



SORBONNE UNIVERSITÉ
MASTER ANDROÏDE

Interaction Humain-Machine & Sciences cognitives

Étude sur la gratification différée - UE de projet M1

Éric ZHANG – Kate ZHANG

2022

Table des matières

1	Introduction	1
2	État de l’art	2
3	Contribution	3
3.1	Modélisation	3
3.1.1	Présentation de l’interface	3
3.1.2	Déroulement du jeu	4
3.1.3	Choix des paramètres	5
3.2	Implémentation	8
3.2.1	Réécriture	8
3.3	Résultats et analyse	9
3.3.1	Mise en place de l’expérience	9
3.3.2	Résultats théoriques	10
3.3.3	Résultats expérimentaux	11
4	Conclusion	15
4.1	Remerciements	15
A	Cahier des charges	18
B	Manuel utilisateur	22
B.1	Prérequis	22
B.2	Mise en place	22
B.3	Configuration et mesures	23
B.3.1	Paramètres	23
B.3.2	Données	23
B.4	Modifier le jeu	24
C	Carnet de bord	25

Chapitre 1

Introduction

Notre projet vise à répondre à la question "Pourquoi ne devenons nous pas des experts ?" sur le sujet des systèmes interactifs. Cette question est légitime dans une multitude de contextes. Pour une multitude de tâches qui sont quotidiennes pour beaucoup de personnes (traitement de texte, montage photo/vidéo), il existe des outils qui sont faits pour les experts (LaTeX, Adobe Photoshop). Ces applications sont conçues avec le but de gagner en efficacité à travers différents vecteurs. Par exemple à travers des raccourcis claviers [1] ou bien l'agencement des éléments de l'interface graphique[2, 3].

Malgré cela, il se trouve que les utilisateurs ne tirent pas avantage de ces améliorations qui font pourtant l'essence de l'efficacité de l'outil de l'outil[4]. Il s'agit d'une opposition inattendue entre l'utilisateur qui choisit un outil pour son efficacité tout en ignorant les fonctionnalités supplémentaires qui la proposent. À la place, les utilisateurs préfèrent se cantonner aux fonctionnalités offertes au premier abord par l'outil, ils décident de rester des Novices.

L'étude de ce paradoxe est au cœur de notre projet, et en particulier l'analyse de la barrière de l'apprentissage, le coût en temps posé par la tâche de la prise en main de ces fonctionnalités.

En somme, l'objectif est d'identifier l'impact de la barrière de l'apprentissage sur les décisions de l'utilisateur. Quels sont les facteurs qui motivent l'utilisateur à choisir une option au dessus d'un autre ? Est-ce une décision faite de manière rationnelle ?

En vue de cet objectif, la réalisation d'un dispositif permettant d'exposer et récolter des données provenant d'utilisateurs réels est nécessaire.

Le projet est encadré par Gilles BAILLY, chercheur du CNRS dans le laboratoire de l'ISIR (Institut des Systèmes Intelligents et de Robotique) spécialisé dans l'Interaction Humain-Machine.

Le [code source du site web](#) est accessible publiquement sur Github.

Le site web est actuellement hébergé par le laboratoire de l'ISIR à l'adresse :

<https://ihm-nodejs.isir.upmc.fr/>

Chapitre 2

État de l'art

Transition novice à expert

En interaction humain-machine, le mode Novice d'un système interactif est l'ensemble des interactions par défaut, souvent visuels prévues pour les utilisateurs débutants afin de faciliter la prise en main d'une interface, comme des menus déroulants ou des icônes. Le mode Expert est lui destiné à des utilisateurs avancés et peut se présenter par exemple sous la forme de raccourcis clavier ou des interactions gestuelles.

La transition de novice à expert est un problème complexe qui comprend les notions d'apprentissage, de prise de décision et de biais cognitifs. Des recherches [5] ont montré que même si les utilisateurs connaissent les avantages du mode expert, ils ne vont pas nécessairement faire l'effort de consacrer du temps à apprendre. Et même après avoir pris en main le mode Expert, les utilisateurs sont susceptibles d'oublier son utilisation.[6].

Des expériences[7] et hypothèses[8] ont déjà été proposées pour expliquer ce phénomène, et quelques solutions ont été proposées pour par exemple régler le paradoxe de l'utilisateur actif.[9] C'est à dire le fait que les utilisateurs favorisent la productivité à court terme au détriment de l'efficacité à long terme.

Gratification différée (Time Discounting)

La gratification différée est le concept de ne pas tirer avantage de bénéfices futurs en échange d'avantages présents mais moindre en comparaison. Par exemple une personne aurait tendance à préférer recevoir 10 Euros aujourd'hui plutôt que 11 Euros le jour ou l'année suivante.[10]. Ce concept a déjà été étudié en profondeur dans le domaine de l'économie [11-13] et de la psychologie[14] et intervient fréquemment dans la vie quotidienne, que ce soit pour la procrastination[15], ou pour d'autres choix du quotidien [16].

Le but de notre projet de d'appliquer le concept de gratification différée sur le domaine de l'interaction humain-machine plus particulièrement sur la transition novice-expert. Pour voir si nous pouvons expliquer l'irrationalité des utilisateurs par la gratification différée. Ces principes sont le plus souvent mis en évidence à travers une échelle plus large, de plusieurs jours ou plusieurs mois. Nous étudions ce phénomène à une échelle plus locale de quelques minutes.

Chapitre 3

Contribution

L'expérience étudiée est décrite dans ce papier[17] fourni par notre professeur encadrant.

3.1 Modélisation

3.1.1 Présentation de l'interface

Dans le but de pouvoir réaliser un système supportant une expérience distribuée à des utilisateurs facilement, la solution d'un site web a été retenue.

L'interface se décline en plusieurs éléments :

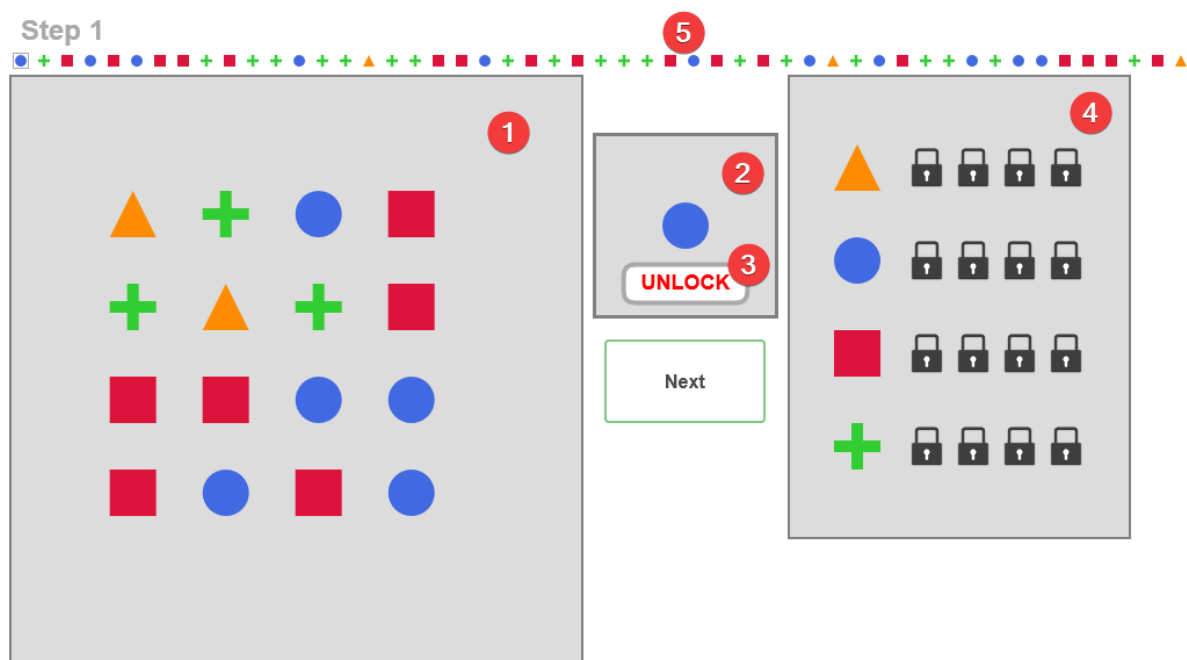


FIGURE 3.1 – Interface de jeu

(1) **Grille de jeu** Contient les formes à sélectionner parmi d'autres formes aléatoires

- (2) **Panel cible** Affiche la forme cible de l'essai courant
- (3) **Bouton déverrouillage** Un bouton qui une fois cliqué révèle l'interaction à effectuer pour déverrouiller un cadenas pour une forme
- (4) **Panel d'apprentissage** Montre le nombre de cadenas déverrouillé pour chaque forme ainsi que le nombre de cadenas restants
- (5) **Frise temporelle** Indique les formes prochaines

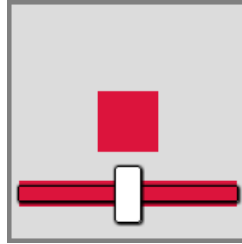


FIGURE 3.2 – Tâche de déblocage

Une fois les formes sélectionnées, l'utilisateur peut opter pour utiliser le mode Apprentissage (ici simulé par une petite tâche ayant une durée fixe pour l'essai) afin de pouvoir graduellement débloquer le mode Expert. L'utilisateur doit faire des aller-retours avec le curseur pendant un certain temps afin de pouvoir compléter la tâche.

