

Apprendre la logique algorithmique et la programmation de base en jouant

Étapes préalables obligatoires / Connaissances préalables des élèves	Aucune
Objectifs d'apprentissage	Familiariser les étudiants avec la logique algorithmique. Apprendre à utiliser les instructions de base du flux de contrôle : IF, IF-else (SI, SI-sinon) dans les puzzles.
Matières	Programmation
Âge recommandé	10 - 14
Matériel nécessaire	Ordinateurs qui peuvent exécuter "Robots: create AI"
Durée de la séquence	90 minutes
Activité individuelle ou collective	Single Game Player en activité de groupe
Compétences développées	Coopération et travail d'équipe, communication, créativité, apprendre à apprendre, résolution de problème, planification
Fourchette du prix du jeu	0.99
Jeux similaires à utiliser pour cette séquence	Scratch, EXAPUNKS, 7 billion humans
Conseils pour une séquence plus courte	L'étape 3 peut être réalisée à la maison

Conseils sur l'accessibilité et l'inclusivité de la séquence

Aucun son n'est utilisé. La résolution de l'écran peut être modifiée pour les personnes ayant des problèmes de vue mineurs.

Étape par étape : comment mettre en œuvre la séquence

Dans cette séquence pédagogique, les élèves vont utiliser un casse-tête de programmation appelé « Robots : create AI ». Dans ce jeu, la programmation est le véritable élément de jeu. Les commandes données permettent à une simple figure (un robot) de se déplacer sur un certain chemin de cases, en évitant les cases vides. Le jeu commence par les étapes très simples de la logique algorithmique et se poursuit avec les instructions IF, IF - else. L'instruction IF-else est la plus basique de toutes les instructions de flux de contrôle. Elle contrôle un programme de manière à exécuter un code spécifique uniquement si une déclaration particulière s'avère être vraie.

Nous considérons qu'il s'agit d'un jeu assez simple qui convient pour un jeu en paire (avec un élève contrôlant et l'autre en soutien) en classe mais aussi pour une pratique personnelle comme devoir à la maison. Cela pourrait également enseigner le travail d'équipe et la communication. En effet, il est plus facile pour les élèves d'être créatifs lorsqu'ils réfléchissent ensemble pour expliquer les solutions recherchées par exemple.



- **Étape 1 : Explication de la procédure de base (10 minutes)**

L'enseignant explique le but et le processus du jeu à tous les élèves en commençant à jouer en classe. Dans un premier temps, le jeu propose un tutoriel du gameplay. Le professeur le suit et, en même temps, ajoute des explications supplémentaires si nécessaire.

