1re et 2e années du primaire)** |
| Je peux développer des programmes avec des séquences et des boucles simples pour exprimer des idées ou régler un problème.  **1A-AP-10** | Je peux définir des problèmes simples, ainsi que décrire et suivre une séquence d'étapes et de décisions (algorithmes) nécessaires à leur résolution.  **ACTDIP010** | Je peux comprendre ce que sont les algorithmes, comment ils sont intégrés aux programmes des appareils numériques, et comment les programmes les exécutent en suivant des instructions claires et précises.  **Key Stage 1 (1re et 2e années du primaire)** |
| Je peux déboguer (déceler et réparer) les erreurs d'un algorithme ou d'un programme qui comprend des séquences et des boucles simples.  **1A-AP-14** | Je peux concevoir, modifier et suivre des algorithmes simples qui comprennent des séquences d'étapes, des embranchements et des itérations (répétitions).  **ACTDIP019** | Je peux utiliser un raisonnement logique pour prédire le comportement de programmes simples.  **Key Stage 1 (1re et 2e années du primaire)** |
| Je peux comparer la vie et le travail des gens avant et après l'implémentation ou l'adoption de nouvelles technologies informatiques.  **1A-IC-16** | Je peux concevoir des algorithmes représentés par diagrammes et en français, et tracer des algorithmes pour prédire la sortie d'une entrée donnée ou cibler des erreurs.  **ACTDIP029** | Je peux utiliser un raisonnement logique pour expliquer le fonctionnement de certains algorithmes simples, ainsi que pour détecter et corriger des erreurs dans les algorithmes et les programmes.  **Key Stage 2 (3e à 6e année du primaire)** |
| Je peux créer des programmes qui comprennent des séquences, des événements, des boucles et des conditionnelles.  **1B-AP-10** | Je peux définir des problèmes simples, ainsi que décrire et suivre une séquence d'étapes et de décisions (algorithmes) nécessaires à leur résolution.  **ACTDIP01** | Je peux utiliser la séquence, la sélection et la répétition dans des programmes, ainsi que travailler avec des variables et diverses formes d'entrée et de sortie.  **Key Stage 2 (3e à 6e année du primaire)** |
| Je peux comparer plusieurs algorithmes pour la même tâche, déterminer lequel est le plus approprié, et le peaufiner.  **1B-AP-08** | Je peux implémenter des solutions numériques simples, comme des programmes visuels, avec des algorithmes qui impliquent un embranchement (décisions) et une entrée utilisateur.  **ACTDIP011** | Je peux concevoir, rédiger et déboguer des programmes qui réalisent des tâches spécifiques, y compris le contrôle ou la simulation de systèmes physiques, ainsi que résoudre des problèmes en les décomposant en plus petites parties.  **Key Stage 2 (3e à 6e année du primaire)** |
| Je peux tester et déboguer (déceler et réparer les erreurs) un programme ou un algorithme afin d'assurer qu'il fonctionne comme prévu.  **1B-AP-15** | Je peux implémenter des solutions numériques, comme des programmes visuels simples qui impliquent un embranchement, une itération (répétition) et une entrée utilisateur.  **ACTDIP020** | Je peux comprendre plusieurs algorithmes clés qui reflètent la pensée informatique, et utiliser un raisonnement logique pour comparer l'utilité d'algorithmes alternatifs pour le même problème.  **Key Stage 3 (1re à 3e secondaire)** |
| Je peux discuter des technologies informatiques qui ont changé le monde et m'exprimer sur la manière dont ces technologies influencent – et sont influencées par – nos pratiques culturelles.  **1B-IC-20** | Je peux implémenter et modifier des programmes avec des interfaces utilisateur qui impliquent un embranchement, une itération et des fonctions dans un langage de programmation universel.  **ACTDIP030** | Je peux utiliser deux langages de programmation ou plus (dont au moins un qui est textuel) pour résoudre une variété de problèmes informatiques; j'utilise bien les structures de données; et je peux concevoir et développer des programmes modulaires qui emploient des procédures ou des fonctions.  **Key Stage 3 (1re à 3e secondaire)** |
| Je peux utiliser des organigrammes et/ou du pseudocode pour résoudre des problèmes complexes comme les algorithmes.    **2-AP-10** | Je peux recueillir, accéder et présenter différents types de données en utilisant un logiciel simple pour créer des informations et résoudre des problèmes.  **ACTDIP009** |  |
| Je peux concevoir et développer des programmes avec itération qui combinent des structures de commande, notamment des boucles imbriquées et des conditionnelles composées.  **2-AP-12** |  |  |
| Je peux décomposer des problèmes et des sous-problèmes en parties afin de faciliter la conception, l'implémentation et la vérification de programmes.  **2-AP-13** |  |  |

