



UNIVERSITÉ PIERRE ET MARIE CURIE

MASTER 2

IMAGE ET SON POUR LES SYSTÈMES INTELLIGENTS

---

Conception d'un jeu sérieux qui s'adapte aux  
capacités cognitives de l'utilisateur

---

*Étudiant :*

AMINE BENAMARA

*Encadré par :*

BRUNO MASSONI SGUERRA

PIERRE JOUVELOT

*Responsable pédagogique :*

BRUNO GAS

30 août 2018



## Remerciements

Je voudrais tout d'abord remercier toutes les personnes autour de moi qui m'ont directement ou indirectement aidé durant mon stage, apporté leur soutien ou leur présence, et m'ont permis de faire de cette période une agréable expérience.

Je tiens à remercier toutes les personnes du CRI à Fontainebleau de l'école des M, qui m'ont accueilli toujours avec la bonne humeur au sein de leur équipe de chercheurs et étudiants passionnés par leur domaine.

Je remercie en particulier mes encadrants Bruno Massoni Sguerra et Pierre Jovelot pour m'avoir accompagné tout au long de mon stage, guidé lors de mes recherches bibliographiques et aidé à mieux réfléchir dans la recherche de solutions. Je les remercie également pour les séances de brainstorming toujours très enrichissantes qui ont vu naître une multitude d'idées, et qui finissaient souvent par trouver une ou plus de connexions à divers domaines, comme de la philosophie, de la physique quantique, de l'art ou de la thermodynamique.

Je remercie aussi toute l'équipe du CEN STIMCO, en particulier Samuel Benvenite, pour m'avoir accueilli au sein du living lab et pour avoir su mettre les mots sur mes idées. Ses conseils m'ont permis de prendre plus de recul pour toujours garder l'objectif principal en vue.

Un grand merci à l'ensemble de l'équipe du living lab de Broca, ainsi que tous les stagiaires que j'ai pu y rencontrer pour l'ambiance toujours agréable et les échanges sur les travaux de chacun, les déjeuners animés de fous rires et tous les moments qu'on a partagés ensemble.

Je dédie également un remerciement supplémentaire à Bruno Massoni Sguerra pour sa présence, son écoute, pour toutes les idées qu'on a échangées, pour les bons moments qu'on a passés, et pour notre laboratoire et son tableau à idées.

Enfin je remercie toute l'équipe pédagogique et administrative du Master SPI pour m'avoir accompagné et guidé tout au long de ma formation m'offrant l'opportunité d'effectuer ce stage.

# Table des matières

<b>1</b>	<b>Structures d'accueil</b>	<b>1</b>
1.1	Centre de Recherche en Informatique (CRI) de MINES ParisTech . . . .	1
1.2	LUSAGE : Living Lab . . . . .	1
1.3	CEN STIMCO . . . . .	2
<b>2</b>	<b>Contexte et état de l'art</b>	<b>3</b>
2.1	Projet LOUISE . . . . .	3
2.2	Mémoire de travail . . . . .	5
2.2.1	Définition . . . . .	5
2.2.2	Modèles existants . . . . .	6
2.2.3	Impact sur la maladie d'Alzheimer . . . . .	8
2.3	Interfaces Homme-Machine et modélisation cognitive . . . . .	9
<b>3</b>	<b>MATCHS : Memory Adaptation Through Cognitive Handling Simulation</b>	<b>11</b>
3.1	Framework . . . . .	11
3.1.1	Modélisation de l'oubli . . . . .	11
3.1.2	Système de contrôle . . . . .	14
3.1.3	Simulateur . . . . .	15
3.2	Limitations du modèle . . . . .	18
3.2.1	L'encodage et la mémoire de travail . . . . .	18
3.2.2	Mémoire de travail verbale . . . . .	20
3.3	Extensions du framework . . . . .	21
3.3.1	Une dégradation temporelle inéluctable . . . . .	21
3.3.2	Modélisation de l'encodage . . . . .	23
<b>4</b>	<b>Expériences</b>	<b>26</b>
4.1	Cas d'utilisation 1 : MATCH <sup>2</sup> S . . . . .	26
4.1.1	Description du jeu . . . . .	26