3.1.2 Déroulement du jeu

Pour modéliser le passage du mode novice au mode expert, nous avons suivi les consignes du papier[17]. Nous avons donc trois modes :

- Le mode Novice : l'utilisateur sélectionne toutes les instances d'une forme cible et clique sur "Suivant" ("Next") pour passer à la prochaine grille. Ce mode représente les manipulations de l'utilisateur lorsqu'il souhaite faire une commande sur un système interactif.
- Le mode Expert : débloqué lorsque l'utilisateur a ouvert tous les verrous d'une forme, il n'a plus qu'à cliquer sur une seule instance et toutes les instances seront automatiquement sélectionnées. Il peut ensuite cliquer sur le bouton "Suivant" ("Next"). Ce mode correspond au moment où l'utilisateur est à l'aise avec les commandes expertes du système.
- Le mode Apprentissage : l'utilisateur sélectionne toutes les instances d'une forme cible puis clique sur un bouton "Déverrouiller" ("Unlock"), il doit ensuite agiter une barre de défilement pendant un certain temps T_{slider} et cliquer sur "Suivant" ("Next") pour ouvrir un verrou. Ce mode correspond aux manipulations qu'il fait lorsqu'il souhaite apprendre à utiliser les raccourcis du système interactif. Lui et le temps minimum passé dessus représentent la barrière de l'apprentissage.

Pour une bonne modélisation, il faut alors que le mode apprentissage soit plus long que le mode novice et le mode expert soit le plus rapide de tous les modes.

Le temps novice et le temps expert sont influencés par les paramètres de l'expérience. Le temps d'apprentissage, lui est l'aspect le plus important de l'expérience. Car on veut savoir si, avec un même temps d'apprentissage, comment le nombre de verrous va affecter le choix de l'utilisateur.

L'utilisateur se voit donc proposé d'abord une série de trois tutoriels, présentant les trois modes utilisables dans le jeu : Le mode Novice, le mode Expert et le mode Apprentissage.

Une fois les tutoriels passés, l'utilisateur commence l'expérience. Un chronomètre caché est lancé et l'utilisateur doit compléter le plus d'essais dans le temps imparti. Les essais sont générées à l'infini et l'utilisateur doit alors mettre en œuvre les mécaniques apprises dans les tutoriels.

En plus du nombre de verrous, nous ajoutons un autre paramètre, la fréquence des formes, qui va représenter la fréquence d'utilisation des commandes. Ainsi une commande peu utilisée sera modélisée par une fréquence plus faible. Ce paramètre va nous permettre de différencier les cas où l'utilisateur va décider d'apprendre.

Chaque forme ayant une fréquence, les essais sont générés par blocs pour garantir l'apparition de certaines formes à certains intervalles. Chaque bloc comporte un nombre d'essais égal à la fréquence de chaque forme. Ainsi pour quatre formes de fréquence respectives (1, 2, 3, 4), un bloc de 10 essais sera généré, chaque forme ayant un essai par unité de fréquence. Le bloc est ensuite mélangé et ajouté à une queue qui est défilée à chaque fois que l'utilisateur complète un essai.

Des pauses régulières sont proposées, et une fois le temps écoulé, les données sont enregistrées et l'utilisateur a la possibilité de donner son avis afin de potentiellement améliorer l'expérience.

3.1.3 Choix des paramètres

Nos paramètres sont donc : Le temps d'apprentissage total, les fréquences des différentes formes et le nombre de verrous pour chacune d'elles. Nous souhaitons mesurer le temps mis par l'utilisateur pour passer un essai, et quand il décide de passer au mode expert.

Nous avons donc choisi arbitrairement un temps d'apprentissage total d'une minute, c'est à dire que peu importe les autres paramètres, il faut exactement une minute pour débloquent le mode expert pour une forme.

Pour la fréquence des verrous, nous avons étudié plusieurs cas théoriques et avons choisi les fréquences 1, 4, 6 et 8. C'est à dire que pour chaque apparition de la forme de fréquence 1 il y aura 8 apparitions de la forme de fréquence 8.

Cela force l'utilisateur à évaluer l'avantage de débloquent une forme, en effet plus une forme est commune plus débloquent le mode Expert pour elle rapporte en gain de temps. À l'inverse, débloquent le mode Expert pour les formes plus rares peut s'avérer être une perte de temps.

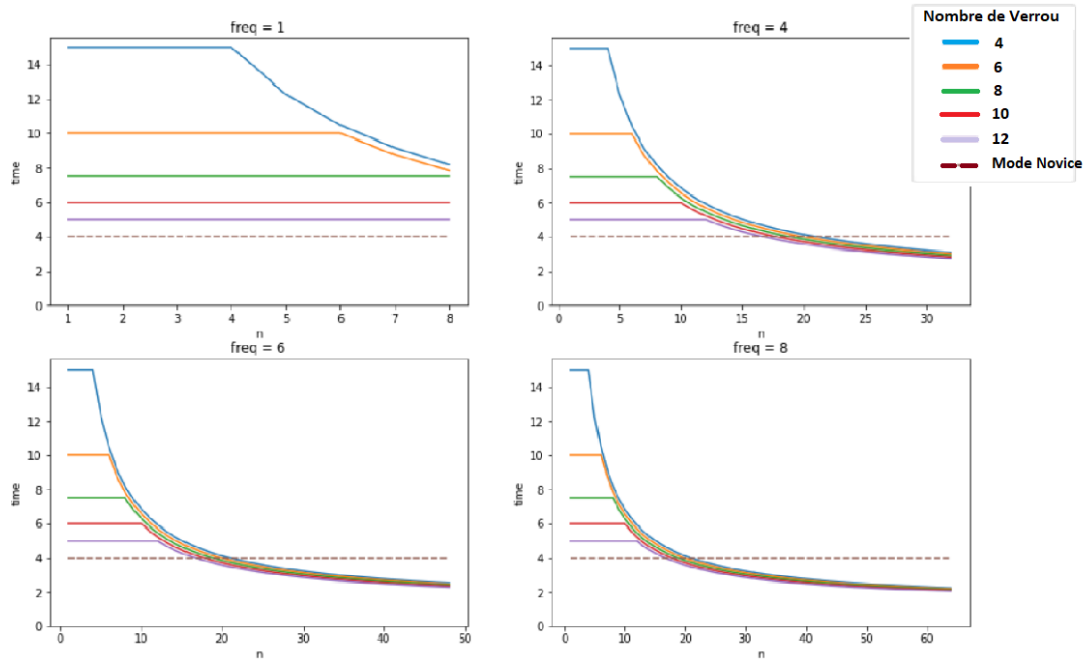


FIGURE 3.3 – Évolution théorique du temps moyen en fonction du nombre d'épreuve, si l'utilisateur n'utilise que le mode expert

Ici, il n'y a pas d'avantage à débloquer la forme de fréquence 1. Pour celle de fréquence 4, le résultat est un peu plus rapide si les utilisateurs déverrouillent mais c'est assez négligeable. Enfin pour les formes de fréquence 6 et 8, l'utilisateur gagne du temps à les débloquer.

Remarque : Si l'utilisateur est rapide, il pourrait potentiellement obtenir un gain en temps si il décide de débloquer la forme de fréquence 4, celle de fréquence 1 toutefois restera désavantageuse à débloquer.

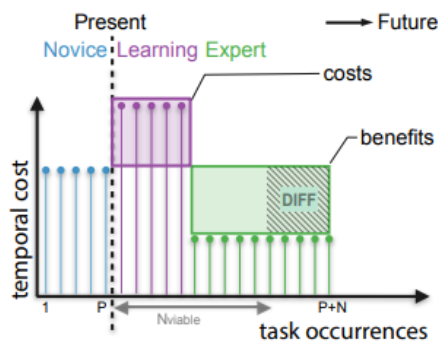


Figure 4. Illustrating a case where learning is the optimal strategy. At the *Present* time (P), the prospective total learning cost (purple box) is smaller than the *future* benefits that will gain with the expert interface (green box). The viability of transition towards expertise thus depends on the expected tasks N which must be more than N_{viable} .

FIGURE 3.4 – Figure dans le papier présentant un cas où il est avantageux de débloquer

Enfin pour le nombre de verrous, dans un premier temps nous les avons choisi arbitrairement.

Une condition d'arrêt était nécessaire pour l'expérience : limiter le nombre de essais maximum était préconisé tout en encourageant l'utilisateur à aller le plus vite possible. Ainsi, nous avons suivi ce qui était écrit et avons délimité l'expérience par un nombre d'essais maximal.

