

## Découvrez ou illustrez des concepts de programmation avec un jeu de puzzle dystopique

Étapes préalables obligatoires / Connaissances préalables des élèves	Bases de codage ou une bonne supervision de l'enseignant requise
Objectifs d'apprentissage	Découvrez les premiers pas de la programmation ou complétez un cours sur les concepts de codage avec les niveaux les plus difficiles.
Matières	Littératie numérique, programmation, maths, anglais langue étrangère
Âge recommandé	15 - 18
Matériel nécessaire	Jeu : 7 billion humans (Windows, MacOs, Linux, AppStore, GooglePlay)
Durée de la séquence	60 minutes
Activité individuelle ou collective	Activité collective
Compétences développées	Résolution de problème, planification
Fourchette de prix du jeu	<15 €, <6 € (application mobile)
Jeux similaires pouvant être utilisés pour cette séquence	Human Resource Machine ( <a href="https://tomorrowcorporation.com/humanresource-machine">https://tomorrowcorporation.com/humanresource-machine</a> )



Conseils pour une séquence plus courte	Pour raccourcir cette séquence, vous pouvez parcourir les premiers niveaux vous-même et vous faire une idée précise des concepts de jeu que vous souhaitez montrer à vos élèves. Ensuite, au lieu de passer ces niveaux avec eux, vous pouvez commencer un de niveau supérieur et leur expliquer tous les concepts par essais et erreurs ("comment pouvons-nous faire ceci ?") puis, en utilisant la fonction de débogage, vous pouvez leur montrer leurs erreurs et essayer de les corriger jusqu'à ce qu'ils réussissent.
Conseils sur l'accessibilité et l'inclusivité de la séquence	Le jeu n'est pas disponible en grec. Le jeu ne comporte pas beaucoup de vocabulaire en anglais. Vous devez donner les explications vous-même au lieu de lire les textes en anglais à vos élèves. Vous devez également expliquer le sens des commandes ("pickup"...).

## Étape par étape : comment mettre en pratique la séquence

Dans cette séquence, vous allez jouer à un jeu de programmation avec votre classe. Le jeu utilise un humour noir et un décor mignon mais dystopique, ce qui en fait une satire légère du travail de bureau moderne. En progressant dans des niveaux de difficulté croissante, vous découvrirez de nouveaux concepts de programmation. Les premiers niveaux sont assez simples, mais vous devrez superviser et donner plus de conseils pour les niveaux suivants.



Il est préférable de préparer la session en jouant seul aux 10 premiers niveaux. Cela vous permettra de passer à des niveaux plus difficiles et d'éviter les explications de jeu si les compétences de vos élèves sont suffisantes.

- **Étape 1 : Présentation du jeu et de la séquence (5 minutes)**

Dites à vos élèves que vous allez jouer à un jeu de puzzle afin d'aborder les principes de programmation de manière ludique. Si vous avez déjà présenté des concepts de programmation à votre classe, utilisez ce jeu pour les illustrer de manière amusante. Lancez et démarrez le jeu. Il vous demandera de sélectionner un identifiant d'employé (c'est-à-dire un emplacement de sauvegarde du jeu) et un avatar. Vous pouvez passer rapidement ce processus.

Une séquence cinématique est jouée, présentant l'univers satirique du jeu, suivie de l'écran de sélection des niveaux.

Choisissez le niveau suivant (« Bienvenue aux nouveaux employés »).

- **Étape 2 : Démonstration des mécanismes du jeu (10 minutes)**

Expliquez l'interface et les mécanismes du jeu en parcourant les 3 premiers niveaux. Pour cette étape, effectuez vous-même les actions et commentez-les. Assurez-vous que vos élèves comprennent les mécanismes du jeu.

## Analyse de l'interface :

Les petites notes en jaune expliquent votre objectif :



Capture d'écran en jeu : mécanique de base étape 1

La boîte avec les commandes de programmation se trouve juste à côté.

Lorsque vous déposez des commandes dans la fenêtre située en dessous, vous pouvez être invité à choisir certains paramètres :



Capture d'écran en jeu : mécanismes de base étape 2

Chaque étape-commande correspond à une tuile de plancher dans la salle des travailleurs.



Exécutez votre programme en cliquant sur le bouton de lecture dans le coin inférieur gauche :



Capture d'écran en jeu : exécution du programme

Le niveau 4 (« Livraison à distance ») présente le concept des boucles :



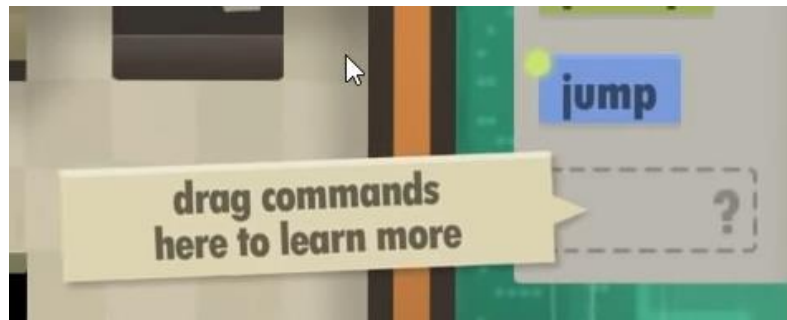
Capture d'écran en jeu : introduction des boucles

Jouez avec votre classe de manière interactive à partir du niveau 4 (étape suivante).

