

CY Cergy-Paris Université

## RAPPORT

pour le projet de Génie Logiciel  
**L2 Informatique, 2024-2025**

sur le sujet

# Simulation d'une Bourse

rédigé par

**Omar Hachani**  
**Bacem Maziz**  
**Arslane-Mohamed Hamlat**



Avril 2025

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Contexte du projet . . . . .	3
1.2	Objectif du projet . . . . .	3
1.3	Organisation du rapport . . . . .	3
<b>2</b>	<b>Spécification du projet</b>	<b>4</b>
2.1	Notions de base et contraintes du projet . . . . .	4
2.1.1	Fonctionnement général du logiciel . . . . .	4
2.1.2	Notions et terminologies de base . . . . .	4
2.1.3	Contraintes et limitations connues . . . . .	5
2.2	Fonctionnalités attendues du projet . . . . .	5
<b>3</b>	<b>Conception et mise en œuvre</b>	<b>7</b>
3.1	Architecture globale du logiciel . . . . .	7
3.2	Classes de données . . . . .	8
3.3	IHM Graphique . . . . .	8
3.4	Conception des traitements (processus) . . . . .	9
<b>4</b>	<b>Manuel utilisateur</b>	<b>13</b>
4.1	Lancement de l'application . . . . .	13
4.2	Création d'entreprise depuis la fenêtre principale . . . . .	14
4.3	Consultation du marché . . . . .	14
4.4	Gestion du portefeuille . . . . .	14
4.5	Réaliser des transactions . . . . .	16
4.6	Visualisation graphique . . . . .	16
4.7	Simulation automatique . . . . .	17
<b>5</b>	<b>Déroulement du projet</b>	<b>19</b>
5.1	Réalisation du projet par étapes . . . . .	19
5.2	Répartition des tâches, communication et intégration . . . . .	19
<b>6</b>	<b>Conclusion et perspectives</b>	<b>20</b>
6.1	Résumé du travail réalisé . . . . .	20
6.2	Améliorations possibles du projet . . . . .	20

## Table des figures

1	Schéma de l'architecture globale . . . . .	7
2	Package "data" . . . . .	8
3	IHM de la fenêtre principale . . . . .	9
4	Fenêtre principale au lancement du simulateur . . . . .	13
5	Création d'une entreprise depuis la fenêtre principale . . . . .	14
6	Portefeuille utilisateur : actifs détenus et performances . . . . .	15
7	Portefeuille utilisateur : affichage des statistiques . . . . .	15
8	Exemple d'achat d'une action via le menu contextuel . . . . .	16
9	Vente d'un actif depuis le portefeuille . . . . .	16
10	Graphique de suivi de l'évolution des prix . . . . .	17
11	Portefeuille utilisateur : actifs détenus et performances . . . . .	17

## Liste des tableaux

## Remerciements

Nous aimerions remercier nos parents, sans qui rien n'aurait été possible, et qui ont beaucoup sacrifié pour nous. Nous aimerions également remercier le Professeur de ce cours Monsieur Tianxiao Liu pour son approche pédagogique et sa patience durant tout le semestre.

# **1 Introduction**

## **1.1 Contexte du projet**

La bourse est un marché organisé où s'échangent des actions, des obligations et d'autres instruments financiers. C'est un espace où entreprises, investisseurs et institutions se croisent, cherchant soit à lever des fonds, soit à faire fructifier leur capital. Les prix y sont déterminés par l'offre, la demande, mais aussi par l'anticipation des événements futurs.

Aujourd'hui, cet équilibre est plus instable que jamais. Les tensions commerciales entre les États-Unis et la Chine, sous l'impulsion de décisions politiques imprévisibles comme celles de Donald Trump par le passé, continuent de peser sur les marchés mondiaux.

## **1.2 Objectif du projet**

Aujourd'hui, comprendre les mécanismes de la bourse devient essentiel. Notre projet propose un simulateur de bourse permettant à l'utilisateur de s'initier aux dynamiques financières, sans risque financier réel, en prenant en compte non seulement l'évolution naturelle des marchés mais aussi l'impact d'événements économiques majeurs.

Grâce à ce simulateur, l'utilisateur peut expérimenter différentes stratégies d'investissement, voir les conséquences de ses choix, et apprendre à composer avec les aléas du marché.

Nous cherchons à former une intuition boursière, sans passer par la case "perte réelle".

## **1.3 Organisation du rapport**

Ce rapport s'ouvre sur une présentation des notions essentielles liées à la bourse et des contraintes que nous avons retenues pour encadrer le projet.

Nous détaillons ensuite la conception du simulateur, son architecture logicielle et l'interface utilisateur envisagée. Un manuel d'utilisation est proposé pour guider la prise en main de l'application.

Enfin, nous revenons sur le déroulement du projet avant de conclure par un bilan du travail accompli et des améliorations possibles.

## 2 Spécification du projet

Nous avons exposé l'objectif général du projet dans la section 1. Dans cette partie, nous détaillons la spécification fonctionnelle de notre application. Elle s'appuie principalement sur le document de spécification initial (cahier des charges) et définit les concepts de base, les contraintes à respecter, ainsi que les principales fonctionnalités attendues.

### 2.1 Notions de base et contraintes du projet

#### 2.1.1 Fonctionnement général du logiciel

Le logiciel simule un environnement boursier interactif où l'utilisateur peut gérer un portefeuille financier en temps réel, avec les mécanismes suivants.

**Initialisation de la simulation :** Le système crée un marché boursier virtuel composé de  $N$  entreprises configurées avec leurs caractéristiques financières : prix d'action initial, nombre d'actions disponibles et obligations associées (taux fixes ou variables). Les données peuvent être générées aléatoirement ou chargées depuis un fichier CSV.

L'utilisateur a la possibilité d'ajouter manuellement des entreprises en précisant leur secteur, taille et niveau de risque. Le module calcule automatiquement les indicateurs clés (volatilité, volume d'échanges) pour établir un environnement initial de marché réaliste et personnalisable.

**Boucle principale de simulation :** Notre système repose sur une boucle de traitement itérative qui pilote la dynamique de marché.

À chaque cycle, le moteur de simulation :

1. Génère des événements aléatoires.
2. Recalcule l'ensemble des prix d'actions.
3. Met à jour les portefeuilles utilisateur.

**Mise à jour de bourse :** Selon les événements générés, il y aura une mise à jour de la bourse en termes de prix d'actions ce qui permet une évolution progressive et réaliste du marché, tout en gardant une trace complète des fluctuations.

**Interactions utilisateur :** L'utilisateur peut acheter ou vendre des actions et obligations à tout moment. Toutes les transactions sont enregistrées dans son portefeuille, avec un suivi en temps réel des performances et historique des opérations. Le système affiche l'impact des décisions sur la valeur du portefeuille et les conditions du marché.

