

TP d'ACVL

Jeu de simulation boursière

Sommaire

Sommaire	2
I - Cahier des charges	3
Préambule	3
But du projet	3
Besoins fonctionnels	3
Solutions proposées	3
II - Document d'analyse	5
Cas d'utilisations	5
Diagrammes de séquences systèmes	6
Diagramme de séquence système n°2 - Lancement du Jeu	7
Diagramme de classes d'analyses	8
III - Document de conception	9
Architecture logique	9
Description de l'incrément choisi	9
Diagramme de classe logicielles	10
Diagramme d'état transition	10
V - Manuel utilisateur	12
1. Lancement du jeu	12
Pour lancer la simulation, exécuter les deux commandes suivantes sur le dossier du projet :	
\$ant -f . -Dnb.internal.action.name=rebuild clean jar	
\$java -cp dist/acvl.jar:jar/* Simulation	
2. Présentation de l'interface	12
Partie Gauche	12
Partie Droite	13
3. Remarques	13
Bilan	14
Outils de modélisation utilisés	14
Problèmes rencontrés	14

I - Cahier des charges

Préambule

Ce document a pour but de formuler les besoins du client, au moyen de fonctions détaillant les services rendus par le produit et les contraintes auxquelles il est soumis.

But du projet

Le but de ce projet est d'implémenter une simulation de la bourse du CAC 40 pour un client. Le client pourra ainsi effectuer l'ensemble des actions classiques via la bourse avec des transactions financières, des achats, ventes de produits financiers.

Précision : l'ensemble des transactions s'effectueront sur le marché primaire uniquement, les transactions de titres financiers sur le marché secondaire pourront faire l'objet d'une autre simulation.

Besoins fonctionnels

La simulation devra comporter l'ensemble des éléments nécessaires à la simulation réaliste et simple de la bourse.

La simulation devra permettre :

- De visualiser l'ensemble des produits financiers disponibles sur le marché
- De composer un portefeuille virtuel spécifique au client.
- De pouvoir acheter et vendre les produits financiers disponibles et à disposition
- De mettre à jour les prix des titres financiers au cours du temps
- De toucher des coupons (dividendes) avec les actions détenues

Solutions proposées

Pour pouvoir effectuer l'ensemble des opérations voulus, un certain nombre de fonctionnalités seront implémentées.

Au lancement de la simulation, le client pourra :

- Se connecter à la bourse et pouvoir visualiser les titres financiers disponibles et leurs caractéristiques. Une version future de la simulation pourrait directement se connecter à la bourse et avoir un rendu en temps réel des titres financiers.
- Une fois connecté , le client pourra à travers sa session : acheter des titres financiers via une fonction d'achat, vendre des titres qu'il possède via une fonction de vente, consulter son solde, son portefeuille et enfin quitter la simulation lorsqu'il aura fini.
- Concernant la mise à jour des prix des titres, elle se fera via une fonction `random()` modulo une certaine valeur, mais avec une application future et avec l'accès au prix en temps réel de la bourse, on pourrait calculer la vraie valeur des titres financiers selon l'état du marché via les équations fondamentales de la finance (Black-Scholes notamment).

L'ensemble de la simulation sera piloté par une interface graphique simple où l'utilisateur pourra naviguer facilement parmi les options du simulateur décrites plus haut. L'interface affichera la liste des titres financiers détenus et sur le marché, et l'utilisateur pourra acheter des actions ou les vendre, les cours des actions du marchés seront affichés et rafraîchit tous les 0.5 s.

II - Document d'analyse

Cas d'utilisations

1. Ouverture du Jeu

1.1. Initialisation de l'acheteur avec une trésorerie et un portefeuille d'action vide. La richesse globale correspond à la somme de l'argent présent en trésorerie et de la valeur courante des titres présent dans le portefeuille.