**Préparation à l'enseignement :**

* Installez Minecraft: Education Edition en vous rendant sur le site <https://aka.ms/HourofCode2021>.
* Regardez [la vidéo d'introduction](https://aka.ms/HOC2021Trailer) au sujet de l'Heure de Code de cette année.
* Regardez [la vidéo des procédures pas à pas](https://aka.ms/HOC2021Walkthrough) pour mieux comprendre la leçon et comment naviguer dans le monde Minecraft.
* Téléchargez le [guide de l'enseignant](file:///C:\Users\laylahbulman\Downloads\aka.ms\HOC2021educatorguide) qui est conçu pour vous fournir toutes les informations dont vous avez besoin pour aider vos élèves à profiter pleinement de cette Heure de Code, y compris toutes les ressources externes ainsi que des suggestions de directives pour des environnements d'apprentissage guidé, autonome et hybride/à distance.
* Utilisez les [diapositives de présentation](C:\\Users\\laylahbulman\\Downloads\\aka.ms\\HOC2021presentationslides) destinées aux élèves pour des directives détaillées.
* Téléchargez une copie des [solutions de code](https://aka.ms/HOC2021CodingSolutions).
* Vous avez d'autres questions? Consultez la [FAQ](https://aka.ms/HOC2021FAQ).

**Aperçu du sujet :**

En tant qu'informaticiens et informaticiennes de l'Institut des grandes erreurs temporelles, votre travail consiste à corriger les mystérieuses failles temporelles qui surviennent dans l'histoire et de trouver qui (ou quoi!) en est la cause.

Aiderez-vous à réparer les failles temporelles et à sauver le cours de l'histoire à l'aide de vos superpouvoirs de programmation?

Trouverez-vous la vérité au sujet de la cause de ces folles modifications de l'histoire?

Dans votre mission TimeCraft, vous devrez :

* remonter le temps jusqu'à de grands moments de l'histoire;
* programmer votre agent temporel pour réparer les failles temporelles;
* utiliser les indices pour trouver le malfaiteur (qui ou ce qui cause les failles temporelles).

Les élèves peuvent utiliser le langage Python ou du code à base de blocs.

**Objectifs d'apprentissage :**

* Comprendre l'importance et les bénéfices de l'informatique dans tous les aspects de la vie.
* Analyser et résoudre des problèmes en utilisant la pensée algorithmique et la décomposition des problèmes.
* S'exercer à utiliser des concepts informatiques comme les séquences, les événements, les boucles et le débogage.
* Créer des solutions codées pour réaliser une tâche ou résoudre un problème avec succès.
* Reconnaître le grand choix de carrières liées au domaine de l'informatique.

**Activités des élèves :**

**Activités d'introduction (5 minutes) :**

1. Les élèves réfléchissent aux trois questions suivantes :

* Qu'est-ce que l'informatique?
* Comment l'informatique (ou les compétences en informatique) est-elle utilisée à l'école?
* Comment l'informatique est-elle utilisée en milieu de travail (dans différents domaines d'emploi)?

1. Les élèves consultent les termes clés du glossaire qui les aideront à mieux comprendre cette expérience d'apprentissage.

**Activités de codage (30 à 40 minutes) :**

Les élèves commencent leur périple de programmation au portail d'Éphémère Un, le principal vaisseau spatial de l'Institut de surveillance des erreurs temporelles. Ils sont des informaticiens que IARRAT attend avec impatience. Cette intelligence artificielle (IA) gère le vaisseau et ses agents temporels, des robots qui voyagent dans le temps vers différents moments de l'histoire afin de réparer les failles temporelles à l'aide du code.

Les élèves auront à sélectionner entre le langage Python ou la programmation par blocs pour leurs activités de codage. Ils doivent ensuite sélectionner leur agent temporel. Nous recommandons aux débutants de commencer avec le codage par blocs.

Après avoir sélectionné leur agent, les élèves relèveront deux défis de codage intégrés.

DÉFIS DE CODAGE INTÉGRÉS

Défi 1 : Appelez votre agent temporel. Obtenez votre appareil de communication Di.RA de IARRAT et appelez votre agent temporel à votre lieu.

