A picture containing toy

Description automatically generated

**PLAN DE LEÇON**

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

**Âges :** %7+

**Domaines :** Informatique, maths, arts du langage, histoire, arts visuels, science, ingénierie

Compétences du **21e siècle :** créativité, collaboration, communication et pensée critique

**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.

**Multijoueur / un seul joueur :** : un seul joueur

**Niveau d'expérience de Minecraft requis (Enseignant) :** Débutant

**Description de la leçon :** Tutoriel d'introduction à l'informatique amusant et créatif. Voyagez dans le temps et utilisez du code basé sur les blocs ou en Python pour résoudre de mystérieux incidents dans l'histoire de la Terre. Rassemblez des indices et faites preuve de pensée critique pour identifier le coupable dans cette aventure à travers le temps. Les élèves découvrent également l'intégration de l'informatique dans la vie professionnelle et les centres d'intérêt personnels.

**Temps total nécessaire :** 1 heure

**Normes d'enseignement :**

|  |  |  |
| --- | --- | --- |
| Normes | | |
| États-Unis | Australie | Royaume-Uni - Angleterre |
| Je suis capable de modéliser des processus quotidiens en créant et suivant des algorithmes pour réaliser des tâches.  **1A-AP-08** | Je suis capable de suivre, décrire et représenter une séquence d'étapes et de décisions (algorithmes) nécessaires à la résolution de problèmes simples.  **ACTDIP004** | Je suis capable de créer et déboguer des programmes simples.  **Étape clé 1** |
| Je suis capable de développer des programmes qui comprennent des séquences et des boucles simples afin d'exprimer des idées ou de résoudre un problème.  **1A-AP-10** | Je suis capable de suivre, décrire et représenter une séquence d'étapes et de décisions (algorithmes) nécessaires à la résolution de problèmes simples.  **ACTDIP010** | Je comprends ce que sont les algorithmes et comment ils sont utilisés en tant que programmes sur les appareils numériques. Je sais que les programmes s'exécutent en suivant des instructions précises et sans ambiguïté.  **Étape clé 1** |
| Déboguer (identifier et corriger) les erreurs d'un algorithme ou un programme qui inclut des séquences et des boucles simples.  **1A-AP-14** | Concevoir, modifier et suivre des algorithmes simples intégrant des enchaînements d'étapes, des arborescences, et itérations (répétitions)  **ACTDIP019** | Je suis capable d'utiliser un raisonnement logique pour prédire le comportement de programmes simples.  **Étape clé 1** |
| Je suis capable de comparer les modes de vie et de travail avant et après la mise en œuvre ou l'adoption de nouvelles technologies informatiques.  **1A-IC-16** | Concevoir des algorithmes représentés sous forme de diagrammes et en anglais, et tracer des algorithmes afin de prédire la sortie pour une entrée donnée et d'identifier des erreurs  **ACTDIP029** | Je suis capable d'utiliser un raisonnement logique pour expliquer le fonctionnement d'algorithmes simples et détecter et corriger les erreurs dans des algorithmes et programmes.  **Étape clé 2** |
| Je suis capable de créer des programmes qui comprennent des séquences, des événements, des boucles et des conditions.  **1B-AP-10** | Je suis capable de suivre, décrire et représenter une séquence d'étapes et de décisions (algorithmes) nécessaires à la résolution de problèmes simples.  **ACTDIP01** | Je suis capable d'utiliser des séquences, sections et répétitions dans des programmes, de travailler avec des variables et différentes formes d'entrée et de sortie.  **Étape clé 2** |
| Je suis capable de comparer et perfectionner plusieurs algorithmes pour une même tâche et de déterminer lequel est le plus approprié.  **1B-AP-08** | Je suis capable de mettre en œuvre des solutions numériques simples sous la forme de programmes visuels avec des algorithmes, intégrant du branchement (décisions) et une entrée utilisateur.  **ACTDIP011** | Je suis capable de concevoir, écrire et déboguer des programmes qui accomplissent des objectifs précis, comme contrôler ou simuler des systèmes physiques. Je peux résoudre des problèmes en les décomposant en parties plus petites.  **Étape clé 2** |
| Tester et déboguer (identifier et corriger) les erreurs d'un algorithme ou d'un programme afin de s'assurer qu'il fonctionne correctement.  **1B-AP-15** | Je suis capable de mettre en œuvre des solutions numériques simples sous la forme de programmes visuels avec des algorithmes, intégrant du branchement (décisions), de l'itération (répétition) et une entrée utilisateur.  **ACTDIP020** | Je suis capable de comprendre plusieurs algorithmes fondamentaux représentatifs de la pensée computationnelle. Je suis capable d'utiliser un raisonnement logique pour comparer l'utilité d'algorithmes alternatifs pour résoudre le même problème.  **Étape clé 3** |
| Je suis capable de discuter des technologies informatiques qui ont changé le monde et de présenter la manière dont ces technologies influencent et sont influencées par les pratiques culturelles.  **1B-IC-20** | Je suis capable de mettre en œuvre et modifier des solutions numériques simples sous la forme de programmes visuels avec des algorithmes, intégrant du branchement, de l'itération et des fonctions dans un langage de programmation généraliste.  **ACTDIP030** | Je suis capable d'utiliser au moins 2 langages de programmation, dont l'un au moins est textuel, pour résoudre divers problèmes computationnels ; de faire un usage approprié des structures de données ; de concevoir et développer des programmes modulaires utilisant des procédures ou des fonctions.  **Étape clé 3** |
| Je suis capable d'utiliser des organigrammes et/ou du pseudo-code afin de résoudre des problèmes complexes en tant qu'algorithmes.    **2-AP-10** | Je suis capable de collecter, d'accéder à et de présenter différents types de données en utilisant des logiciels simples pour créer des informations et résoudre des problèmes.  **ACTDIP009** |  |
| Je suis capable de concevoir et de développer de façon itérative des programmes combinant des structures de contrôle, dont des boucles imbriquées et des instructions conditionnelles.  **2-AP-12** |  |  |
| Je suis capable de décomposer des problèmes et des sous-problèmes en plusieurs parties afin de faciliter la conception, la mise en œuvre et l'examen des programmes.  **2-AP-13** |  |  |