1.2. Émissions des actions dans le marché

2. Choix du titre

2.1. Sélectionne le titre choisi

2.2. Affiche la courbe d'évolution du titre sélectionné

3. Achat

3.1. Achète le titre sélectionné

3.2. Soustrait à la trésorerie de l'acheteur le prix courant du titre

3.3. Affiche le titre dans le portefeuille de l'acheteur, avec sa valeur, et son évolution depuis la date d'émission.

4. Vente

4.1. Vends le titre sélectionné

4.2. Ajoute à la trésorerie de l'acheteur le prix courant du titre

4.3. Retire le titre de l'affichage du portefeuille

5. Quitter la simulation

5.1. Appuie sur la croix du haut pour quitter la simulation, les données sont réinitialisées

Diagrammes de séquences systèmes

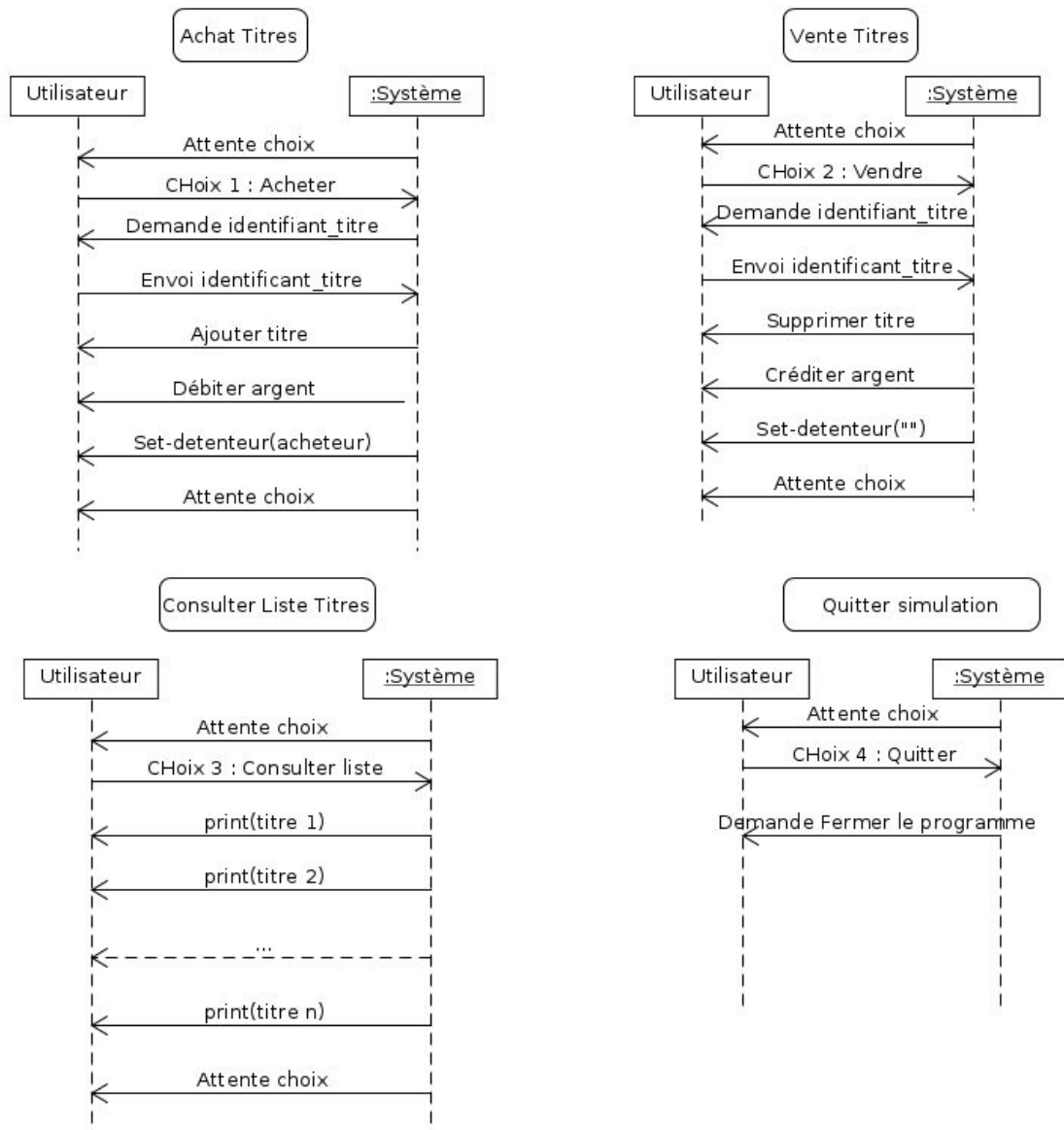


Diagramme de séquence système n°1

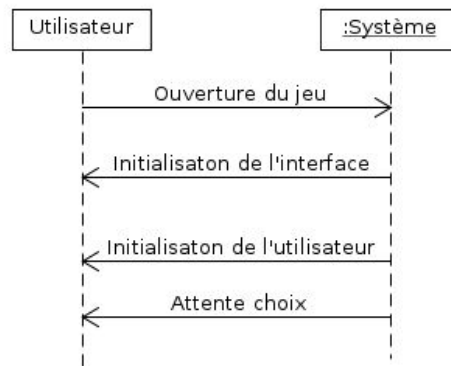


Diagramme de séquence système n°2 - Lancement du Jeu

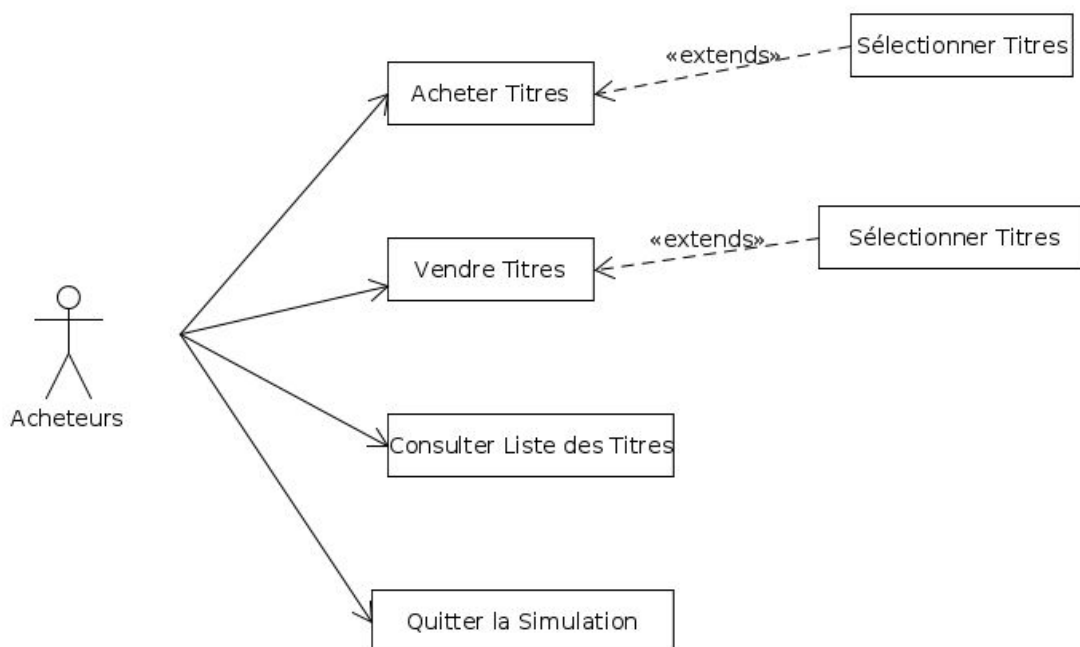
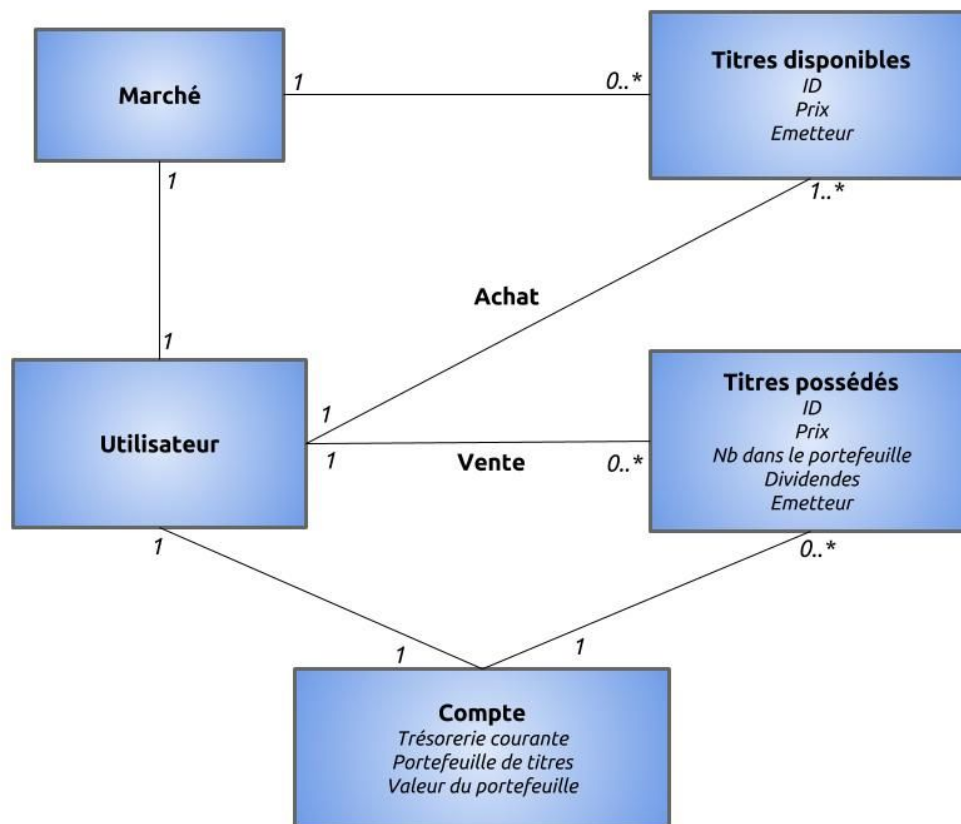


