TP Jafar : module *devin*

Thomas Lemaire

13 décembre 2004

1 Pour commencer

1.1 But du TP

Ce TP va vous familiariser avec Jafar en vous guidant tout au long de la réalisation d'un module très simple. Le jeu du *devin* consiste à deviner un nombre choisi par votre partenaire (en l'occurence l'ordinateur) dans un intervalle pré-défini (par exemple $[0, 100]$). Pour cela vous devez proposer un nombre, et votre partenaire vous répond GAGNÉ, PLUS PETIT ou PLUS GRAND. Le but du jeu est de deviner le nombre avec le moins de tentatives possible.

1.2 Installation de Jafar

Vous pouvez consulter la documentation HTML de jafar à `file:///usr/local/jafar/doc/html/index.html`. Ce sujet de TP ne reproduit pas la documentation jafar, mais de nombreuses références y sont faites... Une autre source d'inspiration peut être le module helloworld.

Une branche (au sens subversion tu terme) particulière a été créé pour le TP, afin de récupérer une copie de cette branche :

```
svn co svn+ssh://pollux.laas.fr/home/tlemaire/svnroot_jafar/branches/tp_devin tp_jafar
```

Dans le répertoire tp_jafar/doc/tp_devin, le fichier tp_env.sh peut vous aider à configurer votre environnement (gnu, gcc, tcl, swig, boost, eltclsh,...).

Si besoin est, consulter la page *Install* pour configurer jafar.

Vous pouvez maintenant lancer *configure* et *make* pour les modules que vous venez de récupérer (kernel, jmath et helloworld) et vérifier que tout fonctionne bien. Dans votre shell tcl favori (par exemple l'alias *jafar*) :

```
source tp_jafar/share/jafarInit.tcl
package require helloworld
help helloworld::demoHello
helloworld::demoHello 4
```

2 Module *devin*

2.1 Création du module

Comme indiquer sur la page *Jafar module*, créer un nouveau module *devin*, puis "configure" dans ce module.

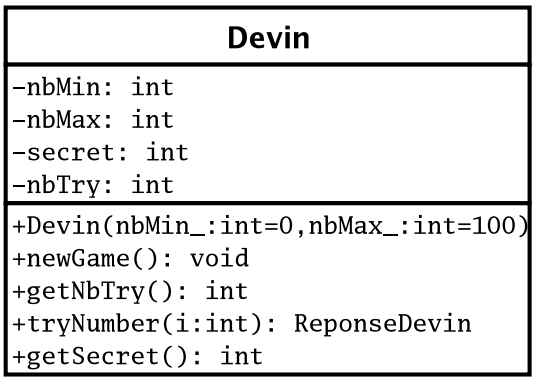

FIG. 1 – Classe Devin

2.2 Classe *Devin*

Recopier les fichiers JAFAR_DIR/doc/tp_devin/devin.cpp,devin.hpp dans les répertoires correspondants de votre module, puis éditez-les si nécessaire. Ces fichiers définissent la classe *Devin* (figure 1). Vérifier que tout compile bien.

Aidez vous de la page *Jafar module* afin de wrapper la classe devin (éditer le fichier include/devin.i)

Compiler... Vous pouvez maintenant essayer de jouer au devin depuis un shell tcl !

2.3 Ajouter une erreur

Nous allons rajouter une erreur lorsque le nombre proposé n'est pas dans $[nbMin, nbMax]$. Consultez *Jafar error* afin d'ajouter une erreur de type INVALID_NUMBER (*per module simple error*), puis modifier la méthode `Devin::tryNumber()`. Vérifier le comportement dans un shell tcl.

2.4 Définir des macros tcl pour rendre le jeu plus conviviale

En C++ comme en C, les types énumérés sont représentés par un entier, le retour de la méthode `Devin::tryNumber()` n'est donc pas très explicite. Écrire quelques macros tcl afin de rendre le jeu plus conviviale. Penser à définir ces macros dans l'espace de nom tcl de votre module (voir la page *Jafar module*).

2.5 Ajouter quelques tests

Dans le répertoire test_suite, ajouter quelques tests pour la classe devin, la page *Unit test in Jafar* vous guidera. Quelques idées de tests basiques :

- le nombre secret est bien dans l'interval $[nbMin, nbMax]$
- la reponse est correcte
- le nombre de coups évolue correctement
- ...

2.6 Documentation

Éditer devin.doxy dans le répertoire doc/ de votre module, puis utiliser la commande *doxygen* dans le répertoire JAFAR_DIR pour générer la documentation. Aller voir le résultat dans JAFAR_DIR/doc/html !

2.7 Utilisation d'un autre module (jmath)

Comme vous l'avez sans doute remarqué, le nombre à deviner n'est pas choisi au hasard. Utiliser la classe UniformDistribution définie dans le module jmath afin de tirer un nombre au hasard.