4.1.2	Protocole et campagne de test . . . . .	31
4.2	Cas d'utilisation 2 : Extension au cas séquentiel . . . . .	34
4.2.1	Simulation de l'encodage en séquence . . . . .	34
4.2.2	Jeu de mémoire de travail verbale . . . . .	35
<b>5</b>	<b>Résultats</b>	<b>37</b>
5.1	Cas d'utilisation 1 . . . . .	37
5.1.1	Simulateur de joueur virtuel . . . . .	37
5.1.2	Campagnes de test . . . . .	39
5.2	Cas d'utilisation 2 . . . . .	40
5.2.1	Encodage et dégradation en séquence . . . . .	40
5.2.2	Discussion sur le jeu de mémoire de travail verbale . . . . .	43
	<b>Bilan</b>	<b>45</b>

# Table des figures

2.1	Avatar de LOUISE . . . . .	3
2.2	Premier modèle de Baddeley & Hitch . . . . .	7
2.3	Evolution du modèle de Baddeley & Hitch . . . . .	7
2.4	Représentation du système de mémoire de Cowan . . . . .	8
3.1	Exemples d'exécution de la politique de maintenance . . . . .	13
3.2	Système de contrôle de MATCHS . . . . .	15
3.3	Courbes d'oubli $r(t)$ en fonction du nombre de simulations. . . . .	17
3.4	Courbe d'oubli . . . . .	18
3.5	Courbes de rappel pour des séquences de mots. . . . .	23
4.1	Menu principal du jeu . . . . .	27
4.2	Captures d'écran du tutoriel 1246 . . . . .	27
4.3	Menu principal du jeu . . . . .	28
4.4	Illustration d'un tour de jeu de Match <sup>2</sup> s . . . . .	28
4.5	Fenêtre de login . . . . .	30
4.6	Exemples de fichiers de données. . . . .	31
4.7	Interface de l'automate . . . . .	32
4.8	Interface du jeu de mémoire de travail verbale . . . . .	36
5.1	Résultat simulateur . . . . .	38
5.2	Résultat simulateur . . . . .	39
5.3	Résultat jeu : Evolution de l'erreur absolue moyenne pendant 6 itérations de jeu . . . . .	40
5.4	Simulation $v=0.5$ s/mot . . . . .	42
5.5	Simulation $v=1$ s/mot . . . . .	43

# Introduction

La dernière décennie a été témoin d'une émergence des technologies assistives, avec des outils et systèmes toujours plus performants et innovants. Cela peut s'expliquer par un vieillissement de la population, augmentant significativement le nombre de personnes de plus de 65 ans. Un besoin se fait ainsi ressentir dans le domaine de la santé, afin d'améliorer la qualité de vie des patients mais aussi réduire la charge de travail des soignants.

Il est toutefois difficile de remplacer l'interaction soignant-patient. En effet, sa connaissance du métier et des patients permet au soignant d'adapter instinctivement son discours et son comportement aux capacités cognitives du patient, et ainsi adapter sa manière de communiquer.

De ce fait, l'intérêt de la société à développer des technologies assistives a permis une plus grande volonté de la communauté scientifique à proposer des solutions et des systèmes capables de mesurer, grâce à des technologies comme la reconnaissance d'émotions, l'IRMf (imagerie par résonance magnétique fonctionnelle) ou l'eye tracker, les capacités cognitives d'une personne, pour pouvoir ensuite s'y adapter.

Le but de ce stage est de développer, en se basant sur les modèles et théories actuels, un système non invasif sous la forme d'un jeu, capable d'estimer la capacité cognitive de l'utilisateur pour y adapter son interface.

# Chapitre 1

## Structures d'accueil

Dans cette partie sont présentées les structures qui m'ont accueilli pour effectuer mon stage.

### 1.1 Centre de Recherche en Informatique (CRI) de MINES ParisTech

L'école MINES Paristech est une école d'ingénieur faisant partie des réseaux Paris-Tech et PSL. Ses domaines de recherche s'étendent de l'énergétique aux matériaux, en passant par les mathématiques appliquées, les géosciences et les sciences économiques et sociales.

Elle possède un centre de recherche en informatique situé à Fontainebleau, qui, selon son site web ([www.cri.mines-paristech.fr](http://www.cri.mines-paristech.fr)), se consacre à l'étude de langages utilisés par les technologies de l'information et développe des techniques d'analyse sémantique et de transformation automatiques destinées à répondre aux besoins industriels et aux besoins administratifs et sociétaux.