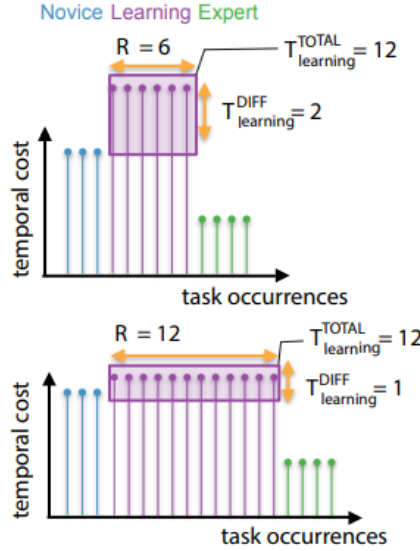


Figure 3. Two users' behaviors. Both users first adopted the novice strategy (blue), then the learning strategy (purple) and finally the expert strategy (green). The temporal cost of the learning strategy ($T_{learning}^{TOTAL}$) is the same for both users (purple box), but it differs in the distributions across the parameters number of repetition (N) and extra learning cost at each task ($T_{learning}^{DIFF}$).

FIGURE 3.5 – Hypothèse initiale sur la répartition du temps d'apprentissage

Mais on s'est rendu compte que cette condition ne convenait pas, car cela pénalisait les configurations avec un grand nombre de verrous. Car sachant que le temps d'apprentissage est identique, et que le nombre d'essais est constant, nous avons moins d'utilisations du mode expert dans les configurations avec un grand nombre de verrous que dans les configurations avec un petit nombre.

Nous avons donc décidé de changer la modalité de la condition d'arrêt, et de mettre un temps limite de 10 minutes à la place. Dans ce laps de temps, l'utilisateur doit compléter le plus d'essais possible.

Nous avons dû donc adapter le nombre de verrous R pour correspondre aux calculs suivants

$$T_{learningTotal} = T_{learningUnique} * R \text{ et}$$

$$T_{learningUnique} = T_{novice} + T_{diff}$$

Alors afin d'éviter d'avoir une tâche de déverrouillage à durée négative, nous avons choisi 4, 6, 8, 10 et 12 verrous.

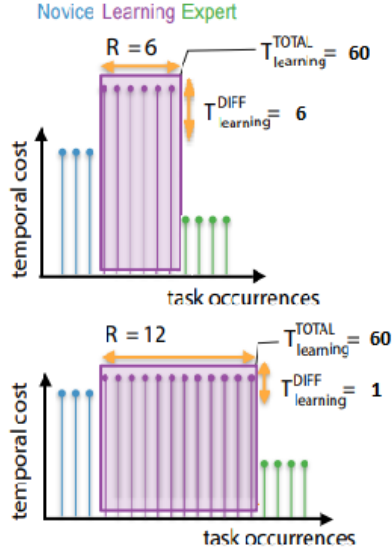


Figure 3. Two users' behaviors. Both users first adopted the novice strategy (blue), then the learning strategy (purple) and finally the expert strategy (green). The temporal cost of the learning strategy ($T_{learning}^{TOTAL}$) is the same for both users (purple box), but it differs in the distributions across the parameters number of repetition (N) and extra learning cost at each task ($T_{learning}^{DIFF}$).

FIGURE 3.6 – Graphe après le changement

Là, on obtient bien des configurations équitables, et il nous restait plus qu'à faire passer des utilisateurs réels et d'analyser les résultats.

3.2 Implémentation

Le site web est réalisé à l'aide de HTML pour donner la forme du site web, Javascript pour gérer les interactions et utilise des feuilles CSS pour contrôler le style de la page.

Toutefois ces technologies s'exécutent intégralement sur le côté client de l'utilisateur, c'est-à-dire son navigateur. Afin de pouvoir capturer et sauvegarder les métriques primordiales à notre recherche ainsi que d'assurer une distribution égale des modalités sur l'ensemble des utilisateurs, une technologie s'exécutant du côté serveur est nécessaire. Node.js est le moteur qui permet la gestion de ces éléments, permettant ainsi de maintenir un serveur capable d'enregistrer les données envoyées par le navigateur de l'utilisateur.

Le code source est initialement basé sur le travail réalisé par des étudiants de Sorbonne Université pour l'UE de Projet sur le même sujet.

3.2.1 Réécriture

Le code initial a été difficile à prendre en main, et il était compliqué pour nous d'identifier exactement où et comment les différentes étapes du jeu étaient mises en place. Dans le but de nous faciliter l'ajout de potentiels nouveaux composants ou la modification de la logique existante, nous avons décidé de réécrire la majorité du code existant.

Un des objectifs étant d'établir une structure plus succincte représentant où les fonctions du jeu et les composants individuels sont identifiables avec un fonctionnement plus in-

dépendant que précédemment. Cela nous a également permis de mieux nous familiariser avec l'architecture d'un site web dynamique et plus précisément avec le Javascript.

En particulier, le code initial utilisait un module Node.js nommé "nedb" afin de pouvoir gérer les paramètres du jeu et la sauvegarde des données utilisateur. Ce module n'est néanmoins plus maintenu depuis plus de 5 ans ([Dernière mise à jour le 15 Février 2016 sur npm, le gestionnaire de paquets de Node.js](#)). Cela signifie donc que le module n'a plus reçu de mises à jour de sécurité de ce moment là. Il se trouve qu'une faille de sécurité a été découverte et publiée le 15 Juin 2021 ([CVE-2021-23395](#)).

Cette vulnérabilité nommée **Prototype Pollution** permettrait de potentiellement modifier des propriétés de la classe générique **Object** de Javascript, causant ainsi un déni de service en redéfinissant par exemple la valeur retournée par **Object.prototype.toString** pour être un nombre entier, faisant ainsi planter le serveur. Cette faille peut aussi être escalader en exécution arbitraire de code dans le cas où un attribut était évalué. Une des dépendances de "nedb" nommée "underscore" est également vulnérable à des injections arbitraires de code ([CVE-2021-23358](#)).

Le module a donc été remplacé par un fichier CSV où les données sont écrites avec le module standard "fs" de Node.js.

Tout d'abord, le code portant sur les composants graphiques du jeu sont séparés des fonctions exécutant la logique du jeu. Les différents composants du jeu ainsi que leurs propriétés sont séparés. Chaque élément peut donc opérer indépendamment des autres. Le fichier principal contenant précédemment l'intégralité du code contient simplement l'initialisation et la liaison des composants.

Ainsi il est possible de centraliser la prise de statistiques à la portion du code gérant la logique du jeu.

3.3 Résultats et analyse

Les résultats que nous avons récolté sont sous format CSV et contiennent une variété de métriques comme :

- Nombre d'essais complétés
- Mode utilisé pour chaque essai
- Temps passé sur chaque essai

Nous avons ensuite utilisé Jupyter Notebook et des outils comme Pandas et Seaborn pour faire l'analyse. Nous avons écrit un Notebook où toute la visualisation est automatique des lors que l'on fournit la base de données

3.3.1 Mise en place de l'expérience

Avant de commencer l'expérience, nous avons dû estimer le temps que l'utilisateur va prendre dans chaque mode. Nous avons donc fait passer quelques utilisateurs sur le site avec pour but de mesurer le temps en mode novice, en mode apprentissage et en mode expert.

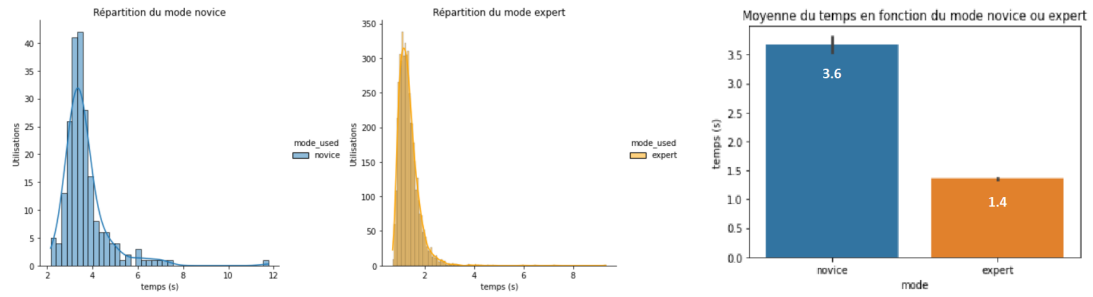


FIGURE 3.7 – Répartition du temps des utilisateurs dans les modes novice et expert

Nous avons donc mesuré (population jeune et dans le domaine de l’informatique) un temps moyen de 3,6s en mode novice et 1.4s en mode expert.

Ensuite il était nécessaire de vérifier que nos temps d’apprentissage correspondaient bien aux temps voulu.