**Visualisation des fluctuations et les indicateurs économiques :** Le système propose une interface interactive affichant en temps réel les fluctuations boursières sous forme de graphiques dynamiques (cours historiques, tendances sectorielles). Les principaux indicateurs financiers (P/E, volatilité, rendements) sont automatiquement calculés et présentés sous différents formats (tableaux, histogramme, diagrammes ...) pour faciliter l'analyse comparative.

Cette visualisation claire et paramétrable permet à l'utilisateur d'identifier rapidement les opportunités de marché et d'évaluer l'efficacité de ses stratégies d'investissement.

#### 2.1.2 Notions et terminologies de base

**Bourse :** La bourse est un marché organisé où s'échangent des instruments financiers, tels que des actions, des obligations, des devises, des matières premières, etc. Elle permet aux entreprises de lever des capitaux en émettant des actions ou des obligations, et aux investisseurs d'acheter et de vendre ces instruments. Les principales Bourses mondiales incluent le New York Stock Exchange (NYSE), le NASDAQ, la Bourse de Paris (Euronext), etc.

**Entreprise :** Une entreprise est une entité économique qui produit des biens ou des services dans le but de les vendre sur un marché. Elle peut être de différentes tailles (petite, moyenne, grande) et de différents types (privée, publique, coopérative, etc.). Pour notre projet, les entreprises sont cotées en Bourse, ce qui signifie qu'il est possible d'acheter des actifs de cette dite entreprise. Chaque entreprise devrait avoir un capital positif.

**Action :** Une action est un titre de propriété représentant une partie du capital d'une entreprise. Lorsqu'un investisseur achète une action, il devient actionnaire de l'entreprise et a droit à une partie des bénéfices (sous forme de dividendes) qui ne peuvent être distribués que si l'entreprise réalise des bénéfices. La valeur d'une action fluctue en fonction de l'offre et de la demande sur le marché boursier.

**Obligation :** Une obligation est un titre de créance émis par une entreprise, un État ou une autre entité pour emprunter de l'argent sur les marchés financiers. L'acheteur d'une obligation prête de l'argent à l'émetteur en échange d'un intérêt (le coupon) et du remboursement du principal à une date d'échéance prédéfinie (contrainte d'un contrat). Les obligations sont généralement considérées comme moins risquées que les actions.

**Achat :** L'achat en Bourse consiste à acquérir des titres financiers (actions ou obligations) en ayant les fonds nécessaires sur le marché de cette Bourse. L'acheteur devient propriétaire des titres et peut en tirer des bénéfices (dividendes pour les actions, coupons pour les obligations) ou espérer une plus-value lors de leur revente.

**Vente :** La vente en Bourse consiste à céder des titres financiers que l'on détient (actions ou obligations) à un autre investisseur sur le marché. L'objectif de la vente est de réaliser une plus-value (si le titre a pris de la valeur), en évitant de faire une moins-value (si le titre a perdu de la valeur).

**Actualité :** L'actualité désigne l'ensemble des événements et informations récents ayant un impact sur la société, l'économie ou la politique. Dans notre contexte de la bourse, elle englobe les annonces économiques, les décisions politiques et les innovations technologiques qui influencent les marchés. Sa diffusion continue façonne les tendances et les comportements des acteurs économiques.

### 2.1.3 Contraintes et limitations connues

#### Contraintes techniques

Outils de développement :

1. Langage de programmation : Java
2. IDE : Eclipse
3. Outil de rédaction : Latex

Nous sommes également contraints de n'utiliser que ce que Le Professeur T.Liu nous transmet dans ses supports pour les CM [Liu25], en passant outre la documentation sur Internet.

## 2.2 Fonctionnalités attendues du projet

#### Fonctionnalités du programme :

- Créer N entreprises avec des caractéristiques définies (capital, secteur, taille).
- Générer et afficher des événements aléatoires impactant le marché.
- Visualiser l'impact de ces événements sur actions, obligations et indices boursiers.
- Acheter et vendre des actions sur le marché.
- Acheter et vendre des obligations émises par les entreprises.
- Visualiser les cours en temps réel et afficher l'évolution historique.
- Recevoir des dividendes pour les actions détenues.

- Gérer un portefeuille diversifié d'actifs financiers.
- Suivre la performance du portefeuille en temps réel (valeur, gains/pertes, dividendes).
- Consulter un historique détaillé des transactions et frais de courtage.

### 3 Conception et mise en œuvre

C'est une longue section, les sous-sections contenant les sous-sous-sections sont acceptables.

#### 3.1 Architecture globale du logiciel

- **Package data** : Contient les classes de base représentant les entités du domaine (par exemple, Entreprise, Action, Obligation, Portfeuille). Ces classes encapsulent les données essentielles et leurs relations, servant de fondation pour les autres couches du logiciel.
- **Package gestion** : Regroupe les classes responsables de la logique métier, telles que la mise à jour des prix des actifs (Bourse), la gestion des transactions (GestionPortfeuille), et la génération d'événements influençant le marché (Actualite). Ce package agit comme le moteur du simulateur, en orchestrant les calculs et les interactions entre les entités.
- **Package gui** : Fournit l'interface graphique (IHM) avec des composants Swing pour afficher les données, interagir avec l'utilisateur, et visualiser les résultats sous forme de tableaux et de graphiques (via JFreeChart). Ce package assure une expérience utilisateur fluide et intuitive.
- **Package test** : Contient la classe Test qui lance l'application en instanciant la fenêtre principale.

En résumé dans la figure 1 :

1. "test" initialise l'application.
2. "gui" communique avec gestion pour afficher et interagir avec la logique métier.
3. "gestion" manipule les données de chez "data" (actions, entreprises, événements, portefeuille).

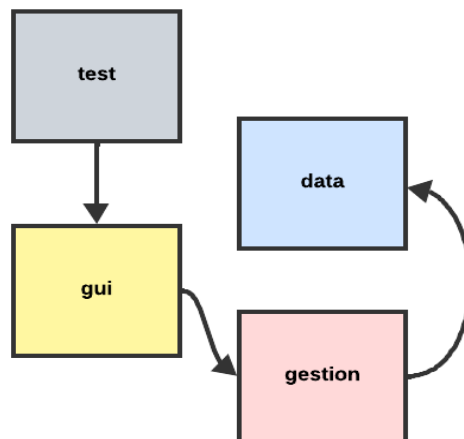


FIGURE 1 – Schéma de l'architecture globale

L'architecture suit une approche orientée objet, avec une séparation claire entre les couches de données, de traitement et de présentation. Les fichiers CSV (`entreprises_bourse.csv` et `events.csv`) permettent une extensibilité des données, tandis que l'utilisation de collections Java (comme `HashMap` et `ArrayList`) garantit une gestion efficace des entités dynamiques.