### 1.2 LUSAGE : Living Lab

Un Living Lab est un environnement reproduisant de façon réaliste le milieu de l'utilisateur final dans lequel sera amené à utiliser un produit ou un service. C'est dans une étape de conception ou de validation que sont menées des expériences qui consistent à ce qu'un utilisateur teste le produit dans cet environnement.

Le Living Lab est souvent utilisé dans le domaine de la santé publique afin de conce-



voir de nouvelles technologies visant à améliorer les soins ou aider les personnes malades.

Le laboratoire LUSAGE utilise l'approche Living Lab afin de développer et tester des technologies d'assistance aux personnes souffrant de troubles cognitifs. Plusieurs méthodes et technologies y sont développées et testées.

Ce laboratoire fait partie de l'hôpital Broca, un établissement de l'Assistance Publique - Hôpitaux de Paris (AP-HP) spécialisé en gériatrie. Il possède un hôpital de jour, où les soins de réadaptation et de longue durée sont effectués, et où se trouve une unité cognitivo-comportementale (UCC), qui permet le retour d'un patient souffrant de troubles du comportement et faisant preuve d'agressivité à son domicile, par un programme individualisé de réhabilitation cognitive. Des activités d'enseignement, rattachées à l'université Paris Descartes, principalement axées sur le vieillissement y sont également proposées. Enfin, des travaux de recherche portant sur la maladie d'Alzheimer ainsi que les technologies assistives s'y déroulent, entre autres par le laboratoire LUSAGE.

### 1.3 CEN STIMCO

Le Centre d'Expertise National en Stimulation Cognitive (CEN STIMCO) est une association de loi 1901 créée le 1er janvier 2011, financée par la Caisse Nationale de Solidarité pour l'Autonomie (CNSA), et présidée par le professeur Anne-Sophie Rigaud, psychiatre, gériatre et directrice du service de gériatrie de l'hôpital.

Les missions du CEN STIMCO s'articulent autour de trois axes :

- Information et actualité : État de l'art de la stimulation cognitive, information sur les méthodes et l'utilisation des nouvelles technologies.
- Expertise : Test et évaluation des nouveaux produits.
- Réseau : Mise en réseau et facilitation des échanges entre les différents acteurs de la stimulation cognitive.

Étant situé au sein de l'hôpital Broca, le CEN STIMCO travaille en étroite collaboration avec les professionnels de la santé et est également proche des patients.

# Chapitre 2

## Contexte et état de l'art

Cette partie consiste à présenter le projet dans lequel s'inscrit le stage, ainsi que les différentes notions et recherches qui permettent de se situer dans les travaux sur le sujet.

### 2.1 Projet LOUISE

LOUISE (LOvely User Interface for Servicing Elders) est un projet qui a pour but d'assister et d'interagir avec les personnes ayant des troubles cognitifs [23]. Le système se présente sous la forme d'un agent conversationnel animé (ACA), un avatar humain virtuel capable de communication verbale et non-verbale.



FIGURE 2.1 – Avatar de LOUISE

Le choix d'un ACA comme interface s'explique par la nature sociale de l'homme qui le rend particulièrement sensible et efficace à décoder les signaux sociaux [46]. Il est plus aisé pour des personnes âgées, pas forcément familières avec certaines technologies,

d'interagir avec un personnage imitant un comportement humain.

Une personne atteinte d'un déficit cognitif présente également souvent des troubles de l'attention [15], ce qui rend l'interaction plus compliquée et demande beaucoup de patience. Il faut souvent répéter plusieurs fois au patient l'information qu'on veut lui transmettre ou la question qu'on veut lui poser, lui faire maintenir son attention pour qu'il suive le fil de la conversation, et à cause de pertes de mémoire récurrentes, répondre plusieurs fois par jour à la même question [17]. Les ACA ne perdent pas patience et ne se fatiguent pas, contrairement aux aides soignants ou à la famille du patient, les rendant ainsi les partenaires idéals pour ces personnes.

De plus, du fait de leur déficit, les patients sont de moins en moins engagés dans des activités, ont beaucoup moins d'interactions avec d'autres personnes, et sont donc très peu stimulés, ce qui a pour effet d'accentuer leur maladie, en les plongeant dans l'ennui, la solitude et parfois la dépression. De ce fait, un ACA présente un gros avantage, car il est toujours disponible pour interagir, occuper et porter toute son attention au patient.

