



# Programmez votre propre jeu



Présenté par TryEngineering – [www.tryengineering.org](http://www.tryengineering.org)

## Objet de la leçon

Cette leçon explique comment les ingénieurs en logiciels conçoivent des jeux vidéos et autres logiciels. Les élèves travaillent en équipes pour développer un simple programme informatique à partir d'un logiciel gratuit disponible en plusieurs langues. Chaque équipe évalue les jeux développés par les autres équipes et présente ses observations à la classe.

## Sommaire de la leçon

L'activité Programmez votre propre jeu explore le travail des ingénieurs en logiciels et permet aux élèves de travailler en équipes afin de développer leur propre jeu vidéo à partir d'un simple logiciel gratuit. Les équipes présentent leur jeu à la classe, évaluent les autres jeux et réfléchissent aux questions d'ingénierie.

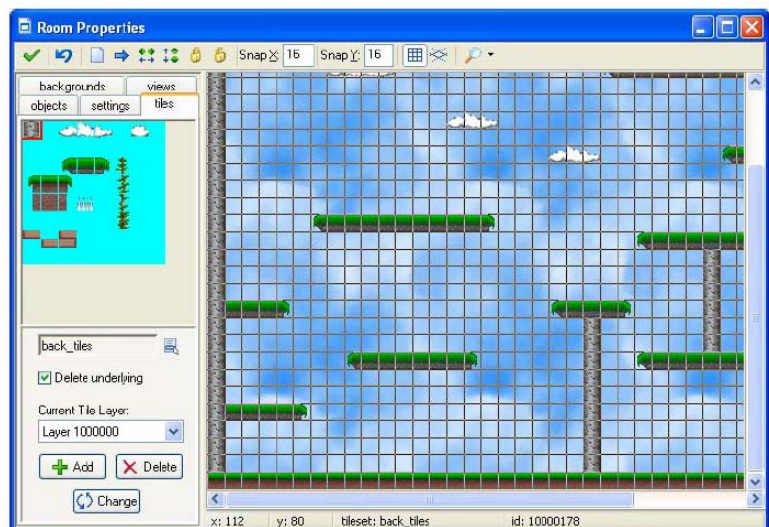


## Niveaux d'âge

11 à 18 ans (Remarque : la difficulté de cette leçon peut être adaptée en fonction de l'âge et du niveau des élèves, de la simple programmation ou modification de programmes à un niveau de programmation plus poussé.)

## Objectifs

- ✦ Apprendre comment les ingénieurs en logiciels développent des jeux vidéo.
- ✦ Etudier le processus de réingénierie d'un produit.
- ✦ Etudier comment les ingénieurs abordent en équipes la résolution des problèmes.
- ✦ Apprendre le travail d'équipe et la résolution des problèmes en groupes.



## Résultats escomptés à la fin de la leçon

Au terme de cette activité, les élèves devraient acquérir une compréhension des sujets suivants :

- ✦ le génie logiciel et la programmation
- ✦ la conception et l'ingénierie des produits

## Programmez votre propre jeu

Elaboré par IEEE dans le cadre de TryEngineering  
[www.tryengineering.org](http://www.tryengineering.org)

- ✦ la résolution des problèmes
- ✦ le travail d'équipe

---

## Activités de la leçon

Les élèves étudient les notions élémentaires de la programmation informatique et le travail des ingénieurs en logiciels. Les élèves travaillent en équipes pour développer un simple programme informatique à partir d'un logiciel gratuit disponible en plusieurs langues. Les élèves réalisent leurs propres jeux et évaluent ceux développés par les autres équipes.

---

## Ressources/Matériaux

- ✦ Documents de ressource aux enseignants (en pièces jointes)
- ✦ Tutoriels Game Maker ([www.yoyogames.com/make/tutorials](http://www.yoyogames.com/make/tutorials))
- ✦ Feuilles de travail des élèves (en pièces jointes)
- ✦ Fiches de ressource aux élèves (en pièces jointes)

---

## Alignement sur les structures des programmes scolaires

Voir la fiche ci-jointe décrivant l'alignement des programmes scolaires.