Le flux de données commence par le chargement des entreprises et des événements depuis les fichiers CSV, suivi par des mises à jour périodiques des prix et des événements via la classe `Bourse`. L'utilisateur interagit avec le système à travers l'IHM, qui affiche les informations en temps réel et permet d'effectuer des transactions.



### 3.2 Classes de données

Le package data définit les entités fondamentales du simulateur, chacune encapsulant des attributs et des comportements spécifiques. Voici un aperçu des principales classes de données (Nous omettrons volontairement de noter les getters/setters) :

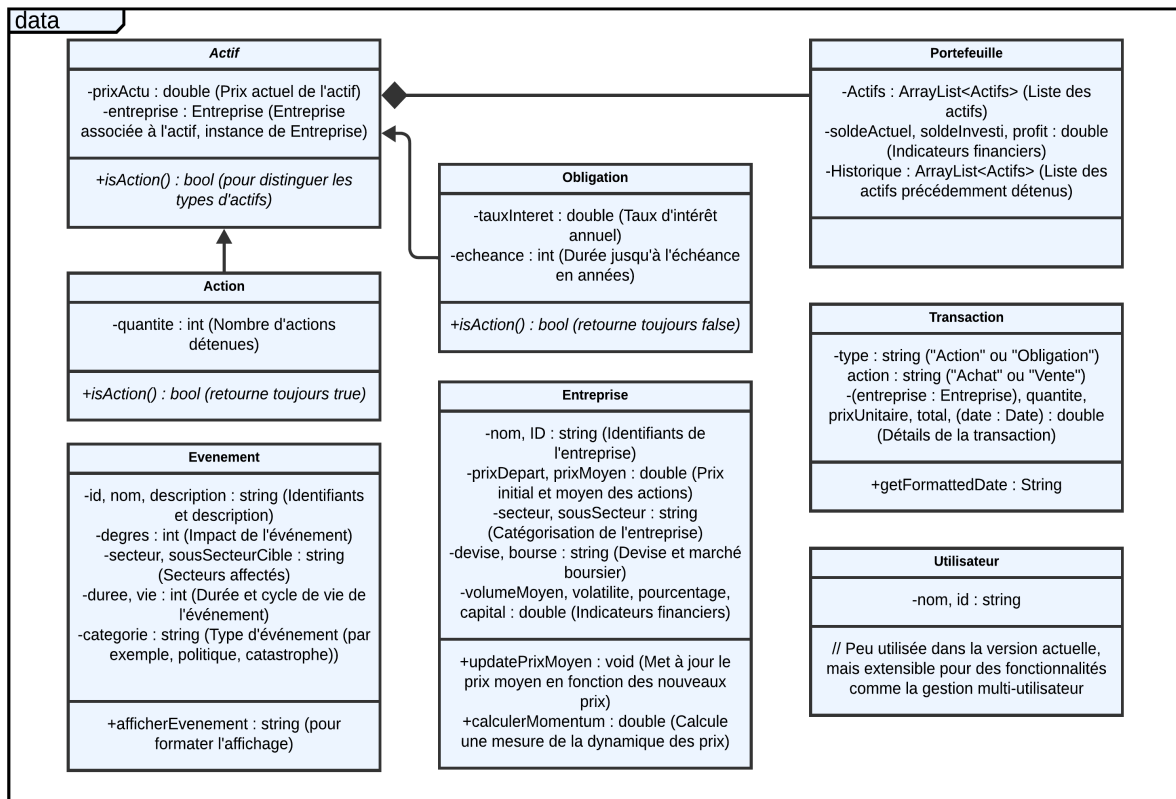


FIGURE 2 – Package "data"

### 3.3 IHM Graphique

**Conception et mise en œuvre de l'IHM graphique de votre projet :** L'interface graphique du simulateur a été conçue pour garantir une expérience utilisateur fluide, en s'appuyant sur une structure claire et modulaire.

L'application repose sur une **fenêtre principale** qui regroupe l'ensemble des fonctionnalités nécessaires à la gestion d'un portefeuille boursier. Elle est organisée autour de plusieurs éléments majeurs :

- **Panneaux d'informations (en haut) :** affichage du solde disponible, du montant investi, du bénéfice/perte et de la santé globale du portefeuille.
- **Barre d'onglets (JTabbedPane) :** navigation entre trois vues principales :
  - **Liste des Actifs :** accès aux actions et obligations disponibles sur le marché.
  - **Mon Portefeuille :** suivi des actifs détenus et des performances financières.
  - **Graphiques :** visualisation des tendances de marché par entreprise et par secteur.
- **Zone de recherche et de filtres :** située sous les onglets, permettant de filtrer les actifs par secteur ou par événement.
- **Panneau des événements :** affichage des événements économiques impactant le marché.
- **Panneau de contrôle (en bas) :** boutons pour avancer le temps dans la simulation ("Tour suivant"), activer la lecture automatique ("Lecture Auto"), ou créer une nouvelle entreprise.

Toutes les informations sont rafraîchies en temps réel pour refléter l'évolution dynamique du marché simulé.

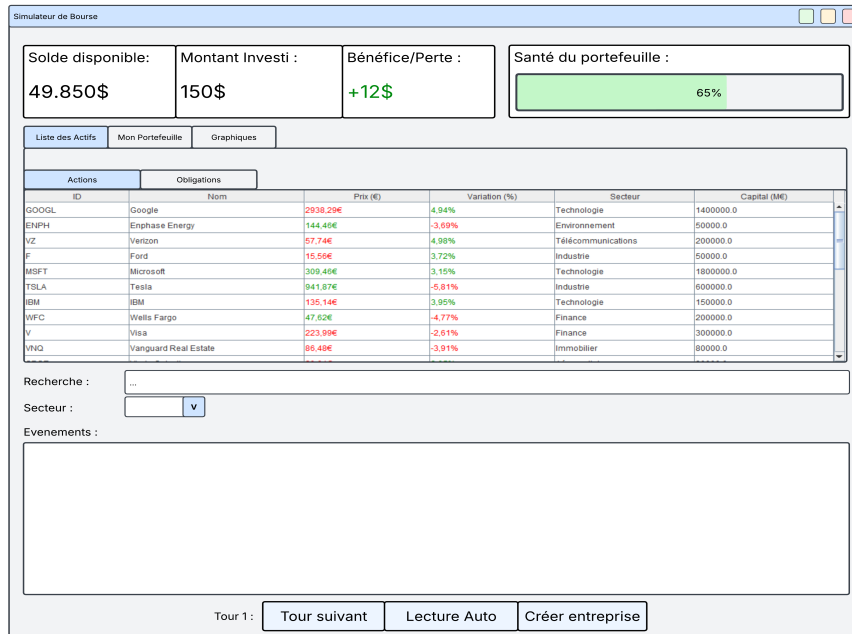


FIGURE 3 – IHM de la fenêtre principale

La figure 3 illustre l’organisation générale de l’interface : la séparation claire entre les différentes zones fonctionnelles assure une navigation intuitive et une bonne lisibilité des données pour l’utilisateur.

