

Licence Informatique 1^e année

Algorithmique et Programmation

Projet 2012-2013

1 Organisation

Le projet est à réaliser en binôme. En fin de semestre, lors de la dernière séance de TP, les binômes présenteront leur travail aux encadrants de TP. Lors des présentations, chaque binôme rendra un rapport qui présentera le programme réalisé (structures de données utilisées, principe des algorithmes implémentés, ...), les résultats obtenus, les problèmes rencontrés, les solutions apportées et tout élément permettant d'évaluer le travail réalisé. Chaque binôme devra également, au moment des soutenances, envoyer le code des programmes écrits aux responsables de TP (*lei.wu@u-picardie.fr*, *marisela.hernandez@u-picardie.fr*, *vincent.villain@u-picardie.fr* et *frederic.furst@u-picardie.fr*).

2 Sujet

Le projet consiste à réaliser un jeu de Snake (Serpent). Le principe du jeu est décrit sur la page: [fr.wikipedia.org/wiki/Snake_\(jeu_vidéo\)](http://fr.wikipedia.org/wiki/Snake_(jeu_vid%C3%A9o)). Une classe *InterfaceSnake.java* est fournie pour l'affichage du jeu (son mode d'emploi est donné à la fin du sujet). La réalisation du jeu nécessite d'identifier les données à représenter, de définir des structures de données permettant de stocker ces données et de coder l'algorithme permettant de faire tourner le jeu.

Au final, le projet doit permettre de jouer au Snake, avec différents niveaux de jeu (au moins 3). Au premier niveau, le plateau de jeu ne contient aucun obstacle, au deuxième niveau le plateau contient comme obstacle une barre horizontale placée au milieu du plateau, au troisième niveau le plateau contient deux barres verticales. Chaque niveau a son propre plateau mais aussi sa propre vitesse, ainsi que la position initiale du serpent. Chaque niveau définit également le nombre de fruits à manger avant de passer au niveau suivant.

Pour temporiser le jeu, on peut utiliser l'instruction `try{Thread.sleep(n);} catch(InterruptedException e){}` qui met le programme en attente *n* millisecondes.

Remarque importante: des programmes de Snake écrits en Java peuvent sûrement être trouvés sur le Web, mais ils seront écrits en Java objet. Aucun projet écrit dans le paradigme objet ne sera pris en compte lors des soutenances.

3 Pour aller plus loin ...

Les extensions du projet décrites dans cette partie sont optionnelles, mais en traiter au moins une augmentera très sensiblement la note attribuée, SI LA PARTIE OBLIGATOIRE A ÉTÉ TRAITÉE

Il est possible d'étendre le jeu pour que deux joueurs puissent jouer, avec chacun un serpent. Le but du jeu est alors à la fois de manger les fruits et de bloquer l'autre serpent pour l'obliger à se cogner à un obstacle.

La description des niveaux peut être stockée dans un fichier et chaque niveau est alors chargé lorsque c'est nécessaire.

On peut sauver les scores des joueurs dans un fichier, et les trier pour afficher les meilleurs scores.

On peut ajouter des obstacles mouvants, ou des mangoustes (qui courent après les serpents pour les manger).

Il est possible d'ajouter de la musique (voir le tutoriel <http://docs.oracle.com/javase/tutorial/sound/index.html>).

4 Mode d'emploi de la classe *InterfaceSnake*

Créer une interface : `InterfaceSnake is = new InterfaceSnake(grille);`
où `grille` est de type `Color[][]`. `Color` est une classe de l'API Java qu'on peut utiliser en ajoutant `import java.awt.Color` en tête de fichier.

Changer la grille de couleurs affichée par l'interface : `is.changeGrille(g);`
où `g` est la nouvelle grille de couleurs.

Forcer le rafraichissement de l'affichage : `is.refresh();`

Récupérer la touche tapée par l'utilisateur : `is.pressedKey();` qui renvoie un entier, 1 pour flèche gauche, 2 pour flèche droite, 3 pour flèche bas, 4 pour flèche haut, 5 pour pause, 0 pour les autres touches. Il est possible de rajouter des touches si besoin (en particulier pour le jeu à deux joueurs).

Afficher un message dans une boîte de dialogue : `is.afficheMessage(c)`
où `c` est une chaîne de caractères.