---

## Liens Internet

- ✦ TryEngineering ([www.tryengineering.org](http://www.tryengineering.org))
- ✦ YoYo Game – Logiciel Game Maker ([www.yoyogames.com/gamemaker/try](http://www.yoyogames.com/gamemaker/try))
- ✦ ITEA Standards for Technological Literacy: Content for the Study of Technology (en anglais) ([www.iteaconnect.org/TAA](http://www.iteaconnect.org/TAA))
- ✦ *National Science Education Standards* (en anglais) ([www.nsta.org/standards](http://www.nsta.org/standards))
- ✦ *NCTM Principles and Standards for School Mathematics* (en anglais) (<http://standards.nctm.org>)

---

## Lecture recommandée (en anglais)

- ✦ Game Creation For Teens (ISBN: 159863500X)
- ✦ Getting Started with Game Maker (ISBN: 1598638823)

---

## Activité d'écriture facultative

- ✦ Rédigez une dissertation ou un paragraphe décrivant les implications, sur le plan éthique, si vous deviez adapter un logiciel développé par quelqu'un d'autre. La notion de « propriété intellectuelle » est un terme générique désignant les droits légaux associés à certains noms, contenus multimédias écrits et enregistrés et inventions. Argumentez pour ou contre la nécessité de rétribuer financièrement ou de toute autre manière le développeur d'origine d'un logiciel que vous adaptez/modifiez en vue de créer un nouveau logiciel. Imaginez que le logiciel d'origine ne se soit pas bien vendu, mais que votre version, en revanche, ait eu un franc succès. Cet exercice de rédaction pourrait éventuellement donner lieu à un débat sur les avantages et les inconvénients et sur la notion de droits de propriété intellectuelle.

# Programmez votre propre jeu



## Pour les enseignants : Alignement sur les structures des programmes scolaires

Remarque : Tous les plans de leçons de cette série sont alignés sur les normes nationales pour l'enseignement des sciences (*National Science Education Standards*), établies par le Conseil national de recherche des Etats-Unis (National Research Council) et approuvées par l'Association nationale des enseignants des sciences des Etats-Unis (National Science Teachers Association), et le cas échéant, sur les normes internationales d'enseignement de la technologie pour l'alphabétisation technologique (International Technology Education Association's Standards for Technological Literacy) ou sur les principes et normes en matière de mathématiques scolaires établis par le Conseil national américain des enseignants en mathématiques (National Council of Teachers of Mathematics' Principals and Standards for School Mathematics).

### ◆ Normes nationales pour l'enseignement des sciences de la CM2 à la quatrième (10 à 14 ans)

#### **NORME DE CONTENU A : Enquête scientifique**

**Au terme de leurs activités, tous les élèves devraient acquérir :**

- ✦ Les aptitudes nécessaires pour réaliser des enquêtes scientifiques

#### **NORME DE CONTENU E : Science et technologie**

**Au terme des activités effectuées de la CM2 à la quatrième, tous les élèves devraient acquérir :**

- ✦ Des aptitudes de conception technologique
- ✦ Une compréhension de la science et de la technologie

#### **NORME DE CONTENU G : Histoire et nature de la science**

**Au terme de leurs activités, tous les élèves devraient acquérir une compréhension de :**

- ✦ L'histoire de la science

### ◆ Normes nationales pour l'enseignement des sciences de la troisième à la terminale (14 à 18 ans)

#### **NORME DE CONTENU A : Enquête scientifique**

**Au terme de leurs activités, tous les élèves devraient acquérir :**

- ✦ Les aptitudes nécessaires pour réaliser des enquêtes scientifiques

#### **NORME DE CONTENU E : Science et technologie**

**Au terme de leurs activités, tous les élèves devraient acquérir :**

- ✦ Des aptitudes de conception technologique
- ✦ Une compréhension de la science et de la technologie

#### **NORME DE CONTENU G : Histoire et nature de la science**

**Au terme de leurs activités, tous les élèves devraient acquérir une compréhension :**

- ✦ Des perspectives historiques

---

## Pour les enseignants : Alignement sur les structures des programmes scolaires (suite)

### ◆ Normes pour l'alphabétisation technologique – Tous âges

#### La nature de la technologie

- ✦ Norme 2 : Les élèves acquerront une compréhension des concepts fondamentaux de la technologie.
- ✦ Norme 3 : Les élèves acquerront une compréhension des relations entre les technologies et des liens entre la technologie et d'autres champs d'étude.