Afin de répondre à ces besoins, le système doit être capable d'interagir de manière naturelle avec un patient selon son trouble cognitif, et donc de s'adapter aux différents niveaux et types de maladie. Pour ce faire, LOUISE est doté d'un estimateur d'attention, basé sur la position de la tête de l'utilisateur, d'un système de reconnaissance de la parole et est capable de générer des animations naturelles (posture, expressions faciales et gestes) selon des scénarios prédéfinis. Grâce à ces outils, LOUISE est capable d'interagir avec un utilisateur assez naturellement, tout en recaptant l'attention du patient lorsque ce dernier est distrait.

Toutefois, la démence, qui est un terme désignant l'ensemble des maladies touchant les capacités cognitives d'une personne, est progressive, et les symptômes et déficiences s'aggravent avec le temps [26]. Il faut donc être capable d'adapter son discours en parlant plus lentement, d'utiliser des phrases plus courtes avec des mots simples et se répéter de plus en plus souvent selon le niveau de la pathologie, comme le font les soignants. Cela requiert une connaissance de l'état cognitif, qui peut être très variable d'un patient à l'autre. Ainsi, il est peu adapté de concevoir un système qui parlera très lentement en employant des phrases simples et courtes, car certains patients moins atteints trouveront cela ennuyeux et inintéressant.

Par conséquent, l'adaptation au niveau cognitif de l'utilisateur devient une question primordiale dans la conception d'un système destiné aux personnes présentant des troubles cognitifs, rendant l'interface plus flexible, intuitive et naturelle, améliorant

ainsi la qualité de vie des personnes touchées par ce type de maladie, celle de leurs proches, et celle des soignants.

## 2.2 Mémoire de travail

### 2.2.1 Définition

Dans les démences, une des parties de la cognition responsable des déficits des personnes atteintes s'appelle la mémoire de travail [5][21][19]. Il faut donc en comprendre les mécanismes afin de pouvoir déterminer la capacité cognitive et donc le niveau de maladie de ces personnes.

La mémoire de travail est la partie de la cognition responsable du stockage et de la manipulation d'informations à court terme. Elle est essentielle dans l'accomplissement de tâches cognitivement complexes, comme résoudre une équation, suivre une recette de cuisine, lire un livre ou encore suivre une conversation. Elle est également sollicitée dans l'apprentissage ou la réalisation de nouvelles activités, et est souvent associée à "l'intelligence fluide"[24], qui permet d'identifier les différences de capacités cognitives individuelles.

La mémoire de travail est souvent confondue avec la mémoire à court terme. Pour expliquer la différence entre les deux, on peut imaginer que la mémoire est une bibliothèque où sont rangés des livres qui contiennent les souvenirs et informations que l'on a à disposition. Lorsqu'on veut accéder à une information, on ouvre le livre qui la contient et on le pose sur notre bureau. C'est la mémoire à court terme.

La mémoire de travail, elle, peut être considérée comme la lampe de bureau qui éclaire les lignes et qui nous permet de porter notre attention sur la portion d'information spécifique que l'on recherche ou qu'on souhaite manipuler. En suivant cette métaphore, on peut expliquer la difficulté à exécuter deux tâches concurrentes qui requièrent toutes deux l'utilisation de la mémoire de travail.

Plusieurs études ont démontré les limites en capacité et en temps de la mémoire de travail [2][30][14], à l'aide de tests d'empan, où un individu doit se rappeler d'un maximum d'informations pendant un instant donné. Cette capacité pour une personne saine est estimée dans la littérature à  $7 \pm 2$  informations [30] et parfois à 4 [14], pendant une durée maximale de 20 secondes. Ceci peut s'expliquer par des stimuli de type différents dans les expériences (visuel ou verbal).

## 2.2.2 Modèles existants

La compréhension du fonctionnement de la mémoire de travail reste encore incomplète, et plusieurs modèles ont été établis pour tenter d'expliquer les mécanismes de cette partie de la cognition.

