

UCL

Université
catholique
de Louvain

Louvain School of Management (LSM)



LOUVAIN
School of Management

Programme d'expérimentation sur les comportements en bourse

Mémoire réalisé par

Orlando Palermo

Promoteur

Rudy De Winne

Année académique 2017-2018

Master 120 en Sciences de gestion

Résumé

Ce mémoire décrit la création d'un marché expérimental qui permettra à ses utilisateurs de capter les comportements de ses traders. De plus, une expérimentation a eu également lieu pour à la fois, tester le logiciel ainsi que pour vérifier, à travers son utilisation, l'impact d'un biais comportemental sur des participants. Ce document est décomposé en cinq grands volets.

La première partie comporte **une brève explication sur la finance classique et la finance comportementale**. Elle met en évidence le pourquoi de l'existence d'un tel logiciel. Après cela, le biais comportemental « **priming effect** » est également étudié. C'est ce biais qui sera testé lors des expérimentations. Pour conclure ce chapitre, **une série de marchés expérimentaux ont été explorés** pour comprendre comment ceux-ci fonctionnent.

La seconde partie **explique brièvement les différents mots techniques** et présente la manière dont le marché opère d'une manière générale.

Le troisième chapitre concerne **l'analyse** qui a été exécutée avant le développement de l'outil. Grâce à elle, il sera possible de comprendre : les intentions des différents utilisateurs, le format des données, les états par lesquels passe le marché expérimental et ainsi de suite.

La quatrième partie **va plus loin dans les détails et explicite comment le marché fonctionne en profondeur**. Tous les mécanismes seront montrés et le mode de fonctionnement des robots n'aura plus aucun secret.

Enfin, le dernier chapitre présente **l'expérimentation** ainsi que les résultats obtenus.

Remerciements

M. De Winne, merci. Ce mémoire m'est d'une importance capitale ; il signifie la fin de ma vie d'étudiant pour rentrer pleinement dans la vie professionnelle. Grâce à vous et à vos conseils avisés, ce document a pu être corrigé et finalisé. Une nouvelle fois, merci pour votre aide et votre disponibilité.

Je remercie également M. Tylski pour son aide lors du déploiement du logiciel qui a eu un impact non négligeable sur ce mémoire.

La construction de ce document final a été épaulée par plusieurs autres personnes : ma famille, mes amis ainsi que mon entourage. Sans eux et leur relecture attentionnée, ce mémoire ne serait pas ce qu'il est aujourd'hui. Merci.

Table des matières

ARTICLE I. INTRODUCTION.....	1
ARTICLE II. REVUE DE LA LITTÉRATURE.....	3
SECTION 2.01 FINANCE TRADITIONNELLE ET COMPORTEMENTALE	3
SECTION 2.02 LE BIAIS COMPORTEMENTAL « PRIMING EFFECT ».....	5
(A) COMMENT POURRAIT-ON COMBINER LE « PRIMING EFFECT » A LA FINANCE ?.....	5
(B) APPROFONDISSEMENT DE NOS CONNAISSANCES	6
(C) OMNIPRESENCE DE CE BIAIS	8
SECTION 2.03 MARCHES EXPERIMENTAUX	9
(A) POURQUOI REINVENTER LA ROUE ?	9
(B) MARCHES EPROUVES	10
(C) COMPARATIFS DES MARCHES.....	17
ARTICLE III. LE FONCTIONNEMENT D'UN « DOUBLE AUCTION MARKET »	18
SECTION 3.01 LES ACTIFS FINANCIERS	18
(A) LES ACTIONS.....	18
SECTION 3.02 LE COMMISSAIRE-PRISEUR	19
SECTION 3.03 LE CARNET D'ORDRES	19
SECTION 3.04 LES ORDRES	19
(A) ACHAT OU VENTE (RESPECTIVEMENT BID OU ASK).....	19
(B) BEST BID ET BEST ASK.....	20
(C) LES TYPES D'ORDRES	21
(D) LA QUANTITE.....	22
(E) LE « COUNT »	23
SECTION 3.05 LES TRANSACTIONS	24
ARTICLE IV. PHASE D'ANALYSE.....	25
SECTION 4.01 LE DIAGRAMME DES CAS D'UTILISATION	26
(A) LES PARTICIPANTS	26
(B) LES EXPERIMENTATEURS	27
SECTION 4.02 LA CARTE DE L'APPLICATION	28
(A) SERVER.....	28
(B) CLIENT.....	28

(C) PARTICIPANT	29
(D) CONSUMER.....	29
(E) SYSTEM.....	30
(F) LES RESSOURCES.....	30
SECTION 4.03 LES ETATS DU MARCHE.....	31
SECTION 4.04 LE DIAGRAMME DE CLASSE.....	32
(A) UNE ATTENTION PARTICULIERE SUR LA CLASSE ORDER	33
SECTION 4.05 ANALYSE DE LA BASE DE DONNEES	34
(A) ORDER	34
(B) TRANSACTION.....	35
 <u>ARTICLE V. LA CONCEPTION DE L'OUTIL</u>	 <u>36</u>
SECTION 5.01 INTRODUCTION.....	36
SECTION 5.02 LES OUTILS UTILISES	36
(A) DJANGO	37
(B) OTREE	38
SECTION 5.03 ARCHITECTURES.....	39
(A) LE MARCHE EXPERIMENTAL.....	39
(B) LES ROBOTS.....	42
SECTION 5.04 REcul CRITIQUE/PERSPECTIVES	49
 <u>ARTICLE VI. EXPERIMENTATION.....</u>	 <u>50</u>
SECTION 6.01 QUID DE LA PROBLEMATIQUE ?.....	50
SECTION 6.02 HYPOTHESES.....	51
SECTION 6.03 MISE EN PLACE INITIALE	52
(A) LES PARTICIPANTS	52
(B) CONFIGURATION D'UNE PHASE D'EXPERIMENTATION	52
(C) OUTILS STATISTIQUES UTILISES.....	53
(D) FORMATAGE DES DONNEES.....	54
SECTION 6.04 TRAITEMENT DES DONNEES.....	54
(A) METHODE DE CREATION DES VARIABLES.....	54
(B) ANALYSE DES VARIABLES.....	57
SECTION 6.05 RESULTATS DE L'EXPERIMENTATION	57
(A) ANALYSE APPROFONDIE DE « QT_MARKET_ASK »	58
SECTION 6.06 REcul CRITIQUE.....	59

<u>ARTICLE VII.</u>	<u>CONCLUSION</u>	<u>60</u>
<u>ARTICLE VIII.</u>	<u>BIBLIOGRAPHIE</u>	<u>61</u>
<u>ARTICLE IX.</u>	<u>ICONOGRAPHIE.....</u>	<u>63</u>
<u>ARTICLE X.</u>	<u>ANNEXES</u>	<u>65</u>
SECTION 10.01	COMMENT LANCER UNE EXPERIMENTATION ?.....	65
SECTION 10.02	OU ET COMMENT CREER UN NOUVEAU FICHIER OU SERONT STOCKES LES SYMBOLES ?.....	67
SECTION 10.03	DIAGRAMME DE CLASSE COMPLET	68
SECTION 10.04	FICHIER RECAPITULANT LE BEST BID ET LE BEST ASK DES CARNETS D'ORDRES	68
SECTION 10.05	OU PUIS-JE RECUPERER LES DONNEES DE CONFIGURATION ?.....	68
SECTION 10.06	Liste exhaustive des variables obtenues.....	69
SECTION 10.07	LIEN DE LA VIDEO POUR EXPLIQUER LES PRINCIPAUX CONCEPTS DU MARCHE EXPERIMENTAL.....	70

Liste des tableaux

Tableau 1 - Résultats des temps d'attentes des différents groupes.	7
Tableau 2 - Tableau comparatif des marchés étudiés dans ce mémoire.	17
Tableau 3 - Démonstration du "d" aléatoire du "Loader Bot".	44
Tableau 4 - Démonstration du "d" constant du "Loader Bot".	44
Tableau 5 - Tableau des préférences du "Player bot".	45
Tableau 6 - Informations de départ en vue de calculer le nouveau prix BID/ASK.	47
Tableau 7 - Démonstration de la formule sans ou avec l'injection du spread.	48
Tableau 8 - Liste des effets supputés par l'utilisation du "priming effect".	51
Tableau 9 - Liste des actions utilisées lors de l'expérimentation.	53
Tableau 10 - Liste non exhaustive des variables créées pour tester les hypothèses + procédures pour les obtenir.	55
Tableau 11 - Vue partielle des variables créées.	56
Tableau 12 - Résultats de l'expérimentation.	57
Tableau 13 - Liste exhaustive des variables créées pour tester les hypothèses.	69

Liste des illustrations

Figure 1 - Arbre décisionnel du marché de Maslov.	12
Figure 2 - Image provenant du document «Building the model of artificial stock market based on JASA » de Kang et de Sun.	14
Figure 3 - Exemple d'un carnet d'ordres.	20
Figure 4 - Carnet d'ordres avant d'avoir reçu un ordre ASK.	22
Figure 5 - État du carnet d'ordres après avoir reçu un ordre ASK.	23
Figure 6 - Augmentation du "count".	24
Figure 7 - Diagramme de cas d'utilisation d'un participant.	26
Figure 8 - Diagramme de cas d'utilisation d'un expérimentateur.	27
Figure 9 - Carte du marché expérimental de ce mémoire.	28
Figure 10 - Présentation des états du marché expérimental.	31
Figure 11 - Diagramme de classe.	32
Figure 12 - Détails de la classe "Order".	34
Figure 13 - Détails de la classe "Transaction".	35
Figure 14 - Formulaire pour placer un ordre.	40
Figure 15 - Affichage des BID, ASK et des transactions.	40
Figure 16 - Illustration des mécanismes « Insertion » et « Rotation ».	41
Figure 17 - Exemple d'un fichier pour créer des symboles.	42
Figure 18 - Présentation de la configuration du symbole "AAPL" avec une distribution aléatoire.	43
Figure 19 - Présentation de la configuration du symbole "AAPL" avec une distribution constante.	44
Figure 20 - Arbre décisionnel du robot "Player" lorsqu'il n'a pas de préférence.	46
Figure 21 - Arbre décisionnel du robot "Player" lorsqu'il a une préférence pour les BID.	46
Figure 22 - Données brutes provenant de la base de données.	54
Figure 23 - Graphique de la variable "qt_market_ask".	58
Figure 24 - Accueil pour créer une session.	65
Figure 25 - Formulaire pour créer une session.	65
Figure 26 - Bouton pour configurer une session.	65
Figure 27 - Paramétrage d'une session.	66
Figure 28 - Exemple d'un fichier pour créer des symboles.	67
Figure 29 - Diagramme de classe complet.	68
Figure 30 - Représentation du fichier "StateOrderBook".	68

Article I. Introduction

« Confirmation bias », « hindsight bias » ou encore « anchoring effect »... Pour peu que vous soyez dans le monde de la finance, vous avez déjà très probablement entendu ces noms. Ceux-ci explicitent soit des raccourcis que nous prenons, soit des habitudes innées qui faussent notre objectivité. La plupart d'entre eux proviennent de nos ancêtres qui devaient analyser avec une rapidité accrue les informations environnantes afin d'éviter de devenir le casse-croûte d'un carnivore aux dents aiguës.

Analyser nos comportements nous permet de réfléchir avant d'agir, de mieux cerner nos automatismes ; de nous souvenir que nous ne sommes pas infallibles. Des points cruciaux surviennent ainsi : comment analyser tout cela ? Par quels moyens ou outils ? Comment ne pas biaiser les données récoltées alors qu'elles-mêmes concernent des biais ?

C'est pour répondre à ces questions que ce mémoire a vu le jour. Le thème permettant d'aborder ce contexte est celui des marchés expérimentaux. Plus précisément, il s'agira d'un logiciel boursier simplifié permettant de passer différents types d'ordres. Les utilisateurs seront guidés par les différentes fluctuations, visibles, d'un carnet d'ordres où les expérimentateurs pourront mesurer et capter les erreurs humaines souhaitées ou non.

L'arborescence de ce document commencera par la littérature scientifique qui se déclinera elle-même en trois volets. Le premier volet comportera des explications sur la finance comportementale ainsi que sur son existence. Le deuxième traitera du biais comportemental « priming effect » qui sera, par la suite, étudié dans ce mémoire à travers une expérimentation se déroulant dans notre marché expérimental. Le troisième et dernier volet concernera l'étude des marchés expérimentaux existants et en quoi leur compréhension a été bénéfique pour le nôtre.

Après avoir lu la partie théorique sur les marchés expérimentaux, il se pourrait que certains termes techniques paraissent obscurs. En effet, le monde de la bourse regorge de notions importantes à assimiler. Dans le but de dissiper tous les problèmes de compréhension, un chapitre sera dédié à l'explication de tous les principaux concepts que l'on pourrait y retrouver.

Ensuite, nous alternerons avec l'analyse de l'outil tout en prenant soin d'évoquer son architecture, ses capacités et ses limites. Son développement joue un rôle on ne peut plus important dans ce mémoire : il incarnera d'une part l'interface entre les utilisateurs et le marché créé de toute pièce, et d'autre part, sera l'éponge numérique autorisant les expérimentateurs à analyser leurs réactions et leurs choix.

Enfin, une expérimentation sera également lancée pour tester la stabilité du logiciel et sera aussi l'occasion pour lui, de faire ses premiers pas dans le monde de la finance comportementale en étudiant le biais du « priming effect ».

Article II. Revue de la littérature

Section 2.01 Finance traditionnelle et comportementale

La finance classique repose sur deux piliers que sont les « beautiful people » et les « beautiful markets » (De Bondt, Muradoglu, Shefrin, & Staikouras, 2008). Le premier pilier cité est caractérisé par des individus ayant des pensées parfaitement rationnelles écartant tous types d'erreurs et qui traitent infailliblement les nouvelles informations tout en ajustant leurs croyances de façons précises et correctes. Pour ce faire, leur gestion des probabilités est sans faille et annihile l'hésitation. Ces derniers persévèrent dans une quête de maximisation de richesse tout en minimisant les risques encourus. Quant au second, il atteste que les marchés sont parfaits, liquides et compétitifs (De Bondt et al., 2008).

De nombreux auteurs se sont interrogés sur ce paradigme qui a été employé pendant plusieurs décennies. Selon leurs expérimentations, il a été avéré que les agents ne sont pas toujours logiques et commettent des erreurs. Une liste dressée par Wagdi (2007) met en évidence cette réalité et montre que l'étude de la non-rationalité des acteurs sur le marché prend ses racines en 1738 avec Bernoulli et se poursuit par d'autres illustres auteurs tels que Kahneman, Tversky, Thaler, etc.

Ceux-ci ont démontré que les agents ne suivent pas ce modèle et s'exposent à des fautes qui peuvent amener à une baisse des gains ou, dans le pire des cas, à des pertes. En citant De Winne et D'Hondt (2017) : « les « homo sapiens » que nous sommes ne ressemblent pas à cet « homo economicus » décrit par Thaler et Sunstein (2009) ». En soi, l'homo economicus reste toujours un Homme et reste soumis à la célèbre maxime « *errare humanum est* ».

Néanmoins, la finance comportementale n'est pas là uniquement pour clamer haut et fort ces incohérences. Son existence réside dans le fait qu'elle propose une approche différente pour comprendre ces incohérences qui causent pertes et erreurs. En outre, elle permet de passer de l'homo economicus à l'homo sapiens. Nous pouvons la voir comme une sorte de pont qui relie le domaine de la finance classique à celle de la psychologie (De Bondt et al., 2008).

Afin de démontrer cette phrase, utilisons un exemple concret. Au sein de la finance classique, les agents se servent d'une ligne directrice connue sous l'appellation de « l'utility theory ». Quand ceux-ci doivent faire un choix, ils appliquent cette théorie en suivant minutieusement d'autres règles appelées axiomes. Comme Kahneman l'explique : « Cette théorie joue un double rôle : c'est une logique qui prescrit comment les décisions devraient être opérées et indique également le mode de fonctionnement des agents¹ ».

Pour contraster les propos sur « l'utility theory », il faut changer de point de vue et se diriger vers la psychologie. En cette matière, Kahneman et Tversky (1979) ont remarqué à travers leurs travaux que la sensation ressentie en cas de pertes et de gains n'est pas du tout la même. Pour mentionner Kahneman : « losses loom larger than gains ». En « layman's terms », nous souffrons plus lorsqu'il y a des pertes et nous nous réjouissons moins quand il y a des gains.