### 3.4 Conception des traitements (processus)

Les processus du simulateur sont gérés par le package gestion, qui implémente la logique métier pour la simulation du marché, la gestion des portefeuilles, et la génération d’événements.

Voici les principaux processus :

**Chargement des données :** (Bourse.chargerEntreprises et Actualite.genererlisteevenements)

1. **Description** : Au démarrage, la classe **Bourse** lit le fichier **entreprises\_bourse.csv** pour créer des instances de **Entreprise** (jusqu’à 25 entreprises sélectionnées aléatoirement). La classe **Actualite** lit **events.csv** pour générer des événements.
2. **Mécanisme** : Utilisation de **BufferedReader** pour lire les fichiers CSV. Parsing des lignes en objets **Entreprise** et **Evenement** avec validation des données.
3. **Impact** : Permet une initialisation flexible du marché avec des données externes.

**Mise à Jour du Marché :** (Bourse.Miseajour)

1. **Description** : Met à jour les prix des actions à chaque tour en fonction de plusieurs facteurs :
  - **Événements** : Impact des événements actifs (via **Actualite.Miseajour**) sur les secteurs et sous-secteurs.
  - **Tendance macroéconomique** : Variation aléatoire globale (basée sur une distribution normale).
  - **Volatilité et momentum** : Calculs basés sur les attributs **volatilite** et **pourcentage** des entreprises.
  - **Retour à la moyenne** : Ajustement des prix pour éviter des écarts extrêmes.
2. **Mécanisme** :
  - Calcul d’un nouveau prix pour chaque entreprise en combinant un terme de dérive (*drift*), un terme stochastique et un facteur de retour à la moyenne.

- Mise à jour des attributs `prixDepart`, `pourcentage`, `prixMoyen`, et `capital`.
  - Lissage des variations extrêmes (supérieures à 8%) pour plus de réalisme.
3. **Impact** : Simule un marché dynamique influencé par des facteurs internes et externes.

Le code qui permet cela :

---

**Algorithm 1** Mise à jour du prix d'une entreprise

---

```

1: Calculer  $\mu Momentum \leftarrow \text{entreprise.calculerMomentum}()$ 
2: Calculer  $\mu \leftarrow 0.25 \times \mu Secteur + 0.35 \times \mu Evenement + 0.15 \times \mu Momentum + 0.25 \times$ 
    $tendanceMacro$ 
3: Calculer  $volatilite \leftarrow volatiliteBase + volatiliteEvenement \times 0.8$ 
4: Calculer  $termeDrift \leftarrow (\mu - \frac{volatilite^2}{2}) \times \Delta t$ 
5: Calculer  $secteurEffect \leftarrow \text{secteurShock}[\text{secteur}]$  (par défaut 0.0)
6: Calculer  $termeStochastique \leftarrow (0.65 \times volatilite \times \sqrt{\Delta t} \times \mathcal{N}(0, 1) + 0.35 \times secteurEffect) \times 0.85$ 
7: Calculer  $prixMoyen \leftarrow \text{entreprise.getPrixMoyen}()$ 
8: Calculer  $meanReversion \leftarrow 0.03 \times \ln\left(\frac{prixMoyen}{prixInitial}\right) \times \Delta t$ 
9: Calculer  $nouveauPrix \leftarrow prixInitial \times \exp(termeDrift + termeStochastique + meanReversion)$ 
10: Calculer  $variationPourcentage \leftarrow \frac{(nouveauPrix - prixInitial)}{prixInitial} \times 100$ 
11: if  $|variationPourcentage| > 8$  then
12:   Calculer  $facteurLissage \leftarrow \frac{8}{|variationPourcentage|}$ 
13:   Mettre à jour  $nouveauPrix \leftarrow prixInitial \times (1 + \text{sign}(variationPourcentage) \times 0.08 \times$ 
      $facteurLissage)$ 
14: end if
15:  $\text{entreprise.setPrixDepart}(nouveauPrix)$ 
16:  $\text{entreprise.setPourcentage}(\frac{(nouveauPrix - prixInitial)}{prixInitial} \times 100)$ 
17:  $\text{entreprise.updatePrixMoyen}(nouveauPrix)$ 
18: Calculer  $capitalInitial \leftarrow \text{entreprise.getCapital}()$ 
19: Calculer  $ratioPrix \leftarrow \frac{nouveauPrix}{prixMoyen}$ 
20:  $\text{entreprise.setCapital}(capitalInitial \times ratioPrix)$ 

```

---

**Modèle de calcul des prix :** La mise à jour des prix des entreprises repose sur une adaptation du modèle du mouvement brownien géométrique (GBM, Geometric Brownian Motion), classiquement utilisé en finance pour modéliser l'évolution stochastique des prix d'actifs.

Le prix  $P(t)$  d'une entreprise à l'instant  $t$  est actualisé selon la formule suivante :

$$P(t + \Delta t) = P(t) \times \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}Z\right)$$

où :

- $\mu$  est le taux de drift (tendance moyenne attendue du prix),
- $\sigma$  est la volatilité ajustée,
- $\Delta t$  est l'intervalle de temps (ici fixé à 1 tour),
- $Z$  est une variable aléatoire suivant une loi normale  $\mathcal{N}(0, 1)$ .

Dans notre projet, ce modèle a été enrichi et adapté pour mieux correspondre au contexte d'une simulation ludique :

- **Effets d'événements économiques** : des chocs sectoriels ou macroéconomiques, issus des événements simulés, modifient localement la valeur du drift  $\mu$  et de la volatilité  $\sigma$ .
- **Retour à la moyenne (Mean Reversion)** : une force de rappel est appliquée pour empêcher des écarts de prix trop extrêmes, favorisant une stabilité réaliste autour du prix moyen historique de l'entreprise.

- **Lissage des fortes variations** : si la variation instantanée d'un prix dépasse 8% en un seul tour, un facteur de correction est appliqué pour adoucir le mouvement et conserver la plausibilité du marché.

