TDLOG – séance n° 4
Compléments sur Python & interfaces graphiques : TP

Xavier Clerc – xavier.clerc@enseignants.enpc.fr
Cédric Doucet – cedric.doucet@inria.fr
Thierry Martinez – thierry.martinez@inria.fr

14 octobre 2015

À rendre au plus tard le 18 octobre 2015

Cette séance se divise en trois parties : une partie introductive portant sur l'installation de Qt4 pour *Python*, une partie présentant le sujet du TP, et enfin une annexe proposant des liens vers de la documentation pour Qt4.

1 Installation

1.1 Avec conda

L'installation s'effectue simplement en exécutant dans un terminal (ou *shell*, ou encore *invite de commande*) :

```
conda install pyqt
```

ou, pour forcer l'utilisation de la version 4 :'

```
conda install pyqt=4
```

1.2 Sous Windows sans conda

L'installation se fait simplement en téléchargeant et exécutant un installeur téléchargé à l'adresse suivante : http://www.riverbankcomputing.co.uk/software/pyqt/download
Attention : bien choisir un installeur pour *Python* 3, Qt 4, et compatible avec la version de *Python* installée (32 ou 64 bits).

1.3 Sous Linux sans conda (Debian et Ubuntu)

L'installation se fait simplement par le gestionnaire de paquets, en exécutant les commandes suivantes, pour installer respectivement Qt4 pour *Python* 3 et QtDesigner :

— `apt-get install python3-pyqt4`
— `apt-get install qt4-designer`

1.4 Sous Mac OS X sans conda

L'installation est un peu compliqué, et est assez longue (plusieurs minutes) car elle nécessite la compilation de nombreux éléments. La manière la plus simple d'installer l'ensemble des éléments est de suivre les étapes suivantes :

1. installation de Homebrew (http://brew.sh) par `ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`
2. installation de *Python* 3 : `brew install python3`
3. installation de Qt : `brew install qt`
4. installation de SIP : `brew install sip --with-python3`
5. installation de PyQt : `brew install pyqt --with-python3`

Attention : bien suivre les différentes instructions données par `brew`. Typiquement, `brew` demande de modifier le fichier `.bash_profile` pour y définir une variable `PYTHONPATH`.

2 Match 3

2.1 Présentation

L'objectif de ce TP est de vous initier à la programmation des interfaces graphiques, en vous proposant d'implémenter un jeu simple nommé *match 3*, dont vous trouverez un exemple à l'adresse

`http://www.pandajs.net/games/match3`

Ce jeu se présente sous la forme d'une grille composée d'éléments de différents types, différenciés par leur forme ou leur couleur. Pour augmenter son score, le joueur doit permuter dans la grille des éléments adjacents, afin d'aligner, verticalement ou horizontalement, au moins trois éléments de même type. Le nombre de points gagnés grandit avec le nombre de lignes formées. Les éléments alignés sont ensuite éliminés de la grille et, en conformité avec la loi de la pesanteur, les éléments situés au-dessus tombent pour combler l'espace libéré par l'élimination. À chaque instant, la grille satisfait aux conditions suivantes :

— toutes les cases de la grille contiennent un élément ;
— aucune ligne de plus de deux éléments de même type n'est formée ;
— au moins une permutation permet d'aligner trois éléments de même type au moins.

Le jeu se termine quand le temps imparti est écoulé, ou quand le nombre maximum de coups est atteint.

2.2 Travail à effectuer

Le but de l'exercice est de programmer une version simple du jeu *match 3*, dans lequel vous êtes libres de faire les choix qui vous semblent pertinents pour rendre votre application à la fois esthétique, ergonomique et ludique. Vous êtes également libres d'enrichir votre application de fonctionnalités supplémentaires par rapport aux fonctionnalités minimales requises :

- le jeu doit pouvoir se jouer sur une grille 7×7 avec 7 types d'éléments différents ;
- un mode de jeu consistant à faire le maximum de points en 20 coups doit être implémenté ;
- les permutations des éléments doivent pouvoir être réalisables à l'aide de la souris ;
- la formule de calcul du score doit avantager les joueurs qui arrivent à former des lignes de plus de trois éléments, et à former plusieurs lignes avec une même permutation ;
- le jeu ne doit jamais s'arrêter sans l'intervention du joueur, en particulier si vous levez des exceptions.

Pour simplifier l'implémentation, il est raisonnable de représenter le jeu à l'aide de boutons portant des lettres représentant les différents éléments, et de placer ces boutons à l'aide d'une grille en s'inspirant de l'exemple d'utilisation de la classe `QtGui.QGridLayout` présenté à l'adresse suivante :

`http://zetcode.com/gui/pyqt4/layoutmanagement/`

Vous serez principalement évalués sur la qualité d'écriture de votre code. Votre implémentation doit être modulaire : le code de l'interface graphiques doit appeler les fonctions qui définissent le jeu et ses règles, mais pas le contraire. Idéalement, il devrait être possible de coder une interface graphique complètement différente de la vôtre en utilisant uniquement la partie du code qui définit votre jeu.

3 Annexe : Documentation Qt4

Le site officiel de *Python* contient quelques pages à propos de PyQt :

- https://wiki.python.org/moin/PyQt
- https://wiki.python.org/moin/PyQt4
- https://wiki.python.org/moin/PyQt/SampleCode

Une documentation complète, avec notamment la liste des classes disponibles est accessible aux adresses suivantes :

- http://pyqt.sourceforge.net/Docs/PyQt4
- http://pyqt.sourceforge.net/Docs/PyQt4/classes.html

Enfin, un wikilivre (en français) et un tutoriel (en anglais) sont accessibles aux adresses suivantes :

- http://fr.wikibooks.org/wiki/PyQt
- http://zetcode.com/gui/pyqt4