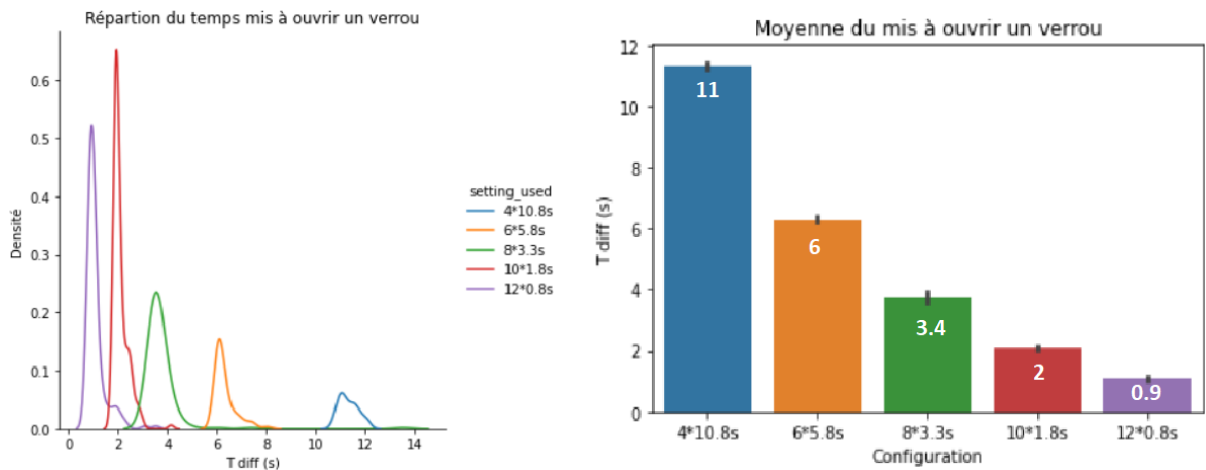


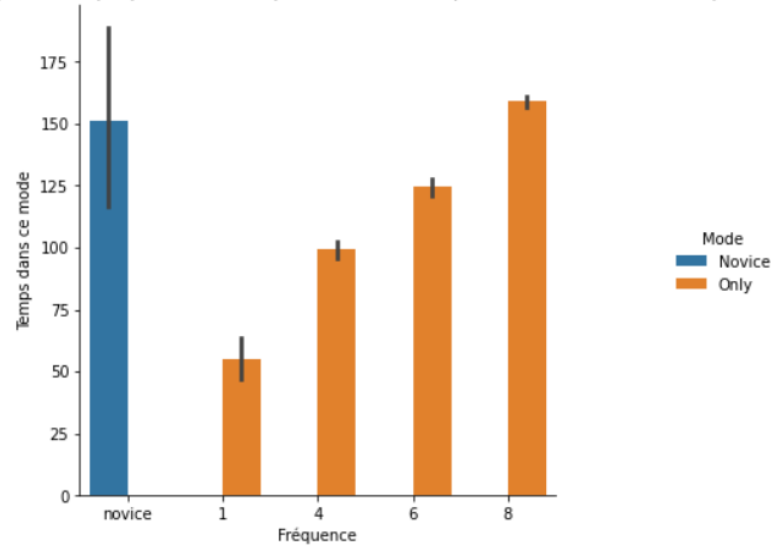
FIGURE 3.8 – Répartition du temps des utilisateurs dans le mode apprentissage

Nous obtenons des temps un peu au dessus de nos paramètres, nous avons alors réduit artificiellement de 0.1s la durée de la tâche de déverrouillage. Toutefois même avec cet ajustement, en fonction de l’utilisateur, la variance entre les utilisateurs peuvent aller jusqu’à 2 secondes.

3.3.2 Résultats théoriques

Notre hypothèse est que les utilisateurs ne sont pas rationnels et ne feront pas les choix optimaux pour aller le plus vite possible. Sous cette hypothèse, puisque le temps d’apprentissage est le même peu importe le nombre de verrous, le temps passé à compléter les essais sera alors le même pour chaque fréquence. Et donc voila le graphe théorique si les utilisateurs ne débloquent qu’une fréquence. Alors, si l’utilisateur ne débloquent qu’une seule des formes, le graphe du temps passé devrait avoir l’allure suivante :

Graphe théorique du temps que l'utilisateur passe sous une fréquence en fonction de la fréquence



Nous avons déterminé que pendant le temps imparti de 10 minutes, les utilisateurs sont en moyenne capable de faire 8 blocs en utilisant seulement le mode Novice, et 14 blocs en utilisant le mode Expert.

Donc si les utilisateurs sont rationnels, ils débloquent dès le départ les formes de fréquences 4, 6 et 8 et compléter environ 14 blocs, soit 280 essais.

3.3.3 Résultats expérimentaux

Un total de 20 utilisateurs ont participé à notre expérience, ils ont été réparti du mieux possible sur chacune des conditions.

Voici le graphe obtenu expérimentalement comparant pour chaque forme le temps passé dans chacun des modes. Il était attendu pour chaque fréquence un temps d'utilisation similaire. C'est le cas ici, mais toutefois on voit que toutes les mesures prises sont supérieures à celles obtenues par la théorie (Car les utilisateurs ne déverrouillent pas qu'un seul verrou), même pour la forme de fréquence 1 qui représente pourtant une perte de temps. On remarque également une grande variance sur les conditions avec des verrous plus long, une base d'utilisateur plus grande pourrait minimiser cet effet car chaque configuration a en moyenne 4 cas d'études.

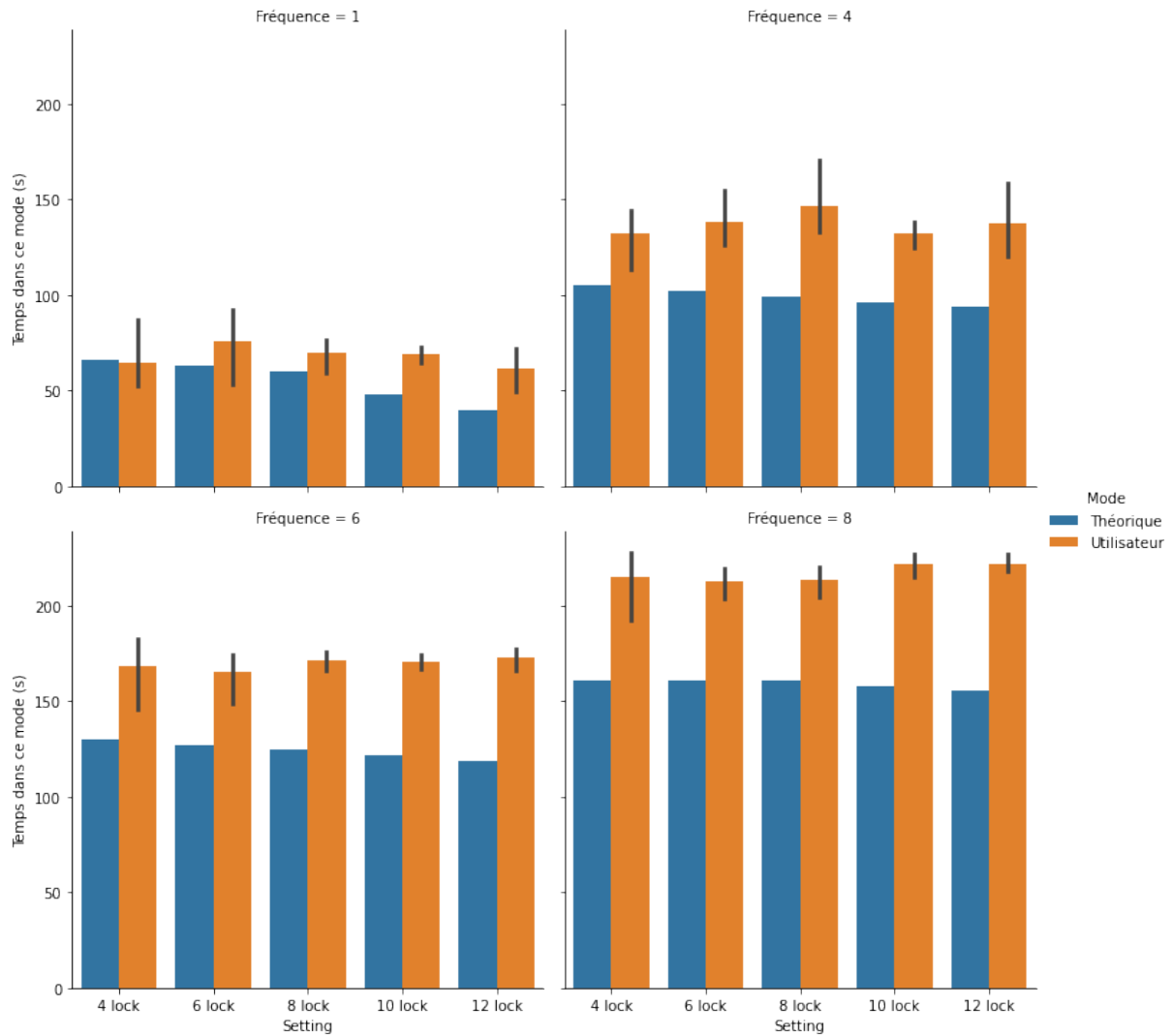


FIGURE 3.9 – Graphe du temps que l'utilisateur passe sous une fréquence en fonction du nombre de verrous

Sur cette figure portant sur le nombre de verrous ouvert à la fin de l'expérience, on observe que pour toute les configuration de verrous un schéma semblable pour les formes de fréquence 6 et 8, les verrous ont tous été ouverts. Pour 4, il y a eu des hésitations, certains l'on débloqué autres non. Par contre pour la fréquence 1, on a différentes situations : Certains utilisateurs décident de ne pas déverrouiller le mode Expert tandis que d'autres le font alors que cela ne leur est pas avantageux. Cela montre un comportement irrationnel de l'utilisateur.