**Modèle de Baddeley & Hitch :** Le modèle de Baddeley & Hitch [4], qui est le modèle le plus répandu et le plus utilisé, est un modèle à composantes multiples. Il est composé de deux systèmes responsables de la rétention des informations, la boucle phonologique et le calepin visio-spatial, qui vont le transmettre à l'administrateur central qui s'occupera du traitement et de la manipulation de ces informations. Le modèle est donc divisé en trois modules :

- **La boucle phonologique (phonological loop) :** Responsable du stockage et du traitement d'informations verbales (phrases, mots, chiffres), elle est impliquée dans des activités telles que la compréhension orale, la lecture, l'écriture et le calcul mental. Elle comprend également un système de répétition, permettant de conserver l'information plus longtemps.
- **Le calepin visio-spatial (visuo-spatial sketchpad) :** Ses fonctions sont similaires à la boucle phonologique, mais pour les informations visuelles et spatiales. Il permet l'encodage d'une image mentale d'un objet.
- **L'administrateur central (central executive) :** C'est le module le plus complexe du modèle. Son rôle est de contrôler et de coordonner les systèmes esclaves. Il permet d'intégrer les informations envoyées par la boucle phonologique et le calepin visio-spatial, en agissant comme un contrôleur d'attention, afin de traiter les informations, de concentrer et diviser l'attention sur les éléments pertinents, et de mettre en relation les informations recueillies avec les connaissances contenues dans la mémoire à long terme.

Le modèle initial (Figure 2.2) a connu une modification majeure [6] (Figure 2.3), en ajoutant un quatrième module au système, le tampon épisodique (episodic buffer). Comme son nom l'indique, il permet de retenir des épisodes ou morceaux de mémoires, avec une représentation multidimensionnelle, intégrant les différents aspects des autres modules. Il sert d'interface entre la mémoire à long terme et les différents modules présentés, en regroupant les informations utiles pour créer une représentation de l'information par rapport au contexte. Il possède également une limitation en capacité, qui est estimé à 4, en accord avec les travaux de Cowan [14]. Avec ce nouveau module, les auteurs ont ajouté une connexion entre les systèmes dits fluides et cristallisés. Les systèmes fluides correspondent aux activations temporaires, comme le traitement d'une nouvelle information, tandis que les systèmes cristallisés correspondent aux savoir et connaissances ancrées, qui désignent la mémoire à long terme.

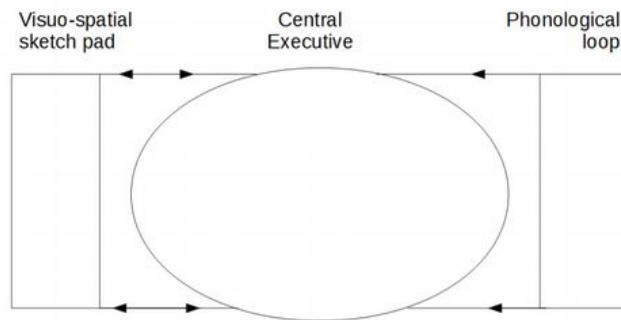


FIGURE 2.2 – Premier modèle de Baddeley & Hitch<sup>1</sup>.

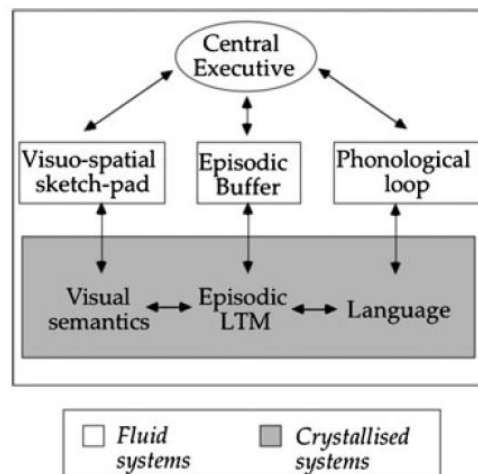


FIGURE 2.3 – Evolution du modèle de Baddeley & Hitch<sup>2</sup>.

**Modèle de Cowan [13]** : Cowan a proposé un autre type de modèle, un modèle unitaire. Il considère, contrairement à Baddeley, que la mémoire de travail ne s'appuie pas sur une représentation structurale, mais est uniquement le résultat de l'activation de certaines parties spécifiques de la mémoire. Pour lui, la mémoire de travail n'est pas

1. Source : A. D. Baddeley. Working Memory. *Science*, 255(5044) :556–559, 1995.

2. Source : A. D. Baddeley. The episodic buffer : A new component of working memory? *Trends in Cognitive Sciences*, 4(11) :417–423, 2000.

séparée de la mémoire à long terme, mais en serait une sous-partie.