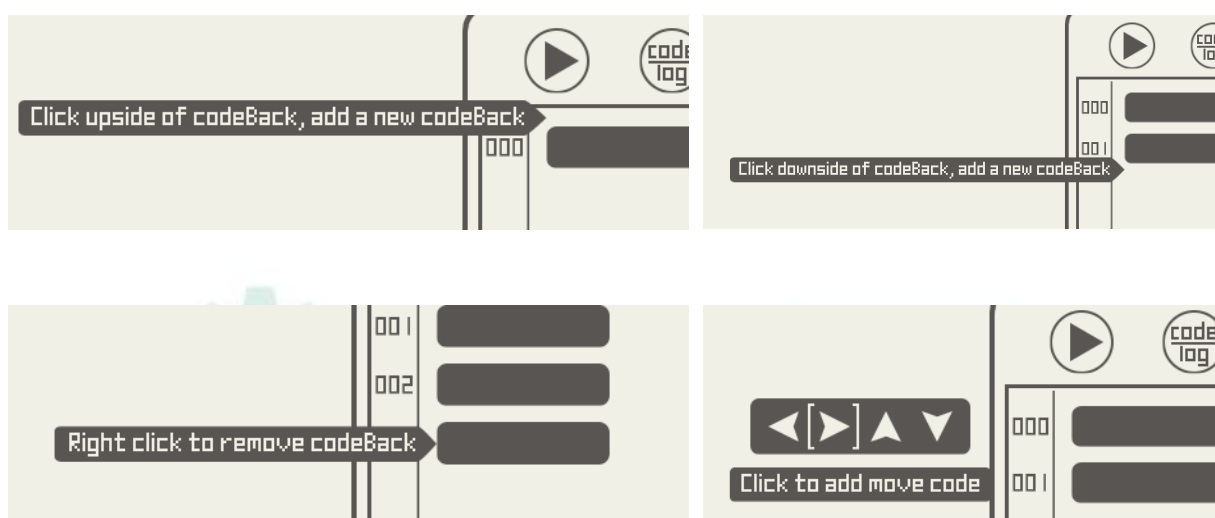


Figure 1. Captures d'écran du tutoriel du jeu (« Robots: create AI », Xuan, 2017)

- **Étape 2 : Découverte du jeu en groupe (25 minutes)**

Les élèves sont répartis en binômes et tentent de résoudre les énigmes des différents niveaux du jeu. Un élève joue le jeu tandis que l'autre observe et aide à la résolution des problèmes. Au cours des cinq premiers niveaux, les élèves se familiarisent avec le jeu et la pratique algorithmique de base présentée.

Une aide structurée est fournie au cours du jeu à l'aide d'exemples et de conseils, aidant les élèves à apprendre par eux-mêmes la structure et à utiliser les instructions de flux de contrôle dans les nouveaux problèmes posés à chaque niveau.

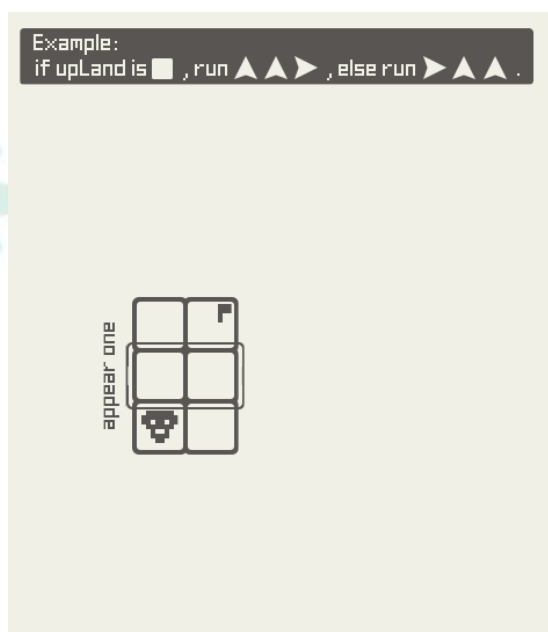


Figure 2. Capture d'écran d'un exemple donné (« Robots: create AI », Xuan, 2017)

🤖 Robots

Hint:

.....

```
if [ ][ ][ ]
```

.....

```
else
```

.....

```
if end
```

Hint:

```
if [ ][ ][ ]  
.....(→P)
```

```
else
```

```
.....(→P)
```

```
if end
```

```
if [ ][ ][ ]
```

.....

```
else
```

.....

```
if end
```

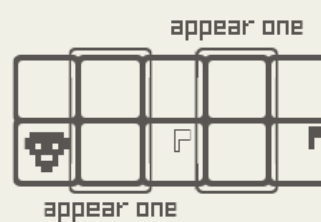


Figure 3. Captures d'écran de deux conseils pour différents niveaux fournis (« Robots: create AI », Xuan, 2017)



- **Étape 3 : Progression de niveaux par des élèves (35 minutes)**

Si un élève de l'équipe ne peut pas résoudre le puzzle seul, la deuxième personne de l'équipe l'aidera. S'ils ne peuvent pas résoudre le puzzle ensemble, ils peuvent demander de l'aide à l'enseignant. Chaque solution sera discutée au sein du groupe.