#### Technologie et société

- ✦ Norme 7 : Les élèves acquerront une compréhension de l'influence de la technologie sur l'histoire.

#### Conception

- ✦ Norme 9 : Les élèves acquerront une compréhension de la conception technique.
- ✦ Norme 10 : Les élèves acquerront une compréhension du rôle de la recherche des défaillances, de la recherche et du développement, de l'invention et de l'innovation, et de l'expérimentation dans la résolution des problèmes.

#### Aptitudes pour un monde technologique

- ✦ Norme 12 : Les élèves acquerront des aptitudes d'utilisation et de maintenance des produits et systèmes technologiques.

#### Le monde, objet de conception

- ✦ Norme 17 : Les élèves acquerront une compréhension et des aptitudes de sélection et d'utilisation des technologies d'information et de communication.

### ◆ Principes et normes en matière de mathématiques scolaires

#### Normes relatives aux nombres et opérations

##### Au terme de leurs activités, tous les élèves doivent pouvoir :

- ✦ Comprendre les nombres, les modes de représentation des nombres, les relations entre les nombres et les systèmes de nombres.
- ✦ Calculer facilement et effectuer des estimations raisonnables.

#### Normes relatives aux liens

##### Au terme de leurs activités, tous les élèves doivent pouvoir :

- ✦ Comprendre comment les principes mathématiques sont interconnectés et interdépendants pour produire un ensemble cohérent.
- ✦ Reconnaître et appliquer les mathématiques dans des contextes extérieurs aux mathématiques.

# Programmez votre propre jeu



## Pour les enseignants : Ressources aux enseignants

### ◆ But de la leçon

Explorer la résolution de problèmes d'ingénierie à travers la programmation, en équipes, d'un nouveau jeu vidéo. Les élèves étudient les notions élémentaires de la programmation informatique et le travail des ingénieurs en logiciels. Les élèves travaillent en équipes pour développer un simple programme informatique à partir d'un logiciel gratuit disponible en plusieurs langues. Les élèves réalisent leurs propres jeux et évaluent ceux développés par les autres équipes.

### ◆ Objectifs de la leçon

- ✦ Apprendre comment les ingénieurs en logiciels développent des jeux vidéo.
- ✦ Étudier le processus de réingénierie d'un produit.
- ✦ Étudier comment les ingénieurs abordent en équipes la résolution des problèmes.
- ✦ Apprendre le travail d'équipe et la résolution des problèmes en groupes.



### ◆ Matériaux

- ✦ Fiches de ressources et feuille de travail des élèves
- ✦ Connexion Internet ou accès à des ordinateurs (le logiciel gratuit peut être téléchargé et installé sur un ordinateur Windows ne disposant pas de connexion Internet ; le logiciel est disponible en plusieurs langues)