C'est à partir de ce moment que la finance comportementale entre en piste. Il fallait mettre en évidence le pourquoi du comment des réactions non logiques des agents et croiser ces résultats avec la psychologie. *In fine*, Kahneman et Tversky ont mis au point des explications et ont donc créé la « prospect theory ». Celle-ci montre que nous effectuons nos prises de décisions sur base d'un point de référence et non sur base de la richesse finale.

Penser que la finance comportementale est une alternative à la classique est une erreur. Toutes les deux possèdent à la fois des défauts ainsi que des arguments pour répondre à certains types de problèmes.

L'objectif de ce mémoire est d'élaborer un marché expérimental qui sera utilisable à de multiples fins. D'une part, il a un objectif résolument tourné vers des notions quantitatives. En effet, de nombreuses données pourront être gérées et sauvegardées. D'autre part, des notions plus axées vers la psychologie pourront y être intégrées. Il permettra ainsi de vérifier si oui ou non, les agents sur le marché sont sensibles au biais comportemental du « priming effect ».

¹ Les agents qui suivent la finance classique.

Section 2.02 Le biais comportemental « priming effect »

Lors du chapitre traitant de l'expérimentation, le « priming effect » sera analysé afin de savoir s'il peut altérer le comportement de joueurs sur un marché.

Les définitions se trouvant dans les livres ou sur Internet vont à l'unisson. La synthèse de celles-ci indique que lorsqu'un individu est soumis à un stimulus implicite, ses actions *a posteriori* seront influencées lorsque ceux-ci se déclencheront à cause d'un autre stimulus.

Cette section sera découpée en deux points : comment le combiner avec le domaine de la finance et la démonstration de ce biais avec l'appui de diverses études.

(a) Comment pourrait-on combiner le « priming effect » à la finance ?

Il est totalement légitime de se demander en quoi ce biais pourrait être combiné avec le domaine de la finance. Avant d'en expliquer la raison, il est important de connaître l'existence du « pre-holiday effect » se déroulant sur les marchés boursiers. Selon une page Internet traduite de Quantpedia², il existe un effet qui affecte les marchés juste avant les vacances (d'où le nom « pre-holiday effect »). L'observation de cet effet indique que le rendement des marchés est multiplié presque par 10 qu'en temps normal ; ce mouvement étant entraîné par **un état psychologique optimiste**.

En sachant cela, l'utilisation du « priming effect » essaiera de produire chez les participants un sentiment de bien-être mêlé à un soupçon d'optimisme. L'objectif étant de confirmer si ce biais comportemental impacte bel et bien les agents. Afin de démontrer cela, le marché expérimental de ce mémoire sera utilisé.

Naturellement, le « priming effect » a été analysé et de nombreuses études ont vu le jour pour en confirmer son existence. Ci-dessous se trouvent les résultats de plusieurs études afin d'approfondir notre compréhension à ce sujet.

² Source citée : <https://quantpedia.com/Screener/Details/83>

(b) Approfondissement de nos connaissances

(i) *Ralentissez ! Il y a des personnes âgées !*

Bargh, Chen et Burrows (1996) ont montré à leurs sujets des mots relatifs aux personnes âgées (par exemple : vieux, seul, bingo, dépendant, sage, prudent, etc³). Il s'est avéré que ceux qui ont vu ces mots se déplaçaient plus lentement que le groupe neutre.

Dans leur expérimentation, ils n'ont aucunement utilisé des mots se rapportant à la vitesse. Par conséquent, les individus « primed » ont vu leur vitesse de marche diminuée et ont automatiquement fait le lien entre la vieillesse et le manque de vivacité.

(ii) *« Business is business »*

Kay, Wheeler, Bargh, & Ross (2004) ont prouvé qu'en exposant des individus à des images ou à des mots clés se rapportant directement au monde du business, cela oriente leurs pensées vers l'appât du gain, la compétitivité, etc.

Deux résultats peuvent être cités :

- Les individus exposés aux images « business » ont moins tendance à partager équitablement les sommes d'argent et préfèrent en garder plus pour eux ;
- Si l'on demande à ces mêmes personnes de compléter, par exemple, le mot « c__p__tive », ils devineront souvent « competitive⁴ »

En résumé, l'exposition à ce type d'images a modifié ultérieurement leur comportement.

³ Ces mots ont été traduits de l'anglais en français.

⁴ Soit compétition en français.

(iii) Interrompez-moi si vous osez !

En 1996, Bargh, Chen et Burrows ont réalisé une étude sur ce concept en faisant participer des étudiants de l'université de New York.

Il leur avait été demandé de décrypter des mots en utilisant comme prétexte que c'était pour tester leur capacité de la manipulation de la langue. Cet exercice était découpé en deux devoirs. Dès qu'ils avaient fini le premier, les apprenants devaient le signaler à leur professeur.

Il avait constitué trois groupes. Le premier devait faire face à des termes qui étaient associés avec des notions d'agressivité et d'impolitesse (par exemple : effronté, perturbé, brutalité...). Le second devait déchiffrer des vocables qui avaient un rapport direct avec la politesse (par exemple, la courtoisie, bien se tenir...). Le dernier incarnait le groupe de contrôle ou de transition et devait analyser des mots neutres tels qu'« exercice » et « préparation ».

Ensuite, lorsqu'un étudiant se dirigeait vers l'expérimentateur, ce dernier était constamment en pleine discussion avec un acteur qui faisait mine de ne pas comprendre le problème qu'il devait résoudre.

C'est à partir de ce moment-là que les chercheurs ont commencé à récupérer des données. Dans le tableau ci-dessous, vous trouverez, en moyenne, le temps qu'un étudiant de chaque groupe a attendu avant d'interrompre le professeur et l'acteur.

Tableau 1 - Résultats des temps d'attentes des différents groupes.

	Groupe avec mots agressifs	Groupe avec mots neutres	Groupe avec mots polis
Temps d'attente	5,4 minutes	8,7 minutes	9,4 minutes

Nous pouvons constater que le groupe ayant vu les mots agressifs a seulement attendu 5,4 minutes avant d'interrompre le jeu de rôle du professeur et de l'acteur. À contrario, le groupe « poli » incarne le groupe qui a attendu le plus longtemps avant d'interrompre leur dialogue fictif.

(c) Omniprésence de ce biais

Nombreuses sont les autres expériences qui en viennent toujours à la même conclusion : nous sommes bien plus influencés qu'on ne le pense. Il suffit de voir comment ce biais est utilisé dans la vie de tous les jours. Prenons le cas des mots *réussite*, *bonheur*, *courage* et *amour* qui sont parfois tapissés en grand sur nos murs. Ou encore ces délicieux parfums de croissants, qui vagabondent près de la boulangerie, susurrant à votre estomac une mystérieuse faim. Nous ne pouvons oublier les petites assiettes qui dès lors qu'elles reçoivent une louche de nourriture, nous donnent l'impression qu'elles sont remplies à ras bord.

Maintenant, nous possédons assez de connaissances sur la finance comportementale ainsi que sur le « priming effect ». Nous savons également comment ces deux notions seront combinées, mais la manière de procéder reste encore obscure. Pour apporter plus de lumière à ce sujet, il ne manque plus qu'un point essentiel à gérer : les marchés expérimentaux. Grâce à la littérature scientifique à leur sujet, il sera possible d'acquérir suffisamment d'informations pour :

- Créer notre marché expérimental ;
- Effectuer une jointure entre la finance comportementale et le « priming effect ».

Section 2.03 Marchés expérimentaux

(a) Pourquoi réinventer la roue ?

Un document rédigé par Sunder (2007) regorge d'interpellations à la fois très simples et très pertinentes. Il commence tout d'abord par poser la question suivante : « Pourquoi encore plus de données ? ». Après lecture de cette question, on ne peut que s'interroger sur la nécessité de construire un énième marché expérimental qui ne fait qu'imiter voire même effleurer ce qui se déroule dans la réalité.

Pourquoi se frister les neurones à essayer de réinventer la roue afin de grappiller des informations alors que ces dernières peuvent très bien être achetées, consultées et analysées *a posteriori* ? L'auteur répond qu'il est plus aisé de connaître les paramètres et d'émettre des attentes raisonnables quand cela touche un marché expérimental. Il ajoute également « qu'il est plus facile de conduire de puissants tests dans ce genre de cas ».

De la même façon, nous pouvons faire une parabole avec une des célèbres maximes de Sun Tzu : « connais ton ennemi et connais-toi toi-même ; eussiez-vous cent guerres à soutenir, cent fois vous serez victorieux ». Replacé dans notre contexte, mieux on maîtrisera le marché expérimental et ses configurations, mieux on éprouvera les hypothèses et les futurs tests engendrés.

Toujours selon le même auteur, en utilisant une version simplifiée d'un marché, on évite par la même occasion les bruits générés par de nombreux facteurs, ces bruits pouvant intervenir lors de l'analyse des résultats et rendre cette tâche plus ardue.

Finalement, la réplication de certains systèmes nous permet d'apprendre plus facilement et de repérer avec plus de précision ce qui nous intéresse.

(b) Marchés éprouvés

S'inspirer du travail d'autrui n'est point un acte de copie, mais plutôt une quête d'apprentissage réfléchi. Ici, il s'agira avant tout de comprendre comment les auteurs ont construit leur outil, en observer la forme finale et surtout prendre en considération les différents reculs critiques.

Cette partie présentera quelques marchés expérimentaux. Le fonctionnement de chacun d'eux sera explicité et leur apport potentiel en lien avec ce mémoire sera souligné.

(i) *Le marché de Douglas (2012)*

1) Explication du travail de l'auteur

Le but de sa conception était de montrer à ses étudiants le processus de l'offre et de la demande à travers des transactions se déroulant dans un marché boursier minimaliste. En dépit de la date de publication de cet article (soit 2012), l'application est employée depuis 1997.

Au sein de son document, il présente la manière dont sa machinerie fonctionne.

Tout d'abord, il explique pourquoi il préfère utiliser des noms d'actions fictifs tels que des cookies, bonbons, tartes, etc. Il justifie son choix en disant qu'avec l'aide de ces noms, il n'était pas nécessaire aux étudiants de se renseigner sur les nouvelles qui concernent la presse économique ou financière. De cette manière, ce seront les participants qui fixeront, eux-mêmes et inconsciemment, les prix des actions.

Ensuite, il passe par une phase où des enchères sont données par les agents. Celle-ci ressemble énormément à une enchère hollandaise. En effet, celui qui a déposé le BID le plus haut remporte les actions pour lesquelles il a réalisé l'ordre.

Après cela, il présente l'interface de son programme en expliquant comment les ordres interagissent entre eux. Nous y apprenons qu'un système de *priorité* existe. Nous pouvons synthétiser cela de la façon suivante : les offres d'achat ayant les prix les plus hauts auront plus de chance de rencontrer un vendeur. Au contraire, les ordres de vente auront une priorité haute

de rencontrer un acheteur si leur prix est au plus bas. Ainsi, la différence entre l'offre d'achat avec le plus grand prix et l'offre de vente avec le plus bas prix formeront le « bid-ask spread ».

Finalement, son logiciel est capable d'enregistrer les ordres, de consulter son portefeuille ainsi que toutes les transactions des actions.

2) Quid des apports dans ce mémoire ?

Toutes ces fonctionnalités ont été intégrées au programme de ce mémoire à l'exception d'une seule : celle des enchères au départ pour lancer le marché. En effet, nous verrons par la suite que le marché expérimental présent dans ce mémoire est effectué par des robots qui ont été configurés préalablement pour le rendre vivant.

De plus, la manière dont Douglas gère les actions ainsi que leur correspondance a été littéralement implémentée. Ce mécanisme est totalement décrit sur la partie qui traite de l'architecture du logiciel.

L'utilisation de noms plus communs pour les actions a été retenue. Cette façon de faire se révélera utile lors de l'expérimentation sur le « priming effect ». En effet, les participants sélectionnés n'ont pas ou peu de connaissances sur le monde financier et de cette manière, le software paraîtra plus facile d'accès.

(ii) Le marché de Maslov (2008)

1) Explication du travail de l'auteur

Cette fois-ci, la bourse n'est plus constituée d'agents humains, mais de robots exerçant automatiquement leur droit de passer des ordres. Le terme technique anglophone le désignant se nomme « Algorithmic trading ».

Dès lors, le chercheur a dû sélectionner plusieurs notions de véritables marchés afin de les faire coïncider avec le sien.

Tout d'abord, les robots peuvent exécuter des ordres à la fois en « market » ou en « limit » et également choisir s'ils souhaitent en acheter ou en vendre [BID/ASK]. Ces mêmes agents ne peuvent pas personnaliser la portion qu'ils souhaitent vendre ou acheter. En effet, tout passage d'ordre est régi par un système de lot fixe. Par exemple, si le lot a été défini à cinq, alors, dès que l'on passera un ordre, la quantité sera automatiquement établie à cinq.

Toujours concernant ces ordres, ceux-ci sont valables jusqu'à ce qu'ils soient exécutés⁵.

Ensuite, les traders robotisés ont une chance équiprobable d'exécuter des actions soit en « limit » ou en « market ». Dès que ce choix a été fixé, le robot se doit de choisir entre un ordre d'achat [BID] ou de vente [ASK]. Une nouvelle fois, il possède une chance sur deux de choisir l'un ou l'autre. En simplifiant ce paragraphe par une image, nous obtenons l'arbre décisionnel suivant :

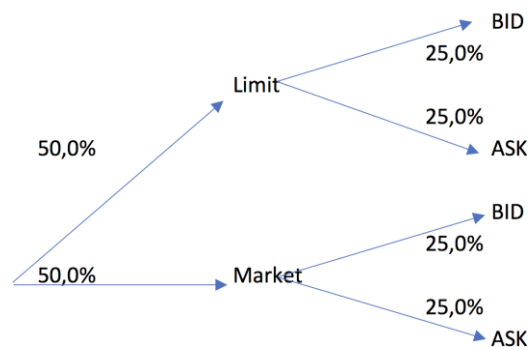


Figure 1 - Arbre décisionnel du marché de Maslov.

Nous pouvons donc remarquer que le robot a 25 % de chance de tout aussi bien placer un ordre d'achat en « limit » qu'un ordre de vente en « market ».

Par ailleurs, le cas d'un ordre de type « limit » mérite d'être explicité. Son prix est fixé avec l'aide d'un chiffre aléatoire dénommé, ici, Δ dont les valeurs sont comprises entre les nombres un et quatre. L'auteur spécifie qu'avec ce *modus operandi*, les robots agissent d'une manière **non stratégique** [*grosso modo*, ils ne réagissent pas d'une manière rationnelle aux différentes fluctuations des prix et agissent à travers un procédé stochastique].

⁵ Une exception est à signaler : si un ordre n'a pas été rempli après 1000 étapes, alors celui-ci est supprimé.

Ce Δ est respectivement retiré ou ajouté aux « highest BID » et « lowest ASK ». Nous obtenons ainsi les formules suivantes :

$$\text{New price limit BID} = \text{Highest BID} - \Delta$$

$$\text{New price limit ASK} = \text{Lowest ASK} + \Delta$$

De cette manière, les ordres ne se chevauchent jamais et aucune transaction n'est enregistrée. *De facto*, la seule façon pour le système de créer une transaction est d'utiliser les ordres « market » [qui a une chance sur deux de se produire].

2) Quid des apports dans ce mémoire ?

Maslov transmet des informations d'une importance capitale qui auront un impact majeur sur un des robots développés dans ce mémoire. En résumé, il indique comment et quand ces robots effectuent des ordres ainsi que les formules mathématiques qu'ils emploient.

Plusieurs de ces points ont été par conséquent programmés. Tout d'abord, le robot de notre marché expérimental adoptera aussi une façon d'agir non stratégique. *De facto*, il aura une chance équiprobable :

1. De passer un ordre d'achat [BID] ou de vente [ASK].
2. Puis, d'effectuer un ordre en « limit » ou en « market ».

Quant aux formules, les bases ont été reprises, mais une variable a été ajoutée. En effet, selon sa solution, ces robots ne créeront jamais de nouveaux « highest BID » ou « lowest ASK ». Dans le cadre du marché fraîchement construit, il était nécessaire que cela se produise. La modification apportée est décrite plus longuement dans le chapitre qui traite de l'architecture logicielle⁶.

⁶ C'est-à-dire au chapitre qui se penche sur le « [Player bot](#) »

(iii) *Le marché de Kang et de SUN (2011)*

1) Explication du travail de l'auteur

Avant de commencer les explications sur leur travail, les chercheurs ont développé des extensions pour un logiciel dénommé « JASA⁷ » [Java Auction Simulator API].