Diagramme de séquence système n°3

Dans le diagramme n°3, on fait la synthèse des cas d'utilisation. En effet, une fois le jeu lancé, on a 5 choix possibles : l'attente, consulter la listes des titres (puis en choisir un), acheter un titre, en vendre un ou quitter la simulation. Enfin, Pour vendre ou acheter un titre X, il faut l'avoir sélectionner au préalable.

Diagramme de classes d'analyses



Ce diagramme a pour but de définir les grands axes du programmes et les interactions qu'ils ont.

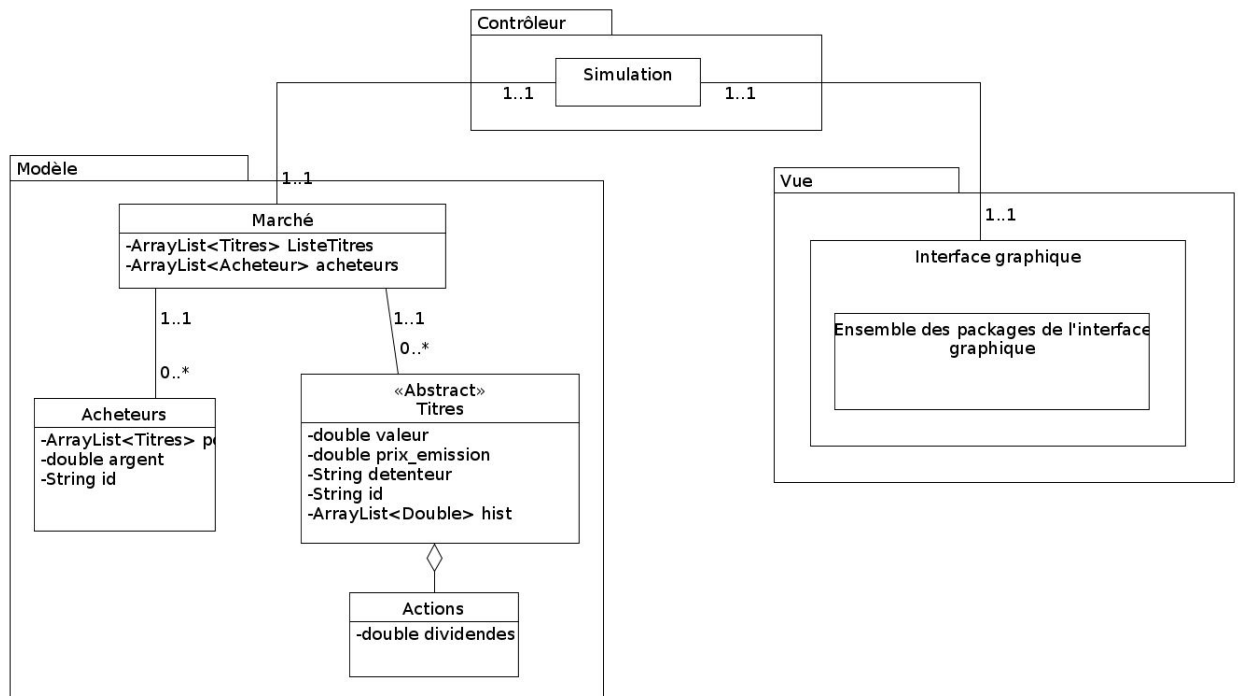
Le jeu permet à un utilisateur d'interagir avec un marché. L'utilisateur possède un compte. Ce compte regroupe la trésorerie courante (l'argent disponible), une liste des titres qu'il possède et la valeur totale de son portefeuille.

Son compte regroupe donc les différents titres. L'utilisateur peut à tout instant vendre un titre qu'il possède.

Le marché contient différents titres. A tout instant, l'utilisateur peut acheter un titre, qui s'ajoute alors à son compte.

III - Document de conception

Architecture logique



Une interface graphique permet à l'utilisateur de d'opérer à diverses actions (achat et vente de titres, consultation des titres disponibles, fin de la simulation). La classe *Simulation* contrôle la simulation.

Description de l'incrément choisi

4 fonctionnalités ont été programmées :

- L'achat de titres : après avoir entré l'identifiant du titre qu'il souhaite acheter, le solde de l'utilisateur est débité du montant du titre, le titre est ajouté à son portefeuille et le champs "possesseur" du titre est désormais initialisé à l'utilisateur.
- La vente de titres : après avoir entré l'identifiant du titre qu'il souhaite vendre, le solde de l'utilisateur est crédité du montant du titre, le titre est supprimé de son portefeuille et le champs "possesseur" du titre est désormais initialisé à 'null'.
- La consultation des titres disponibles : l'utilisateur peut consulter les entreprises dont il peut acheter des titres.
- Sortie de la simulation : l'utilisateur met fin à la simulation.