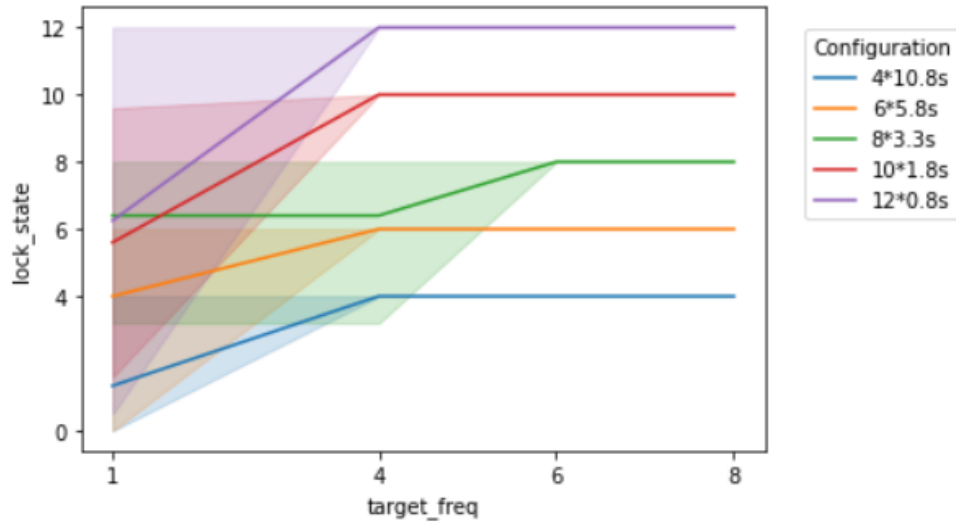


FIGURE 3.10 – Nombre de verrous ouvert à la fin de l'expérience en fonction de la fréquence de la forme

Sur cette figure, nous observons que les utilisateurs recherchent l'accès au mode Expert le plus rapidement possible. Donc le nombre de verrous déverrouillés augmente jusqu'à ce que le mode Expert soit débloqué. On voit donc que sur toutes les configurations des verrous les formes de fréquence 6 et 8 ont toujours été déverrouillé.

La plupart des utilisateurs ont décidé de déverrouiller le mode Expert pour la forme de fréquence 4, à gain nul ou moindre.

Toutefois, certains utilisateurs décident de débloquent le mode Expert pour la forme de fréquence 1 sans se rendre compte qu'il s'agit d'une perte de temps.

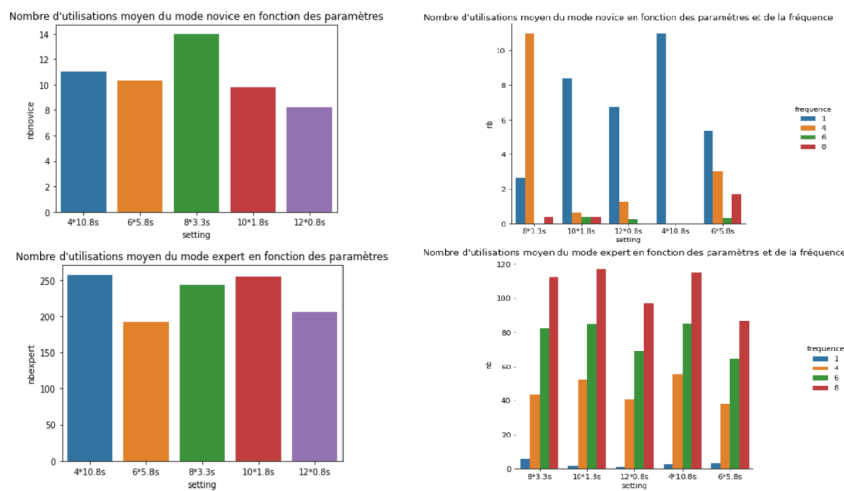


FIGURE 3.11 – Utilisation moyenne des modes novices et experts de nos utilisateurs

Enfin nous remarquons que certaines personnes commencé a déverrouiller pour une forme sans arriver au bout, s'arrêtant en cours de chemin ainsi que des utilisateurs qui entre deux ouvertures font une épreuve en mode novice.

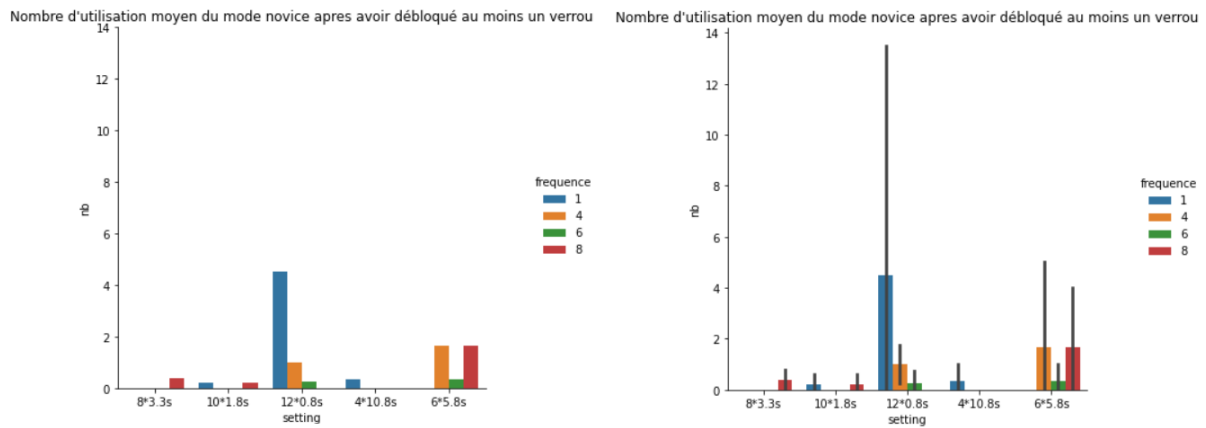


FIGURE 3.12 – Nombre d'utilisations du mode Novice après avoir ouvert au moins un verrou

En somme, nous pouvons souligner quelques comportements irrationnels des utilisateurs. Parfois oubliant de continuer à déverrouiller ou ne considérant pas la perte de temps à déverrouiller le mode Expert pour les formes à fréquence d'apparition faible.

Chapitre 4

Conclusion

Notre expérience a pu souligner des irrationalités auprès des utilisateurs. Parfois décidant de prendre des décisions allant à l'encontre de leur intérêt.

Une étude plus compréhensive aurait pu être possible avec significativement plus d'utilisateurs que nous avons. De plus, les données que nous avons mesuré ne s'alignent parfois pas exactement avec les mesures théoriques. Il a également été nécessaire de changer la manière dont l'expérience était réalisée et l'utilisation de technologies étrangères qui ont prouvés être des obstacles chronophages. Nous n'avons pas pu plus exploiter tous nos métriques et d'autres hypothèses sont formulables.

Néanmoins le prototype final est fonctionnel et modulable, il serait possible de mieux souligner la barrière de l'apprentissage avec plus d'utilisateurs et des paramètres plus précis.

4.1 Remerciements

Nous tenons à remercier spécialement notre encadrant Monsieur Gilles BAILLY pour son aide et sa grande implication durant tout le semestre pour ce projet, en particulier pour pour ses conseils qui nous a énormément servi dans nos moments de doutes.

Nous tenons aussi à remercier tous les utilisateurs d'avoir dédié leur temps à tester notre prototype et de nous permettre de collecter des données sur eux pour notre projet.