Sur son site [Web officiel](http://jasa.sourceforge.net/)⁸, il est noté que ce dernier est « un simulateur d'enchères qui a été conçu pour opérer des expériences au sein d'une économie centrée sur les agents [...] Il est possible de mettre en œuvre des marchés réels tels que les bourses [...] ».

De base, « JASA » donne l'occasion d'effectuer des enchères anglaises et hollandaises ainsi que des « sealed-bid auctions ». En plus de cela, il est composé de trois éléments principaux dont vous pouvez observer les mécanismes à la Figure 1.

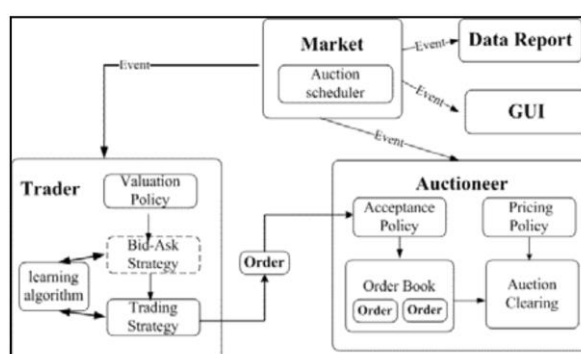


Figure 2 - Image provenant du document « Building the model of artificial stock market based on JASA » de Kang et de Sun.

Premièrement, il y a le « Market ». Celui-ci réagit aux différents événements ayant lieu sur le marché et les transmet au service approprié.

Concernant le « Trader », il faut savoir qu'il représente les robots. Ceux-ci peuvent enregistrer des ordres d'achat ou de vente, gérer leur compte, passer en revue leurs actions ou encore changer ou améliorer leur stratégie. À la vue de cela, un détail doit être souligné. Lorsque le

⁷ Ce software a été, quant à lui, développé par Steve Phelps.

⁸ <http://jasa.sourceforge.net/>

programme est dans une phase d'enchère, les robots doivent obligatoirement opter pour devenir **soit** un acheteur ou un vendeur. Dès que cette phase est terminée, la condition précédente s'annule et les agents sont de nouveau libres de choisir le statut qu'ils souhaitent occuper.

Enfin, l'« Auctioneer » ou le commissaire-priseur dans la langue de Molière est le responsable des adjudications et des ordres. En outre, il essaie de trouver des correspondances, de déterminer les prix de transactions, de récupérer tous les « BID » ou « ASK » non clôturés ou de communiquer sur l'état actuel du marché.

Après toutes ces informations, il serait légitime de se poser la question suivante : « pourquoi les auteurs ont-ils ajouté ces fonctionnalités supplémentaires ? ».

Par défaut, « JASA » ne possède pas les indicateurs qu'ont les véritables marchés à savoir : la quantité d'actions échangées, le prix d'ouverture, la meilleure offre d'achat ou de vente ou encore l'obtention en retour des graphiques montrant la fluctuation des prix. Dans les marchés boursiers, ces indicateurs restent assez basiques et ont été donc implémentés dans « JASA ».

Pour en imiter le comportement normal, les chercheurs ont implémenté de nouveaux éléments : la limitation des prix et trois stratégies pour les agents. Le premier permet de préciser un niveau de prix dans lequel un prix peut augmenter ou diminuer lors d'une même journée boursière. Les trois stratégies viennent renforcer les stratégies déjà existantes : la stratégie aléatoire [celle-ci a été employée pour l'élaboration de ce mémoire et sera explicitée par la suite], l'adaptabilité [utilise du « machine learning »] et le mode de prix.

2) Quid des apports dans ce mémoire ?

La lecture de ce document a eu un double impact sur le développement de l'application finale. Manifestement, ce travail présente deux volets distincts : « JASA » et les améliorations des auteurs.

À propos de « JASA », ses trois « core classes » ont été reprises. Cependant, elles portent un tout autre nom et leurs objectifs diffèrent aussi. Dans l'ordre, la classe « Market » a été remplacée par « Consumer ». Dès qu'un évènement se produit au sein de la bourse, un « Consumer » le prendra en charge et le dispatchera. Ensuite, la classe « Trader » a été

remplacée par « Participant » et n'effectue que les tâches les plus basiques : acheter, vendre des actions ainsi que consulter le solde de son compte. Pour terminer, « Auctioneer » se prénomme « System » et permet également de trouver les correspondances entre les ordres.

En ce qui concerne le plan des ajouts des auteurs sur ce programme⁹, seule l'une d'entre elles a été introduite dans le cadre de ce mémoire. Il s'agit de la **stratégie aléatoire**. Celle-ci sera expliquée plus longuement dans la partie qui décrira le « [Player bot](#) ».

⁹ JASA

(c) Comparatifs des marchés

Dans l'un de ses travaux, Stefan Palan (2015) expose une série de critères qu'il a utilisés pour comparer des marchés expérimentaux. Le tableau qui se trouve ci-dessous se base sur le sien. Néanmoins, plusieurs détails diffèrent : seuls les marchés éprouvés dans ce mémoire ont été comparés et certains des éléments ont été supprimés car aucun des marchés n'en présente les caractéristiques¹⁰.

Tableau 2 - Tableau comparatif des marchés étudiés dans ce mémoire.

Intitulés/Nom des marchés	Douglas	Maslov	JASA + Kang & Sun	Palermo
Marché				
Double auction ¹¹	X ¹²	X	X	X
Affichage d'un prix indicatif ¹³	X		X	X
Market orders		X	X	X
Trading possible de plusieurs actifs	X			X
Marchés établis sur plusieurs périodes				X (grâce à oTree et à son système de sessions)
Existence de plusieurs marchés				X
Possibilité de fixer les rôles des traders ¹⁴		X	X	
Graphique sur les prix	X		X	
Trading par algorithme ¹⁵		X	X	X
Présence de taxes sur les transactions		X		
Vente à découvert	X			
Autre				
Est une application Web ?	X			X
Possibilité d'importer les données de configurations				X
Open source			X (Uniquement JASA)	? ¹⁶

¹⁰ À savoir : les coûts d'utilisation du logiciel, le support de plusieurs langues, le système de « call auction ».

¹¹ Ce type d'enchère est décrit au chapitre « [Le fonctionnement d'un « double auction market »](#) ».

¹² Le **X** signifie que la fonctionnalité y est disponible.

¹³ Par exemple, combien obtiendrait-on si l'on pouvait vendre nos actions ou combien devrions-nous payer pour autant d'actions ?

¹⁴ Si un logiciel possède cette fonctionnalité, cela veut dire que l'expérimentateur peut fixer le rôle des participants (soit ils peuvent acheter, vendre ou faire les deux) ;

¹⁵ Est-ce que le software propose des robots qui passent des ordres ?

¹⁶ L'UCL possède les sources de l'application.

Article III. Le fonctionnement d'un « double auction market »

Comme indiqué dans le titre et selon [un article d'Investopedia](#)¹⁷, il s'agit d'un marché à double enchère où l'on peut émettre des ordres d'achat ou de vente représentés respectivement par les termes BID ou ASK.

Après avoir étudié les raisons d'être d'un tel logiciel, la prochaine étape consiste à comprendre tous les mécanismes sous-jacents quant à son fonctionnement.

À cette fin, tous les concepts pour bien assimiler ce qu'est un « double auction market » sont spécifiés ci-dessous.

Section 3.01 Les actifs financiers

Sur les marchés réels, plusieurs types d'actifs financiers peuvent circuler. On y retrouve des actions, des obligations, des options, des contrats futurs et ainsi de suite. Le marché expérimental développé dans le cadre de ce mémoire proposera uniquement d'échanger des actions d'entreprises fictives.

(a) Les actions

Étant donné que le logiciel développé traite uniquement ce type d'actif, il est essentiel d'expliquer brièvement leur nature.

Lors de l'acquisition d'une action d'une entreprise, vous achetez une partie de son capital. Grâce à cela, vous porterez dorénavant le nom d'actionnaire. Un actionnaire (selon [lesechos.fr](#)¹⁸) peut, s'il possède une part importante du capital de l'entreprise, intervenir dans sa gestion.

¹⁷ <https://www.investopedia.com/terms/a/auctionmarket.asp>

¹⁸ https://www.lesechos.fr/finance-marches/vernimmen/definition_actionnaire.html

Certaines actions concèdent aux actionnaires un retour à travers des dividendes. Celles-ci ne sont pas implémentées dans le modèle boursier ici présent. Cependant, il faut savoir que ces actifs peuvent être revendus par leur acheteur à tout moment.

Section 3.02 Le commissaire-priseur

Le commissaire-priseur est toujours¹⁹ la personne de contact lorsque vous envisagez de participer à la vie de la bourse. Ce dernier enregistrera dans son carnet d'ordres toutes vos actions et vous trouvera aussi les meilleurs individus avec lesquels vous pourrez conclure des affaires.

Section 3.03 Le carnet d'ordres

Le carnet d'ordres est l'outil du commissaire-priseur. Dès que celui-ci recevra une requête d'un client, il ouvrira son cahier, marquera l'identifiant de la personne, ce qu'elle veut, la quantité, le prix souhaité et enfin, la date à laquelle il a réceptionné sa demande.

Par exemple, on pourrait très bien lire au sein de ce carnet ce genre d'écrit : « **À la date du 12 juin 2018 à 10 h**, monsieur **Dupont** souhaite **acheter directement 30 actions** de l'entreprise **Apple**. »

Section 3.04 Les ordres

(a) Achat ou vente (respectivement BID ou ASK)

Comme monsieur Dupont, vous pouvez passer un ordre : cela signifie votre intention d'achat ou de vente au marché.

¹⁹ Aujourd'hui, les commissaires-priseurs sont très souvent remplacés par des ordinateurs qui jouent leur rôle.

Si l'action d'Apple vous intéresse et que vous souhaitez en acquérir, vous effectuerez par conséquent un ordre d'achat en précisant la quantité voulue. Dans le jargon des traders, cela s'appelle un « **BID order** ».

Dans le cas contraire, vous pouvez tout aussi bien aspirer à vendre vos actions afin d'en dégager des bénéfices. À partir de ce moment-là, vous opterez pour un ordre de vente ou autrement dit un « **ASK order** ».

(b) Best BID et best ASK

Au premier coup d'œil, sous ces noms qui peuvent paraître étranges se cachent des traductions très simples à comprendre. Remplacez « best BID » par meilleur acheteur et « best ASK » par meilleur vendeur.

Reprenons nos exemples précédents. Logiquement, quand vous voulez effectuer un achat, vous cherchez toujours le vendeur qui vous propose le prix le plus avantageux. Ce prix très alléchant se nomme « best ASK ».

Maintenant, considérez-vous comme étant un vendeur : votre objectif est de vendre vos actions pour engranger un maximum de bénéfice. Il est évident que votre recherche va uniquement s'articuler autour des acheteurs qui vous donneront le plus d'argent en échange de vos actions. Le meilleur acheteur qui vous donnera le plus d'argent se présentera sous le nom de « best BID ».

Ci-dessous se trouve une représentation graphique d'un carnet d'ordres où sont mis en évidence le « best BID » et le « best ASK ».

BID			ASK		
Count	Size	Price	Count	Size	Price
1	4	57.00 €	1	4	59.00 €
1	7	56.50 €	1	8	59.50 €
1	7	56.00 €	1	5	60.00 €
1	8	55.50 €	1	10	60.50 €
1	5	55.00 €	1	9	61.00 €

Figure 3 - Exemple d'un carnet d'ordres.

(c) Les types d'ordres

(i) Market order

Contrairement à ce que l'on pourrait imaginer, il n'est pas obligatoire de spécifier un prix lorsque l'on passe un ordre d'achat ou de vente.

Si un ordre ne comporte aucun prix, vous effectuerez ce que l'on appelle un « **market order** ». Cela signifie que le commissaire-priseur identifiera automatiquement quelle est la meilleure offre pour vous.

Supposons que vous vous rendez au sein de votre marché favori et que vous ayez la subite envie de consommer des pommes. Tout naturellement, vous vous dirigerez vers le commissaire-priseur et vous lui ferez part de votre besoin. Vous formulerez une phrase équivalente : « J'aimerais effectuer un **ordre de type BID** sur les actions **POMMES** dont la **quantité** a été assignée à **5** ».

Grâce à ces informations, le spécialiste vous désignera le meilleur vendeur de pommes qui vous proposera le prix le plus avantageux.

Décomposons ensemble cette requête :

Un ordre de type **BID** sur les actions **POMMES** en **market** dont la **quantité** a été assignée à **5**

BID = vous proposez une offre d'achat. Un raccourci possible serait de dire que vous êtes un acheteur.

POMMES = ce que vous désirez acquérir.

Market = vous acceptez le meilleur prix que le commissaire-priseur vous propose.

Quantité de 5 = vous lui en prenez 5.

(ii) Limit order

Par opposition, un « limit order » est un ordre dans lequel vous avez spécifié un prix auquel vous voulez vendre ou acheter une action.

Reprenons notre exemple avec le commissaire-priseur et les marchands de pommes. Supposons que vous trouviez ses prix de vente trop élevés et vous lui rétorquez : « Je les achèterai uniquement si vous me les laissez à un prix par unité de 80 centimes ». Si le vendeur vend ses pommes à partir de 1 €/pièce, alors, il enregistrera votre commande, mais vous ne recevrez pas en retour vos fruits. En effet, le vendeur considèrera votre offre d'achat insuffisante et ne vous cèdera pas ses biens. En revanche, si un jour le vendeur baisse ses prix et égalisent ou passent en dessus de votre offre de 80 centimes, alors, il vous cèdera automatiquement ses pommes.

Par l'utilisation des « limit order », vous pouvez éviter d'acheter des actions à des prix trop hauts ou encore de les vendre à des prix trop bas. En résumé, cela agit comme une protection. Toutefois, contrairement au « market order », il y a des chances que vous ne receviez pas vos biens immédiatement (voire même jamais). Toujours à travers nos pommes, il est possible d'illustrer cela. Prenons le cas extrême où un autre acheteur souhaite les acheter à 0,01 €/pièce. Il va sans dire que cette personne a peu de chance de recevoir un jour des pommes à ce montant-là car les vendeurs auront tendance à ignorer son offre.

(d) La quantité

Sur le tableau de droite, une colonne portant le nom « Size » nous renseigne les quantités disponibles pour le prix indiqué sur sa droite.

Le « best BID » est à 57 € et possède une quantité disponible de 4 unités.

BID		
Count	Size	Price
1	4	57.00 €
1	7	56.50 €
1	7	56.00 €
1	8	55.50 €
1	5	55.00 €

Figure 4 - Carnet d'ordres avant d'avoir reçu un ordre ASK.

Supposons qu'un vendeur souhaite en échanger une dizaine le plus rapidement possible (il donne un « market order » de type ASK avec une quantité de 10). Au sein de ce scénario, la première ligne disparaîtrait et il

resterait exactement 1 ordre valant 56,50 €. De plus, ce dernier deviendrait automatiquement le nouveau « best BID ».

BID

Count	Size	Price
1	4	57.00 €
1	1	56.50 €
1	7	56.00 €
1	8	55.50 €
1	5	55.00 €

Figure 5 - État du carnet d'ordres après avoir reçu un ordre ASK.

Ce changement de « best BID » nous permet d'aborder une nouvelle notion qui est rattachée à la quantité : la profondeur du marché²⁰. Dans cet exemple, notre marché n'est absolument pas profond. En effet, il a fallu un seul et unique ordre pour faire bouger le « best BID » de 57 € à 56,50 €. Au contraire, si ce marché avait eu une profondeur élevée, il aurait fallu, au contraire, plusieurs ordres pour faire bouger le « best BID ».

Dernière remarque, la quantité n'affecte en rien le positionnement d'un ordre dans le carnet. Seuls leur prix et le temps²¹ sont pris en considération lors du classement des ordres.

(e) Le « Count »

Ceci est la dernière notion sur les ordres à souligner. Dans tous les scénarios décrits précédemment, l'acheteur était toujours seul. En réalité, c'est loin d'être le cas. Il se pourrait que plusieurs personnes décident d'acquérir ou vendre un bien à un même prix.

²⁰ Source : <https://www.investopedia.com/terms/m/marketdepth.asp>

²¹ Lorsque l'on parle du temps, nous sous-entendons la date et l'heure de l'enregistrement de l'ordre.

