Table des matières

Bibliographie.....1

Objectifs.....1

Travaux pratiques : le juste prix 2
- **Introduction**.....2
- **étape n°1 : organiser et planifier**.....2
- **étape n°2 : réaliser**.....3

Annexe : les fichiers de gestion de projet.....4
- **Changelog**.....4
- **TODO** 4
- **README**.....4

Bibliographie

Tutoriel bouml : http://bpages.developpez.com/tutoriels/bouml/classes-generation/?page=page_4#LIV
Guide utilisateur bouml : http://bouml.free.fr/doc/index.html
Les documents concernant l'utilisation sont sur le serveur de la section.

Objectifs

Ce TP a pour objectif de mettre en application l'utilisation de **Subversion** pour le développement en équipe d'un projet informatique.

Le projet consiste à réaliser le jeu du juste prix : un nombre est choisi aléatoirement entre deux bornes et le joueur doit le deviner en un minimum d'essais. A chaque essai, si le nombre n'est pas le bon, on indique au joueur si celui recherché est supérieur ou inférieur à celui énoncé.

A terme, ce jeu devra être utilisé par plusieurs personnes jouant sur des ordinateurs différents. Ainsi, pour pouvoir les mettre en compétition, un fichier contenant les meilleurs scores devra être partagé et maintenu automatiquement.

Concrètement, voici les fonctionnalités qui devront être disponibles au joueur :

- saisir le nom du joueur
- jouer au jeu
- afficher le nombre de coups utilisés pour arriver à trouver le juste prix
- afficher l'historique des parties
- afficher un classement provenant d'un fichier
- afficher la position du résultat du joueur dans le classement
- sauvegarder le résultat du joueur dans le fichier
- synchroniser automatiquement les modifications apportées sur le fichier

Remarque : ce projet est issu d'un document de Michael Jégat (Corexpert) .

Travaux pratiques : le juste prix

Introduction

Ce projet devra être réalisé sous forme de quatre versions (itérations) contenant les fonctionnalités suivantes :

- Version 1.0 :
  - saisir le nom du joueur
  - jouer au jeu
  - afficher le nombre de coups joués pour trouver le juste prix
  - afficher l'historique des parties

- Version 2.0 :
  - afficher un classement provenant d'un fichier
  - afficher la position du résultat du joueur dans le classement

- Version 3.0 :
  - sauvegarder le résultat du joueur dans le fichier

- Version 4.0 : (non réalisée)
  - synchroniser automatiquement les modifications apportées sur le fichier

Chaque version sera mise en production après validation des tests unitaires. La version suivante de ne sera pas auditée tant que la version précédente ne sera pas complètement validée.

Étape n°1 : organiser et planifier

Après avoir formé des équipes de 3 ou 4 personnes, la première étape va consister à désigner un chef de projet. Cette personne sera responsable de la gestion de projet et servira d'intermédiaire avec le client. Une fois le chef de projet désigné, vous devez réfléchir au projet, aux fonctionnalités et aux choix techniques (structure de données, algorithmes, ...).

Vous devez aussi définir une organisation de travail (dépôt Subversion, gestion des bugs, nommage, ...). Enfin, vous devez planifier les tâches et les attribuer aux membres de l'équipe de projet.

Toutes ces informations devront être écrites sous format informatique (les fichiers Changelog, TODO et README) et devront être disponibles à chaque membre du projet.

A la fin du temps attribué à cette étape, le chef de projet présentera brièvement son plan d'action et des questions pourront être posées au client concernant les choix techniques, le besoin, etc.

Étape n°2 : réaliser

Cette étape va consister à réaliser le projet.

N'oubliez pas de préparer votre dépôt Subversion et de mettre en place vos procédures de compilation (Makefile).

Lorsqu'une version est terminée, vous pouvez contacter le client qui jouera au juste prix. Si des bugs sont trouvés lors de l'utilisation, ils devront être corrigés dans les plus brefs délais. Si plus aucun bug n'est trouvé, la version sera acceptée définitivement et la version suivante pourra être testée.

Ainsi, une version possède trois états :

- en développement : la version est en cours de développement (trunk).
- livrée : la version est livrée chez le client qui l'utilise (tags).
- acceptée : le client a utilisé la version et n'a plus détecté de problèmes d'utilisation. Lorsqu'une version est acceptée, la version suivante peut être livrée si elle est terminée.

Il vous faudra maintenir les fichiers Changelog, TODO et README (ces fichiers faisant partie d'une version livrable au client). La description de ces fichiers est fournie en Annexe.

Annexe : les fichiers de gestion de projet

Changelog

Un ChangeLog, littéralement **Journal des modifications** (en anglais), désigne souvent un fichier qui contient l'énumération de ce que les personnes collaborant à un projet ont effectué comme travail sur ce dernier. Ce fichier est souvent un simple fichier texte, assez brut, avec éventuellement des sections correspondant aux différentes sous-parties du projet. On peut également y trouver des noms de personnes qui ont réalisé ces tâches. Ce terme est directement issu du monde des développeurs de logiciels, notamment celui des développeurs de logiciels libres, afin que tout le monde puisse savoir dans quelle direction le projet a évolué à travers le temps, quelle est sa vitalité (s'il avance beaucoup et vite, s'il est en plein essor ou abandonné depuis un an, ou seulement en phase de corrections de bugs). Dans tous les cas, cela constitue également une invitation à contribuer en sachant quelles sont les dernières évolutions, ce qu'il y a à tester...

TODO

Une todo list (ou **une liste des choses à faire**) est un procédé simple et efficace qui permet de se concentrer sur une tâche d'un projet sans pour autant perdre de vue les autres tâches à accomplir. Les listes à faire se déclinent de multiples façons : par exemple, un chef de projet qui note les bogues à corriger et les fonctionnalités à programmer construit une todo list. Plus trivialement, un post-it avec une liste de courses à faire est aussi une todo list.

Une todo list désigne parfois un fichier qui contient l'énumération de ce que les personnes se fixent comme tâches à réaliser. Ce fichier est souvent un simple fichier texte, assez brut, avec éventuellement des sections correspondant à différents domaines d'action ou aux différentes sous-parties d'un projet.

Pour le travail en équipe, on peut également y trouver des noms de personnes à qui ces tâches échoient.

Les éléments présents dans une todo list sont généralement biffés une fois réalisés, afin de mesurer rapidement l'avancement global. Pour des projets logiciels, ces éléments devraient à terme se retrouver dans le journal des améliorations et modifications apportées au programme ChangeLog.

README

Un fichier readme (**lisezmoi** en français) est, en informatique, un fichier contenant des informations sur les autres fichiers d'un répertoire.

Un tel fichier est généralement un fichier texte appelé « README.TXT », « README.1ST », « READ.ME » ou simplement « README », parfois localisé dans les distributions françaises en « LISEZMOI.TXT », etc. Son contenu varie mais inclut d'ordinaire des instructions d'exploitation, une liste des noms et utilités des autres fichiers, des informations sur la personne les ayant créés, la version de l'application, sa description, voire la licence applicable.

 ©BTS IRIS 2009 [tv]