**Préparation de l'enseignant :**

* Installez Minecraft: Education Edition en vous rendant sur <https://aka.ms/HourofCode2021>
* Regardez [la vidéo d'introduction](https://aka.ms/HOC2021Trailer) du thème d'Heure de Code de cette année.
* Regardez [la vidéo de présentation](https://aka.ms/HOC2021Walkthrough) pour mieux comprendre comment commencer la leçon et naviguer dans le monde de Minecraft.
* Téléchargez le [Guide de l'enseignant](file:///C:\Users\laylahbulman\Downloads\aka.ms\HOC2021educatorguide), conçu pour vous fournir toutes les informations dont vous avez besoin pour aider vos élèves à s'amuser et à réussir avec Heure de code, notamment toutes les ressources externes et des suggestions d'instructions pour environnements d'apprentissage guidé, indépendant et hybrides/à distance.
* Utilisez les [diapositives de présentation](file:///C:\Users\laylahbulman\Downloads\aka.ms\HOC2021presentationslides) face aux élèves pour obtenir des instructions pas à pas.
* Téléchargez un exemplaire des [solutions de programmation](https://aka.ms/HOC2021CodingSolutions).
* Vous avez d'autres questions ? Découvrez la [FAQ](https://aka.ms/HOC2021FAQ).

**Aperçu du thème :**

En tant qu'informaticien à l'Institut des Erreurs temporelles majeures, votre mission consiste à corriger les mystérieuses brèches qui font irruption dans le cours de l'histoire et de découvrir qui en est la cause (ou quoi !).

Parviendrez-vous à corriger les brèches et à sauver le cours de l'histoire à l'aide de vos superpouvoirs de codage ?

Découvrirez-vous la vérité sur la personne ou la chose à l'origine de ces altérations délirantes de l'histoire ?

Dans votre mission TimeCraft, vous devrez :

* Voyager dans le passé jusqu'à des moments passionnants de l'histoire mondiale.
* Programmer votre Agent pour corriger les brèches.
* Utiliser des indices pour identifier le coupable (la personne ou chose à l'origine des brèches)

Les élèvent peuvent utiliser du code Python ou basé sur les blocs.

**Objectifs d'apprentissage :**

* Comprendre l'importance et les avantages de l'informatique dans tous les aspects de la vie.
* Analyser et résoudre des problèmes à l'aide de raisonnement algorithmique et en décomposant les problèmes.
* Appliquer des concepts d'informatique comme les séquences, les événements, les boucles et le débogage.
* Créer des solutions de programmation pour accomplir une tâche ou résoudre un problème.
* Comprendre que l'informatique offre des opportunités de carrière supplémentaires.

**Activités des élèves :**

**Activités de présentation (5 minutes) :**

1. Les élèves sont amenés à réfléchir aux trois questions suivantes :

* Qu'est-ce que l'informatique ?
* Comment sont utilisées l'informatique et les compétences informatiques à l'école ?
* Comment est utilisée l'informatique au travail (dans différents métiers) ?

1. Les élèves découvriront différents termes de vocabulaire fondamentaux qui les aideront dans leur compréhension de l'expérience d'apprentissage.