### ◆ Marche à suivre

1. Téléchargez et installez le logiciel gratuit GameMaker de YoYo Games ([www.kidsprogramminglanguage.com](http://www.kidsprogramminglanguage.com)) sur plusieurs ordinateurs ou dans un laboratoire, afin de permettre aux élèves de développer leurs jeux en équipes. Notez qu'il existe une version limitée gratuite et une autre version plus évoluée pour une somme modique. La version gratuite est adaptée à une utilisation en classe par des débutants.
2. Vous pouvez également consulter les didacticiels proposés sur [www.yoyogames.com/make/tutorials](http://www.yoyogames.com/make/tutorials).
3. Montrez aux élèves les divers documents de référence à leur disposition. Ces documents peuvent être lus en classe ou donnés à lire à la maison la veille. Les élèves doivent avoir lu les pages sur l'« initiation à la programmation » avant d'utiliser le logiciel sur un ordinateur.
4. Répartissez les élèves en groupes de 2 ou 3 (ou plus selon le nombre d'ordinateurs dont vous disposez) et distribuez un jeu de matériaux à chaque groupe.
5. Expliquez-leur qu'ils forment une équipe d'ingénieurs logiciels et qu'ils doivent développer un nouveau jeu vidéo destiné à des élèves âgés entre 6 et 10 ans.
6. Les équipes d'élèves développent un jeu simple qu'ils présenteront à la classe.
7. Chaque groupe d'élèves évalue ensuite les jeux développés par les autres équipes, puis remplit une fiche d'évaluation/de réflexion.

# Programmez votre propre jeu



---

## Pour les enseignants : Ressources aux enseignants (suite)

### ◆ Conseils

Cette leçon peut donner lieu à un projet s'étendant sur toute la durée du semestre, ou être simplifiée en demandant aux élèves de créer ou d'améliorer l'un des jeux de démo fournis par le développeur du logiciel. Le jeu « Space Cleaner », par exemple, peut être modifié en une seule session. Si vous optez pour la simple modification d'un jeu, demandez aux élèves d'explorer au préalable le jeu tel qu'il est actuellement conçu, de déterminer en équipe les modifications qu'ils souhaitent apporter, puis de mettre leur plan à exécution.

### ◆ Temps nécessaire

Une ou deux sessions de 45 minutes

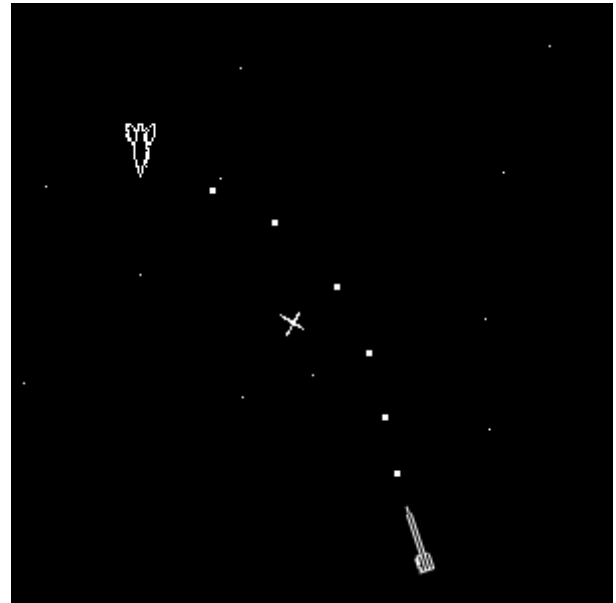
# Programmez votre propre jeu



## Ressources aux élèves Génie logiciel : l'histoire des jeux vidéo

### ◆ La naissance des jeux vidéo

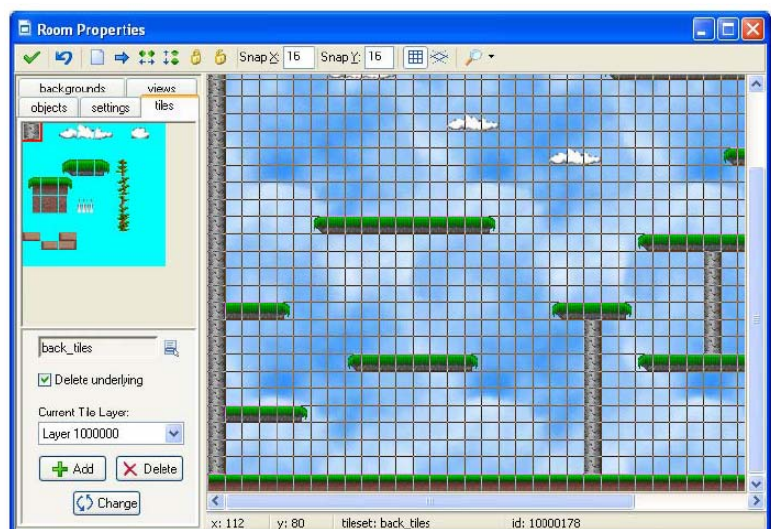
Même si les ordinateurs personnels n'ont acquis leur popularité qu'avec l'émergence du microprocesseur, les ordinateurs centraux et les mini-ordinateurs sont utilisés pour le développement de jeux vidéo depuis au moins les années 60. Spacewar! est l'un de ces premiers jeux vidéo. Il a été développé en 1961 sur un ordinateur qui servait à effectuer des calculs statistiques, par Martin Graetz et Alan Kotok, étudiants du MIT (Massachusetts Institute of Technology) et Stephen Russell, employé du MIT. Comme le montre l'illustration de droite, deux joueurs s'affrontaient pour piloter chacun son vaisseau spatial autour d'une étoile centrale, dans le but de détruire l'autre.