Défi 2 : Déplacez l'agent. Faites avancer votre agent de façon à ce qu’il monte sur le bloc vert.

DÉROULEMENT DES DÉFIS DE CODAGE

Les élèves seront dirigés vers le premier défi, le Grand orchestre de jazz, qui est une expérience de programmation guidée. Les élèves recevront deux indices pour résoudre chacune des activités de codage avant que celles-ci ne se résolvent d'elles-mêmes. Chaque expérience de programmation est suivie d'une brève recherche d'indices pour trouver le malfaiteur.

Après avoir terminé la première faille temporelle, le Grand orchestre de jazz, les élèves sont téléportés au lobby pour sélectionner un autre défi. Les élèves peuvent choisir n'importe quelle faille temporelle, mais il est recommandé de suivre les failles temporelles affichées sur le grand écran au-dessus de la chronologie.

|  |  |
| --- | --- |
| **Faille temporelle 1**  Grand orchestre de jazz  (C'est le défi guidé.) | Le grand musicien de jazz a perdu sa trompette bien-aimée et l'a remplacée par un gazou! Programmez votre agent temporel pour parcourir le labyrinthe, retrouver la trompette du musicien et sauver le jazz. |
| **Faille temporelle 2**  Les Grandes Pyramides de Gizeh | Les Grandes Pyramides sont maintenant des cubes!  Programmez votre agent temporel pour aider le concepteur à créer une structure sécuritaire qui perdurera des milliers d'années... la pyramide. |
| **Faille temporelle 3**  Mission lunaire | Les astronautes ont besoin de votre aide. Utilisez votre agent temporel pour réaliser un labyrinthe de conviction et remettez les calculs aux astronautes pour les aider à alunir! |
| **Faille temporelle 4**  La Grande Muraille de Chine | La Grande Muraille n'est pas si grande que ça... En fait, elle est très courte!  Les pandas affamés n'arrêtent pas de manger l'échafaudage en bambou, alors les travailleurs ne peuvent pas construire la grande muraille qu'ils souhaitent. Programmez un jardin de bambou pour rediriger les pandas affamés! |
| **Faille temporelle 5**  La Joconde | La Joconde ne sourit plus. En fait, elle est très contrariée. Le malfaiteur du temps a remonté le temps pour piétiner le jardin de la Joconde. Son célèbre sourire n'est plus. Programmez votre agent temporel pour lui redonner le sourire en plantant des fleurs. |
| **Faille temporelle 6**  Premiers vols | Le malfaiteur a fait plein de trous dans la piste de l'aéroport. Au lieu d'avoir des avions, nous n'avons que des montgolfières. Programmez votre agent temporel pour réparer la piste afin que les inventeurs puissent faire leur premier vol! |
| **Faille temporelle 7**  Premier programme informatique | Le premier programme informatique a joué une chanson, mais le malfaiteur a ruiné le code. Utilisez votre agent temporel pour réparer le code afin que joue la musique! |
| **Faille temporelle 8**  Meilleur ami de l'homme | Le meilleur ami de l'homme n'est plus le chien, mais bien l'ours! Utilisez votre agent temporel pour apprivoiser un chiot et mener le chiot vers les humains pour qu'ils soient à nouveau les meilleurs amis. |
| **Faille temporelle 9**  Casse-tête paléontologique | Oh non! Le malfaiteur a remonté le temps pour voler des fossiles, ce qui a fait en sorte que le brachiosaure a un petit cou. Utilisez votre agent temporel pour remplacer les fossiles dérobés afin que les paléontologues puissent reconstruire le brachiosaure avec un long cou! |
| **Faille temporelle 10**  Éléments de découverte | Le malfaiteur s'est faufilé à l'intérieur du labo de sciences et a caché certains des éléments. Utilisez votre agent temporel pour trouver les éléments cachés afin que les scientifiques puissent faire de grandes découvertes! |

Après avoir terminé les trois activités de codage, les élèves auront résolu la première boucle du jeu et trouvé le malfaiteur. Ils seront en mesure de prolonger leurs activités de codage en sélectionnant un nouvel agent temporel et en programmant avec de nouvelles failles temporelles.

**Évaluation :**

* Quelle partie de l'Heure de Code avez-vous préférée?
* Quelle partie de l'Heure de Code était la plus difficile?
* Comment avez-vous utilisé les compétences en informatique aujourd'hui?
* Qu'avez-vous appris de nouveau aujourd'hui?
* Pourquoi l'informatique est-elle importante pour tout le monde?
* Aimeriez-vous essayer Minecraft: Education Edition à nouveau?