Ainsi, l'évolution d'un prix au sein du simulateur est déterminée par une combinaison de :

$$P(t + \Delta t) = P(t) \times \exp(Termededrift + Termestochastique + Retourlamoyenne)$$

Cette approche hybride permet de conserver un réalisme suffisant tout en introduisant des dynamiques de marché amusantes et variées pour l'utilisateur.

### Gestion des Événements : (Actualite.Miseajour)

1. **Description** : Gère le cycle de vie des événements, en réduisant leur durée à chaque tour et en générant de nouveaux événements lorsque les anciens expirent.
2. **Mécanisme** :
  - Décrémentation de la durée des événements actifs.
  - Suppression des événements dont la durée atteint zéro.
  - Génération d'un nouvel événement (favorisant les événements positifs avec une probabilité de 60%) à partir de `events.csv`.
  - Ajout de notifications pour informer l'utilisateur des nouveaux événements.
3. **Impact** : Crée une dynamique de marché en introduisant des perturbations périodiques.

### Gestion des Transactions (GestionPortfeuille)

1. **Description** : Gère les opérations d'achat et de vente d'actions et d'obligations.
2. **Mécanisme** :
  - Achat : Vérification du solde disponible, ajout de l'actif au portefeuille, mise à jour des soldes (`SoldeActuel`, `SoldeInvesti`) et enregistrement de la transaction.
  - Vente : Suppression ou réduction de la quantité d'actifs, mise à jour des soldes, et enregistrement de la transaction.
  - Utilisation de `JOptionPane` pour les interactions utilisateur (par exemple, saisie de la quantité).
3. **Impact** : Permet à l'utilisateur de construire et de gérer un portefeuille en fonction des conditions du marché.

### Simulation Automatique (MainWindow.startAutoSimulation)

1. **Description** : Exécute la simulation en continu avec un intervalle de 1,2 seconde par tour.
2. **Mécanisme** :
  - Utilisation d'un `Thread` pour appeler `nextTurn` à intervalles réguliers.
  - Mise à jour de tous les panneaux (`ListeActionsPanel`, `PortfeuillePanel`, `GraphiquePanel`, `EvenementsPanel`, `InfoPanel`) à chaque tour.
  - Possibilité de mettre en pause ou d'arrêter la simulation via des boutons.
3. **Impact** : Offre une expérience immersive en simulant un marché en temps réel.

## Visualisation des Données (GraphiquePanel et PortfeuillePanel)

1. **Description** : Affiche les données du marché et du portefeuille sous forme de graphiques.
2. **Mécanisme** :
  - **GraphiquePanel** : Mise à jour des graphiques linéaires (évolution des prix) et en histogramme (performance par secteur) à chaque tour.
  - **PortfeuillePanel** : Mise à jour du graphique en anneau pour la répartition des investissements par secteur.
  - Utilisation de **JFreeChart** pour une visualisation professionnelle.
3. **Impact** : Aide l'utilisateur à analyser les tendances et à prendre des décisions éclairées.

Le simulateur garantit sa robustesse grâce à des vérifications systématiques, notamment sur le solde disponible lors des achats, ainsi qu'à des mécanismes de lissage pour éviter des comportements de marché irréalistes. Son extensibilité est assurée par l'utilisation de fichiers CSV et la modularité des classes, facilitant l'ajout de nouvelles données. Enfin, la performance est maintenue grâce à l'utilisation de structures efficaces comme **HashMap**, optimisant la gestion d'un nombre modéré d'entreprises.

## 4 Manuel utilisateur

Cette section présente le manuel d'utilisation du simulateur de bourse. Elle décrit les principales fonctionnalités disponibles à travers l'interface graphique, et explique comment naviguer et interagir avec l'application. Des captures d'écran du logiciel y sont intégrées afin d'illustrer concrètement les différentes actions possibles.

### 4.1 Lancement de l'application

Pour lancer le simulateur, il suffit d'exécuter la classe **Test** (contenue dans le package **test**), qui instancie la fenêtre principale (**MainWindow**). Au démarrage, l'interface affiche la liste des entreprises disponibles et permet à l'utilisateur de commencer immédiatement la simulation.

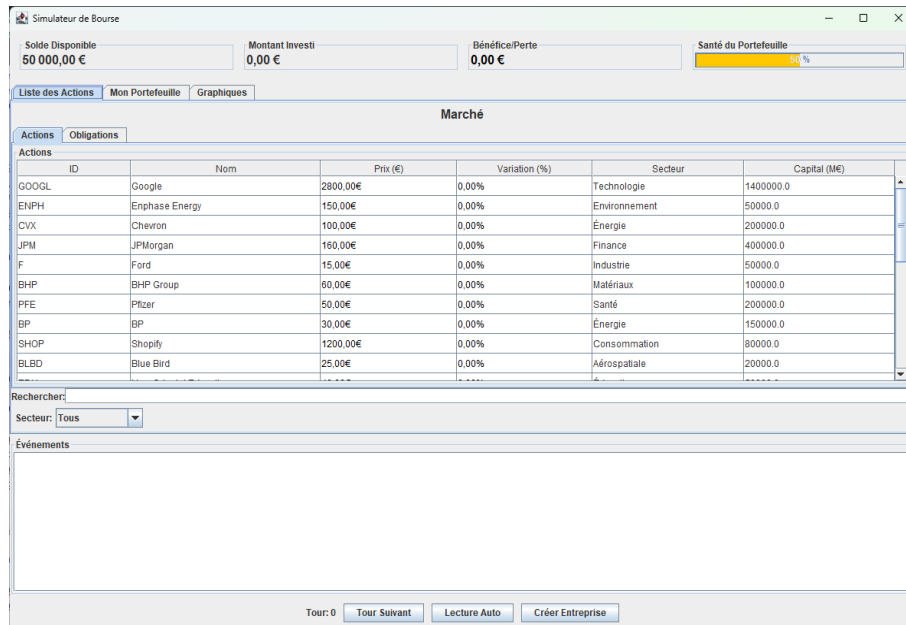


FIGURE 4 – Fenêtre principale au lancement du simulateur

## 4.2 Création d'entreprise depuis la fenêtre principale

Depuis la fenêtre principale, l'utilisateur a la possibilité de créer de nouvelles entreprises afin d'enrichir le marché boursier simulé. Un bouton dédié ouvre une fenêtre de saisie où l'utilisateur peut définir les caractéristiques de l'entreprise : nom, secteur d'activité, capital initial, volatilité et nombre d'actions disponibles.

Cette fonctionnalité permet de personnaliser l'environnement de simulation et d'observer l'impact de nouvelles entreprises sur l'évolution du marché.