Les jeux pour PC de première génération étaient des aventures textuelles ou des fictions interactives qui consistaient à communiquer avec l'ordinateur en entrant des commandes à l'aide d'un clavier. Au milieu des années 70, ces jeux étaient développés et distribués par le biais de groupes amateurs et de magazines américains spécialisés tels que Creative Computing et plus tard, Computer Gaming World. Ces publications fournissaient le code des jeux, que les lecteurs/joueurs saisissaient sur un ordinateur et reproduisaient, encourageant ainsi ces derniers à développer leurs propres logiciels dans le cadre de concours.

### ◆ Le métier d'ingénieur logiciels

Les ingénieurs logiciels qui travaillent au développement d'applications ou de systèmes analysent les besoins des utilisateurs en vue de concevoir, construire, tester et gérer les applications logicielles ou systèmes informatiques. Les ingénieurs logiciels peuvent également participer à la conception et au développement de nombreux types de logiciels, y compris ceux utilisés par les systèmes d'exploitation et pour la distribution en réseau, ainsi que les compilateurs, qui convertissent les programmes en vue de leur exécution sur un ordinateur.



# Programmez votre propre jeu

---

## Ressources aux élèves

### Génie logiciel : l'histoire des jeux vidéo (suite)

En programmation, ou codage, les ingénieurs logiciels élaborent les instructions, ligne par ligne, qui permettront à un ordinateur d'exécuter une fonction. Ils résolvent également les problèmes techniques qui se produisent. Les ingénieurs logiciels doivent posséder de solides compétences en programmation, mais leur rôle consiste plus à développer des algorithmes et à analyser et résoudre les problèmes de programmation qu'à écrire les codes proprement dit.





## Ressources aux élèves Génie logiciel : les algorithmes

### ◆ Qu'est-ce qu'un algorithme ?

En mathématiques, en informatique, en linguistique et dans d'autres disciplines apparentées, un algorithme est une liste finie d'instructions clairement définies pour accomplir une tâche précise qui, compte tenu d'un état initial, se terminera dans un état final défini. La notion d'algorithme est née comme moyen d'enregistrer des procédures de résolution de problèmes mathématiques, tels que la recherche du diviseur commun de deux nombres ou la multiplication de deux nombres.

Le concept fut formalisé en 1936 avec les machines d'Alan Turing et le lambda-calcul d'Alonzo Church, qui s'avèrent par la suite le fondement de l'informatique.

L'organigramme, qui est essentiellement une séquence logique d'étapes visant la résolution d'un problème, est un exemple d'algorithme.

