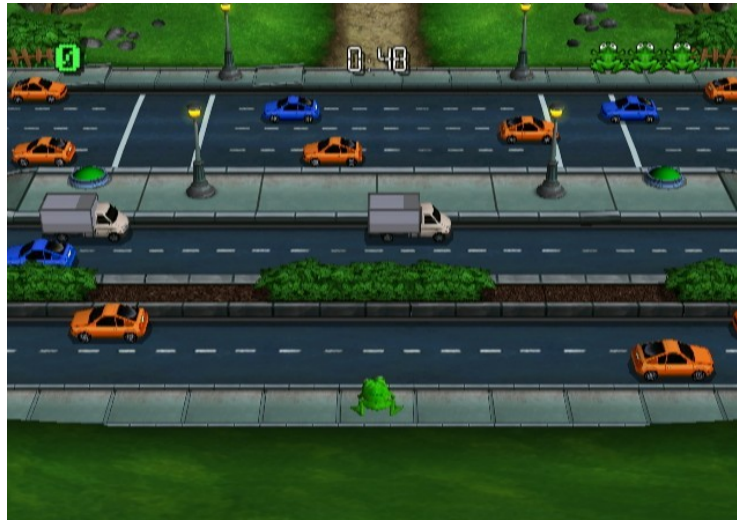


Projet de jeu vidéo : projet de développement en équipe



Source¹

Conception et développement d'un jeu vidéo d'arcade avec Greenfoot

Version 1.2 / 16.02.2017

Présentation

Le présent projet consiste à, par groupe de deux, concevoir (algorithme, modélisation), mettre en oeuvre (programmation, implémentation), tester et documenter une application de votre choix. L'application doit répondre à un tableau de spécifications minimales (voir ci-dessous). Le projet sera évalué selon un certain nombre de critères issus, entre autre, de ces spécifications. Un accent particulier devra être mis sur le travail en équipe et sa planification. Il sera notamment important de subdiviser le travail à accomplir et de le répartir judicieusement entre les coéquipiers. Le logiciel sera développé en Java dans le cadre de l'IDE Greenfoot utilisé au cours.

Le jeu vidéo à programmer peut être choisi parmi la multitude de jeux vidéos déjà existants. On veillera cependant dans ce cas de se lancer dans un projet dont la difficulté est adaptée au niveau de compétences en programmation des coéquipiers. Il est bien sûr également possible de créer un tout nouveau jeu vidéo. Dans le cas de la reproduction d'un jeu vidéo existant, il est demandé d'y rajouter une touche d'originalité en y introduisant une nouvelle règle ou une nouvelle fonction². S'il est possible de s'inspirer de projets Greenfoot existants et disponibles sur Internet, il est alors demandé d'indiquer dans le projet les parties de code reprises d'un ou de projets existants³.

¹ <http://www.project-management-software.biz/how-to-select-project-management-software-objectives.html>

² Un projet portant sur le développement d'un jeu déjà existant consiste à concevoir une nouvelle version (même simplifiée) de ce jeu et non de prendre un projet Greenfoot existant et de simplement en modifier son code. Tout emprunt (limité) d'extrait code à un autre projet disponible sur Internet doit être clairement désigné en indiquant ses sources (date, nom du site web, adresse de téléchargement du projet existant).

³ Un cas de plagiat serait sanctionné par la note 1 (!)

Eléments d'un jeu vidéo d'arcade

Un jeu vidéo d'arcade est composé des éléments suivants :

- une **scène** (fond d'écran)
- des **acteurs** (images) :
 - des acteurs fixes (ex : des obstacles tel un mur ou un arbre)
 - des acteurs animés :
 - par le programme (mouvements gérés par l'application)
 - par le ou les joueurs (à l'aide du clavier, d'une manette de jeu, etc)
- des **collisions** d'acteurs provoquant des événements tels que :
 - l'arrêt du mouvement d'un acteur (obstacle)
 - le changement d'état d'un acteur
 - changement de visage (image) d'un acteur
 - changement de vitesse et/ou d'orientation d'un acteur en mouvement
 - sortie de scène de l'acteur (ex : explosion, destruction, disparition)
 - émission d'un son (ex : bruit d'une explosion)
- un **scénario** avec :
 - un ou des *objectifs* (ex : faire traverser une route à une grenouille, sortir d'un labyrinthe, etc)
 - une situation de *départ* (ex : une grenouille devant une route avec du trafic)
 - des *règles* ou des contraintes (ex : impossibilité de passer par dessus des obstacles)
 - un *score* et/ou des *ressources* limitées (ex : nombre de 'vies' d'un personnage, compte à rebours, etc)
 - une gestion croissante de la *difficulté* (ex : niveaux de jeu, augmentation de la vitesse des voitures, etc)
 - une situation de *fin* (ex : la grenouille a réussi à traverser la route)

Les images et les sons utilisés peuvent être trouvés sur Internet, mais dans ce cas il est impératif de citer les sources (date, nom du site web, adresse de téléchargement du fichier) dans un écran de remerciements. Les jeux incluant des images ou sons trouvés sur Internet ne peuvent être mis publiquement à disposition (sur Internet via un site web notamment) sans l'autorisation des ayants-droits des images et sons (auteurs, propriétaires).