Supposons que M. Dupont souhaiterait en acheter 5 pour le prix de 56 €. Dans la même optique, M. Grimm voudrait en prendre 2 pour le même prix de 56 €. En procédant à une addition, on se retrouve avec 2 ordres d'achat à 56 € dont la quantité vaut 7²².

Visuellement, il est impossible de savoir qui a mis telle ou telle quantité. Seul le système informatique est capable de faire une distinction.

BID

Count	Size	Price
1	4	57.00 €
1	1	56.50 €
2	7	56.00 €
1	8	55.50 €
1	5	55.00 €

Figure 6 - Augmentation du "count".

Section 3.05 Les transactions

Finalement, dans la circonstance où le vendeur de fruits et vous échangez vos biens, il y a une transaction qui se déroule.

Une transaction contient les informations suivantes : la date où elle a eu lieu, le montant, le produit véhiculé, la quantité²³ et le prix auquel il a été échangé.

²² À noter que l'addition est juste appliquée pour donner une indication visuelle aux utilisateurs. Dans la base de données, aucune addition n'est opérée.

²³ En vulgarisant, on pourrait comparer la transaction à un ticket que vous recevez du marchand.

Article IV. Phase d'analyse

Avant de commencer à programmer, il est primordial de bien conceptualiser les besoins des différents utilisateurs de même que le problème donné.

Une partie de ce chapitre sera consacrée à l'analyse UML²⁴ 2.0 qui établira le diagramme des cas d'utilisation et de classe. Respectivement, le premier montre les objectifs d'utilisation de chaque utilisateur tandis que le second décrira d'une manière plus technique les différents composants de l'application.

Quant à l'autre partie, elle sera composée d'un schéma qui aura pour but de montrer toutes les veines et ressources du logiciel ainsi que les différentes phases par lesquelles le marché transite.

L'ordre de présentation de ces analyses sera fait par ordre croissant de technicité :

- 1. Diagramme des cas d'utilisation ;**
- 2. Carte de l'application ;**
- 3. Les phases du marché ;**
- 4. Diagramme de classe.**

²⁴ UML signifie : **Unified Modeling Language**.

Section 4.01 Le diagramme des cas d'utilisation

Les diagrammes des cas d'utilisation permettent de représenter simplement et rapidement les différents usagers de l'application ainsi que les actions qu'ils pourront effectuer. Pour chaque usager, ce diagramme détermine son départ depuis une question type : « pourquoi tel ou tel utilisateur se rendrait-il sur l'application ? ».

In fine, deux acteurs principaux ont été décelés : les participants et les expérimentateurs.

(a) Les participants

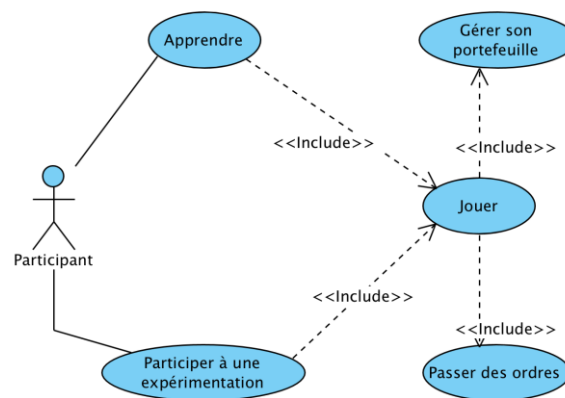


Figure 7 - Diagramme de cas d'utilisation d'un participant.

En reprenant notre question type, pourquoi un participant voudrait-il se rendre sur ce marché expérimental ?

La première raison suppose que celui-ci désire **apprendre** et souhaite s'entraîner à mieux appréhender les marchés boursiers en utilisant ce marché expérimental. Tandis que le second atteste qu'il fait partie d'une **expérimentation** et qu'il doit y participer.

Les mots clés « Include » nous renseignent sur le côté obligatoire d'un cas d'utilisation²⁵. En d'autres termes, si par exemple un agent souhaite tester le marché, il devra inévitablement y jouer. « Jouer » implique aussi les concepts « Gérer son portefeuille » ainsi que « Passer des ordres ». En d'autres mots, si un agent souhaite « Jouer » sur le marché expérimental, il sera tôt ou tard obligé de passer des ordres et de consulter son portefeuille.

²⁵ Un cas d'utilisation est représenté par une bulle bleue.

(b) Les expérimentateurs

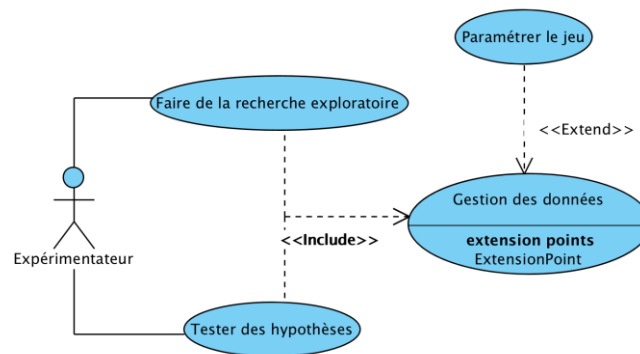


Figure 8 - Diagramme de cas d'utilisation d'un expérimentateur.

Le point de vue de cet acteur est résolument tourné vers la compréhension, le test d'hypothèses et la recherche scientifique.

En toute logique, celui-ci organisera des sessions boursières pour, dans un premier temps, obtenir des informations et par la suite les analyser.

Le cas d'utilisation « Gestion des données » inclut un sous-ensemble qui est, ici, dissimulé. Il implique la notion de CRUD (**C**reate, **R**ead, **U**ppdate and **D**ele) sur toutes les données qu'il a récoltées. Pour chacun des points du CRUD, voici ce que l'expérimentateur pourrait éventuellement faire :

- **Create**: pouvoir produire de nouveaux jeux de données à partir de l'existant. Par exemple, grâce aux variables stockées dans la base de données, il pourrait créer de nouvelles variables telles que la valeur boursière des portefeuilles des différents participants ou la quantité moyenne d'offres enregistrées ;
- **Read**: consulter la base de données (lecture d'informations) ;
- **Update**: mettre à jour l'application, modifier les paramètres de son lancement, etc. ;
- **Delete**: supprimer les ordres non nécessaires qu'ont exécutés les robots.

« Paramétrer le jeu » est un cas particulier dont son association a été affublée de l'étiquette « Extend ». En d'autres termes, il indique simplement que l'application peut très bien fonctionner sans altérer les configurations initiales.

Section 4.02 La carte de l'application

L'image que vous trouverez ci-dessous synthétise les principaux canaux de communication du logiciel, les agents, le marché et les fichiers nécessaires à son fonctionnement.

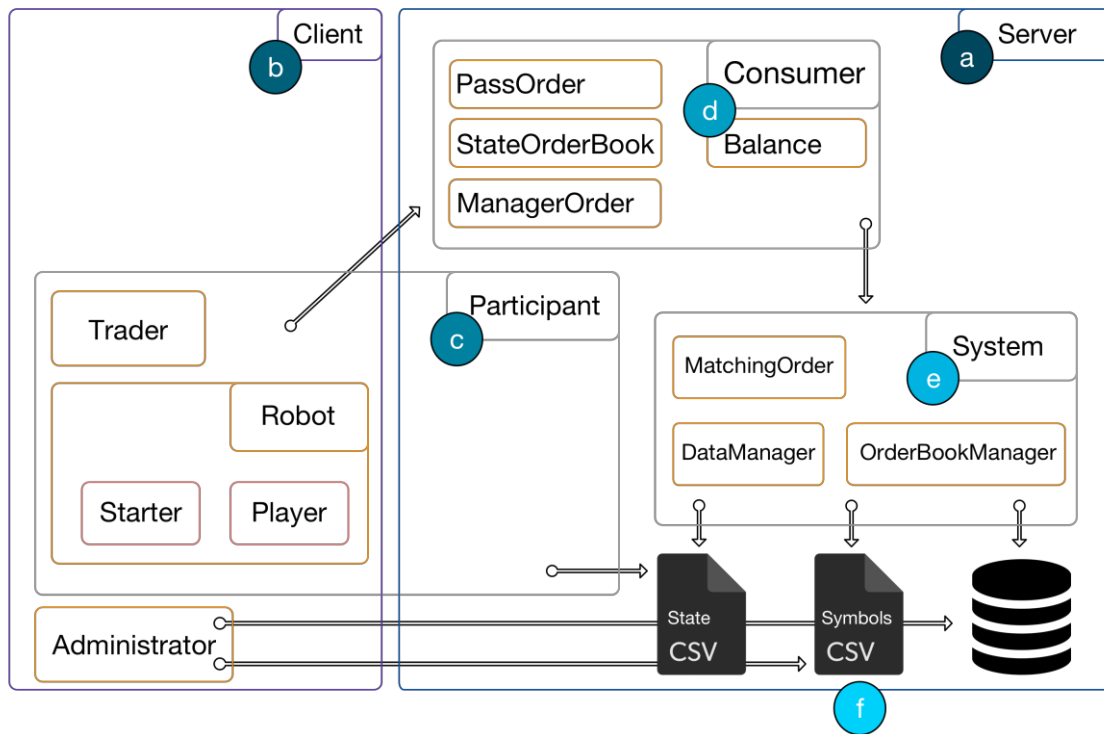


Figure 9 - Carte du marché expérimental de ce mémoire.

(a) Server

C'est le logiciel où sont centralisées toutes les données. Dès qu'un utilisateur, par exemple, envoie une requête, celle-ci sera directement adressée au serveur qui la réceptionnera et la traitera.

(b) Client

Dans le jargon des développeurs, un client est un logiciel qui crée des demandes pour le serveur. Ici, dès qu'un utilisateur se connectera au marché, il demandera d'abord d'afficher la page Web et les scripts. Après cela, le client pourra seulement commencer la réception des informations des ordres, des transactions, etc.

Au sein de ce rectangle, d'autres sont compris : l'« Administrator », « Trader » et « Robot ». Concernant « Administrator », il s'agit de l'expérimentateur. Il peut télécharger et modifier à sa guise toutes les données qui sont stockées sur le serveur (des fichiers de configuration et la base de données).

Concernant « Trader » et « Robot », ils font partie de l'élément « Participant ».

(c) Participant

« Participant » comprend les traders ainsi que les robots. En utilisant les termes adéquats, ces deux éléments sont des **classes**. En vulgarisant, nous pouvons dire qu'un trader est une personne physique qui échange des biens sur un marché tandis qu'un robot est son homologue numérique. À noter que le serveur n'arrive pas à distinguer des nuances de cette sorte. On a besoin de les lui signaler à travers une version « simplifiée » de la réalité. Cette version simplifiée porte le nom de **classe**.

Par contre, une autre nuance extrêmement fine existe entre « Client » et « Participant ». Le premier ne fait qu'envoyer des requêtes au serveur. Tandis que le second, par contre, fait par extension la même chose que « Client » mis à part qu'il peut participer à la vie du marché.

(d) Consumer

Sous ce nom un peu étrange se cache un concept simple. Il s'agit des portes d'entrée du programme.

Imaginons qu'un trader souhaite passer un ordre. Il va d'abord remplir le formulaire et envoyer ses informations. Le « Client », connaissant dès le départ toutes les portes accessibles, enverra sa requête au serveur, plus particulièrement au portail « PassOrder ».

Le mécanisme est exactement le même pour les autres « consumers » :

- StateOrderBook : il prévient tous les clients des dernières informations des carnets d'ordres (les cinq derniers BID, ASK et transactions) ;

- ManagerOrder : il permet aux clients de supprimer leurs ordres²⁶ ;
- Balance : il permet aux clients de connaître leur solde disponible ainsi que leur portefeuille d'actions à jour.

Après réception d'une requête par un des « Consumer » disponible, ce dernier répartira l'information vers le service adéquat. Généralement, l'information est envoyée au service s'appelant « System ».

(e) System

Cette sous-partie du serveur incarne le commissaire-priseur du marché. Il gère les requêtes entrantes, alimente les robots avec les dernières informations sur les carnets d'ordres, effectue les correspondances et vérifie si tout est en état de fonctionnement.

(f) Les ressources

Enfin, les ressources ont plusieurs objectifs. Premièrement, elles servent à indiquer les actions qui seront disponibles aux agents. Deuxièmement, dès lors qu'une requête a été traitée par le « System », ce dernier enregistre le best BID et le best ASK dans un fichier. Ce fichier sera lu et analysé par les robots afin qu'ils gardent une certaine cohérence dans leur procédure. Troisièmement, il est possible d'arrêter les mouvements des robots à partir d'un fichier. Pour chacune des sessions, un fichier stocke la valeur 0 ou 1 si l'on souhaite respectivement les arrêter ou les démarrer.

²⁶ Seulement ceux qui ne sont pas encore totalement exécutés.

Section 4.03 Les états du marché

Tout au long de son cycle de vie, le marché doit passer par une série d'étapes.

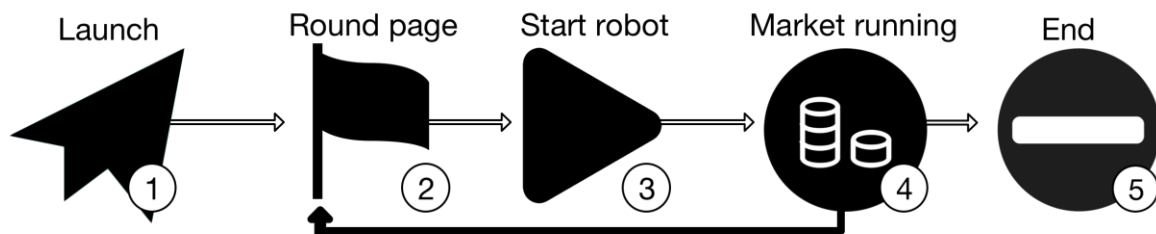


Figure 10 - Présentation des états du marché expérimental.

Tout d'abord, le serveur crée les données initiales **(1)**. En résumé, il crée le fichier où seront sauvegardées les futures informations sur les carnets d'ordres (qui alimenteront les robots) et, si l'administrateur l'a spécifié, active le robot qui sera responsable de produire les premiers BID et ASK. Naturellement, cette étape est effectuée juste une fois par expérimentation.

Ensuite, lors de l'étape « Round page » **(2)** le début d'une session boursière sera introduit. Plusieurs processus sont implicitement exécutés : d'abord, il empêche la visualisation du marché et empêche les robots de passer des ordres. Aussi, lors du premier round, **il crée le fichier où sera stocké leur état de fonctionnement (0 en arrêt ou 1 en état de marche)**.

L'étape « Start robot » **(3)** ne fait que les activer.

Lors du « Market running » **(4)**, les traders pourront enfin passer des ordres et voir le marché fonctionner. Dans l'hypothèse où l'expérimentateur a spécifié plusieurs sessions boursières, alors le système redirige tous les utilisateurs vers l'étape numéro **(2)**.

Enfin, lorsque tous les rounds se sont déroulés, la dernière étape **(5)** signale à tous les utilisateurs la fin de la session boursière et coupe par la même occasion tous les robots.

Section 4.04 Le diagramme de classe

Sous ce nom légèrement barbare se cache une idée enfantine. Lorsqu'un artiste décide de dessiner, par exemple, une personne, il commencera toujours par mettre en évidence les principaux traits avec l'aide de son crayon.

Le diagramme de classe incarne le crayon d'un concepteur de logiciel. Il permet de structurer les différents objets informatiques entre eux et de révéler leurs interactions. On peut également s'en servir de fondation pour construire les bases de données : ce qui a été fait ici.

Ci-dessous se trouve le diagramme de classe de ce marché dans sa version simplifiée²⁷.

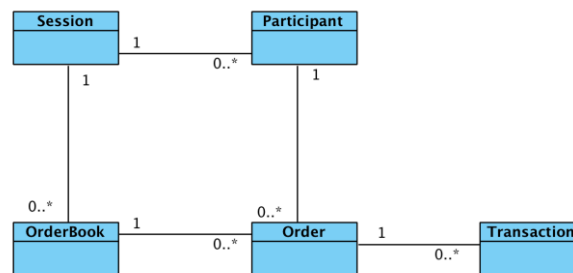


Figure 11 - Diagramme de classe.

Cinq objets sont d'une importance capitale.

La classe « **OrderBook** » comprend le symbole²⁸ (par exemple : AAPL), une brève description de l'actif financier et fait la liaison avec la classe « **Session** ».