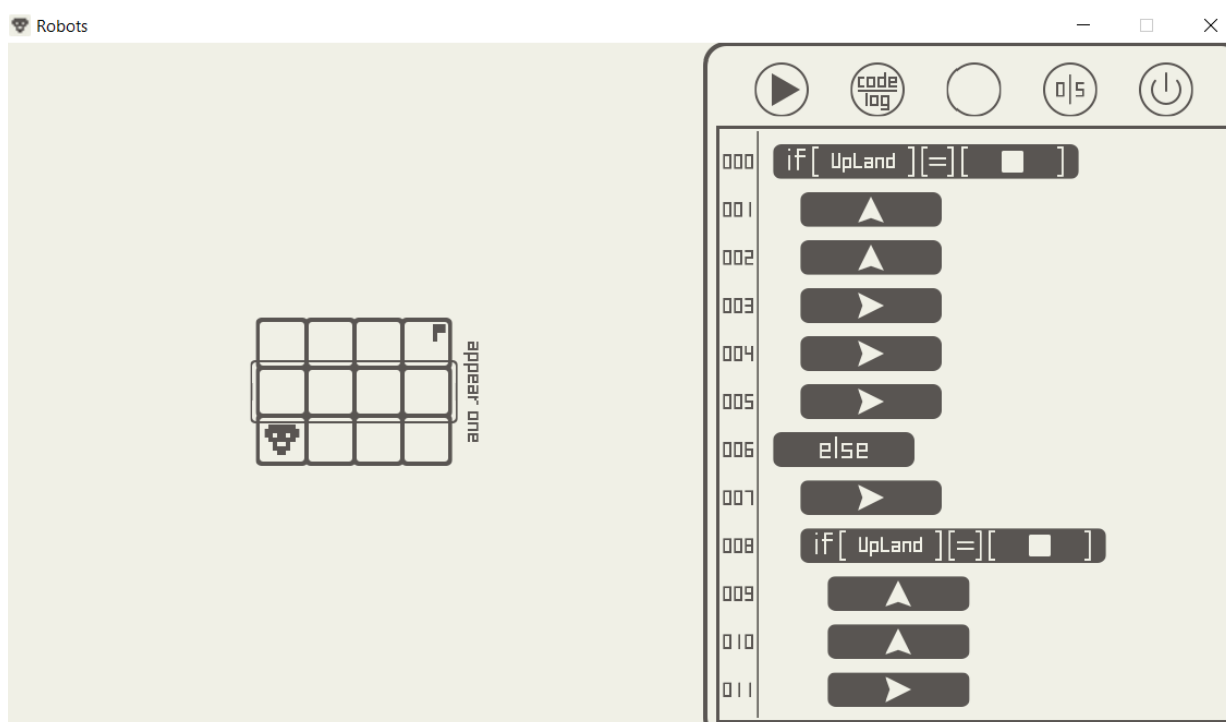


Figure 4. Capture d'écran de la section d'un code ("Robots: create AI", Xuan, 2017)

- **Étape 4 : Débrief (15 minutes)**

L'éducateur résume la structure des instructions IF - else, else et utilise le bouton code/log pour expliquer la série de commandes exécutées dans le dernier niveau.

```

running line 000 if -no
running line 001 >
running line 008 if -yes
running line 009 ▲
running line 010 ▲
running line 011 >
running line 012 >
finish running
running line 000 if -no
running line 001 >
running line 008 if -no
running line 014 >
running line 015 if -no
running line 020 >
running line 021 if -yes
running line 022 ▲
running line 023 ▲

```

Figure 5. Capture d'écran de la commande **code/log** (« Robots: create AI », Xuan, 2017)

- **Étape 5 : L'enseignant propose de continuer les niveaux suivants à la maison comme devoir (5 minutes)**

Le professeur attribue les niveaux suivants comme devoirs à la maison afin de consolider les instructions « IF-else ». Le nombre exact de niveaux proposés dépend de son point de vue sur les capacités des élèves à utiliser l'instruction dans de nouveaux problèmes.

Remarque : les étapes 2 à 5 peuvent être répétées pour les niveaux avancés avec plus de boucles de rupture et des interactions plus complexes.

Obtenir le jeu : https://store.steampowered.com/app/676620/Robots_create_AI/



Bibliographie

O'Connor, A. (2017, September 11). Program robots to run mazes in Robots: Create AI. Rock Paper Shotgun. <https://www.rockpapershotgun.com/robots-create-ai-programming-game-released>

