



Projet CVDA

MINI-JEU SNAKE

Amaury Guyonnet – Yohann Vernet | G6S2 | 12/12/2017

Caractéristiques du projet

| | |
|------------------------------|---|
| Nom du projet | Snake |
| Langage de développement | Java |
| Framework de tests unitaires | Junit |
| Gestionnaire de version | GIT |
| Lien du dépôt GIT | https://github.com/Yohann29/Snake |
| Dates de développement | 07/08/2017 - 10/12/2017 |

Description et explications

Nous avons voulu développer un mini-jeu connu de (presque) tous pour tester nos connaissances en Java et s'entraîner à l'utilisation des outils de l'interface Swing, actuellement étudiés et cours d'IHM.

Il consiste en le déplacement par l'utilisateur d'un serpent via les touches haut/bas/droite/gauche. Ce dernier doit éviter les obstacles symbolisés par des carrés oranges, et « attaquer » les ennemis symbolisés en vert pour grandir et ainsi augmenter son score de jeu. Plus le serpent est grand, plus le score est élevé et plus la vitesse de déplacement du serpent augmente : la difficulté est alors également augmentée. La partie est perdue lorsque le serpent touche un obstacle ou lorsqu'il touche les bords de la fenêtre.

Utilisation du gestionnaire de versions

Nous avons fait le choix d'utiliser GIT pour gérer les différentes versions de notre projet.

Le développement s'est déroulé en deux étapes. La première consistait à la création des différentes fonctions principales de fonctionnement du programme. Nous avons ensuite décidé, une fois ces fonctions écrites, d'ajouter la fonctionnalité des « obstacles » qui renforce la difficulté du jeu, d'où la création d'une seconde branche parallèle à la branche `master` qui constitue la seconde étape du développement : la branche `obstacles-feature`. Une fois la fonctionnalité développée et fonctionnelle, cette dernière a été fusionnée avec la branche principale.

Plan de tests effectués

| N° test | Type du scénario | Description | Résultat du test : Le 10/12/2017 |
|------------------------|------------------|--|-------------------------------------|
| 1 | Normal | La taille du tableau générant les obstacles est égale à 40X40 (tableau bidimensionnel) | Validé |
| Résultat(s) attendu(s) | | | |
| Retourne true | | | |

| N° test | Type du scénario | Description | Résultat du test : Le 10/12/2017 |
|------------------------|------------------|--|-------------------------------------|
| 2 | Normal | Les données de chaque case du tableau générant les obstacles sont des objets de la classe Obstacle et ne sont pas égaux à null | Validé |
| Résultat(s) attendu(s) | | | |
| Retourne true | | | |

| N° test | Type du scénario | Description | Résultat du test : Le 10/12/2017 |
|--|------------------|--|-------------------------------------|
| 3 | Erreur | Deux ennemis sont créés à la même position (selon les coordonnées (x ; y)) | Validé |
| Résultat(s) attendu(s) | | | |
| Les ennemis ne sont pas créés (retourne false) | | | |

| N° test | Type du scénario | Description | Résultat du test : Le 10/12/2017 |
|--|------------------|--|-------------------------------------|
| 4 | Normal | Deux ennemis sont créés à deux positions différentes | Validé |
| Résultat(s) attendu(s) | | | |
| Les ennemis sont créés (retourne true) | | | |

Explication du choix des tests

Les tests effectués concernent principalement le tableau bidimensionnel de génération d'obstacles, très fréquemment source d'erreur de par leur manipulation particulièrement délicate, ainsi que la création des objets de type Ennemi. Le programme ne possédant que très peu de fonctions renvoyant une valeur via un `return` (pertinentes), il nous est paru difficile d'effectuer d'autres tests que ceux développés dans ce rapport.

Améliorations possibles

Le programme fourni n'est qu'une ébauche du jeu tel qu'on le connaît originellement. De nombreuses améliorations sont possibles, tout comme :

- La création d'une interface « Menu », proposant au joueur de lancer le jeu plutôt que de le démarrer dès l'exécution du programme ;
- La proposition de rejouer lorsque l'utilisateur a perdu ;
- Un système d'enregistrement et d'affichage du score ;
- Un déplacement plus fluide du serpent ;
- Le choix du nombre d'ennemi générés à un instant t lorsque le serpent n'en a pas encore attaqué.

Ces fonctionnalités n'ont pas été développées par manque de temps.