La classe « **Session** » est générée automatiquement par oTree²⁹ et contient des informations telles qu'un code (qui permet de la reconnaître avec plus d'aisance dans une base de données), le nombre de joueurs ou encore d'autres éléments de configuration.

²⁷ Vous pourrez trouver la version complète à l'annexe « [Diagramme de classe complet](#) ».

²⁸ Le symbole équivaut au nom de l'actif financier.

²⁹ Dont son fonctionnement sera décrit dans la partie qui abordera l'architecture logicielle.

Ensuite, la classe « **Participant** » permet l'instanciation³⁰ d'un agent lorsqu'une session boursière est ouverte et active. Par défaut, nous pouvons les identifier à travers un numéro unique. Sinon, l'expérimentateur peut spécifier les noms des participants dans un fichier.

Incontestablement, « **Order** » et « **Transaction** » sont les classes qui regorgent le plus d'informations en tout genre. En effet, lorsqu'un joueur passera un ordre et, s'il a franchi toutes les mesures de sécurité, celui-ci sera sauvegardé à travers l'objet « **Order** ». La liaison entre un acheteur et un vendeur sera scellée dans la classe « **Transaction** ».

(a) Une attention particulière sur la classe Order

Le but de ce marché est de capter toutes les informations que peuvent nous laisser les utilisateurs. Dans cette optique, de nombreux choix ont été opérés.

Premièrement, l'application ne supprimera jamais des ordres.

Deuxièmement, la base de données ne fera pas de différence entre les ordres de types « limit » ou « market ». La seule manière de les repérer sera de regarder si un prix a été fixé ou non. Si tel est le cas, alors l'ordre possède un état « limit ». Sinon, il s'agira donc d'un ordre « market ».

³⁰ « Participant » est une classe. En programmation, une classe est une sorte de moule qui contient une structure simplifiée de la réalité. Une instance est un objet qui provient d'une classe. Supposons que l'on possède une classe « Personne ». Nous savons qu'en réalité une personne peut être désignée par un nom et un prénom. Une instance de cette classe serait un objet « Personne » qui aura respectivement le nom et prénom de Grimm Dupont.

Section 4.05 Analyse de la base de données

Cette dernière partie de l'analyse décortiquera la manière dont les données ont été sauvegardées et formatées. Elle concerne les ordres et les transactions.

(a) Order

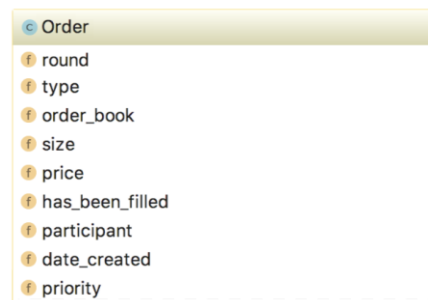


Figure 12 - Détails de la classe "Order".

L'attribut « **round** » permet de détecter à partir de quel tour l'ordre a été passé.

Grâce à « **type** », il est possible de déterminer si c'est un ordre d'achat ou de vente (les valeurs imaginables sont : BID ou ASK).

« **Order_book** », « **size** » et « **price** » nous renseignent respectivement dans quel carnet d'ordres l'ordre a été stocké, sa quantité et son prix.

Si « **has_been_filled** » vaut un, alors l'ordre a été clôturé. *De facto*, lorsque l'ordre sera fraîchement inséré dans la base de données, sa valeur sera de zéro. En cas d'exécution partielle, ce champ vaudra toujours zéro.

Avec l'aide de l'enregistrement « **participant** », nous pouvons remonter jusqu'à la personne qui a opéré l'action.

Comme son nom l'indique, « **date_created** » met en évidence la date ainsi que l'heure exacte à laquelle il a été créé.

Enfin, « **priority** » est une variable numérique qui nous indique le classement de l'ordre dans son carnet d'ordres. Ce classement est effectué par ordre croissant des « **priority** ». Prenons le cas d'un exemple : le meilleur BID est caractérisé par une valeur égale à 1. Par conséquent,

plus le chiffre est petit, plus il aura des chances d'entrer en correspondance avec des ordres opposés (c'est-à-dire des ordres venant de vendeurs).

Exceptionnellement, les ordres de types « market » n'auront pas de « priority » significatifs. Dans la très grande partie des cas, un ordre « market » trouvera toujours une correspondance.

Maintenant, nous pouvons songer à un scénario extrême : un utilisateur effectue un ordre de ce type en BID lorsque le marché vient tout juste de s'ouvrir. Malheureusement, il n'y a encore aucun vendeur de disponible. Alors, son ordre sera tout de même sauvegardé, mais ne trouvera aucune correspondance. Sa « priority » sera donc marquée par la valeur de 0.

Donc, la valeur de ce champ sera toujours fixée à 0.

(b) Transaction

Pour rappel, une transaction est enregistrée lorsqu'un acheteur et un vendeur se sont mis d'accord pour échanger de l'argent contre un bien.

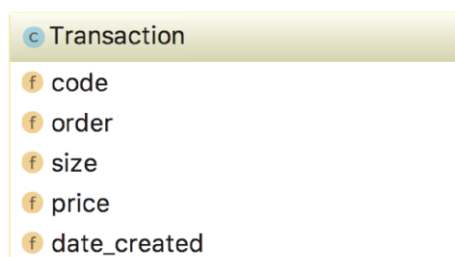


Figure 13 - Détails de la classe "Transaction".

Tout comme pour la classe « Order », nous pouvons récupérer les renseignements basiques tels que l'ordre, le prix et la quantité qui ont été reconnus ainsi que la date et l'heure de création de l'enregistrement.

Une dernière information peut sembler anormale au premier coup d'œil. Il s'agit de l'attribut « **code** ». À chaque fois que deux ordres se correspondent, il y aura toujours deux transactions qui seront sauvegardées, car il y aura à la fois une instance de transaction pour l'acheteur et le vendeur. Il fallait donc être capable de repérer la liaison entre les couples. C'est à ce moment-là que l'attribut « **code** » entre en jeu : les paires se partageront un code identique.

Article V. La conception de l'outil

Section 5.01 Introduction

La conception de ce marché expérimental fut une activité chronophage. Elle a nécessité des recherches et une compréhension minutieuses des diverses actions qu'un commissaire-priseur et un carnet d'ordres opèrent. Seulement après avoir franchi ce cap, il a fallu, tel un puzzle, réassembler le tout à travers l'analyse et la programmation.

Pour rappel, il s'agit de mettre sur pied un logiciel qui autorise les participants à soumettre des ordres de vente ou d'achat (soit un « double auction model ») sur un marché où tous les mouvements sont exécutés en temps réel. Les joueurs doivent pouvoir gérer leur portefeuille à leur guise et visualiser les dernières fluctuations et transactions d'un carnet d'ordres.

Afin d'éviter que le marché paraisse vide lors de son commencement et pour le rendre plus animé, des robots ont été créés. Ceux-ci surveillent les informations et réagissent en conséquence.

Ce chapitre expliquera les notions essentielles de l'application et veillera à ne pas rentrer dans les détails trop techniques de la programmation.

Section 5.02 Les outils utilisés

Avant d'aborder les notions plus théoriques concernant l'architecture du marché et des robots, il est important de présenter brièvement les principaux outils qui ont été utilisés pour mener le projet à son terme.

(a) Django

Django est ce que l'on appelle un « framework » de développement. Il permet aux développeurs d'accélérer leur rendement grâce à des lignes de conduite ainsi que par une multitude d'outils.

Énoncer toutes ses capacités prendrait littéralement plusieurs dizaines de pages et n'est absolument pas l'objectif poursuivi dans ce mémoire. Néanmoins, certaines de ses aptitudes ont permis de gérer et de traiter plus rapidement de nombreux processus. Deux d'entre elles sont explicitées ci-dessous.

Premièrement, celui-ci oblige à adopter une manière de programmer précise et rigoureuse. Certes, sa structure devient plus rigide, mais facilitera les mises à jour ou les futures résolutions de soucis techniques.

Deuxièmement, il gère d'une **manière totalement automatique la base de données** à travers un système d'ORM (Object **R**elational **M**apping).

Grâce à l'ORM, on évite d'encoder des requêtes SQL³¹ pour enregistrer, écrire, supprimer ou mettre à jour des informations de la base de données. En guise de comparaison, voici un exemple qui illustre leur différence :

- En SQL, on réalise une requête de cette manière : « **SELECT * FROM orders WHERE participant_id = 1** ». En décomposant, nous obtenons : **recupère toutes** les données provenant de la table³² « orders » **du participant qui possède l'identifiant 1**.
- Pour obtenir un résultat similaire à travers l'ORM, nous programmons la ligne suivante : « `Order.objects.filter(participant_id=1)` ». L'écriture est plus facile à lire et il est donc plus aisé de se repérer dans le code. En arrière-plan, l'ORM construit pour nous la requête SQL précédemment vue et l'exécute.

³¹ Le langage SQL (Structured Query Language) permet de communiquer avec une base de données.

³² On peut imaginer une « table order » comme une farde où le commissaire-priseur a enregistré tous les ordres.

(b) oTree

oTree (Chen, Schonger & Wickens, 2016) est une plateforme opensource spécialisée dans l'expérimentation. Celle-ci permet de réaliser et de construire des expériences simples avec ses outils qui le sont tout autant. Sur le site officiel, de nombreux exemples y sont présents : public goods game, trust game, le dilemme du prisonnier...

Celui-ci possède de nombreux avantages techniques : il fonctionne sur n'importe quel type d'appareils, aucune installation n'est nécessaire et il a été écrit avec le langage de programmation Python³³. Il est le remplaçant spirituel du programme Z-Tree qui fonctionne sous l'environnement Windows. Ce dernier nécessitait une installation sur un système d'exploitation de Microsoft et devait également être configuré. Avec la solution Z-Tree, il n'était donc pas possible de réaliser des tests sur des appareils mobiles. Or, selon les développeurs d'oTree, il semblait nécessaire de s'adapter au changement d'environnement des supports électroniques : c'est-à-dire l'utilisation quotidienne de notre smartphone en dépit de nos ordinateurs.

Comme cela a déjà été dit plus haut, cette bibliothèque³⁴ a été pensée pour les expérimentations : elle propose donc des outils et des raccourcis pour gérer ces expérimentations avec plus de facilité. À cette fin, elle propose également des mécanismes tels que la **gestion des sessions** qui a été on ne peut plus utilisée lors du développement du marché expérimental. *Grosso modo*, ces sessions permettent aux expérimentateurs de différencier des groupes de participants différents. À titre d'exemple, nous pouvons très bien imaginer une session qui prendra place sur Internet afin de tester la stabilité du système et une autre qui se déroulera dans les locaux de l'UCL où des étudiants y participeront.

Ensuite, il est également facile de mettre en place un **environnement de paramétrage**. Avant de lancer une session, l'administrateur devra alors spécifier certains paramètres. Par exemple,

³³ Ce langage jouit d'une forte notoriété dans le domaine de la recherche et de l'analyse de données.

³⁴ Bibliothèque : en anglais "library" est un ensemble d'outils déjà créés pour éviter au développeur final à devoir réinventer la roue.

il pourra activer des robots, spécifier un temps limite, donner un budget initial à chaque participant, etc.

Enfin, oTree est aussi livré avec un agrégat d'autres bibliothèques. L'une d'entre elles a été extrêmement sollicitée : «django channels». Cette dernière permet d'établir la notion d'instantanéité et d'accessibilité des données dans toute l'application. En d'autres mots, elle ajoute aux informations la possibilité d'être **distribuées**³⁵ **en temps réel** et cette possibilité a été implémentée à travers le passage d'ordres ou la réception des dernières transactions.

Section 5.03 Architectures

L'apparition du pluriel dans le titre n'est pas anodine. Il y a en réalité deux logiciels : le marché expérimental ainsi que les robots.

(a) Le marché expérimental

Afin d'expliquer d'une manière simple l'architecture de cette partie, nous allons poser un contexte durant lequel nous sommes dans une session.

Lors de son commencement, plusieurs mécanismes sont lancés. Le plus important est celui qui initialise tous **les carnets d'ordres**. En effet, l'expérimentateur devra au préalable créer ou réutiliser un fichier où figureront tous les symboles disponibles ainsi que leur prix initial.

Dès que cette étape sera terminée, les utilisateurs pourront accéder à l'application. Une interface leur proposera de **choisir un carnet d'ordres** où ils souhaiteront **passer des ordres** d'achat ou de vente.

Pour exécuter la dernière action citée, les utilisateurs verront un formulaire dans lequel ils devront renseigner les informations suivantes : le type d'ordre (BID/ASK), la quantité et enfin le prix.

³⁵ Pour faire un parallèle avec l'analyse, cette fonctionnalité est accessible grâce à nos « portes » de l'application. C'est-à-dire, grâce à nos « Consumers ».

Place an order

BID ☐ ASK ☐ Quantity : Limit price :

Figure 14 - Formulaire pour placer un ordre.

Si ceux-ci laissent un prix équivalent de zéro, il sera considéré automatiquement par le système comme étant un « **market order** ». Dans le cas contraire, un prix sera renseigné et l'ordre sera considéré comme un « **limit order** ».

Les participants seront **constamment au courant des dernières fluctuations** des carnets d'ordres. Juste au-dessus du formulaire pour encoder un nouvel ordre, ils verront trois tableaux indiquant les données suivantes : cinq « BID » (en partant du « best BID »), cinq « ASK » (en partant du « best ASK ») et les cinq dernières transactions effectuées.

Order book

BID			ASK			TRANSACTION		
Count	Size	Price	Count	Size	Price	Size	Price	Temporal
1	2	57.87 €	1	8	58.10 €	3	57.87 €	17/06/2018 11:58
1	7	57.86 €	1	7	58.65 €	7	57.88 €	17/06/2018 11:58
1	7	57.78 €	1	8	58.74 €			
1	7	57.50 €	1	10	58.87 €			
1	9	57.36 €	1	8	58.93 €			

Figure 15 - Affichage des BID, ASK et des transactions.

Si l'utilisateur possède assez d'argent ou d'actions, son ordre sera enregistré. Dans le cas où le système détecte un vendeur ou acheteur qui convient à l'ordre fraîchement enregistré, alors le logiciel symbolisera la rencontre entre les deux acteurs par une **transaction**. Cet événement entraînera naturellement une mise à jour de leur portefeuille, créditant ou débitant ainsi la somme due.

Pour chapeauter le tout, un commissaire-priseur a été mis en place sous le nom de « **System** ». Il incarne le chef d'orchestre invisible qui garantit entre autres l'association entre les ordres, la création des transactions et la vérification des accès et des actions des utilisateurs. Finalement, il prévient l'utilisateur concerné si un problème s'est passé.

Une **session peut ou non avoir une durée déterminée**. Grâce à cela, les joueurs pourront suivre leur rendement au fur et à mesure du temps et les administrateurs pourront aussi récolter des données supplémentaires, et ce de façon continue.

Enfin, la persistance des ordres se déroule par un système d'insertion et de rotation. Dès qu'un nouvel ordre est réceptionné par le « System », celui-ci prête attention à deux informations : son prix et son heure. Faisons l'hypothèse suivante, un participant décide de créer le « highest BID ». Le « System » l'enregistre et remarque que son prix est le plus haut. Dans ce cas, il procède à une **insertion** et lui affecte une priorité égale à 1 (plus la priorité est basse, plus elle a de chance de trouver une correspondance). Évidemment, l'ancien ordre qui avait une priorité de 1 verra la sienne se faire incrémenter (augmenter de 1). Il s'agit de l'étape de **rotation**. Voici une illustration résumant ces étapes :

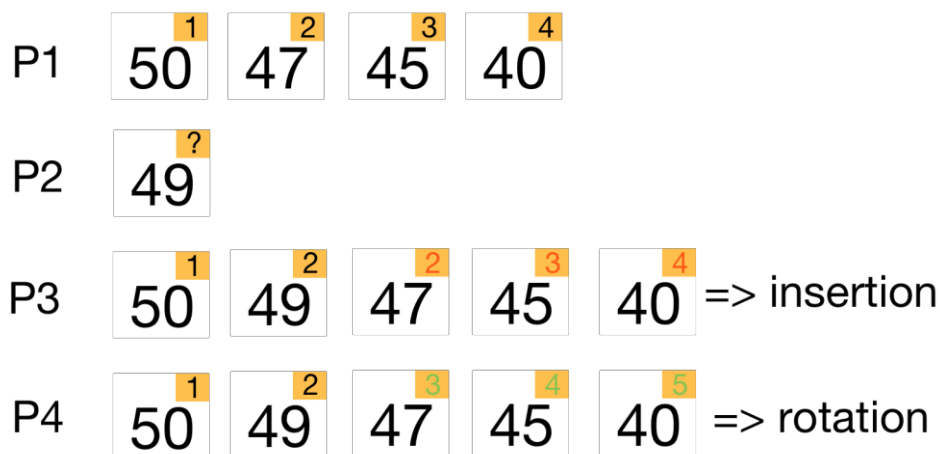


Figure 16 - Illustration des mécanismes « Insertion » et « Rotation ».

