Projet de POCA — Nethack

Université Paris Diderot – Master 2

24 octobre 2012

1 Principe du projet

Le projet de POCA vise à vous faire progresser dans votre capacité *à concevoir un système objet extensible.*

Il se déroule en *deux étapes.* Dans la première étape, un sujet vous est fourni. Ce sujet est *volontairement* décrit de façon informelle, sans vous donner d'indication pour le réaliser. Vous devez donc définir vous-même une *architecture* pour votre projet et vous la décrirez dans le **délivrable 1**.

Une *implémentation* écrite en Scala de cette première version constitue le **délivrable 2**.

Vous recevrez ensuite un *autre sujet* qui est une extension du premier. Cette extension sera objet du **délivrable 3** (plus de détails seront fournis avec le deuxième sujet) et mettra à l'épreuve votre architecture initiale : un projet bien pensé *n'aura pas besoin de modifier* le code de la première version du projet pour traiter cette extension mais seulement de *rajouter de nouveaux composants.*

2 Sujet

Le projet consiste en la réalisation d'un clone du jeu NetHack [3].

Comme souvent est le cas dans l'industrie du jeux vidéos, vous n'avez pas une spécification précise à suivre. Vous devez donc vous documenter sur les règles et les mécanismes du jeu [2, 5, 1] et sur les variantes existantes [4]. À nouveau, comme dans l'industrie du jeux vidéos, toutes manières des mieux comprendre votre objectives sont les bienvenues : étudier le code des jeux *roguelike*, les tester en jouant, etc.

Votre bût initial est de réaliser une implémentation minimale du jeu (p.ex. pas la pêne d'implémenter tout de suite un gazillion des personnages), en vous assurant qui votre architecture permettre d'ajouter plus tard en toute simplicité ce qui manque (p.ex. en ajoutant seulement de donnés, ou en implantant quelques nouvelles classes).

Néanmoins, votre jeu devra permettre de choisir parmi une liste de personnage prédéfinis, de sauvegarder/continuer un jeu, et de jouer un *dungeon* qui change à chaque nouvelle partie. Réfléchissez surtout à comment implémenter et

rendre extensible la liste d'actions que les joueurs peuvent appliquer au contexte dans lequel les joueurs se trouvent.

L'interface du jeu doit être minimale, pas la peine de perdre du temps à préparer des graphismes sophistiqués : si votre jeu aura de succès, des artistes graphiques s'en occuperont plus tard. Pensez donc par exemple à développer une interface minimale ou l'état du jeu est représentez par des caractères textuels comme dans la plus part des variantes existante de NetHack, ... mais rendez simple le changer plus tard avec des graphismes plus sophistiqués ! Pour contenir vos canevas, utilisez un *toolkit* graphique déjà existant, comme par exemple :

- Swing pour Scala : `http://www.scala-lang.org/api/current/scala/swing/package.html`
- Java Curses : `http://sourceforge.net/projects/javacurses/`
- Java GNOME : `http://java-gnome.sourceforge.net/`
- Qt Jambi : `http://qt-jambi.org/`

3 Travail demandé

L'utilisation du SVN est obligatoire. L'historique de ce dernier permettra de déterminer la contribution des membres de l'équipe et la gestion du temps dont vous avez fait preuve.

Avant le 12 novembre 2012

Dans un répertoire `delivrables/architecture/` de votre SVN, vous devez nous soumettre une architecture sous la forme d'un ou plusieurs diagrammes de classes UML et d'autres diagrammes de votre choix accompagnés d'explications justifiant vos choix.

Ce document sera fourni au format PDF et pourra suivre le plan suivant :

interprétation du sujet Vous expliquerez de façon informelle ce que vous avez compris du sujet et de ces enjeux. Quels sont les problèmes techniques et conceptuels que vous avez exhibés ?

concepts Dans cette section, vous définirez les concepts utilisés pour modéliser le problème ainsi que les invariants essentiels du système.

description de l'architecture Vous donnerez ici les diagrammes UML décrivant votre architecture et surtout sa justification.

extensions envisagées Vous énumérez ici les généralisations et les extensions que vous avez imaginées et vous expliquerez pourquoi votre architecture permet de les traiter facilement.

Avant le 10 décembre 2012

Dans un répertoire `delivrables/version1/` de votre SVN, vous devez soumettre une première version de votre système.

Celui-ci doit évidemment compilé à l'aide d'une commande `make` effectuée à la racine de ce répertoire. Un fichier README doit être fourni aussi et doit expliquer comment exécuter votre système.

Votre projet doit aussi être testé. La qualité du code—c'est-à-dire sa correction, sa robustesse et son élégance—sera prise en compte dans la notation.

Références

[1] Absolute beginner's guide for NetHack. http://www.melankolia.net/nethack/nethack.guide.html, 2002. version 1.81.

[2] NetHack homepage. http://www.nethack.org/. retrieved October 2012.

[3] Wikipedia - NetHack. http://en.wikipedia.org/wiki/NetHack. retrieved October 2012.

[4] Wikipedia - roguelike. http://en.wikipedia.org/wiki/Roguelike. retrieved October 2012.

[5] Eric S. Raymond. A guide to the mazes of menace : Guidebook for NetHack. http://www.nethack.org/v343/Guidebook.html. version 3.4.3.