

# ACVL - Jeu de Simulation Boursière

## *Bilan*

Kévin Bonkoski      Roxane Brenier      Jérôme Lapostolet      Chenhao Yan

23 avril 2015

### Table des matières

<b>1</b>	<b>Choix de conception</b>	<b>2</b>
<b>2</b>	<b>Outils utilisés</b>	<b>2</b>
<b>3</b>	<b>Apports du projet</b>	<b>2</b>

## 1 Choix de conception

Dès le départ, nous avons choisi d'éviter de passer notre temps à concevoir et développer une base de données qui aurait servi pour stocker les comptes, les sauvegardes de parties, les caractéristiques du marché du CAC40, etc.

En ce qui concerne le marché, nous aurions également pu récupérer les valeurs du marché via des sites d'informations comme **Yahoo** qui permettent un accès facile aux données financières depuis une application tierce. Cependant cette technique nécessite une connexion à Internet.

C'est pourquoi, nous avons comme idée de ne pas stocker de valeurs et de les prendre aléatoirement à chaque actualisation ou suivant certaines lois de probabilités. Cela favorise la manipulation de grande quantité de données.

Nous avons développé notre programme en C++, car nous voulions dès le départ apprendre à réaliser une interface graphique dans ce langage. En effet, il nous était déjà arrivé, au cours d'un projet, d'en développer une en Java. De plus, nous avons jugé inutile de refaire une base de données comme nous en avons déjà conçue une au premier Semestre.

C'est pourquoi, nous avons préféré travailler d'autres connaissances comme le C++, l'utilisation d'un nouveau logiciel permettant de tracer des diagrammes. En effet, suite au projet de Génie Logiciel, nous avons appris à maîtriser le logiciel *Dia*.

## 2 Outils utilisés

Tout d'abord afin de faciliter le travail en équipe même lorsque nous ne pouvions pas être ensemble nous avons travaillé avec un dépôt *Git* permettant de s'échanger facilement les modifications entre documents et de garder une trace des anciennes versions.

Nous avons donc réalisé les rapports en  $\text{\LaTeX}$ , un des seuls formats d'édition de document facilement compatible avec la gestion d'un dépôt Git. Ce choix permet de plus un rendu très uniforme et une gestion correcte de l'importation des différents diagrammes. Cependant, nous avons réfléchi à l'utilisation d'une application comme *Overleaf* qui aurait permis le partage en ligne et la compilation automatique des fichiers  $\text{\LaTeX}$ . Cependant, cette application en ligne est assez lente quand on manipule des photos et des diagrammes. De plus, cette application n'a pas vraiment d'historique comme peut présenter la commande **gitk** par exemple de git.

Les diagrammes ont été réalisés avec le logiciel *StarUML* qui s'est révélé très pratique et facile à prendre en main. Nous avons choisi ce logiciel car de nombreuses personnes et entreprises le recommandent. Nous avons également beaucoup hésité entre un logiciel que nous pourrions utiliser en ligne en même temps comme *Lucidchart*, mais finalement, nous nous sommes tournés vers un logiciel plus performant.

Enfin l'application a été programmée en C++ à l'aide de l'IDE *Qt Creator* afin de permettre la mise en place d'une interface graphique facilement grâce au framework *Qt*.

## 3 Apports du projet

En parlant avec deux nombreux professionnels du monde de la finance, du logiciel..., nous nous sommes rendu compte que de nombreux anciens étudiants de l'ENSIMAG se retrouvent à

occuper des postes dans les domaines de la maîtrise d'ouvrage et d'œuvre.

Nous nous sommes donc donnés à fond dans ce projet dans l'optique de mieux comprendre et évaluer ce que pouvez être de tels métiers. En effet, nous avons essayé de jouer le jeu en passant beaucoup de temps sur le cahier des charges, la conception et l'architecture en respectant la chronologie des étapes.

Ce projet, nous a permis de prendre conscience que rédiger un cahier des charges détaillé, permettait de clarifier le produit que l'on souhaite concevoir mais facilite également son implémentation.

De plus, parfois dans le monde du travail, certaines entreprises préfèrent faire sous-traiter le développement de l'application une fois que l'architecture a été conçue, le cahier des charges rédigés, etc. C'est dans cette optique que nous avons essayé d'implémenter notre code en pensant qu'il pourrait être amélioré plus tard et en laissant ainsi de façon évidente des parties qui devront être terminées par une personne externe au projet.

En effet, nous avons rédigé le cahier des charges, en gardant en tête par exemple que nous aurions pu quitter le projet en cours et que si un nouveau développeur arrivait sur ce projet il serait en mesure de le reprendre. C'est le cas de notre rendu final, l'application fonctionne mais du travail devra être effectuée en plus, si le client souhaite par exemple que le cahier des charges soit entièrement respecté. Néanmoins, la future équipe en charge possède déjà toute la documentation pour terminer notre travail correctement.