Dans ce modèle, l'attention joue un rôle primordial, car elle permet l'activation plus ou moins importante de certaines informations. C'est ce que Cowan nomme le focus attentionnel. Ce focus attentionnel peut être contrôlé de manière automatique (réaction à un stimulus) ou volontaire, en portant son attention sur une information spécifique. Ce processus d'attention demande une quantité d'énergie cognitive selon l'information, plus coûteuse lorsqu'une information est nouvelle, car elle requiert plus d'attention, contrairement à une information connue.

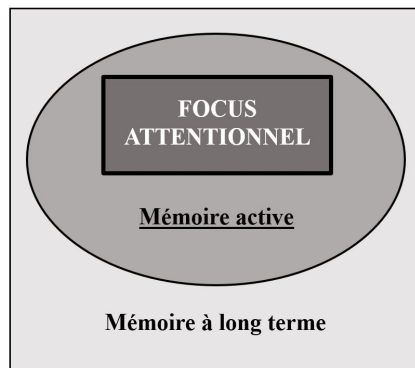


FIGURE 2.4 – Représentation du système de mémoire selon Cowan.

### 2.2.3 Impact sur la maladie d'Alzheimer

La maladie d'Alzheimer est la forme de démence la plus connue. C'est une pathologie neurodégénérative irréversible, au cours de laquelle les cellules du cerveau se détériorent progressivement. Elle apparaît en majorité après 65 ans et s'aggrave au fil des années. Le symptôme principal de la maladie est la difficulté de se rappeler d'informations nouvelles ou récemment apprises. Au cours de la maladie, d'autres symptômes vont apparaître, comme la difficulté à s'exprimer, la désorientation dans le temps et l'espace ou l'apraxie (difficulté ou incapacité à effectuer des gestes du quotidien).

Pour comprendre les implications de la mémoire de travail dans la maladie, plusieurs études ont été menées sur le sujet [5][21][19], et ont permis d'identifier les parties de la cognition impliquées dans la pathologie.

Aujourd'hui, il existe plusieurs tests permettant de diagnostiquer la maladie d'Alzheimer. Ils consistent en grande partie en une évaluation de la mémoire sémantique (connaissances générales sur soi et sur le monde), la mémoire épisodique et la mémoire

de travail. Dans [25], les auteurs démontrent que la détérioration de la mémoire sémantique entraîne des déficits de la mémoire de travail. Par ailleurs, la mémoire épisodique, étant un intermédiaire entre la mémoire de travail et la mémoire à long terme, se retrouve également impactée par le déficit de capacité de mémoire de travail. Ainsi, on peut voir que les déficits cognitifs dans la maladie d'Alzheimer sont fortement corrélés à des déficits de la mémoire de travail.

## 2.3 Interfaces Homme-Machine et modélisation cognitive

Réussir à concevoir des systèmes qui s'adaptent à l'utilisateur est depuis quelques années une préoccupation de beaucoup de chercheurs.

En effet, on trouve déjà de nombreux travaux concernant ce sujet, comme dans [3], où est présenté un système robotique pour la rééducation qui ajuste la difficulté d'une tâche à l'utilisateur. Dans [9], les auteurs proposent un système qui adapte l'interface d'un simulateur d'avion présentée à un pilote en fonction de son état émotionnel déterminé par un EMG (électromyogramme) pour mesurer la contraction des muscles du visage, et le rythme cardiaque pour estimer le niveau d'anxiété. Piloter un avion demande beaucoup de ressources cognitives et de concentration, et faire des erreurs peut être fatal. Il est donc d'un grand intérêt pour l'aviation de pouvoir adapter le contenu d'une interface aux ressources cognitives disponibles du pilote, évitant ainsi la surcharge cognitive et assurant que les tâches importantes puissent être accomplies.

On retrouve aussi des recherches se basant sur l'eye tracking pour inférer les capacités cognitives [39][35] ou la fatigue [7] de l'utilisateur. Le fMRI est aussi un des outils utilisés comme dans [44] où on mesure l'activité cérébrale d'une personne pour en déduire sa concentration et adapter la tâche proposée. Récemment, on retrouve aussi beaucoup d'études qui se basent sur des jeux pour en adapter leur difficulté au profil de l'utilisateur comme par exemple dans [8].