Diagramme de classe logicielles

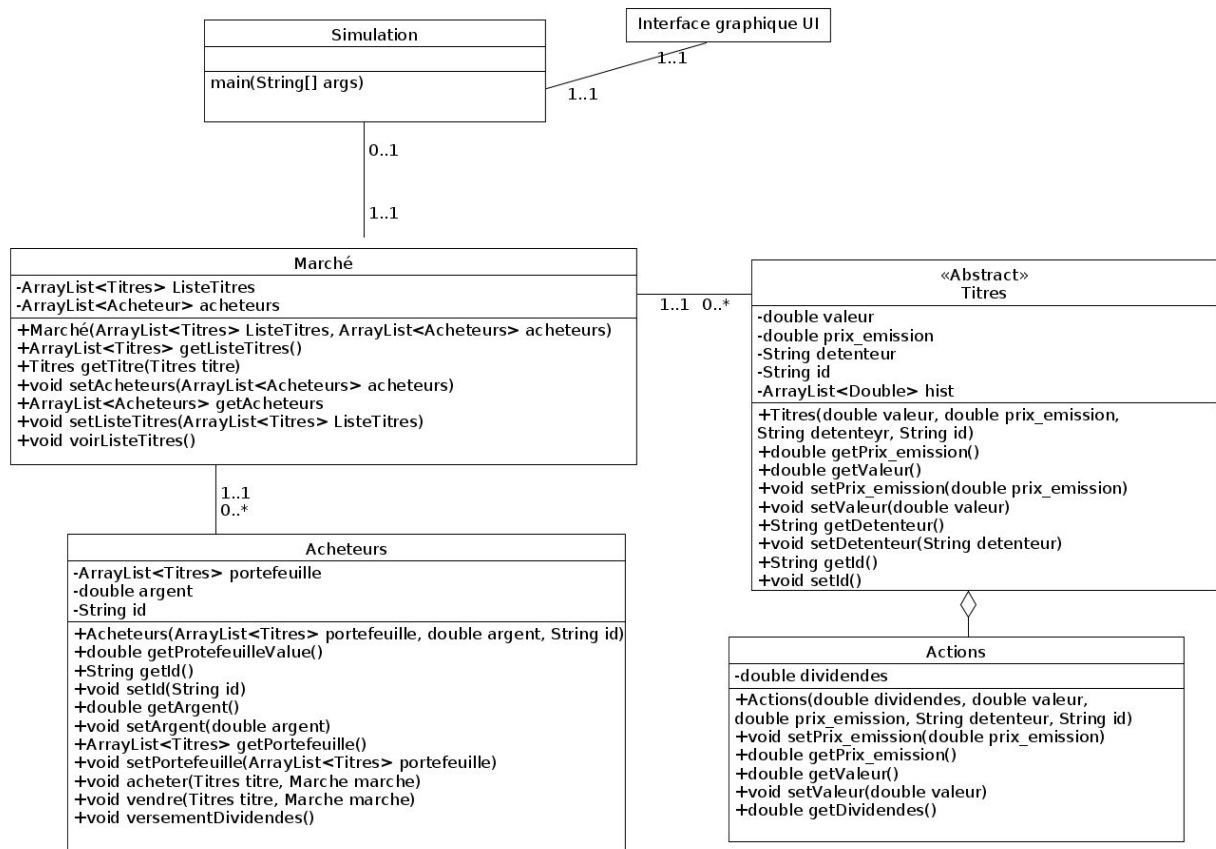
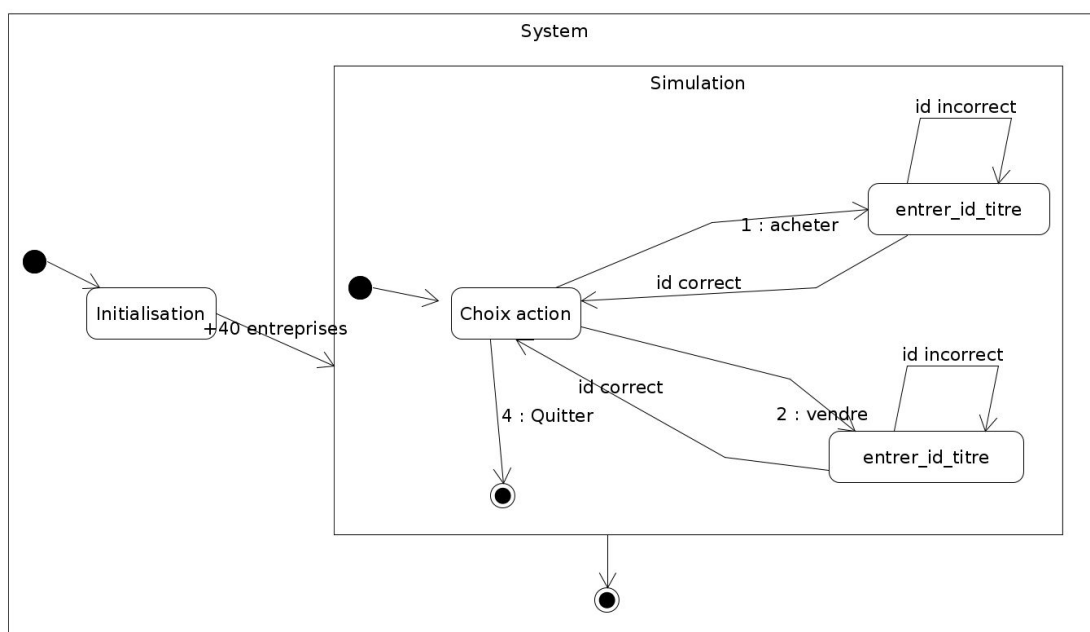