Spécifications et critères d'évaluation

Le projet consiste développer un jeu d'animation en 2d (jeu d'arcade). Celui-ci répond aux caractéristiques suivantes :

A. Caractéristiques minimales

- ☐ attrait du logiciel (les gens ont envie d'y rejouer !)
- ☐ jeu complet (voir les éléments d'un jeu vidéo d'arcade ci-dessus)
- ☐ développement intégrant au maximum les concepts et techniques étudiés durant le cours (programmation, fichiers, éven. bases de données, etc)
- ☐ prise en compte des critères d'évaluation des exercices et des examens (éviter les erreurs réalisées durant les exercices et examens)
- ☐ utilisation de fichier(s) pour stocker les noms et les scores des derniers joueurs ainsi qu'affichage de la liste de ces derniers

B. Documentation (projet OC INF : uniquement le mode d'emploi)

- ☐ documentation du projet (!) : le programme doit être accompagné d'un mode d'emploi⁴ pour l'utilisateur ainsi que d'un bref document présentant le projet. Ce dernier contient, notamment :
 - *Présentation générale* du projet (introduction) : en quoi consiste le jeu développé ? Quel est son but ? Quelles sont ses règles ? Quelles sont ses fonctionnalités ?
 - *La structure des informations internes* traitées par le programme : quelles sont les informations fondamentales traitées par le programme (ex : score, noms du ou des joueurs, localisation des acteurs, etc) ? Où et comment sont-elles stockées dans le programme : types (primitifs, tableaux, structures) et localisation (variables locales, variables d'objet (attributs), variable de classe) ?
 - Le *stockage externe* des informations conservées par le programme : où et sous quelle forme sont stockées les informations conservées par le programme (fichier, base de données) ? Quelle est la structure du fichier (texte avec délimiteurs, xml, ...) ou de la base de données (diagramme E-A et schéma) ?
 - Pour chaque *classe* définie dans le projet (sous-classes de classe World et Actor), la liste des fonctions et méthodes (objets) avec à chaque fois leur *signature* et leur *contrat*.
 - *Organisation du code* (algorithme) : comment le code du projet est-il organisé ? Quelles sont les classes (sous-classes des classes World et Actor) le constituant et quels sont les rôles de chacune d'elles dans le bon fonctionnement du jeu ?
 - Une *conclusion* : quelles sont les limites du projet ? Que pourrait-t-on y améliorer dans une version future ? etc

⁴ Le mode d'emploi peut être directement intégrée dans le programme de jeu et le joueur peut alors le lire directement à partir de l'application

- En annexe :

- Un tableau de *répartition des tâches* précisant quel coéquipier est responsable de quelle(s) partie(s) du projet
- Le code source intégral du projet

C. Code source

- ❑ lisibilité du code source :

- la structure du programme doit ressortir à la lecture du code source (*indentations, espacements, etc*)
 - des **commentaires** pertinents donnant des éléments d'explications figurent aux endroits clés du code source
 - les identifiants (noms) des variables, fonctions, méthodes et classes doivent être pertinents et faciliter ainsi la compréhension du code
 - une tierce personne ayant des compétences équivalentes ou supérieures à celle acquises au cours doit pouvoir comprendre aisément la structure du code afin de le modifier et continuer son développement
- ❑ la manière dont le travail a été réparti (qui a écrit quelle partie du code) doit être précisée (peut être communiquée dans un document séparé)

D. Distribution (optionnel)

- ❑ publication du projet final sur Internet (avec Educanet2 ou Google Sites par exemple)
- ❑ opérationnalité du programme : le programme doit pouvoir être téléchargé à partir du site et utilisé par n'importe qui disposant d'un ordinateur fixe ou portable (pour autant que le projet ne contienne pas de ressources multimédia de tiers protégées par des droits d'auteur)

E. Extension possibles ou recommandées (facultatif)

- ❑ recherche et développement : la mise en application de concepts ou techniques non présentés au cours est un plus. Ce projet est également une opportunité d'un approfondissement dans un domaine intéressant les coéquipiers.
- ❑ publication du projet sur un site de partage (idéalement : www.greenfoot.org)

Ressources à utiliser

Sauf accord préalable avec le professeur, les applications suivantes sont à utiliser pour concevoir les éléments du projet :

- ❑ Greenfoot (IDE)
- ❑ Google Docs (documentation)

Planning

Les échéanciers suivant doit être respecté :

Date	Objet
Jeudi 2 novembre	Présentation intermédiaire
Jeudi 22 mars	Restitution des projets
Jeudi 29 mars (à confirmer)	Présentations des projets

Restitution des projets

Le projet doit être restitué sous la forme d'un fichier *.zip contenant tous ses fichiers et dont le nom indique ses auteurs et le nom abrégé du projet (ex : SteinauerP_DuruzS_Frogger.zip). Le fichier est à déposer dans le classeur du groupe 'CSM INF OC4' d'Educanet2. Dans le cas où le fichier serait trop volumineux, on le déposera sur Google Drive, avec accès limité aux seules personnes disposant du lien de partage et on placera dans le classeur du groupe Educanet2 une fichier *.txt (avec la même nomenclature que pour le fichier *.zip) contenant le lien de téléchargement du fichier *.zip