**Activités de programmation (30-40 minutes)**

Les élèves démarrent leur découverte de la programmation au portail vers l'Ephemera One, le vaisseau spatial principal de l'Institut de surveillance des erreurs temporelles. Ils incarnent des informaticiens, impatiemment attendus par TARRA, le robot à intelligence artificielle (IA) qui gère le vaisseau et ses Agents temporels, des robots capables de voyager dans le temps jusqu'à différents moments de l'histoire pour corriger les brèches à l'aide de la programmation.

Les élèves devront choisir entre les blocs et le Python pour les activités de programmation, puis choisir leur Agent temporel. Nous recommandons aux programmeurs débutants de commencer par les Blocs.

Après avoir choisi leur agent, les élèves résoudront deux défis de programmation introductifs.

DÉFIS DE PROGRAMMATION D'INTRODUCTION

Défi 1 : Appeler votre Agent. Utilisez le communicateur TALK que vous donne TARRA pour faire venir votre Agent temporel à votre emplacement.

Défi 2 : Déplacer l'Agent. Faites avancer votre Agent jusqu'au bloc vert.

DÉFIS DE PROGRAMMATION HISTORIQUES

Les élèves seront orientés vers le premier défi, Orchestre de jazz, une expérience de programmation guidée. Les élèves auront deux indices pour résoudre chaque activité de programmation avant qu'elle ne soit résolue automatiquement. Chaque expérience de programmation sera suivie d'une rapide recherche d'indices pour identifier le coupable.

Après avoir terminé la brèche 1 : Orchestre de jazz, les élèves sont téléportés au lobby principal où ils pourront choisir un nouveau défi. Les élèves peuvent sélectionner n'importe quelle brèche, mais il leur est conseillé de suivre l'ordre des brèches indiqué sur le grand écran au-dessus de l'historique.

|  |  |
| --- | --- |
| **Brèche 1**  Orchestre de jazz  (il s'agit du défi guidé) | Un grand musicien de jazz a perdu sa trompette préférée, remplacée par un kazoo ! Programmez votre Agent pour traverser le labyrinthe, retrouver la trompette du musicien et ainsi sauver le jazz. |
| **Brèche 2**  Les grandes pyramides de Gizeh | Les grandes pyramides de Gizeh sont maintenant des cubes !  Programmez votre Agent pour aider le concepteur à créer une structure stable qui durera des milliers d'années... la pyramide. |
| **Brèche 3**  Mission lunaire | Les astronautes ont besoin de votre aide. Utilisez votre Agent pour arriver au bout d'un labyrinthe et en rapporter les calculs qui aideront les astronautes à alunir ! |
| **Brèche 4**  La Grande Muraille de Chine | La Grande Muraille n'est pas encore si grande que ça... Elle est même très petite !  Des pandas affamés n'arrêtent pas de dévorer les échafaudages en bambou, et à cause de cela, les ouvriers ne peuvent pas construire une muraille aussi longue et haute que souhaité. Programmez une bambouseraie pour détourner l'attention des pandas affamés ! |
| **Brèche 5**  Mona Lisa | Mona Lisa n'a plus le sourire. Elle est même très contrariée. Le Coupable temporel est retourné dans le passé pour piétiner le jardin de Mona Lisa, transformant son célèbre sourire en grimace. Programmez votre Agent pour lui remonter le moral en replantant des fleurs. |
| **Brèche 6**  Premiers vols | Le Coupable a fait de nombreux trous sur la piste de décollage. Au lieu d'avoir des avions, nous n'avons que des montgolfières. Programmez votre Agent pour réparer la piste, afin que les inventeurs puissent réaliser leur premier vol ! |
| **Brèche 7**  La première informaticienne | Le premier programme informatique jouait une chanson, mais le coupable a endommagé son code. Utilisez votre Agent pour réparer le code et entendre la musique ! |
| **Brèche 8**  Le meilleur ami des humains | Le meilleur ami des humains n'est plus le chien, mais un ours ! Utilisez votre agent pour se lier d'amitié avec un chiot et le ramener aux humains pour qu'ils puissent redevenir les meilleurs amis du monde. |
| **Brèche 9**  Énigme paléontologique | Oh non ! Le Coupable est retourné dans le passé pour voler des fossiles et donner au brachiosaure un cou très court. Utilisez votre Agent pour remplacer les fossiles volés, permettant aux paléontologues de reconstituer un brachiosaure avec son long cou ! |
| **Brèche 10**  Éléments de découverte | Le Coupable s'est introduit dans le laboratoire et a caché certains éléments. Utilisez votre Agent pour retrouver les éléments cachés, afin de permettre aux scientifiques de faire une grande découverte ! |

Après avoir réussi trois activités de programmation, les élèves auront résolu la première boucle du jeu et identifié le coupable. Ils peuvent poursuivre leurs activités de programmation en choisissant un nouvel agent et en programmant des brèches supplémentaires.