Bibliographie

- [1] Roger W REMINGTON, Ho Wang Holman YUEN et Harold PASHLER. « With practice, keyboard shortcuts become faster than menu selection : A crossover interaction. » In : *Journal of Experimental Psychology : Applied* 22.1 (2016), p. 95 (page 1).
- [2] Gilles BAILLY, Eric LECOLINET et Laurence NIGAY. « Visual menu techniques ». In : *ACM Computing Surveys (CSUR)* 49.4 (2016), p. 1-41 (page 1).
- [3] Antti OULASVIRTA, Per Ola KRISTENSSON, Xiaojun BI et Andrew HOWES. *Computational interaction*. Oxford University Press, 2018 (page 1).
- [4] Andy COCKBURN, Carl GUTWIN, Joey SCARR et Sylvain MALACRIA. « Supporting novice to expert transitions in user interfaces ». In : *ACM Computing Surveys (CSUR)* 47.2 (2014), p. 1-36 (page 1).
- [5] John M CARROLL et Mary Beth ROSSON. « 5 Paradox of the Active User1 ». In : () (page 2).
- [6] Benjamin LAFRENIERE, Carl GUTWIN et Andy COCKBURN. « Investigating the post-training persistence of expert interaction techniques ». In : *ACM Transactions on Computer-Human Interaction (TOCHI)* 24.4 (2017), p. 1-46 (page 2).
- [7] Ned AUGENBLICK et Matthew RABIN. « An experiment on time preference and misprediction in unpleasant tasks ». In : *Review of Economic Studies* 86.3 (2019), p. 941-975 (page 2).
- [8] David M LANE, H Albert NAPIER, S Camille PERES et Aniko SANDOR. « Hidden costs of graphical user interfaces : Failure to make the transition from menus and icon toolbars to keyboard shortcuts ». In : *International Journal of Human-Computer Interaction* 18.2 (2005), p. 133-144 (page 2).
- [9] Brian KRISLER et Richard ALTERMAN. « Training towards mastery : overcoming the active user paradox ». In : *Proceedings of the 5th Nordic conference on Human-computer interaction : building bridges*. 2008, p. 239-248 (page 2).
- [10] Shane FREDERICK, George LOEWENSTEIN et Ted O'DONOGHUE. « Time discounting and time preference : A critical review ». In : *Journal of economic literature* 40.2 (2002), p. 351-401 (page 2).
- [11] Gretchen B CHAPMAN. « Temporal discounting and utility for health and money. » In : *Journal of Experimental Psychology : Learning, Memory, and Cognition* 22.3 (1996), p. 771 (page 2).
- [12] Gretchen B CHAPMAN et Arthur S ELSTEIN. « Valuing the future : Temporal discounting of health and money ». In : *Medical decision making* 15.4 (1995), p. 373-386 (page 2).
- [13] Leonard GREEN, Joel MYERSON et Edward MCFADDEN. « Rate of temporal discounting decreases with amount of reward ». In : *Memory & cognition* 25.5 (1997), p. 715-723 (page 2).

- [14] George AINSLIE. « Specious reward : a behavioral theory of impulsiveness and impulse control. » In : *Psychological bulletin* 82.4 (1975), p. 463 (page 2).
- [15] Carolyn FISCHER. « Read this paper later : procrastination with time-consistent preferences ». In : *Journal of Economic Behavior & Organization* 46.3 (2001), p. 249-269 (page 2).
- [16] Forest BAKER, Matthew W JOHNSON et Warren K BICKEL. « Delay discounting in current and never-before cigarette smokers : similarities and differences across commodity, sign, and magnitude. » In : *Journal of abnormal psychology* 112.3 (2003), p. 382 (page 2).
- [17] Gilles BAILLY. « Investigating Time Discounting Bias in User Interfaces ». In : () (pages 3, 4).

Annexe A

Cahier des charges

Cahier des charges

Description du projet :

Nous étudions le problème de la gratification différée dans le domaine de l'informatique. Nous voulons trouver pourquoi les utilisateurs ne font pas l'effort d'apprendre les raccourcis ou des logiciels plus performant pour interagir avec un système interactif. Nous voulons savoir pourquoi ils ne prennent pas le temps d'apprendre pour être plus efficace. Pour cela nous allons mener une expérience et créer une interface web qui simule le fait d'apprendre, puis nous le ferons tester à l'utilisateur pour obtenir des résultats que nous analyserons.

Description de l'expérience:

Dans l'expérience, il y a des formes géométriques, le but, à chaque niveau, est de cliquer le plus rapidement possible sur toutes les formes attendues. Pour simuler l'apprentissage, nous permettons à l'utilisateur de libérer des verrous pour chaque forme. Dès que tous les verrous de la forme sont déverrouillés, l'utilisateur n'aura besoin que d'un seul clic sur la forme pour tous les sélectionner. Pour simuler le temps et l'effort d'apprentissage, nous mettons en place une barre qu'il faudrait agiter pour déverrouiller un seul verrou.

Définition des mots scientifiques dans la description

La gratification différée est une notion d'économie, c'est la capacité de résister à une tentation d'obtenir une récompense immédiate au profit d'une récompense plus tard . Par exemple, quelqu'un aurait tendance à préférer 10 euros aujourd'hui plutôt que 15 euros demain.

Un système interactif est une application informatique qui prend en compte, au cours de son exécution, des informations communiquées par le ou les utilisateurs du système, et qui produit, au cours de son exécution, une représentation perceptible de son état interne.

Historique du projet:

Nous avons repris le code, de nos camarades d'il y a 3 ans, qui est une application hébergée sur Glitch codée en langage JavaScript et Node Js.

Objectif envisagé:

Le but principal est de créer une plateforme simple et malléable pour pouvoir effectuer les expériences, ensuite nous ferons passer des utilisateurs pour visualiser et analyser leurs comportements.

Architecture générale:

- une API Web codé sous NodeJs
- avec pour entrée un fichier de configuration (taux d'apparence des formes, nombres de verrous, temps d'apprentissage, nombre de niveau)
- une interface pour les utilisateurs pour l'expérience
- une interface de tutoriel pour l'utilisateur
- une interface de configuration manuelle
- en sortie une page où l'on peut télécharger un fichier log des résultats

Tâches et organisation des tâches

Tâches principales

- Formater le code (fini d'ici fin février au plus tard)
 - Front end (html, css, design) : Kate
 - Back end (JavaScript, Node Js, fonctionnalités principales) : Eric
- Ajout de fonctionnalités (dans la foulée du codage)
 - L'utilisateur peut ou non avoir un pre-aperçu des formes qu'il va avoir à sélectionner (Savoir si cette fonctionnalité sera utilisée ou pas)
 - Une possibilité d'échec dans l'ouverture du cadenas (simule le fait de ne pas avoir compris la fonctionnalité)
 - Mettre une probabilité qu'un cadenas soit de nouveau verrouillé (oubli des raccourcis)
- Un cycle de l'expérience (si possible le faire 3 à 4 fois une seule fois prend 2 à 3 semaines maximum)
 - Définir des paramètres à évaluer
 - Faire tester aux utilisateurs dès que nous le pouvons avec si possible plusieurs groupes d'utilisateurs. (1 ou 2 semaine)
 - Visualisation et analyse des résultats (1 semaine)
 1. Traiter les données

2. Visualiser les données

- En fonction des résultats, modifier les différents paramètres de l'application (valeur du temps d'apprentissage, variantes)

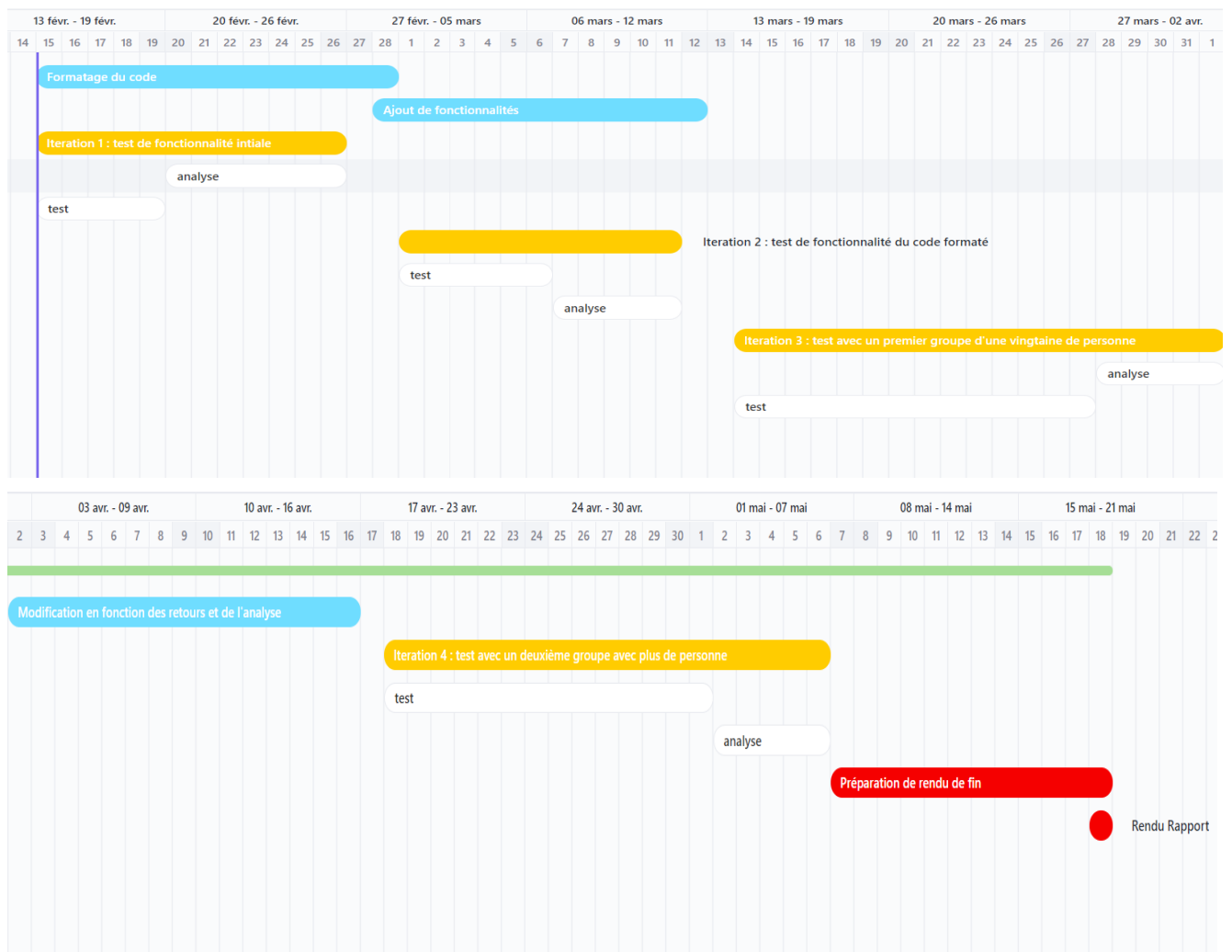
Tâches Optionnelle

faire des variantes qui dépendent des résultats

générer un rapport pour les participants

ajouter questionnaires et âges

Diagramme de Gantt (temporaire)