Un autre secteur particulièrement intéressé par l'adaptation d'interfaces aux capacités cognitives est celui de l'éducation. Tout comme la mémoire, les capacités d'apprentissage varient d'une personne à l'autre, et de plus en plus de travaux se penchent sur des moyens d'identifier les capacités cognitives des élèves pour pouvoir proposer un contenu adapté à leur style cognitif, à travers par exemple un support d'éducation en ligne [27] ou un jeu éducatif [33].



Dans un contexte plus proche du sujet de stage, les auteurs de [22] ont développé un jeu sérieux qui a pour but l'entraînement cognitif et l'estimation des performances cognitives d'un patient atteint d'Alzheimer afin d'adapter la difficulté du jeu, basé sur le NAT (Naturalistic Action Test), qui est un test neuropsychologique pour tester les capacités cognitives dans la vraie vie d'un patient.

# Chapitre 3

## MATCHS : Memory Adaptation Through Cognitive Handling Simulation

Cette partie sera consacrée à la description et l'explication du framework développé afin de modéliser l'oubli dans la mémoire de travail lors d'une tâche. Par la suite, les limites du modèles ainsi que les améliorations apportées seront exposées.

### 3.1 Framework

#### 3.1.1 Modélisation de l'oubli

Afin de modéliser l'oubli dans la mémoire de travail, le framework MATCHS s'appuie sur le modèle établi par J. Suchow [41] qui se base sur des MDP (Markov Decision Process).

Les MDP permettent de modéliser des prises de décision séquentielles dans un environnement partiellement stochastique, où le résultat dépend de l'action de l'agent décisionnel. On définit formellement une MDP par un ensemble  $S$  d'états entièrement observables, un ensemble  $A$  d'actions possible, une fonction de transition :

$$\tau : S \times A \rightarrow \mathcal{P}(S),$$

et une fonction de récompense  $\rho$ .

Le but d'une MDP est de fournir à l'agent décisionnel une politique optimale  $\pi : S \rightarrow A$  qui lie chaque état  $s$  à une action  $a = \pi(s)$  de sorte à optimiser (maximiser ou minimiser) la récompense cumulée de l'agent tout au long de la séquence d'états [34].

Dans son système, afin de décrire le phénomène de dégradation des informations dans la mémoire de travail, Suchow utilise un modèle évolutionnaire : le processus de Moran [31]. Il permet de décrire, pour une population finie de taille constante, la dynamique d'évolution des individus de types différents, en compétition pour la dominance. A chaque itération, un individu aléatoire est sélectionné pour la reproduction, et un autre pour la mort. Il est possible que le même individu soit sélectionné pour la reproduction et la mort, n'apportant ainsi aucune modification à la population. La sélection aléatoire est pondérée par des probabilités assignées aux individus selon leur valeur sélective, ou fitness. Ce modèle est souvent employé en biologie, utilisé par exemple pour décrire la dynamique des fréquences alléliques.

Les informations dans la mémoire de travail sont modélisées par une énergie cognitive, qui peut être divisée en quanta élémentaires. Chaque information disponible dans la mémoire se voit attribuée un certain nombre de quanta. Ce nombre est indicatif de la stabilité de la mémoire. Ainsi, plus il y a de quanta assignés à une information, plus elle a de chance d'être retenue dans la mémoire.

Pour en revenir au MDP, on définit un état de la mémoire  $s = [n_1, \dots, n_k] \in S$  comme étant  $K$  ensembles de cases mémoires  $b_i$ , contenant chacune  $n_i$  quanta qui encodent une information. Le nombre total de quanta  $Q$  attribué est donc tel que  $\sum_{i=1}^K n_i = Q$ . Le modèle utilise également un seuil de stabilité  $L$ , exprimé en nombre de quanta, à partir duquel une information est considérée comme étant perdue et irrécupérable. L'ensemble des distributions possibles des quanta entre les  $K$  cases mémoires définit ainsi l'espace des états  $S$  du MDP.

L'évolution du nombre de quanta dans chaque case mémoire permet de dire si une information est encore présente dans la mémoire de travail. C'est l'agent décisionnel qui choisit quelle information est conservée au cours de la dégradation, en suivant le mode opératoire du processus de Moran. À chaque itération de temps  $\delta_t$ , le mécanisme de maintenance désigne un quantum à dupliquer ou "reproduire" et un autre de manière aléatoire qui va "mourir". Choisir de dupliquer un quantum d'une case mémoire spécifique garantit que l'information qu'elle contient sera conservée dans la mémoire de travail. On se retrouve donc avec une politique de maintenance qui détermine une action  $a_i \in A$  comme étant un ajout d'un quantum dans une case mémoire  $b_i$  et la dégradation d'un quantum choisi aléatoirement parmi les  $K$  cases mémoire.