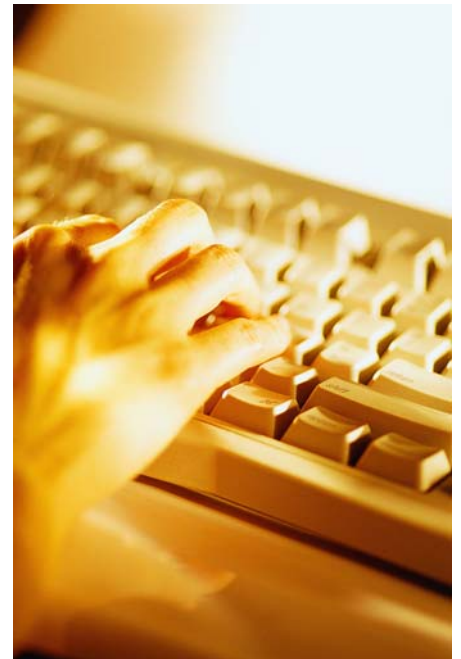
### ◆ Applications informatiques

Les algorithmes sont indispensables au traitement des informations par les ordinateurs. En effet, un programme informatique est essentiellement un algorithme qui instruit l'ordinateur sur les étapes précises à suivre (et dans quel ordre) afin d'accomplir une tâche donnée, telle que le calcul d'une facture, l'impression de bulletins de rendement ou l'exécution d'une analyse budgétaire.

Etant donné qu'un algorithme est une liste précise d'étapes spécifiques, l'ordre de calcul sera presque toujours crucial pour son fonctionnement. Ces instructions sont généralement censées être explicites, et sont décrites comme partant du haut vers le bas. C'est ce que l'on appelle parfois l'« ordre d'exécution ».

A chaque domaine scientifique est associée une série de problèmes qui requiert des algorithmes efficaces. Les problèmes connexes dans un domaine sont souvent étudiés ensemble.

Ils peuvent être classés comme suit : algorithmes de recherche, algorithmes de tri, algorithmes de fusion, algorithmes numériques, algorithmes de graphes, algorithmes de chaînes, algorithmes géométriques de calcul, algorithmes combinatoires, apprentissage machine, cryptographie, algorithmes de compression des données et techniques d'analyse.



# Programmez votre propre jeu



## Feuille de travail des élèves : Maintenant, c'est vous l'ingénieur !

◆ Vous êtes une équipe d'ingénieurs chargés de développer un nouveau jeu vidéo destiné à des enfants âgés entre 6 et 10 ans.

### ◆ Préparation

1. Lisez les différents documents de référence que vous avez reçus.
2. Lisez le guide d'initiation à la programmation qui vous a été remis.

### ◆ Etapes de l'activité

1. En équipe, élaborer un plan ou une idée de jeu (en inventant également un nom). Dans l'encadré ci-dessous, rédigez une description en deux phrases de votre nouveau jeu, qui pourrait être utilisée dans une publicité :

Nom du jeu :

Description :

2. A l'aide du logiciel, travaillez ensemble à la conception de votre jeu.
3. Présentez votre jeu aux autres équipes de la classe en leur montrant comment il fonctionne (vous testerez également leurs jeux).
4. Répondez aux questions suivantes concernant votre jeu et ceux développés par les autres équipes.
5. En équipe, présentez vos observations et votre réflexion à la classe.

### ◆ Questions d'évaluation

1. Comment avez-vous décidé des modifications à apporter à votre jeu une fois que vous étiez prêt à le développer à l'aide du logiciel fourni ?

2. D'après vous, combien de temps faudrait-il pour développer un nouveau logiciel graphique ou de traitement de texte ? Selon vous, combien d'ingénieurs seraient nécessaires pour concevoir ce type de logiciel ? Pour quelle raison ?

3. La programmation d'un jeu vidéo était-elle plus facile ou plus difficile que vous ne le pensiez ? Pour quelle raison ?

4. Quelles difficultés avez-vous rencontrées lors de la conception de votre jeu ?

5. De tous les autres jeux développés dans votre classe, lesquels préféreriez-vous ? Pour quelle raison ? Quelles fonctionnalités vous ont plu ?



6. Le travail en équipe a-t-il selon vous facilité ou compliqué le développement de votre jeu ? Pensez-vous que vous auriez été capable de créer votre nouveau concept si vous n'aviez pas travaillé en équipe ? Quels sont les avantages du travail d'équipe par rapport au travail individuel ?

7. Qu'avez-vous appris dans cette leçon au sujet des méthodes de résolution des problèmes utilisées par les ingénieurs ?