Annexe B

Manuel utilisateur

B.1 Prérequis

- Navigateur web (De préférence Mozilla Firefox, Google Chrome ou navigateur fondé sur Chromium)
- Git
- Node.js v16.14.2

B.2 Mise en place

Le code est disponible à l'adresse :

<https://github.com/KomeRice/temporal-discounting>

Une copie du dépôt Git peut être téléchargée en entrant la commande suivante dans un terminal :

```
git clone https://github.com/KomeRice/temporal-discounting.git
```

Installation de modules Node.js :

```
cd temporal-discounting/app
npm install
cd myexpress-init
npm install
```

Lancer le serveur Node.js (À exécuter depuis l'intérieur du dossier "app" à :

```
npm start
```

Une fois le serveur lancé, une version locale du site web devrait être accessible à l'adresse : <http://localhost/>

Les instructions pour prendre en main l'expérience sera affichée en tant que page d'accueil.

Alternativement, un serveur hébergeant le site se situe à l'adresse :

<https://ihm-nodejs.isir.upmc.fr/>

B.3 Configuration et mesures

B.3.1 Paramètres

Le fichier de configuration par défaut est situé dans "testSettings/testSettings.json". L'emplacement de ce fichier peut être changé dans "app/rsc/assets/js/main.js@36" - Variable `path`. Il s'agit d'un objet JSON (JavaScript Object Notation).

Les paramètres du jeu sont ainsi :

triWeight Fréquence du triangle (int)

cirWeight Fréquence du cercle (int)

squWeight Fréquence du carré (int)

croWeight Fréquence de la croix (int)

nbTargets Nombre de formes cible à générer (int)

timeLearning Temps d'apprentissage en ms (int)

nbSliders (Fragment de l'ancien code, inutilisé) (int)

nbLocks Liste des nombres de verrous possibles (list[int])

gridWidth Largeur de la grille en nombre de formes (int)

gridHeight Hauteur de la grille en nombre de formes (int)

shapeNames Nom des formes (Changement découragé) (list[string])

showTimeline Décide si la frise devrait être affichée (boolean)

easyMode Décide si un clic dans le rectangle contenant la forme est toléré (boolean)

maxStep Nombre d'essais maximum (-1 si infini) (int)

maxTimer Durée maximale de l'expérience en ms (-1 si infini) (int)

noviceTime Offset pour la définition de la durée du déverrouillage en ms (int)

breakTimer Délai avant le début d'une pause en ms (int)

debug Affiche des détails pour le debugging à l'écran (boolean)

B.3.2 Données

Les données sont sauvegardées au format CSV dans "rsc/data/gameData.csv". Chaque ligne correspond à une étape complétée par un utilisateur.

Les données sauvegardées sont ainsi :

date Date local du début de l'expérience

user_ip Adresse IP de l'utilisateur

trial_id Numéro de l'essai de la ligne

block_id Numéro du bloc de la ligne

n_trials Nombre d'essais complétés à la fin de l'expérience

n_block Nombre de blocs complétés à la fin de l'expérience
block_size Taille d'un bloc
target_shape Cible de l'essai courant
target_id ID de la forme cible
target_freq Fréquence de la forme cible
target_n Nombre de formes cible sur la grille
timeLearning Temps d'apprentissage en ms
setting_used Nombre de verrous * Durée d'un verrou individuel
n_locks Nombre de verrous par forme
lock_duration Durée d'un verrou
unlock_action 1 si un déverrouillage est fait pendant l'étape, 0 sinon
lock_state État du verrou lors de l'étape de la ligne
occurrence Nombre de fois que la forme a été la cible jusqu'à la ligne
time Temps mis pour compléter l'étape
time_selected Temps mis pour sélectionner toutes les formes
time_next Temps mis pour cliquer sur le bouton "Next"
slider_display_span Durée d'affichage du slider
n_opened_locker Nombre verrous débloqués jusqu'à la ligne
first_unlock_occurrence Indique à quel occurrence de la forme l'utilisateur a commencé le déverrouillage de la forme cible de la ligne.
first_unlock_trial Indique à quel essai de la forme l'utilisateur a commencé le déverrouillage de la forme cible de la ligne.
nb_total_click Indique le nombre de clics fait par l'utilisateur au cours de l'expérience
exp_total_time Indique le temps total consacré à l'expérience
mode_used Indique le mode utilisé dans l'étape de la ligne

B.4 Modifier le jeu

Les instructions Node.js se trouvent dans le fichier "app/index.js".

La logique du jeu peut être modifiée dans le fichier "app/rsc/assets/js/tdGame.js".

Les éléments graphiques sont dans les dossiers "app/rsc/assets/js/shapes, components.

Annexe C

Carnet de bord

M1 Informatique – UE Projet

Carnet de bord : les coulisses de la recherche documentaire

Les éléments que vous indiquez dans ce carnet donneront lieu à une notation

Noms, prénoms et spécialité :

ZHANG Kate ANDROÏDE

ZHANG Eric ANDROÏDE

Sujet :

IHM & Sciences cognitives

Introduction :

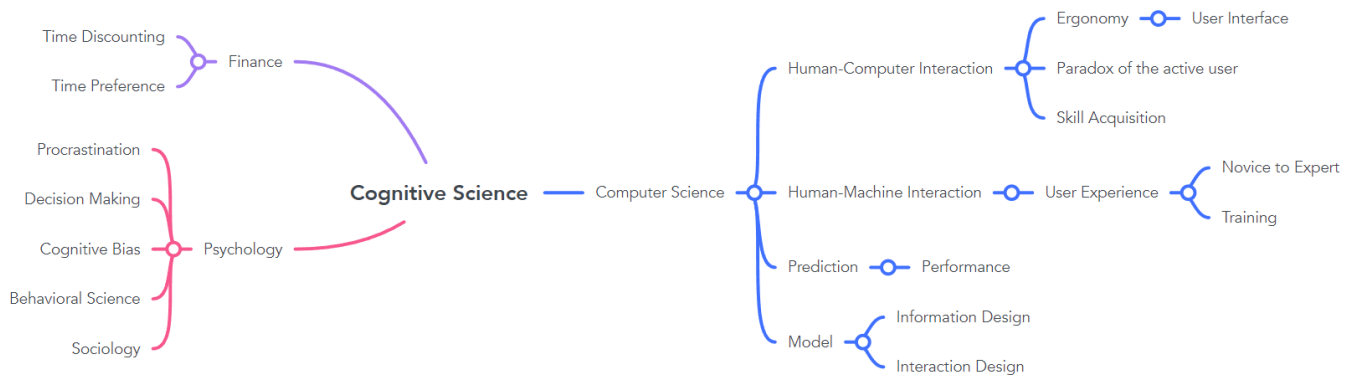
Notre projet vise à répondre à la question "Pourquoi ne devenons nous pas des experts ?". Cette question est légitime dans une multitude de contextes. Pour une multitude de tâches qui sont quotidiennes pour beaucoup de personnes (traitement de texte, montage photo/vidéo), il existe des outils qui sont faits pour les experts (LaTeX, Adobe Photoshop).

Ces applications sont conçues avec le but de gagner en efficacité à travers différents vecteurs. Par exemple à travers des raccourcis claviers ou bien l'agencement des éléments de l'interface graphique. Malgré cela, il se trouve que les utilisateurs ne tirent pas avantage de ces améliorations qui font pourtant partie des piliers de l'outil. Il s'agit d'une opposition inattendue entre l'utilisateur qui choisit un outil pour son efficacité tout en ignorant les fonctionnalités supplémentaires qui la proposent.

L'étude de ce paradoxe est au cœur de notre projet, et en particulier l'analyse de la barrière de l'apprentissage, le coût en temps posé par la tâche de la prise en main de ces fonctionnalités.

Mots Clés :

Computer Science, Human-Machine Interaction, Human-Computer Interaction, Ergonomics, User Interface, User Experience, Time Discounting, Finance, Psychology, Cognitive Science, Procrastination, Prediction, Decision Making, Model, Novice to Expert, Performance, Training, Paradox of the active user, Skill Acquisition, Cognitive Bias, Time Preference, Information Design, Interaction Design, Behavioral Science, Sociology



Descriptif de la recherche documentaire :

Dans un premier temps, nous avons cherché sur des moteurs de recherches et dans des journaux en ligne des articles sur l'état de la recherche pour notre sujet. Ainsi nous avons pu nous familiariser avec le comportement humain vis à vis de l'interaction avec des systèmes informatiques. Ensuite nous avons eu notre premier entretien avec notre professeur encadrant.