- **Étape 3 : Passez des niveaux avec votre classe (35 minutes)**

Pour chaque niveau :

- Indiquer clairement l'objectif du niveau
- Expliquez les nouvelles commandes que le jeu vous permet de faire au fur et à mesure de votre progression :
  - Expliquer leur fonctionnement en utilisant la fente d'aide en bas de la liste des commandes



- Les combiner avec d'autres commandes pour démontrer leur fonctionnement en exécutant un code de test inachevé.
- Donnez à vos élèves quelques minutes pour réfléchir à une solution, puis demandez qui veut faire un essai et mettez en œuvre la solution proposée.
  - Si ça marche, tant mieux ! Assurez-vous que tout le monde a compris avant de continuer.
  - Si ça ne marche pas, passez en mode débogage et essayez de comprendre ce qui n'a pas fonctionné.

Le processus de débogage est expliqué au niveau 6 (« petit exterminateur 1 »). Vous pouvez exécuter votre code ligne par ligne en utilisant le bouton flèche :





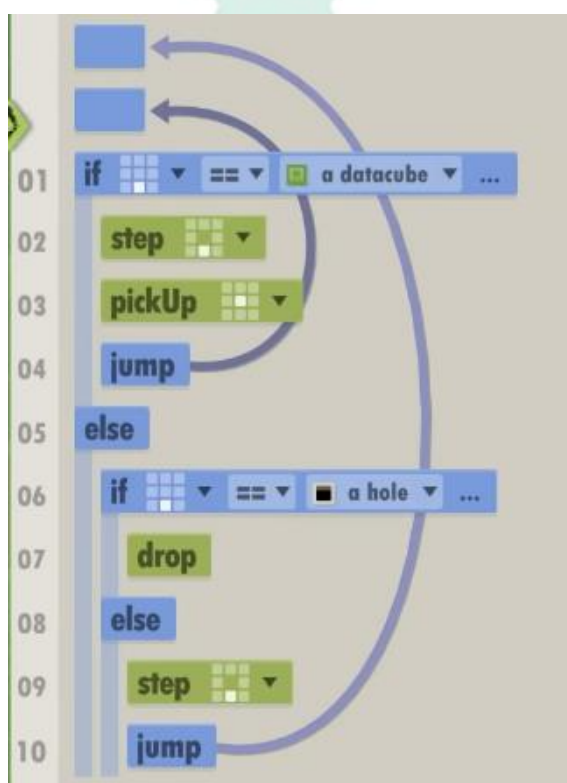
Si vos élèves ont des difficultés, décomposez les étapes pour eux et demandez-leur quelle commande doit être utilisée pour chaque étape.

S'ils sont totalement bloqués, donnez la réponse. Attendre trop longtemps tuerait le dynamisme de la session. Le plaisir de recevoir la solution d'un problème peut être aussi agréable que de le résoudre soi-même (pensez aux romans policiers). Cependant, assurez-vous que les élèves comprennent parfaitement la solution.

- **Étape 4 : Conclusion et ouverture (10 minutes)**

Expliquer le concept de boîte noire en programmation : il n'est pas nécessaire de connaître le fonctionnement interne d'une fonction pour l'utiliser.

Comparez un algorithme en Python avec un algorithme extrait du jeu.



```
x = 0
y = 0
tile = workspace.get(x,y)
character = Character(tile, workspace)
```

```
while(True):
    if(isinstance(workspace.get(character.getCurrentTile().getBottom()), "Datacube")):
        character.step(character.getCurrentTile().getBottom())
        character.pickup()

    else:
        if(isinstance(workspace.get(character.getCurrentTile().getBottom()), "Hole")):
            character.drop()
        else:
            character.step(character.getCurrentTile().getBottom())
```

Il s'agit de différentes représentations (langages) du même algorithme. Nous ne connaissons pas le code à l'intérieur des fonctions, mais il nous suffit de savoir quelle action elles effectuent. De cette façon, nous pouvons les arranger pour produire un algorithme qui répond à nos besoins.



## Bibliographie :

Obtenir le jeu

[https://store.steampowered.com/app/792100/7\\_Billion\\_Humans/](https://store.steampowered.com/app/792100/7_Billion_Humans/)

<https://www.epicgames.com/store/fr/p/7-billion-humans>

<https://apps.apple.com/fr/app/7-billion-humans/id1393923918>

<https://play.google.com/store/apps/details?id=com.tomorrowcorporation.sevenbillionhumans>

## Information

Guevara, W. G. (2018, November 29). What is “black-box code” and why it’s important. thatsoftwaredude.com Retrieved

from <https://www.thatsoftwaredude.com/content/8881/what-is-black-box-code-and-why-its-important>

Bristol, J. B. (2019). 7 Billion Humans: Amusing puzzler challenges kids, teaches programming principles. commonsense.org Retrieved

from <https://www.commonsense.org/education/app/7-billion-humans>

Toutes les captures d’écrans utilisées dans cette séquence sont extraites de 7 Billion Human, Tomorrow Corporation (2018)