Lors de la phase « P1 », nous avons l'état initial de tous nos BID.

En phrase « P2 », le « System » reçoit un nouvel ordre dont son prix a été fixé à 49 €. À ce moment-là, nous pouvons voir que sa priorité actuelle équivaut à un point d'interrogation. En effet, étant donné que cet ordre vient tout juste d'arriver, il n'est pas encore possible de lui fixer une priorité.

En « P3 », nous pouvons voir que cet ordre a été placé juste après celui qui vaut 50 €. Ceci symbolise donc l'étape **d'insertion**. Néanmoins, la priorité des ordres qui valent 47, 45 et 40 € est devenue **obsolète**.

Justement, en « P4 », le « System » réalisera une **rotation** sur toutes les priorités pour effacer ce problème.

(b) Les robots

Au sein du marché, il y a exactement deux robots qui peuvent intervenir selon la configuration initiale mise en place par l'expérimentateur.

(i) *Loader bot*

Tout d'abord, il y a le robot « Loader ». Son rôle est de démarrer le marché en effectuant plusieurs ordres de type BID ou ASK. Grâce à lui, lorsque les joueurs arriveront sur le marché pour la première fois, ils verront que celui-ci est déjà vivant et actif.

Afin qu'il puisse commencer son travail, il est obligatoire de créer³⁶ au préalable un fichier où seront sauvegardés les symboles qui seront utilisés lors des sessions boursières.

Ci-dessous se trouve un exemple :

```
AAPL;Celebre symbole de la pomme;58;0.6;y;10
NVIDIA;Nvidia;40;0.7;n;5
MCST;Microsoft comme on peut s'en douter !;30;0.5;n;4
```

Figure 17 - Exemple d'un fichier pour créer des symboles.

Dans cet ordre, il faut d'abord définir le **nom** du symbole, une **courte description** le représentant, le **prix initial**, le pas maximal de variation du prix [soit « d »]³⁷, la capacité de

³⁶ La procédure pour créer le fichier se trouve à l'annexe suivante : « [Où et comment créer un nouveau fichier où seront stockés les symboles ?](#) »

³⁷ Sa manière d'agir change en fonction du type de distribution voulue. Les points « [Distribution où « d » est aléatoire](#) » et [Distribution où le « d » est constant](#) » en expliquent les différentes nuances.

« **d** » à avoir une **distribution aléatoire ou non**³⁸ et enfin, la **quantité** d'ordre qui sera générée. Tous ces renseignements sont obligatoires pour faire fonctionner cette étape.

Nous pouvons donc nous retrouver avec deux distributions possibles rien qu'en modifiant le caractère « y » par « n ». Ces deux scénarios sont expliqués ci-dessous.

1) Distribution où « **d** » est aléatoire

Nous commencerons par l'illustration suivante :

AAPL;Celebre symbole de la pomme;58;0.6;y;10

Figure 18 - Présentation de la configuration du symbole "AAPL" avec une distribution aléatoire.

En décomposant toutes ces informations avec le formatage présenté ci-dessus, nous obtenons : « un carnet d'ordres dont le nom est **AAPL** dont son prix initial gravitera autour du prix de **58 €**. De plus, son « **d** » sera fixé à **0,6** et suivra également une **distribution aléatoire**. Finalement, il générera à la fois **10 ordres BID et ASK** ».

Ainsi, le robot effectuera au total 20 itérations (10 pour créer les ordres BID et 10 autres pour les ASK). Lors de chacune d'entre elles, une fonction retournera une variation (notée Δ) qui affectera le prix initial. Premièrement, elle déterminera un chiffre compris entre 0,01 et le « **d** » et ensuite, la variation sera calculée en multipliant ce chiffre par le numéro de l'itération actuelle. Au bout du compte, on déduira du prix initial du BID le résultat Δ et inversement dans le cas du ASK.

Tout cela mis en musique, nous obtenons la formule ci-après :

$$\begin{aligned} \epsilon &= \text{montant initial} \\ i &= \text{numéro de l'itération actuelle} \\ \Delta &= \text{variation} = i * d \end{aligned}$$

³⁸ Si l'on suggère au robot d'établir une distribution aléatoire du « **d** », il faut indiquer « y » pour « yes ». Dans le contraire, ça sera « n » pour « no ».

$$\text{futur prix BID} = \epsilon - \Delta$$

$$\text{futur prix ASK} = \epsilon + \Delta$$

Ci-dessous se trouve un tableau montrant les 5 premières itérations lors de la génération des BID :

Tableau 3 - Démonstration du "d" aléatoire du "Loader Bot".

i	« d » aléatoire	Δ	Futur prix BID
1	0,50	0,5	57,5
2	0,22	0,44	57,56
3	0,59	$3 * 0,59 = 1,77$	$58 - 1,77 = 56,23$
4	0,34	1,36	56,64
5	0,08	0,4	57,6

2) Distribution où le «d» est constant

Deuxième illustration :

MCST;Microsoft comme on peut s'en douter !;30;0.5;n;4

Figure 19 - Présentation de la configuration du symbole "AAPL" avec une distribution constante.

Les renseignements essentiels sont : « le «d» a été fixé à 30 et est constant. La quantité, quant à elle, est de 4 BID et ASK. »

D'un point de vue mathématique, il n'y a qu'un seul élément qui est altéré : plus aucun nombre aléatoire ne sera utilisé. Ainsi, uniquement le calcul de la variation change.

$$\Delta = \text{variation} = i * d$$

Nous utiliserons les ordres ASK pour clarifier cette situation :

Tableau 4 - Démonstration du "d" constant du "Loader Bot".

i	Δ	Nouveau prix
1	0,5	30,5
2	1	31
3	$3 * 0,5 = 1,5$	$30 + 1,5 = 31,5$
4	2	32

(ii) *Player bot*

Le « Player bot » participe à rendre le marché dynamique et interactif en utilisant les dernières informations provenant des carnets d'ordres³⁹.

Ce robot peut posséder plusieurs manières d'opérer, mais seule la configuration par défaut sera démontrée et expliquée.

Premièrement, sa première caractéristique concerne sa **vitesse**. Lors de la phase de paramétrage, nous pouvons lui indiquer combien de requêtes par minute nous souhaitons qu'il réalise. Grâce à cette information, chaque seconde, le robot se posera la question suivante : dois-je passer un ordre ? La réponse à cette interrogation lui est apportée par un algorithme dont le fonctionnement est le suivant : celui-ci génère un chiffre aléatoire compris entre 1 et 60 et si celui-ci est plus petit ou égal à la vitesse encodée précédemment, alors un ordre est passé.

Ensuite, nous pouvons également choisir sa **préférence au niveau des ordres** : BID ou ASK. Les différentes probabilités lorsqu'il doit effectuer ce choix sont reprises dans ce tableau :

Tableau 5 - Tableau des préférences du "Player bot".

Type/Alias	BID	ASK
1 (alias pour pref_for_bid)	75%	25%
2 (alias pour pref_for_ask)	25%	75%
3 (alias pour no_pref)	50%	50%

Par défaut, le robot n'a aucune préférence et aura donc une chance équiprobable de choisir entre les deux. S'il passe toutes les conditions précédemment citées, il aura également une chance sur deux de choisir de passer un ordre soit en « market » ou en « limit ».

³⁹ Vous pouvez retrouver dans les annexes le fichier où sont enregistrées les dernières informations des carnets d'ordres : « [Fichier récapitulatif le best BID et le best ASK des carnets d'ordres](#) ».

Deux arbres décisionnels résument ci-dessous ces pourcentages pour plus de clarté.

Le premier arbre décisionnel illustre la configuration suivante : pas de préférence (niveau 3 du tableau précédent), 50 % de chance en moyenne d'effectuer une requête (donc fixé à 30 requêtes/minute).

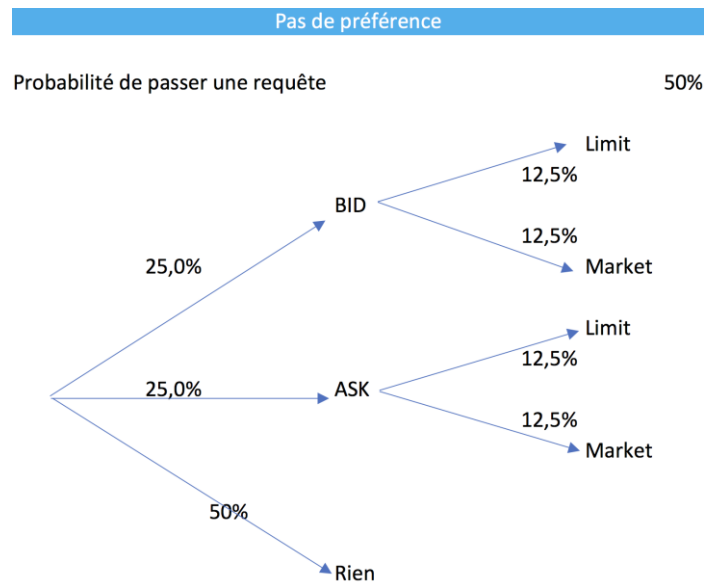


Figure 20 - Arbre décisionnel du robot "Player" lorsqu'il n'a pas de préférence.

Le second arbre illustre une configuration similaire mais qui possède une préférence pour les ordres de type BID.

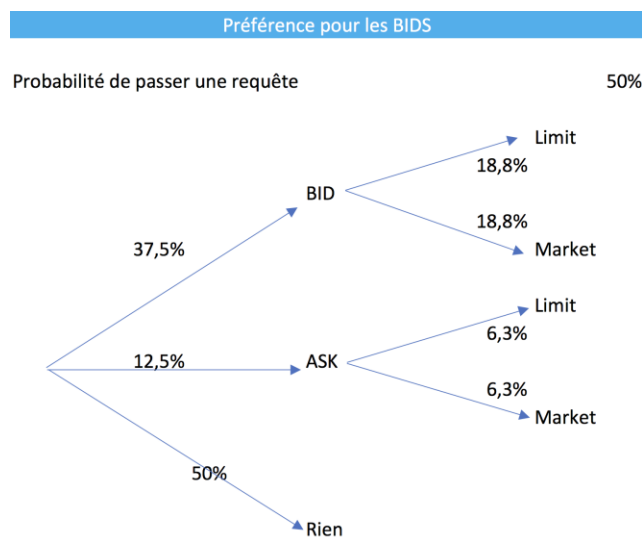


Figure 21 - Arbre décisionnel du robot "Player" lorsqu'il a une préférence pour les BID.

Pour en terminer avec l'architecture des « Player bots », il ne reste plus qu'un point à traiter : le calcul des prix des ordres en « limit ». Lors de la première étape, ils doivent suivre et d'abord aller chercher les dernières informations des carnets d'ordres dans un fichier récapitulatif.

Au sein de ce dernier se trouvent le « highest BID » et le « lowest ASK ». En se basant sur cela, le « Player bot » calculera d'abord le spread (qui est la différence entre le prix du best ASK et du best BID) en suivant la formule :

$$spread = best\ ask - best\ bid$$

Lorsque ce chiffre sera obtenu, une fonction générera un chiffre aléatoire (nommé ici Δ) compris entre zéro et le spread. Dans le cas où le robot émettra un ordre BID, il créera le nouveau prix de l'ordre en additionnant au best BID la moitié du spread puis en déduisant Δ .

En posant toutes ces informations mathématiquement, nous obtenons les formules suivantes :

$$\Delta = \text{chiffreAleatoireEntreZeroEtSpread}(\text{spread})$$

$$\text{nouveau prix BID} = \text{best bid} + \frac{\text{spread}}{2} - \Delta$$

$$\text{nouveau prix ASK} = \text{best ask} - \frac{\text{spread}}{2} + \Delta$$

Nous pouvons illustrer ce raisonnement en prenant ces données comme base pour la suite :

Tableau 6 - Informations de départ en vue de calculer le nouveau prix BID/ASK.

Best BID	80
Best ASK	85
Spread	85 – 80 = 5

La première étape est de faire appel à la fonction « chiffreAleatoireEntreZeroEtSpread » en lui passant comme argument notre spread dont la valeur est de 5.

Suite à cela, celle-ci nous renverra une valeur qui sera comprise entre 0 et 5. Supposons qu'elle nous renvoie la valeur 3,63, Δ vaut 3,63.

Si le robot a décidé de créer un ordre de type BID, alors il exécutera le calcul suivant :

$$\text{Futur prix de l'ordre BID} = 80 + \frac{5}{2} - 3,63 = 78,87$$

Toutefois, il se pourrait que l'étape de division du spread par deux vous titille. Il est plus facile d'expliquer sa nécessité en le retirant de l'équation. Supposons que nous **n'appliquons plus la division** du spread. Dans ce scénario, tous les futurs ordres seront toujours en dessous du best BID ou haut dessus du best ASK. En d'autres termes, notre robot serait tout bonnement incapable de produire un best BID ou un best ASK ! Par contre si nous l'injectons, il aura une chance sur deux d'en créer.

Ci-dessous se trouvent deux tableaux sans (à gauche) et avec l'intervention du spread (à droite) :

Tableau 7 - Démonstration de la formule sans ou avec l'injection du spread.

1) Sans l'intervention du spread			2) Avec l'intervention du spread		
# de tirage	Δ^{40}	Prix de l'ordre BID	# de tirage	Δ	Prix de l'ordre BID
1	3,14	$80 - 3,14 = 76,86$	1	3,73	$80 + 5/2 - 3,73 = 78,77$
2	1,66	$80 - 1,66 = 78,34$	2	4,65	$80 + 5/2 - 4,65 = 77,85$
3	4,41	$80 - 4,41 = 75,59$	3	1,05	$80 + 5/2 - 1,05 = 81,45^{41}$
4	2,55	$80 - 2,55 = 77,45$	4	3,58	$81,45 + 3,55/2 - 3,58 = 79,65^{42}$
5	0,24	$80 - 0,24 = 79,76$	5	0,97	$81,45 + 3,55/2 - 0,97 = 82,26$

Dans le tableau où le spread n'a pas été inclus, nous pouvons observer que tous les prix créés ne dépassent jamais le best BID.

Au contraire, dans le second tableau, le « Player bot » réalise, à deux reprises, des best BID et les données décroissent beaucoup moins rapidement tout en gardant leur cohésion.

⁴⁰ Rappel : Δ est généré aléatoirement entre zéro et le spread.

⁴¹ L'encadrement des chiffres par cette couleur indique l'apparition d'un nouveau best BID.

⁴² À la vue du résultat précédent (soit 81,45), nous étions obligés de modifier le best BID ainsi que le spread. Ceci explique les changements effectués dans le calcul.

Section 5.04 Recul critique/perspectives

En prenant du recul et en comptabilisant ses qualités ainsi que ses défauts, de nombreuses pistes peuvent être évoquées en vue de l'améliorer. Cette partie est divisée en trois sous-points : les qualités, les défauts et les révisions potentielles.

L'objectif initial de ce projet a été atteint : avoir la possibilité d'étudier le comportement des agents sur un marché expérimental. Pour faciliter la phase de l'analyse, les données ont été encodées de telle sorte qu'il soit plus évident de les filtrer, de reconnaître les utilisateurs et d'avoir en sa possession les dimensions temporelles. La configuration se veut également simple à mettre en place : la modification de quelques champs est requise pour altérer le fonctionnement de toute l'application.

Le premier défaut à signaler est l'ajout d'une technologie supplémentaire pour fabriquer l'interface graphique. À cause de cela, il faut d'abord apprendre à la manipuler pour modifier en profondeur le côté utilisateur. Deuxième accroc, on ne peut pas changer le comportement du jeu en pleine partie. En d'autres termes, la configuration d'une session est immuable.

Améliorer un projet de ce genre est une quête « sans fin » car il sera toujours possible d'y ajouter des fonctionnalités. Parmi celles-ci certaines d'entre elles méritent de s'y attarder. La génération de graphiques en temps réel permettrait par exemple de voir rapidement la formation de certains schémas et de réagir en conséquence. Il serait également intéressant de maintenir le marché actif malgré l'absence des participants. Ainsi, observer les réactions de ces derniers quand ils verront les changements dans les carnets d'ordres pourrait piquer notre curiosité.