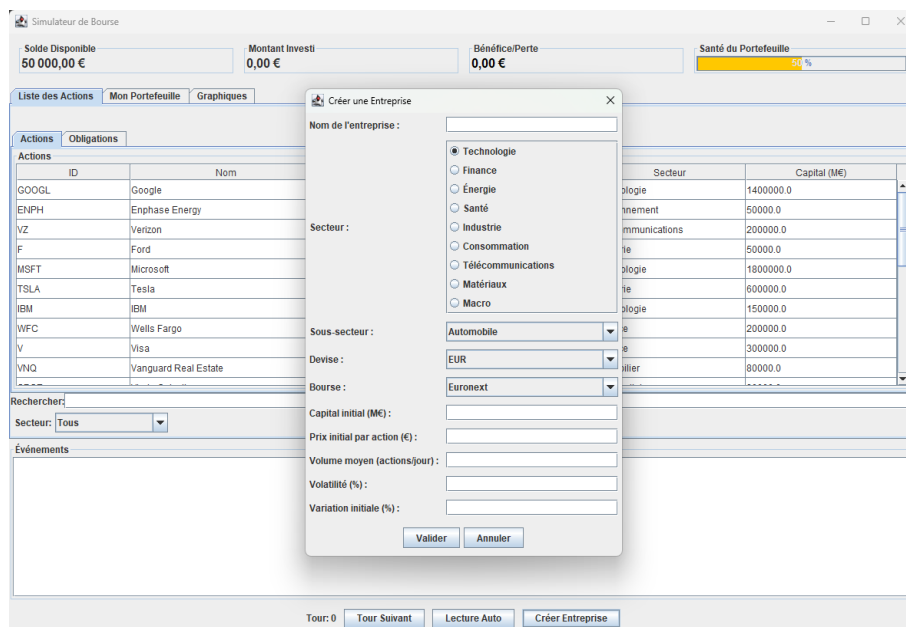


FIGURE 5 – Création d'une entreprise depuis la fenêtre principale

## 4.3 Consultation du marché

L'utilisateur peut consulter la liste des actions et obligations disponibles via l'onglet "Liste des Actions". Chaque actif est présenté avec ses principales caractéristiques (nom, prix actuel, variation, secteur) comme vu sur la figure 4.

## 4.4 Gestion du portefeuille

Dans l'onglet "Mon Portefeuille", l'utilisateur peut suivre les actifs détenus, consulter les statistiques d'investissement (répartition par secteur) et visualiser l'historique des transactions.

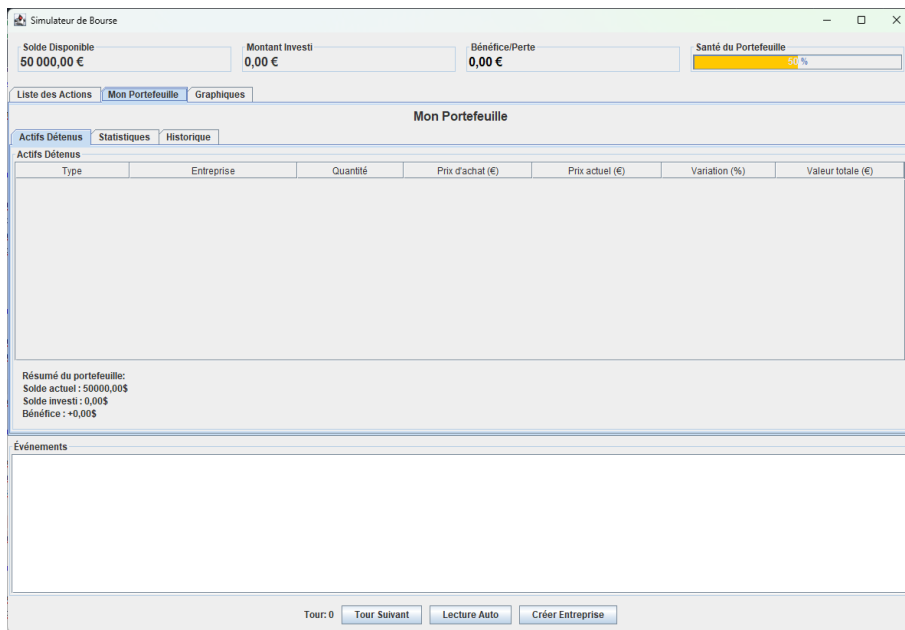


FIGURE 6 – Portefeuille utilisateur : actifs détenus et performances

On peut également consulter les statistiques de notre portefeuille en cliquant sur la rubrique "Statistiques"

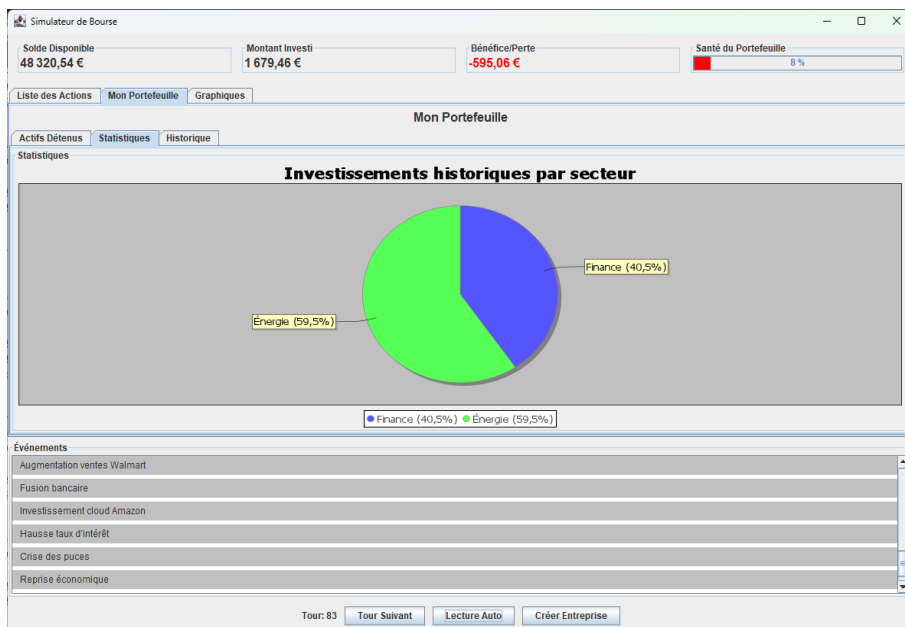


FIGURE 7 – Portefeuille utilisateur : affichage des statistiques



## 4.5 Réaliser des transactions

Depuis la liste des actions ou directement dans le portefeuille, l'utilisateur peut acheter ou vendre des actifs. Des menus contextuels simplifient les opérations (clic droit → acheter/vendre).