Diagramme d'état transition



NB : Dans le diagramme d'état transition, le terme "action" n'est pas utilisé comme un titre financier décrit par notre programme mais comme le fait d'agir.

Lors de l'initialisation, 40 entreprises sont créées. La simulation démarre dès l'initialisation finie. L'utilisateur a alors le choix entre les 4 fonctionnalités décrites précédemment (achat et vente de titres, consultation des titres disponibles et sortie de la simulation).

V - Manuel utilisateur

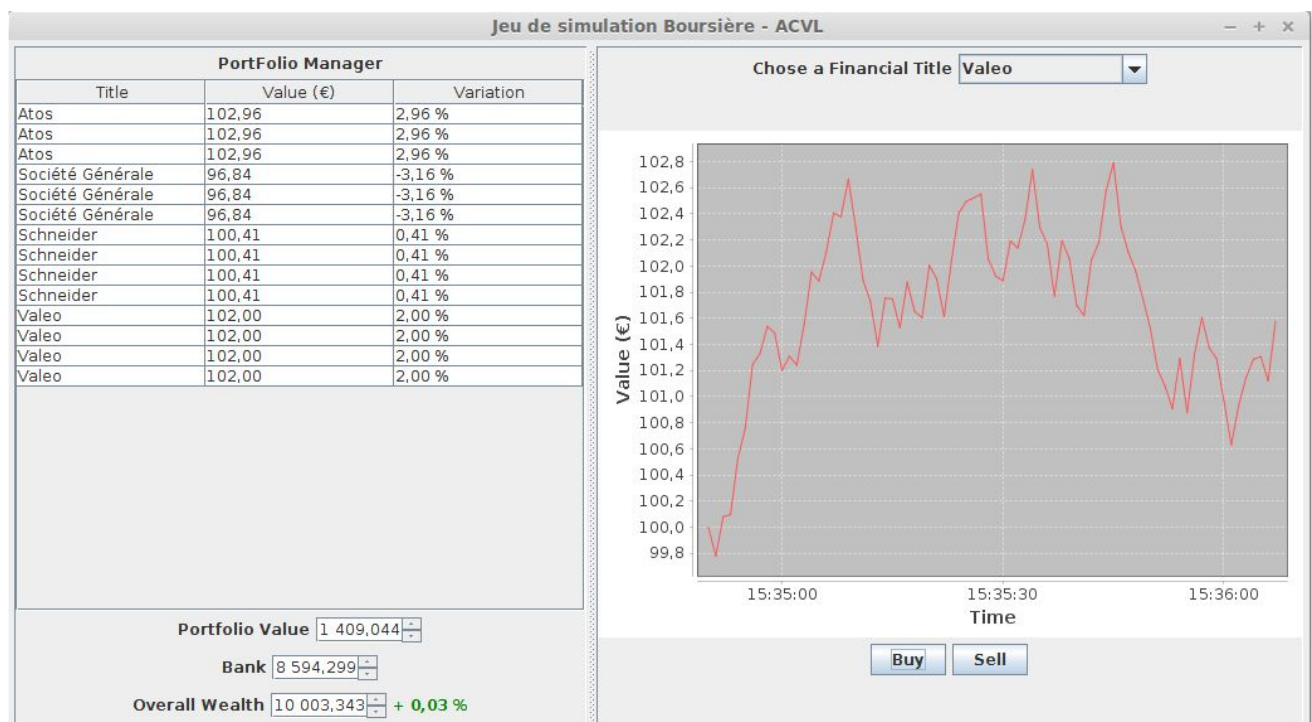
1. Lancement du jeu

Pour lancer la simulation, exécuter les deux commandes suivantes sur le dossier du projet :

```
$ant -f . -Dnb.internal.action.name=rebuild clean jar
```

```
$java -cp dist/acvl.jar:jar/* Simulation
```

2. Présentation de l'interface



L'interface est constituée de deux parties :

❖ Partie Gauche

- La partie à gauche qui présente la composition du portefeuille du joueur : ici on peut voir que le joueur possède 3 actions *Atos*, 3 actions *Société Générale*, 4 actions *Schneider*, et 4 actions *Valéo*.
- Le nombre à côté de *Portfolio value* correspond à la valeur du portefeuille du joueur (en euros), c'est à dire la somme des valeurs de toutes les actions possédées par le joueur.
La valeur à coté de *Bank* correspond à l'argent que le joueur à en banque.

Par défaut ce montant est à 10000 €, et il est débité/crédité à chaque fois que le joueur achète/ vend une action.

- La valeur à côté de `Overall Wealth` correspond à la valeur totale en euros des possessions du joueur, c'est la somme de ce que le joueur a à la banque et dans son portefeuille d'actions. Le but du jeu est de maximiser cette valeur. A côté de cette valeur un indicateur montre si le joueur a gagné ou perdu de l'argent depuis le début de la partie. (ici par exemple le joueur gagne 0,03 % de ce qu'il avait au départ).

❖ Partie Droite

- La partie à droite de l'interface présente les cours des actions disponibles sur le marché. Pour sélectionner une action parmi les celles disponibles utilisez le menu déroulant en haut à côté de la mention `Chose a financial title`.