Article VI. Expérimentation

Dans le cadre de ce mémoire, une expérimentation a été menée pour vérifier si le biais comportemental « priming effect » a un impact sur les participants. Il a été introduit, tout au début de ce document, en partant du principe que l'on peut influencer un individu par un stimulus [des images ou encore des phrases] avant que celui-ci n'exécute une action.

Ce biais a été inséré dans le marché expérimental de ce mémoire en affichant une phrase qui a un caractère joyeux⁴³. Plusieurs hypothèses peuvent d'ores et déjà être posées afin de confirmer si oui ou non, celui-ci affecte les participants.

Avant de rentrer dans le cœur du sujet, il faut être conscient que cette expérimentation est un prétexte pour mettre en pratique les capacités du marché expérimental. Le noyau de ce mémoire est bien le développement d'un programme permettant l'analyse des comportements de sujets sur un marché boursier. L'expérimentation, quant à elle, atteste le fonctionnement de l'outil et montre comment les données peuvent être récupérées et analysées.

Section 6.01 Quid de la problématique ?

La question principale est la suivante : en quoi l'apparition d'une phrase à caractère joyeux, lors d'une session boursière, influence-t-elle le choix des traders ?

L'utilisation d'une phrase joyeuse dans cette expérimentation n'est pas anodine. Son existence réside dans le fait qu'elle déclenchera, par supposition, un sentiment positif chez les personnes qui la verront. Ainsi, voici les impacts supputés :

⁴³ Sens du mot joyeux dans ce contexte : qui provoque au moins un sourire ou projette la personne dans une situation où elle se sent apaisée/bien.

Tableau 8 - Liste des effets supputés par l'utilisation du "priming effect".

Nom de la caractéristique étudiée	Effet attendu sur le groupe « primed ⁴⁴ »
Quantité d'ordres introduits.	Ce groupe en produit plus que le groupe neutre.
Impatience du participant d'acquérir un bien ou de recevoir l'argent après une vente.	Ce groupe aura plus tendance à acheter et à vendre rapidement (soir en réalisant des ordres de type « market »).
La limitation des prix	Les prix renseignés par ce groupe (soit les ordres en « limit ») auront une différence marquée par rapport à l'autre. Cela vaut aussi bien pour les BID que pour les ASK.

Section 6.02 Hypothèses

En nous basant sur le tableau précédent, voici les hypothèses nulles qui ont émergé⁴⁵ :

- **H1_0.** Il n'existe aucun lien entre la présence d'un message positif et l'augmentation du nombre d'ordres exécutés.
- **H1_1.** Il existe un lien positif entre la présence d'un message positif et le nombre d'ordres exécutés.
- **H2_0.** Il n'existe aucun lien entre la présence d'un message positif et la fréquence d'ordres BID.
- **H2_1.** Il existe un lien positif entre la présence d'un message positif et la fréquence d'ordres BID.

⁴⁴ Groupe ayant vu la phrase joyeuse qui est censée changer les résultats.

⁴⁵ Le premier chiffre après le H désigne le numéro de l'hypothèse. Le chiffre après le « underscore » ('_') vaut 0 si l'on aborde l'hypothèse nulle et vaut 1 dans le cas de l'alternative.

- **H3_0.** Il n'existe aucun lien entre la présence d'un message positif et la fréquence d'ordres ASK.
- **H3_1.** Il existe un lien positif entre la présence d'un message positif et la fréquence d'ordres ASK.
- **H4_0.** Il n'existe aucun lien entre la présence d'un message positif et la façon de fixer le prix d'un ordre BID.
- **H4_1.** Il existe un lien positif entre la présence d'un message positif et la limite de prix des ordres BID.
- **H5_0.** Il n'existe aucun lien entre la présence d'un message positif et la façon de fixer le prix d'un ordre ASK.
- **H5_1.** Il existe un lien négatif entre la présence d'un message positif et la limite de prix des ordres ASK.

Section 6.03 Mise en place initiale

(a) Les participants

Un total de 38 participants, venant de milieux hétérogènes et d'âges variés, a été enregistré. *In fine*, la moitié d'entre eux sera exposée à la phrase pouvant provoquer un biais comportemental tandis que l'autre groupe jouera au logiciel non modifié (c'est-à-dire, qu'il ne verra aucune phrase s'afficher sur leur écran).

(b) Configuration d'une phase d'expérimentation

Au préalable, chaque session a été configurée de la manière suivante :

- Durée : chacune est découpée en 3 tours de 3 minutes. Par conséquent, l'expérimentation dure 9 minutes ;
- Phrase induisant potentiellement le biais : « C'est l'été ! Vive le soleil, les sorties et les barbecues ! »

À propos des actions disponibles, voici toutes les informations de configuration s'y rapportant :

Tableau 9 - Liste des actions utilisées lors de l'expérimentation.

	Prix de départ	Variation	Distribution aléatoire ?	Quantité
COOKIE	30	1	Oui	20
ICECREAM	40	0,7	Non	20
CHOCOCAKE	20	0,5	Oui	20
LEMON	7	0.3	Non	20
HOTDOG	10	0,5	Oui	20
SALAD	4	0.2	Non	20
FISH	12	1.2	Oui	20

(c) Outils statistiques utilisés

Un logiciel a été créé pour agréger et filtrer les résultats provenant de la base de données.

Son premier rôle est d'effacer les bruits. Après extraction des données, il y a 2 sources de données qui peuvent en provoquer : les ordres des robots et les participants qui ont effectué moins de 5 ordres.

Son second rôle est de réaliser toute une série de moyennes ou de transformations sur les données. Par exemple, il calcule la quantité moyenne spécifiée lors d'un passage d'ordre, le prix moyen enregistré lors d'un ordre en « limit », le laps de temps entre chaque passage d'ordre, etc. Il scinde également ces résultats en 2 fichiers différents : le groupe ayant été exposé au « priming effect » [alias le groupe variable] et l'autre groupe ayant participé au marché sans avoir subi de modifications [alias le groupe de référence].

Enfin, lorsque des insights⁴⁶ ont été repérés, ceux-ci seront vérifiés afin de constater ou non s'ils sont statistiquement significatifs. Pour ce faire, un test de student unilatéral sera dès lors employé. Quant au seuil de significativité [*alpha*], celui-ci a été fixé à 5%.

⁴⁶ Un insight est la valeur ajoutée que l'on peut tirer des données après avoir opéré des traitements dessus. Nous pouvons également faire une analogie avec le mot « indicateur » en français.

(d) Formatage des données

Dès qu'une session boursière est terminée, nous recueillons uniquement les données qui abordent les ordres passés. Les attributs analysés de la base de données sont : le type de l'ordre (soit BID ou ASK), l'identifiant du participant, la quantité, le prix, le jour et l'heure de la création. En voici un exemple :

id	type	order_book	size	price	has_been_filled	participant_id	datetime	priority	round
1223	BID	31	7	31,1108	false	128	2018-07-02 21:15:50.981935 +02:00	4	3

Figure 22 - Données brutes provenant de la base de données.

Section 6.04 Traitement des données

(a) Méthode de création des variables

Après avoir récupéré et nettoyé les données, nous pouvons désormais créer nos variables qui nous serviront à l'analyse des hypothèses.

Ci-dessous se trouve un tableau exhaustif de toutes les variables créées ainsi que les formules mathématiques attachées.

Tableau 10 - Liste non exhaustive des variables créées pour tester les hypothèses + procédures pour les obtenir.

Nom des variables + liaison avec son hypothèse	Explication	Procédure
« num_orders » + HQ ⁴⁷	Nombre total d'ordres effectués par un participant	
« tot_size »	Nombre total de la quantité qu'un participant a réalisé lors de ses passages d'ordres.	On réalise la somme de l'attribut « size » de tous les ordres d'un participant.
« mean_size »	C'est la quantité moyenne qu'un participant a encodé.	$\frac{tot_size}{n}$
« mean_price_bid » + HFB ⁴⁸ « mean_price_ask » + HFA ⁴⁹	Plusieurs traitements ont été apportés aux variables des prix. Il a fallu normaliser tous les prix des ordres en fonction de leur carnet d'ordres puis en faire la moyenne.	$\frac{tot_prices_normalized}{n}$
« qt_market_[bid ask] » + [HRA ⁵⁰ HRV ⁵¹] & « qt_limit_[bid ask] »	Quantités totales des ordres de type « market » et « bid », « ask » ainsi que « limit » et « bid », « ask ».	Nous filtrons et additionnons tous les ordres de types « market » puis on les refiltre par « bid » ou « ask ». La logique est la même pour les ordres « limit ».
« mean_time_elapsed_before_new_order »	Temps moyen d'attente entre deux passages d'ordres	$\frac{\sum_{i=2}^n (dateOrder_{i-1} - dateOrder_i)}{n - 1}$

⁴⁷ HQ = hypothèse sur la quantité.

⁴⁸ HFB = hypothèse sur le fixage des prix des ordres BID.

⁴⁹ HFA = hypothèse sur le fixage des prix des ordres ASK.

⁵⁰ HRA = hypothèse sur la rapidité à acquérir des actions.

⁵¹ HRV = hypothèse sur la rapidité à vendre ses actions.

Ci-dessous se trouve une partie⁵² des variables utilisées lors des tests statistiques. La création des deux groupes s’effectue grâce à l’utilisation de la dummy variable « primed ? ». En ce qui concerne les « nan⁵³ » visibles dans ce tableau, ces données ne sont pas ignorées et sont remplacées par la moyenne respective de leur colonne.

Enfin, le logiciel d’analyse a retiré 5 enregistrements. Soit 2 provenant du groupe « primed » et 3 du groupe neutre. Il nous reste 33 enregistrements.

Tableau 11 - Vue partielle des variables créées.

num_ords	total_size	mean_size	mean_price_bid	mean_price_ask	qt_market_bid	qt_market_ask	qt_limit_bid	qt_limit_ask	mean_elapsed_time	Prime d ?
6	44	7.33	0.89	nan	0	0	6	0	90	True
9	83	9.22	1.0	0.85	0	0	6	3	39	True
14	170	12.14	nan	1.0	6	7	0	1	38	True
8	61	7.625	1.0	1.0	3	0	3	2	64	True
5	7	2.33	1.0	1.0	0	0	2	1	60	False
4	25	6.25	1.0	0.70	0	0	2	2	39	False
17	80	5.71	nan	1.0	7	4	0	3	31	False
11	60	7.5	1.0	1.0	2	1	4	1	45	False

⁵² La totalité des données se trouvent dans l’annexe : « [Liste exhaustive des variables obtenues](#) ».

⁵³ « nan » signifie « **not a number** ».

(b) Analyse des variables

Après avoir conçu nos variables, il faut passer à l'étape de les charger et de les analyser. Cette partie, d'une importance capitale, est malgré tout réalisée en à peine quelques lignes de code.

En effet, l'outil statistique « [scipy](#)⁵⁴ » fournit un module « stats » qui permet de produire un test bilatéral de student entre 2 groupes. Dans notre cas, il ne nous manque plus que de charger les résultats et de faire appel à la méthode « stats.ttest_ind (groupe1, groupe2) ». Pour rendre le test unilatéral, nous divisons toutes les p-value obtenues par 2.

Section 6.05 Résultats de l'expérimentation

Avec un *alpha* de 5 %, voici les résultats des tests statistiques de student (par ordre croissant de p-value) :

Tableau 12 - Résultats de l'expérimentation.

	p-value (cas unilatéral)	Est-elle significative ?
qt_market_ask	0.11	Non
mean_price_bid	0.16	Non
mean_elapsed_time	0.17	Non
total_size	0.25	Non
num_orders	0.27	Non
qt_market_bid	0.30	Non
mean_size	0.31	Non
qt_limit_ask	0.33	Non

Nous ne pouvons dans aucun des cas rejeter les hypothèses nulles.

⁵⁴ Sur la page l'index de leur [site web](#), nous pouvons y lire (en anglais) que scipy est un écosystème de programmes open source pour les mathématiques, les sciences, l'ingénierie [...].

Maintenant, il serait intéressant d'aller plus en profondeur pour comprendre la première variable (soit « qt_market_ask ») étant donné que sa p-value est la plus « proche » de l'alpha fixé. Pour ce faire, le logiciel [Tableau](#)⁵⁵ sera employé pour en obtenir une représentation graphique.

(a) Analyse approfondie de « qt_market_ask »

Même si les résultats sont criants de vérité, il est important de décrire la procédure qui a été utilisée pour créer ce graphique.

L'axe des abscisses est composé de deux informations : si le groupe a été « primed » ou non et l'identifiant du participant. Grâce à cette formation, il est aisé de faire une séparation visuelle des résultats. L'axe des ordonnées affiche seulement les quantités totales des ordres de type ASK en « market » par participant et par type de groupe.

Après ces courtes explications, il est possible de savoir pourquoi cette variable a la plus petite p-value. Nous relevons que le groupe « primed » a bel et bien effectué plus d'ordres ASK en « market » que l'autre groupe.

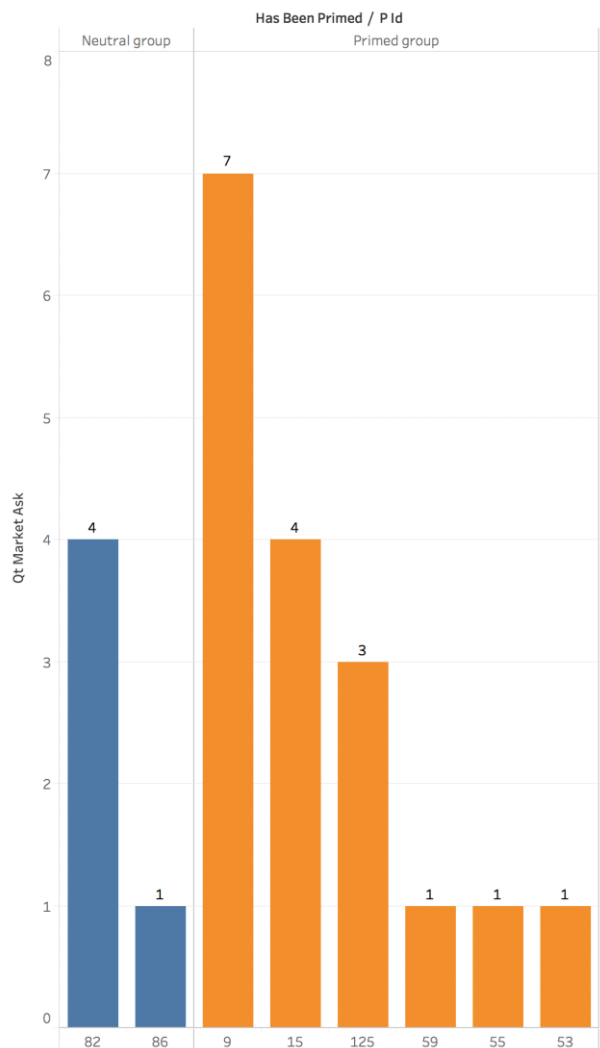


Figure 23 - Graphique de la variable "qt_market_ask".

Toutefois, nous avons trop peu de résultats. Seuls

12 % des participants ont au moins effectué un ordre de ce type tandis que 31 % des participants de l'autre groupe l'ont fait. En dernière analyse, nous remarquons que cette variable n'est pas pertinente à cause de la faible quantité d'informations recueillie.

⁵⁵ [Tableau](#) est un logiciel permettant de créer des graphiques avec une très grande facilité. La plupart du temps, un drag'n'drop suffit pour en produire.

Section 6.06 Recul critique

Tous les résultats précédemment obtenus peuvent être expliqués par deux faits : l'expérience des participants ou tout simplement, les effets attendus du « priming effect » n'existent pas à travers cette mise en place.

Concernant la première partie traitant de leur expérience dans ce domaine, ceux-ci n'ont pas ou peu de connaissance des marchés boursiers. Dans le but d'éviter de les lancer tête baissée vers cet environnement économique, un fichier « foire aux questions » ainsi qu'une [vidéo](#) ont été produits pour cette occasion. Malgré cela et d'après certains retours, tous les concepts n'ont pas été bien cernés. Au préalable, il aurait fallu rassembler des professionnels des marchés ou préparer des cours pour former les personnes.