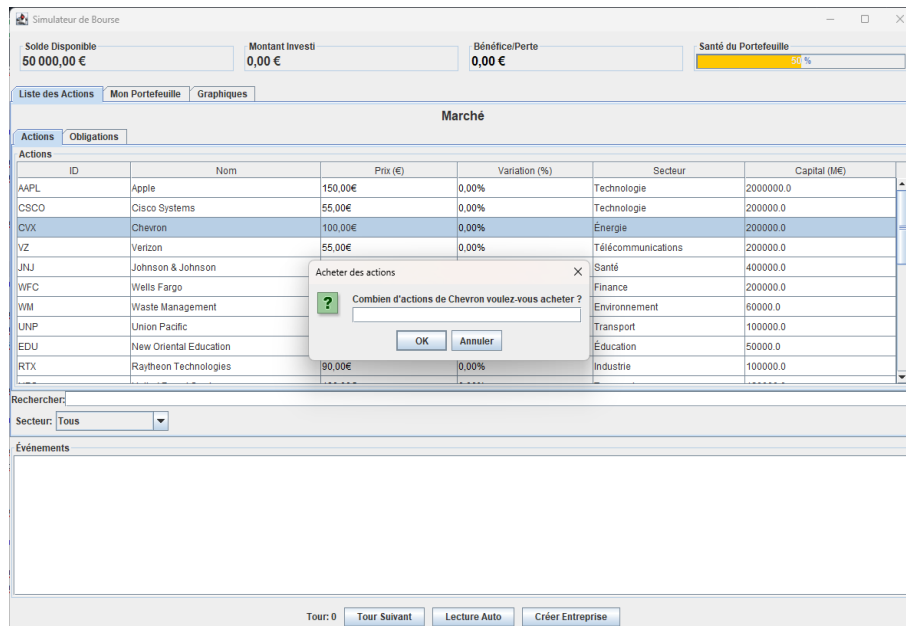


FIGURE 8 – Exemple d'achat d'une action via le menu contextuel

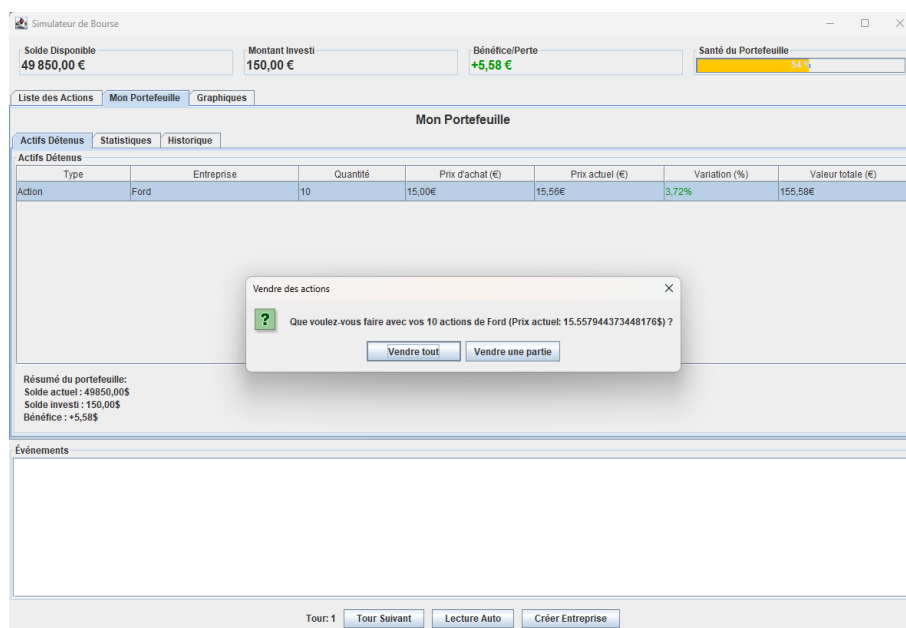


FIGURE 9 – Vente d'un actif depuis le portefeuille

## 4.6 Visualisation graphique

L'onglet "Graphiques" permet de suivre l'évolution des prix par entreprise et la performance par secteur sous forme de courbes et d'histogrammes dynamiques.

Il est possible de filtrer l'affichage des graphiques selon un secteur spécifique en utilisant le menu déroulant prévu à cet effet. Ainsi, l'utilisateur peut concentrer son analyse sur un domaine d'activité particulier, par exemple uniquement les entreprises du secteur technologique ou énergétique.

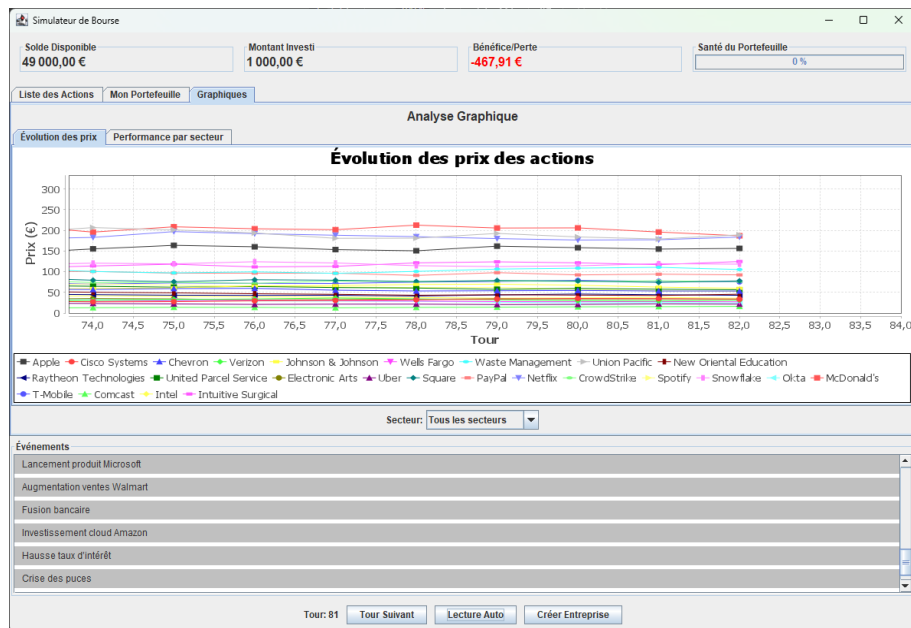


FIGURE 10 – Graphique de suivi de l'évolution des prix

## 4.7 Simulation automatique

L'utilisateur peut activer la simulation automatique pour faire évoluer le marché sans intervention manuelle, en utilisant les boutons prévus dans l'interface de contrôle.