- Une fois une action sélectionnée le joueur peut voir en dessous l'évolution du cours de cette dernière depuis le lancement de la simulation (i.e depuis le lancement du programme).
- Ici par exemple on a choisi d'afficher le cours de l'action *Valéo*.
- Pour pouvoir acheter ou vendre une action, sélectionnez la dans le menu déroulant et appuyez sur les boutons `Buy` ou `Sell` en bas de la partie droite de l'interface.

3. Remarques

- Théoriquement il est possible d'acheter autant d'actions d'une entreprise que vous voulez, la seule limite est votre compte en banque.
- La colonne `Variation` dans le `Portfolio Manager` (partie gauche de la fenêtre) indique les variations des cours des actions possédées par le joueur depuis le début de la partie. Par exemple si l'action *Valéo* valait 100 € en début de partie et qu'elle en vaut 80 € maintenant, cette colonne affichera -20 %
- Au départ le portefeuille du joueur est vide et il possède 10000 € en banque.
- La simulation des cours des actions se lance dès l'ouverture du programme (même si le portefeuille est encore vide).
- La fréquence de rafraîchissement est fixée à 0,5 secondes comme expliqué dans le cahier des charges. C'est à dire que tous les 0,5 secondes le cours des actions évolue, cette durée pourrait être modifiée si elle ne correspond pas au client (pas dans le jeu lui même mais dans le programme).

Bilan

Outils de modélisation utilisés

- ❖ UMLet - Free UML Tool for Fast UML Diagrams : logiciel open source de dessin de diagrammes UML.
- ❖ Google Draw : outil google pour produire des diagrammes. A servi pour le diagramme de classes d'analyses. En effet il est peu développé, mais simple et rapide à utiliser. Il convenait donc parfaitement à la réalisation de ce diagramme.

Problèmes rencontrés

- Etant donné que nous avons commencé à coder l'interface graphique sous netbeans, nous avons perdu du temps pour trouver comment compiler et exécuter notre jeu depuis la ligne de commande.