**Évaluation :**

* Quelle partie d'Heure de Code avez-vous préférée ?
* Quelle partie d'Heure de Code vous a semblé la plus difficile ?
* Comment avez-vous utilisé vos compétences d'informatique aujourd'hui ?
* Qu'avez-vous appris de nouveau aujourd'hui ?
* Pourquoi l'informatique est-elle aussi importante pour tous ?
* Voudriez-vous réessayer Minecraft: Education Edition ?

**Différentiation :**

Afin de faciliter l'accès des élèves à Heure de Code, trois expériences d'apprentissage différentes leur sont proposées : en classe avec un enseignant, en classe en autonomie ou en distanciel (virtuel). Chacune des expériences d'apprentissage dispose de différents niveaux de modifications et d'aide aux enseignants afin d'aider les élèves à participer et réussir à Heure de Code.

**Animé par l'enseignant (en face à face)**

***Avec ce type d'apprentissage, les élèves bénéficient d'un accompagnement maximal de leur enseignant***. En tant qu'enseignant, vous donnerez aux élèves des instructions et une modélisation explicites avant de les laisser travailler en autonomie sur les différentes activités de programmation d'Heure de Code. Vous utiliserez la [Présentation d'Heure de Code](https://aka.ms/HOC2021Presentation) comme un guide des premiers pas dans le jeu, des activités de programmation et de réflexion sur les leçons, pour les élèves et vous-même.

**Élèves en autonomie**

***Avec ce type d'apprentissage, les élèves bénéficient d'un accompagnement minimal de l'enseignant***. En tant qu'enseignant, vous vous assurerez que les élèves parviennent à se connecter à la plateforme Minecraft: Education Edition. Vous fournirez aux élèves la Présentation d'Heure de code, qui les aidera à faire l'expérience du jeu à leur rythme et en autonomie. La présentation donnera aux élèves des instructions explicites pour participer à Heure de Code. Il vous est également conseillé de leur procurer un exemplaire du Glossaire visuel, situé à la fin de ce Guide de l'enseignant. Vous pouvez le leur fournir sous une forme numérisée de votre choix ou en imprimer des exemplaires sur papier pour qu'ils l'aient toujours sous la main.

**Enseignement à distance**

Ce type d'expérience d'apprentissage peut-être accompagné de façon synchrone ou asynchrone.

***Si vos élèves participent à une session virtuelle synchrone (en direct)***, vous conduirez la session exactement comme en classe (avec accompagnement de l'enseignant). En tant qu'enseignant, vous donnerez aux élèves des instructions et une modélisation explicites avant de les laisser travailler en autonomie sur les différentes activités de programmation d'Heure de Code. Vous utiliserez la Présentation d'Heure de Code comme un guide des premiers pas dans le jeu, des activités de programmation et de réflexion sur les leçons, pour les élèves et vous-même. Assurez-vous que votre plateforme de communication virtuelle (p. ex. Microsoft Team) est prête pour le partage de contenu, et que le son est activé.

***Si vos élèves participent à une session virtuelle asynchrone (pas en direct)***, vous conduirez la session exactement comme pour l'expérience en autonomie. En tant qu'enseignant, vous vous assurerez que les élèves ont accès avant la leçon à la plateforme Minecraft: Education Edition. Vous fournirez également aux élèves la Présentation d'Heure de code, qui les aidera à faire l'expérience du jeu à leur rythme et en autonomie. La présentation donnera aux élèves des instructions explicites pour participer à Heure de Code. Il vous est également conseillé de leur procurer un exemplaire du Glossaire visuel, situé à la fin de ce Guide de l'enseignant. Vous devriez leur fournir ces documents (présentation et glossaire) sous une forme numérisée de votre choix (p. ex. un canal Microsoft Teams, un Bloc-notes OneNote, votre système de gestion des apprentissages, etc.).

Si vous envisagez de proposer cette leçon par le biais de l'enseignement à distance, veuillez tenir compte des conseils suivants :

1. Cette leçon n'est pas conçue comme une expérience multijoueur. Chaque élève doit travailler dans sa propre version du monde.

2. Répartissez les élèves par paires ou en petits groupes dans des salles de réunion afin qu'ils puissent s'entraider pour résoudre les activités de programmation.

3. Étant donné que les élèves sont susceptibles d'avoir différents niveaux de familiarité avec Minecraft et la programmation, il peut être utile d'affecter des chefs de groupe d'élèves pour aider leurs camarades qui pourraient être bloqués.

Pour plus d'informations sur l'enseignement à distance dans Minecraft: Education Edition, veuillez vous référer à <https://aka.ms/remote-learning-kit>