Quant à la mise en place de l'expérimentation, l'ajout de plus d'éléments pour altérer leur comportement aurait peut-être été judicieux. Par exemple, on pourrait songer à fixer un fond d'écran, représentant la notion même des vacances, derrière le marché boursier.

À travers ce mémoire et cette courte expérimentation, il n'est pas possible d'affirmer que le « priming effect » altère le comportement des individus. *De facto*, il n'existe pas assez de preuves pour confirmer le lien entre l'achat, la vente d'actions et la façon de fixer le prix des ordres combinés avec un stimulus antérieur.

Article VII. Conclusion

Au début de l'écriture de ce mémoire, seule la rédaction d'un marché expérimental devait être réalisée. Au fur et à mesure des semaines, celui-ci s'enrichissait continuellement : de nouvelles fonctionnalités ont été ajoutées, les robots « Loader » et « Player » ont été développés et un logiciel d'analyse et de filtrage des résultats s'est vu nécessaire pour accélérer les processus de traitement des données récoltées.

Grâce à l'expérimentation, ce marché expérimental a eu l'occasion d'accueillir près de 15 participants en même temps. En quelque sorte, c'était une épreuve qui mesurait les capacités du software à distribuer et à traiter les données en temps réels. De plus, ce fut également un moyen d'observer s'il était possible de tirer des analyses sur le biais comportemental du « priming effect ». Les résultats sont clairs : rien ne prouve, pour l'instant, que celui-ci affecte ou altère les actions des participants.

Dans un futur proche, ce dernier pourra être utilisé à plusieurs fins : enseigner le marché des actions aux étudiants lors d'un cours de finance ou pour faire de la recherche exploratoire. Nul doute qu'il est possible de l'améliorer en rendant les robots intelligents avec l'aide du machine learning, ou en créant dynamiquement des graphiques, etc.

Enfin, il ne reste plus qu'à l'employer pour déceler ou prouver l'existence des mystères encore non résolus venant du monde financier.

Article VIII. Bibliographie

- Bargh, J. A., Chen, M., & Burrows, L. (1996). Automaticity of Social Behavior: Direct Effects of Trait Construct and Stereotype Activation on Action. *Journal of Personality and Social Psychology*, pp. 230–244.
- Bargh, J., Chen, M., & Burrows, L. (1996). Automaticity of Social Behavior: Direct Effects of Trait Construct and Stereotype Activation on Action. *Journal of Personality and Social Psychology*, 71, pp. 230-244.
- Chen, D., Schonger, M., & Wickens, C. (2016). oTree - An open-source platform for laboratory, online, and field experiments. *Journal of Behavioral and Experimental Finance*.
- Daniel, K., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, pp. 263-291.
- De Bondt, W., Muradoglu, G., Shefrin, H., & Staikouras, S. K. (2008). Behavioral Finance: Quo Vadis ?
- De Winne, R., & D'Hondt, C. (2017, 3). La finance comportementale : enjeux et perspectives. *Regards économiques*.
- Downing, D. (2012, 1). Stock Market Simulation. *American Economic Association*.
- Investopedia. (s.d.). *Auction Market*. Récupéré sur Investopedia: <https://www.investopedia.com/terms/a/auctionmarket.asp>
- Kahneman, D. (2011). *Thinking, Fast and Slow*. Penguin Books.
- Kang, W., & Sun, C.-h. (2011). Building the model of artificial stock market based on JASA. *Procedia Engineering*.
- Kay, A., Wheeler, S., Bargh, J., & Ross, L. (2004). Material priming: The influence of mundane physical objects on situational construal and competitive behavioral choice. *Organizational Behavior and Human Decision Processes*, pp. 83-96.
- LesEchos.fr. (s.d.). *Actionnaire*. Récupéré sur lesechos.fr: https://www.lesechos.fr/finance-marches/vernimmen/definition_actionnaire.html
- Maslov, S. (1999, 2). Simple model of a limit order-driven market. *Physica*.
- McRaney, D. (2011). *You Are Not So Smart*. Oneworld Book.
- Palan, S. (2015). GIMS—Software for asset market experiments. *Journal of Behavioral and Experimental Finance*, pp. 1-14.
- Quantpedia. (s.d.). *Pre-Holiday Effect*. Récupéré sur quantpedia.com: <https://quantpedia.com/Screener/Details/83>
- Sunder, S. (s.d.). What Have We Learned From Experimental Finance ?

Tzu, S. (s.d.). *L'art de la guerre*.

Wagdi, O. (2017). Behavioral Finance: Overview. *International Parpers of Finance and Business Administration*.

Article IX. Iconographie

Figure 9 - Dave Gandy - Icône de la base de données, sur le site www.iconfinder.com et téléchargée à l'adresse suivante : https://www.iconfinder.com/icons/1608662/database_icon

Figure 9 – Honza Douzek - Icône du fichier CSV, sur le site www.iconfinder.com et téléchargée à l'adresse suivante : https://www.iconfinder.com/icons/199249/csv_extension_file_format_icon

Figure 10 – Becris . - Icône de l'avion en feuille de papier, sur le site www.iconfinder.com et téléchargée à l'adresse suivante : https://www.iconfinder.com/icons/2205231/launch_online_post_send_start_icon

Figure 10 – Yannick Lung - Icône du drapeau, sur le site www.iconfinder.com et téléchargée à l'adresse suivante : https://www.iconfinder.com/icons/183233/flag_icon

Figure 10 – Anastasya Bolshakova - Icône du signe « start », sur le site www.iconfinder.com et téléchargée à l'adresse suivante : https://www.iconfinder.com/icons/984752/movie_play_player_start_turn_up_icon

Figure 10 – Xicons.co - Icône des pièces de monnaies, sur le site www.iconfinder.com et téléchargée à l'adresse suivante : https://www.iconfinder.com/icons/2024650/buy_cash_coin_coins_money_pay_icon

Figure 10 – asad jawad eshtawe - Icône du signe « stop », sur le site www.iconfinder.com et téléchargée à l'adresse suivante : https://www.iconfinder.com/icons/387517/character_math_minus_sign_stop_icon

Article X. Annexes

Section 10.01 Comment lancer une expérimentation ?

Tout en haut à gauche, dans le menu, se trouve le lien « Sessions ». En cliquant dessus, vous atterrirez sur la page suivante :

Sessions

Create new session ▾

Figure 24 - Accueil pour créer une session.

Pour créer une nouvelle session, il faut, évidemment, valider son choix en appuyant sur le bouton « Create new session ». Dès lors cette étape terminée, vous obtiendrez le contenu ci-dessous :

Create a new session

Session config:

Number of participants

Create

Figure 25 - Formulaire pour créer une session.

Premièrement, il faut faire un clic sur le menu déroulant ayant comme libellé « Session config ». Un seul choix sera disponible et devra être sélectionné : « market_model ».

Pour le second champ, vous devez indiquer le nombre de participants. **Attention**, veuillez toujours ajouter une personne à votre nombre réel de participants. En effet, le dernier de la liste sera remplacé par les robots. Imaginons le scénario où nous devons réaliser une expérimentation sur 10 personnes. Nous encoderons donc 11 car le 11^e participant sera représenté par les robots (« Loader » et « Player »).

Dans le cas où vous souhaitez modifier les paramètres de configuration, il sera nécessaire d'effectuer un clic sur ce lien :

Configure session ⚙

Figure 26 - Bouton pour configurer une session.

Juste à la seconde d'après, vous aurez l'occasion de modifier toutes ces variables :

Custom

initial_budget	1000
load_loader_robot	☒
load_player_robot	☒
max_row_to_display	5
player_robot_favourite_speed	5
player_robot_favourite_type_order	3
priming_effect_sentence	coucou
symbol_file	symbol_file_1.csv
time_market	0

Figure 27 - Paramétrage d'une session.

Voici dans l'ordre les significations de chacune d'entre elles :

« initial_budget » : c'est le budget initial que vous accordez à tous les participants ;

« load_loader_robot » : si coché, le robot « Loader » chargera les premiers BID et ASK lors de la phase « Building market » ;

« load_player_robot » : si coché, le robot « Player » jouera avec les participants ;

« max_row_to_display » : indique le nombre de lignes maximales que contiendront les 3 tableaux (BID, ASK et Transaction) lorsque l'application sera en phrase « Market running » ;

« player_robot_favourite_speed » : indique le nombre de requêtes par seconde que le robot « Player » réalisera ;

« player_robot_favourite_type_order » : indique le type de préférence des ordres que le robot « Player » réalisera. Si la valeur vaut 1, alors le robot aura 75% de chance de faire des BID. Si la valeur vaut 2, alors le robot aura 75% de chance de faire des ASK. Enfin, si elle vaut 3, le robot n'aura aucune préférence ;

« priming_effect_sentence » : il permet d'afficher une phrase précise à tous les utilisateurs dont leur numéro d'identification est impair ;

« symbol_file » : indique le fichier où seront contenus tous les symboles de votre marché expérimental ;

« time_market » : indique en minute la durée d'ouverture du marché lors de chaque session boursière.

Enfin, vous pourrez terminer la configuration de l'application en cliquant sur le bouton « Create ».

Section 10.02 Où et comment créer un nouveau fichier où seront stockés les symboles ?

Veillez d'abord suivre cette arborescence de dossiers : ucl_master_memoire_2 > market_model > symbols_files.

À l'intérieur de ce dossier, vous pourrez y insérer vos propres symboles. Toutefois, il est nécessaire de respecter ce formatage au sein d'un fichier CSV⁵⁶ :

NOM_DU_SYMBOLE ; **DESCRIPTION_DU_SYMBOLE** ; **PRIX_INITIAL** ; d ;
DISTRIBUTION_ALEATOIRE_OU_NON ; **QUANTITE_SOUHAITEE**

Vous trouverez ci-dessous deux exemples :

```
HOTDOG;Miam !;10;0.5;y;10  
SALAD;Good for you !;4;0.2;n;10
```

Figure 28 - Exemple d'un fichier pour créer des symboles.

Comme vous pouvez vous en douter, si l'on souhaite une distribution aléatoire, on indiquera « y » (y signifie yes) et « n » pour son contraire.

⁵⁶ Toutes les valeurs sont séparées par un point-virgule.

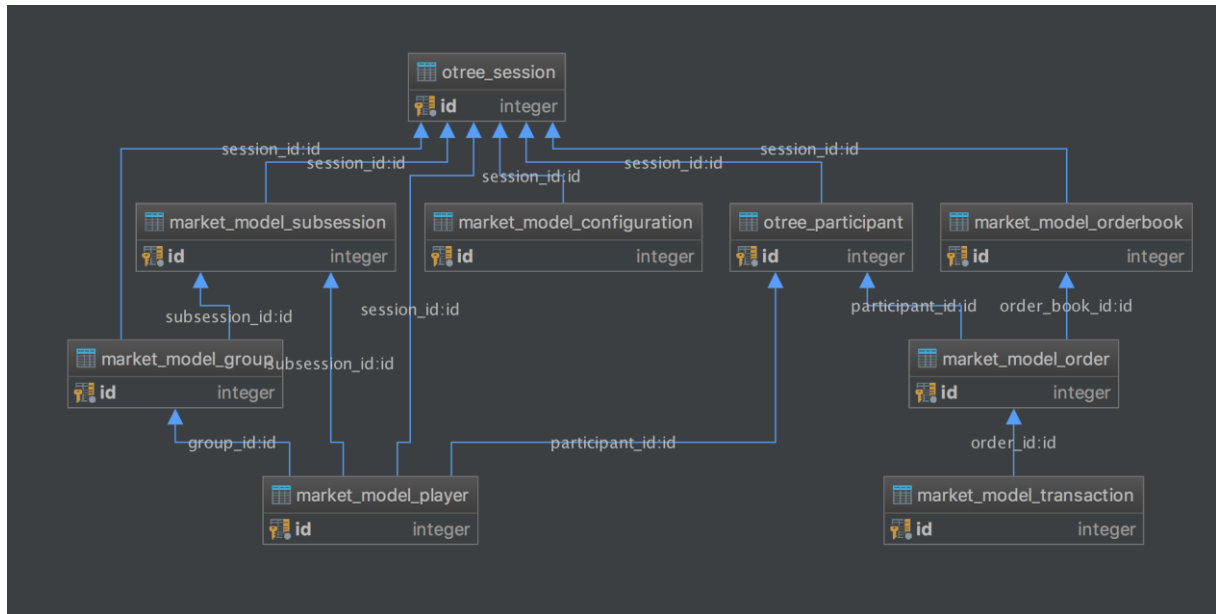


Figure 29 - Diagramme de classe complet.

```

FISH;11.9718;12.3027
SALAD;3.8000;4.4000
HOTDOG;9.8069;10.0524
LEMON;6.4000;7.3000
CHOCOLATE;19.7894;20.2097
ICECREAM;39.3000;40.7000
COOKIE;29.9458;30.1990
  
```

Figure 30 - Représentation du fichier "StateOrderBook".

Toutes ces données sont sauvegardées dans un fichier CSV qui porte le nom du code de la session. Ainsi, on peut avoir un fichier sous cette forme : « great352high.csv ». Concernant le formatage des données, voici comment elles y sont représentées :

NOM_DU_SYMBOLE ; **BEST_BID** ; **BEST_ASK**

Toutes les données sont sauvegardées dans la table « configuration » se trouvant dans la base de données.

Tableau 13 - Liste exhaustive des variables créées pour tester les hypothèses.

num_ords	total_size	mean_size	mean_price_bid	mean_price_ask	qt_market_bid	qt_market_ask	qt_limit_bid	qt_limit_ask	mean_elapsed_time	has_been_primed
17	81	4.76	1.0	0.76	6	3	2	6	30	True
6	44	7.33	0.89	nan	0	0	6	0	90	True
9	83	9.22	1.0	0.85	0	0	6	3	39	True
14	170	12.14	nan	1.0	6	7	0	1	38	True
8	61	7.62	1.0	1.0	3	0	3	2	64	True
11	45	4.09	nan	nan	7	4	0	0	41	True
20	91	4.78	0.92	0.83	4	0	5	10	27	True
10	47	5.22	1.0	1.0	4	0	1	4	54	True
27	165	6.11	1.0	0.76	11	1	2	13	20	True
13	58	5.8	1.0	0.80	4	0	1	5	43	True
15	60	4.0	0.95	0.90	3	1	7	4	35	True
5	20	5.0	1.0	nan	2	1	1	0	101	True
7	71	10.14	0.92	1.0	0	0	5	2	51	True
2	15	7.5	1.0	nan	1	0	1	0	21	True
6	63	12.6	1.0	nan	0	0	5	0	80	True
5	18	3.6	1.0	1.0	0	0	4	1	105	True
8	50	8.33	0.76	nan	0	0	6	0	67	True
13	51	3.92	0.70	0.76	5	0	2	6	42	True
4	15	5.0	0.70	1.0	0	0	2	1	140	True
18	172	9.55	nan	0.83	8	0	0	10	29	False
21	118	6.21	nan	0.79	10	0	0	9	26	False
34	280	8.48	nan	0.45	12	0	0	21	15	False
14	93	7.15	nan	0.92	8	0	0	5	36	False
6	101	20.2	0.89	1.0	0	0	4	1	48	False
16	56	3.73	0.70	0.70	1	0	12	2	34	False

4	6	1.5	0.89	nan	0	0	4	0	71	False
15	30	2.5	1.0	0.94	4	0	2	6	31	False
3	21	7.0	1.0	nan	0	0	3	0	73	False
5	7	2.33	1.0	1.0	0	0	2	1	60	False
8	46	5.75	0.71	1.0	1	0	6	1	58	False
4	25	6.25	1.0	0.70	0	0	2	2	39	False
17	80	5.71	nan	1.0	7	4	0	3	31	False
11	60	7.5	1.0	1.0	2	1	4	1	45	False
13	123	11.18	0.90	1.0	4	0	4	3	42	False
7	50	7.14	0.94	nan	0	0	7	0	81	False
5	30	6.0	1.0	nan	1	0	4	0	78	False
5	38	9.5	1.0	nan	3	0	1	0	80	False
11	42	5.25	0.74	1.0	1	0	5	2	52	False

Les lignes colorées en rouge indiquent que ces données ont été effacées par le logiciel d'analyse des résultats.

Section 10.07 Lien de la vidéo pour expliquer les principaux concepts du marché expérimental

Pour expliquer les principaux concepts de ce marché expérimental, une vidéo a été réalisée. En voici le lien :

<https://www.youtube.com/watch?v=uxeu949-Dnw&t=3s>.