Le professeur encadrant nous a fourni un papier à lire, nous avons donc consulté les papiers présents dans sa bibliographie, ce qui nous a permis de commencer à former nos mots clés. Nous avons également recherché sur des bases de données comme ACM, Google Scholar ou encore Web of Science des articles correspondant à ces mots clés puis à partir de ces articles nous avons pensé à d'autres mots clés et ainsi nous avons réunis cette bibliographie. Nos sources sont toutes spécialisées et viennent de papier de recherche de grands journaux scientifiques (ACM, JEL, RES...).

Bibliographie :

1. George Ainslie. 1975. Specious reward: A behavioral theory of impulsiveness and impulse control. Psychological Bulletin 82, 4 (1975), 463–496. DOI:<http://dx.doi.org/10.1037/h0076860>
2. Ned Augenblick and Matthew Rabin. 2019. An Experiment on Time Preference and Misprediction in Unpleasant Tasks. The Review of Economic Studies 86, 3 (may 2019), 941–975. DOI:<http://dx.doi.org/10.1093/restud/rdy019>
3. Leif Azzopardi and Zuccon Guido. 2018. Economic Models of Interaction. In Computational Interaction, Antti Oulasvirta, Per Ola Kristensson, Xiaojun Bi, and Andrew Howes (Eds.). Vol. 1. Oxford University Press, Chapter 11. DOI:<http://dx.doi.org/10.1093/oso/9780198799603.001.0001>
4. Forest Baker, Matthew W. Johnson, and Warren K Bickel. 2003. Delay discounting in current and never-before cigarette smokers: Similarities and differences across commodity, sign, and magnitude. Journal of Abnormal Psychology 112, 3 (2003), 382–392. DOI:<http://dx.doi.org/10.1037/0021-843X.112.3.382>

5. Gretchen B. Chapman. 1996. Temporal discounting and utility for health and money. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 22, 3 (1996), 771–791. DOI: <http://dx.doi.org/10.1037/0278-7393.22.3.771>
6. Gretchen B. Chapman and Arthur S. Elstein. 1995a. Valuing the Future. *Medical Decision Making* 15, 4 (oct 1995), 373–386. DOI: <http://dx.doi.org/10.1177/0272989X9501500408>
7. Gretchen B Chapman and Arthur S Elstein. 1995b. Valuing the Future: Temporal Discounting of Health and Money. *Medical Decision Making* 15, 4 (oct 1995), 373–386. DOI: <http://dx.doi.org/10.1177/0272989X9501500408>
8. Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2014. Supporting Novice to Expert Transitions in User Interfaces. *Comput. Surveys* 47, 2 (nov 2014), 1–36. DOI: <http://dx.doi.org/10.1145/2659796>
9. Carolyn Fischer. 2001. Read this paper later: procrastination with time-consistent preferences. *Journal of Economic Behavior & Organization* 46, 3 (nov 2001), 249–269. DOI: [http://dx.doi.org/10.1016/S0167-2681\(01\)00160-3](http://dx.doi.org/10.1016/S0167-2681(01)00160-3)
10. Shane Frederick, George Loewenstein, and Ted O’donoghue. 2002. Time Discounting and Time Preference: A Critical Review. *Journal of Economic Literature* 40, 2 (jun 2002), 351–401. DOI: <http://dx.doi.org/10.1257/002205102320161311>
11. Leonard Green, Joel Myerson, and Edward Mcfadden. 1997. Rate of temporal discounting decreases with amount of reward. *Memory & Cognition* 25, 5 (sep 1997), 715–723. DOI: <http://dx.doi.org/10.3758/BF03211314>
12. Brian Krisler and Richard Alterman. 2008. Training towards mastery: Overcoming the Active User Paradox. In *Proceedings of the 5th Nordic conference on Human-computer interaction building bridges - NordiCHI '08*. ACM Press, New York, New York, USA, 239. DOI: <http://dx.doi.org/10.1145/1463160.1463186>
13. Benjamin Lafreniere, Carl Gutwin, and Andy Cockburn. 2017. Investigating the Post-Training Persistence of Expert Interaction Techniques. *ACM Transactions on Computer-Human Interaction* 24, 4 (aug 2017), 1–46. DOI: <http://dx.doi.org/10.1145/3119928>
14. Roger W. Remington, Ho Wang Holman Yuen, and Harold Pashler. 2016. With practice, keyboard shortcuts become faster than menu selection: A crossover interaction. *Journal of Experimental Psychology: Applied* 22, 1 (mar 2016), 95–106. DOI: <http://dx.doi.org/10.1037/xap0000069>
15. Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2016. Visual Menu Techniques. *ACM Comput. Surv.* 49, 4, Article 60 (Dec. 2016), 41 pages. DOI: <http://dx.doi.org/10.1145/3002171>
16. John M Carroll and Mary Beth Rosson. 1987. Paradox of the Active User. In *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. MIT Press, 80–111. <https://pdfs.semanticscholar.org/c011/4ca106ff22cf53aaed57d95adcb7ef306378.pdf>
17. David M Lane, H Albert Napier, S. Camille Peres, and Aniko Sandor. 2005. Hidden Costs of Graphical User Interfaces: Failure to Make the Transition from Menus and Icon Toolbars to Keyboard Shortcuts. *International Journal of Human-Computer Interaction* 18, 2 (may 2005), 133–144. DOI: http://dx.doi.org/10.1207/s15327590ijhc1802_1
18. Gilles Bailly, (unpublished) Investigating Time Discounting Bias in User Interfaces.

Evaluation des sources :

1. George Ainslie. 1975. Specious reward: A behavioral theory of impulsiveness and impulse control.

Date/Fraîcheur : 1975. Le papier date d'il y a peu de temps et il existe des versions plus récentes. Mais ce document ci représente la base de la théorie sur lequel repose notre projet et il nous a paru intéressant de l'ajouter dans nos sources.

Pertinence : L'information contenue dans l'article n'est pas du domaine informatique mais il aide beaucoup à comprendre comment les humains réagissent dans des situations variées. Pour notre projet, le but est justement de comprendre un comportement dans le cadre de l'interaction homme machine.

Provenance : Ce papier vient du journal scientifique Psychological Bulletin Journal, grand journal de psychologie et est ensuite référencé par de nombreux autres papiers. Nous avons trouvé dans les sources du document de notre professeur encadrant.

Son auteur George Ainslie est un psychiatre, psychologue et économiste du comportement qui a étudié et étudié toujours le sujet depuis 40 ans. Il a publié 12 articles dans sa carrière.

Rigueur du contenu : Le document a une structure conforme à celle d'un papier de recherche, il n'y a pas de fautes, les expériences sont bien décrites et sont reproduisibles. L'auteur a de l'expérience dans le milieu.

Objectif : Le papier a été financé en partie par le National Institute of Mental Health et a pour objectif l'étude scientifique du comportement humain.

8. Supporting Novice to Expert Transitions in User Interfaces.

Date/Fraîcheur: Ce document, publié en 2014, est récent et est fiable puisque cité 108 fois selon Google Scholar.

Pertinence : Notre expérience se base sur l'évaluation du comportement de l'utilisateur sur le fait de passer ou non au mode expert, donc cet article sur la transition débutant à expert nous a paru pertinent.

Provenance : Nous l'avons trouvé dans la base de données ACM. Le papier est objectif, bien référencé et très clair. Les auteurs sont des professeurs ou des chercheurs dans le domaine de l'interaction humain-machine et a été publié dans un journal de l'ACM, le journal qui a le plus d'influence dans le domaine de l'informatique.

Rigueur du contenu : Toutes les sources du papier sont vérifiées et vérifiables, il n'y a pas d'erreur, les expérimentations sont reproduisibles.

Objectif: Le papier est clairement en faveur du mode expert contre le mode novice et essaie de trouver pourquoi les utilisateurs favorisent quand même le mode novice.

10. Time Discounting and Time Preference: A Critical Review.

Date/Fraîcheur : Ce papier a été publié en 2002, il n'a pas été réfuté et plusieurs autre papier ont poursuivi et approfondi le travail sur le sujet.

Pertinence : Cette source est la fondation de notre recherche. Il définit et montre ce qu'est la gratification différée en tant que concept économique. Nous cherchons justement à présenter une vision informatique de ce dernier.

Provenance : Les auteurs sont des professeurs d'économie et de psychologie qui ont déjà publié plusieurs articles et le papier a été publié dans le *Journal of Economic Literature* qui est un journal réputé en économie.

Rigueur du contenu: Ce papier de recherche s'appuie sur de nombreuses références et est légitime. Il semble ne pas y avoir d'erreur.

Objectif : L'objectif de cet article est de définir la frontière entre la préférence temporelle et les autres facteurs pouvant influencer les choix intertemporels.