**Différenciation :**

Afin de favoriser l'accès des élèves à l'Heure de Code, trois expériences d'apprentissage différentes ont été mises en place : en classe avec animation de l'enseignant, en classe avec une expérience autoguidée et en mode d'apprentissage à distance (virtuel). Chacune de ces expériences d'apprentissage comporte différents niveaux de soutien et de modification pour l'enseignant afin d'assurer la réussite et la participation des élèves à l'Heure de Code.

**Animation par l'enseignant (en classe)**

***Avec ce type d'expérience d'apprentissage, l'enseignant offrira aux élèves le niveau de soutien le plus élevé***. En tant qu'enseignant, vous fournirez un modèle et des directives explicites avant de laisser vos élèves travailler de manière autonome sur les activités de codage de l'Heure de Code. Vous utiliserez la [présentation de l'Heure de Code](https://aka.ms/HOC2021Presentation) pour vous guider, vos élèves et vous, dans le début du jeu, les activités de codage et les réflexions sur la leçon.

**Élève autoguidé**

***Avec ce type d'expérience d'apprentissage, l'enseignant offrira aux élèves le niveau de soutien le moins élevé***. En tant qu'enseignant, vous vous assurerez que vos élèves puissent se connecter à la plateforme Minecraft: Education Edition. Vous fournirez à vos élèves la présentation de l'Heure de Code qui les aidera à réaliser ces activités à leur rythme, de manière autoguidée. La présentation prodiguera aux élèves des directives explicites sur la manière de participer à l'Heure de Code. Vous devriez également donner aux élèves une copie du glossaire visuel que vous trouverez à la fin de ce guide de l'enseignant. Il est recommandé d'offrir ce matériel aux élèves au moyen du support numérique de votre choix ou sur papier pour qu'il soit immédiatement prêt à être utilisé par les élèves.

**Expérience d'apprentissage à distance**

Ce type d'expérience d'apprentissage peut être présenté de manière synchrone ou asynchrone.

***Si vous avez des élèves qui participent à une session virtuelle synchrone (en simultané)***, vous présenterez cette session comme vous le feriez avec une expérience en classe (avec animation de l'enseignant). En tant qu'enseignant, vous fournirez un modèle et des directives explicites avant de laisser vos élèves travailler de manière autonome sur les activités de codage de l'Heure de Code. Vous utiliserez la présentation de l'Heure de Code pour vous guider, vos élèves et vous, dans le début du jeu, les activités de codage et les réflexions sur la leçon. Assurez-vous que votre plateforme de communication virtuelle (comme Microsoft Teams) est configurée pour partager du contenu et que le son est activé.

***Si vous avez des élèves qui participent à une session virtuelle asynchrone (en différé)***, vous présenterez cette session comme vous le feriez avec une expérience autoguidée en classe. En tant qu'enseignant, vous vous assurerez que vos élèves ont accès à la plateforme Minecraft: Education Edition avant la leçon. Vous devrez aussi fournir à vos élèves la présentation de l'Heure de Code qui les aidera à réaliser ces activités à leur rythme, de manière autoguidée. La présentation prodiguera aux élèves des directives explicites sur la manière de participer à l'Heure de Code. Vous devriez également donner aux élèves une copie du glossaire visuel que vous trouverez à la fin de ce guide de l'enseignant. Il est recommandé d'offrir ce matériel (la présentation et le glossaire) aux élèves au moyen du support numérique de votre choix (par exemple un canal Microsoft Teams, un bloc-notes OneNote, votre système de gestion des apprentissages, etc.).

Si vous prévoyez offrir cette leçon à distance, veuillez considérer ces quelques conseils :

1. Cette leçon n’est pas prévue pour le jeu multijoueur. Chaque élève devrait jouer dans sa propre version du monde.

2. Divisez les élèves par paires ou en petits groupes (par exemple avec les salles de petits groupes) pour qu’ils puissent s’entraider à résoudre les activités de codage.

3. Sachant que le niveau de familiarité avec Minecraft et avec la programmation peut varier d'un élève à l'autre, nous suggérons d’assigner un chef de groupe qui pourra aider ceux qui restent bloqués.

Pour plus d’informations sur l’apprentissage à distance dans Minecraft: Education Edition, consultez la page <https://aka.ms/remote-learning-kit>