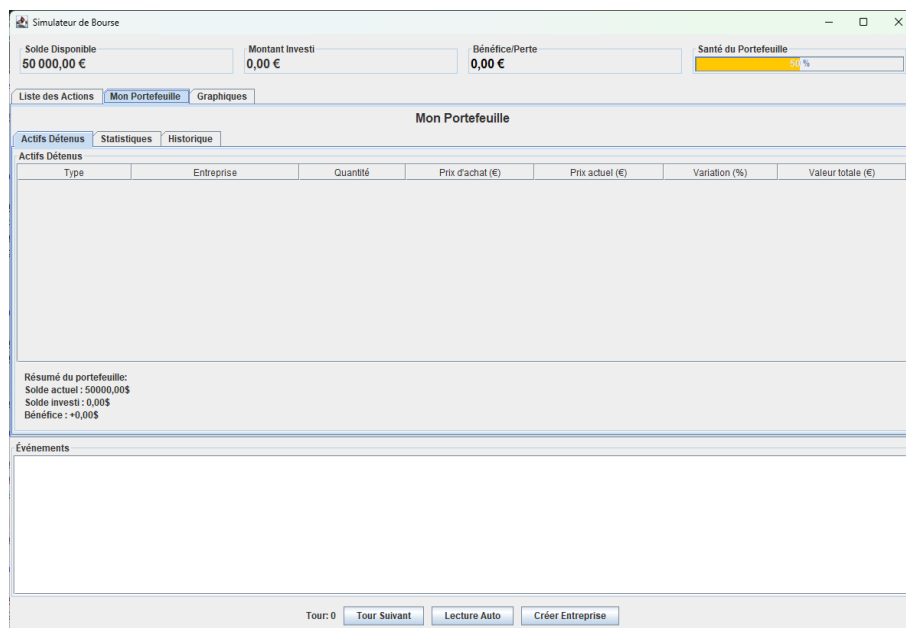


FIGURE 11 – Portefeuille utilisateur : actifs détenus et performances

L'objectif est de réussir à investir pour faire fructifier votre votre solde de départ. L'utilisateur est libre d'adopter différentes stratégies : investissement massif sur un seul secteur, diversification du portefeuille, ou spéculation sur les évolutions rapides des cours. L'outil de simulation permet de tester et d'affiner ses décisions en temps réel, dans un environnement contrôlé.

## 5 Déroulement du projet

Dans cette section, nous décrivons comment le projet a été réalisé en équipe : la répartition des tâches, la synchronisation du travail en membres de l'équipe, etc.

### 5.1 Réalisation du projet par étapes

Le projet a été réalisé en suivant une démarche progressive structurée autour de plusieurs étapes principales :

- **Spécifications fonctionnelles** : définition des besoins du simulateur et rédaction d'un cahier des charges simplifié.
- **Conception des classes et de l'architecture** : création des classes principales (**Entreprise**, **Action**, **Obligation**, **Portefeuille**, etc.) et structuration de l'architecture générale du logiciel.
- **Développement de la logique métier** : mise en œuvre des processus de simulation du marché boursier, gestion des événements, et évolution dynamique des cours.
- **Création de l'interface graphique** : développement de l'IHM en Java Swing, incluant l'affichage des actifs, des portefeuilles, des graphiques et des événements selon ce que nous avons étudié en cours.
- **Tests et validations** : tests réguliers de chaque fonctionnalité ajoutée, corrections de bugs et ajustements ergonomiques au fur et à mesure.

### 5.2 Répartition des tâches, communication et intégration

Même si chacun a contribué à l'ensemble du projet, une spécialisation naturelle s'est formée au fil du développement :

- **Arslane** a travaillé sur la logique des processus internes et sur la création et l'intégration des graphiques.
- **Bacem** s'est occupé principalement de la gestion des données (classes d'entités) et de la gestion dynamique des événements économiques.
- **Omar** s'est principalement chargé de la conception de l'interface graphique et de l'ergonomie de l'application.

La communication entre les membres a été assurée principalement via Discord, complétée par des échanges en présentiel pendant les cours et des sessions de travail à la bibliothèque. Initialement, un dépôt GitHub a été tenté pour le partage du code, mais l'équipe a rapidement basculé sur Discord pour simplifier les échanges et faciliter l'intégration des fichiers.

Des réunions régulières, en présentiel ou en appel, ont permis de suivre l'avancement du projet, de répartir les nouvelles tâches, et de résoudre rapidement les éventuels blocages.

## 6 Conclusion et perspectives

### 6.1 Résumé du travail réalisé

Ce projet de simulation boursière nous a permis de concevoir et de réaliser un logiciel complet, intégrant à la fois la gestion d'un marché d'actions, un portefeuille utilisateur, et des événements dynamiques influençant les cours. Le développement a couvert toutes les étapes classiques d'un projet logiciel : spécifications fonctionnelles, conception de l'architecture, codage modulaire, conception de l'IHM graphique, et mise en œuvre de la logique métier.

Sur le plan technique, nous avons mis en place une architecture orientée objet structurée, facilitant la maintenance et l'évolutivité du code. La simulation s'appuie sur un modèle inspiré du mouvement brownien géométrique, adapté pour intégrer des événements économiques variés et un mécanisme de retour à la moyenne. Une attention particulière a été portée à l'ergonomie de l'interface graphique afin de garantir une expérience utilisateur fluide et intuitive.

Le travail d'équipe a été essentiel pour mener à bien ce projet, avec une communication régulière, une répartition claire des tâches, et une intégration continue des différentes fonctionnalités.

### 6.2 Améliorations possibles du projet

Plusieurs pistes d'amélioration pourraient être envisagées pour enrichir le simulateur :

- **Ajout de nouvelles classes d'actifs** : intégration d'obligations d'État, de matières premières, ou de crypto-monnaies pour diversifier les investissements.
- **Évolution de l'IA du marché** : implémenter des comportements d'achats/ventes automatiques simulant des investisseurs virtuels pour dynamiser encore davantage le marché.
- **Affinage du modèle économique** : améliorer la prise en compte de la corrélation entre secteurs, intégrer des cycles économiques longs et des crises.
- **Amélioration de l'interface** : proposer des tableaux de bord personnalisables, avec des indicateurs avancés (moyennes mobiles, RSI, volatilité implicite).
- **Système de sauvegarde et de chargement** : permettre à l'utilisateur d'enregistrer la progression de sa simulation et de la reprendre ultérieurement.
- **Mode multijoueur** : proposer des compétitions amicales entre plusieurs utilisateurs pour comparer leurs performances d'investissement.

Ces évolutions offriraient un enrichissement considérable de l'expérience utilisateur et prolongeraient l'intérêt de l'application au-delà du cadre académique initial.

## Références

[Liu25] Tianxiao Liu. Introduction au cours de génie logiciel et projet. pages 1–10, 2025.