Projet de POCA — Scénarioo

version 1.0

Université Paris Diderot – Master 2

29 septembre 2015

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 les *requis* (**délivrable 1**), l'*architecture* (**délivrable 2**), et un squelette exécutable de votre projet (**délivrable 3**). Une *implémentation* écrite en Scala de cette première version constitue le **délivrable 4**.

Vous recevrez ensuite un *autre sujet* qui est une extension du premier. Cette extension sera objet du **délivrable 5** (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

Cette année, le sujet porte sur la **génération de scripts de séries télévisées**. Un client a exprimé son besoin de la façon suivante :

> Nous lançons une startup pour vendre un service innovant aux sociétés de production : un assistant à la génération de scénarios de séries télévisées.
>
> Notre système devra simplifier à l'extrême la conception de la trame dramatique de la série, la structuration de chacune de ses saisons, la composition des scripts de chaque épisode, etc. Idéalement, notre client n'aura qu'à choisir les ingrédients initiaux de sa série à partir de bibliothèques thématiques (vampires, morts-vivants, romance, policier, etc) pour que la chaîne de production de la série soit mise en place pour lui. Bien sûr, nous avons une expertise unique dans la construction de scénario : les cliffhangers, la tension dramatique, les profils psychologiques des personnages qui susciteront un maximum d'émotions chez le spectateur, tous les aspects scénaristiques auquels vous pouvez penser n'ont aucun secret pour nous !
>
> Nous prendrons en charge non seulement la création initiale de la série mais aussi la totalité de son cycle de vie : le casting des acteurs, l'organisation du tournage, les changements de scénario en fonction de l'audience et du retour du public, les campagnes de communication sur les réseaux sociaux, ... Tout cela dans l'unique but de maximiser le retour sur investissement de nos clients.
>
> Enfin, notre application sera résolument orientée web : son interface s'appuiera sur les toutes dernières technologies HTML5 pour la construction d'interfaces utilisateurs réactives ! Elle s'utilisera tout aussi facilement sur un ordinateur, une tablette ou un smartphone.

3 Travail demandé

L'utilisation du Git 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. Vous devez utiliser le Gitlab de l'UFR d'Informatique [1].

Le projet est à traiter en groupes de minimum 4, maximum 5 personnes. Compte-tenu de la quantité importante de travail à effectuer, il est essentiel de vous organiser en répartissant le travail entre les différents membres de l'équipe. Vous devez suivre une méthode de conduite de projet et avoir un planning rigoureux vous permettant de tenir les *deadlines*. Les retards seront sanctionnés dans la notation.

3.1 Délivrables

Sauf si on précise le contraire, tous les documents délivrés seront au format PDF.

1. **Manuel d'utilisateur et cahier des charges (deadline : 7 octobre 2015)**
   - Dans un fichier `deliverables/manual.pdf` sur votre dépôt, vous devez nous soumettre un manuel d'utilisateur de votre logiciel. Ce manuel devra décrire précisément les fonctionnalités principales de votre logiciel et comment l'utilisateur de votre application pourra en jouir. Vous pourrez y insérer des captures d'écran imaginaires et toutes illustrations permettant de comprendre le principe d'utilisation du logiciel.
   - Dans un fichier `deliverables/requirements.pdf` sur votre dépôt, vous produirez une liste de propriétés (fonctionnelles et non fonctionnelles) que vous avez identifiées et que votre projet s'engage à respecter. Voir [**?**] et le cours de Génie logiciel avancé (M1).
2. **Architecture et modèle logique (deadline : 15 octobre 2015)**
   - Dans un fichier `deliverables/architecture.pdf` sur votre dépôt, vous devez nous soumettre une architecture sous la forme de plusieurs diagrammes de classes UML et d'autres diagrammes de votre choix accompagnés d'explications justificatives. 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. Attention à fournir différents points de vue à différents niveaux de détails sur votre architecture. Le but de cette partie est de fournir une explication globale de votre application : cette explication doit passer en revue la totalité des fonctionnalités de votre logiciel sans rentrer dans les détails sans intérêt. Choisissez bien le niveau d'abstraction de vos explications !

     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.
   - **Attention :** Ce travail d'analyse et de rédaction est essentielle dans le projet. Vous ne devez pas le négliger et soigner la façon dont vous communiquez avec nous à travers les documents produits.
3. **Version 0 (deadline : 2 novembre 2015)**
   - (en Anglais : *walking skeleton*) Dans un répertoire `deliverables/skeleton/` vous devez nous soumettre une première ébauche de l'implémentation de votre projet. L'implémentation ne devra pas être nécessairement complète, loin de là, mais devra contenir toutes les abstractions prévues par votre architecture (même si beaucoup d'entre elles auront des méthodes non implémentées) et un jeu de tests exécutable pour le code déjà implémenté.

1. http://moule.informatique.univ-paris-diderot.fr:8080/users/sign_in

Le squelette doit évidemment compiler à l'aide d'une commande `make` effectuée à la racine de ce répertoire. Un fichier `README` doit être également fourni et doit expliquer comment compiler et exécuter votre logiciel.
La commande `make check` permettra de lancer les tests unitaires de votre projet. (`ScalaTest` est recommandé pour rédiger ces tests.)

4. **Implémentation (deadline : 14 novembre 2015)**
   Dans un répertoire `deliverables/version1/`, vous devez nous soumettre la première version complète de votre projet.

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