Par conséquent, il existe  $K$  actions possibles, ce qui correspond au nombre de cases mémoires stockées dans la mémoire de travail.

On peut voir en Figure 3.1 des exemples d'exécution de la politique de maintenance décrivant le processus de dégradation de la mémoire.

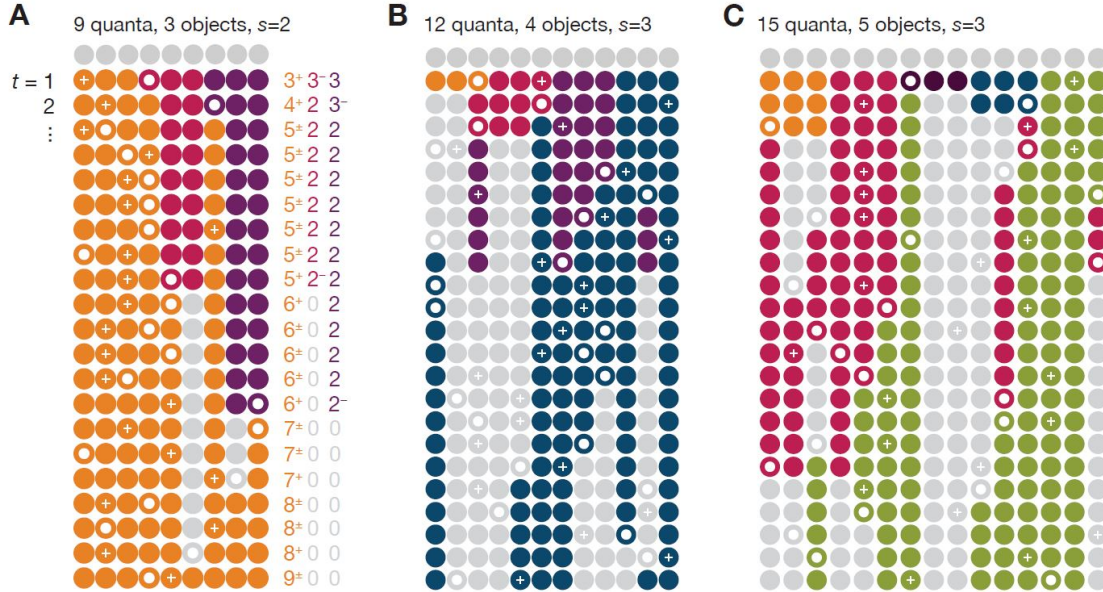


FIGURE 3.1 – Exemples d'exécution de la politique de maintenance<sup>3</sup>.

+ : Quantum sélectionné pour la reproduction  
 o : Quantum sélectionné pour la dégradation

Sachant que le quantum à éliminer est sélectionné de manière aléatoire et uniformément parmi tous les quantas, la probabilité qu'une case mémoire soit choisie pour être dégradée est :

$$P = n_i/Q.$$

La fonction de transition  $\tau(s, a)$ , qui décrit l'évolution dynamique de la population de quantas, est définie via l'ensemble des probabilités  $P(s'|s, a)$  d'aller d'un état  $s$  à l'état suivant  $s'$ , sachant l'action  $a$  de l'agent décisionnel.

De ce fait, si le système se trouve à l'état  $s = [n_1, n_2, \dots, n_k]$  et que l'agent décisionnel choisit l'action  $a_1$  qui duplique un quantum dans la première case mémoire, la probabilité de se trouver à l'état  $s' = [n_1 + 1, n_2 - 1, \dots, n_k] \in \tau(s, a_1)$  est :  $P(s'|s, a_1) = n_2/Q$ , qui est la probabilité de dégrader la deuxième information et de renforcer la première.

Pour ce qui est de la fonction de récompense, elle doit être définie selon la tâche à laquelle la mémoire de travail est confrontée. On rappelle qu'une information dans une case mémoire contenant moins de  $L$  quanta est considérée comme perdue. Cela implique

3. Jordan W Suchow. *Measuring, monitoring, and maintaining memories in a partially observable mind*. PhD thesis, 2014.

que si la tâche requiert de garder un maximum d'informations en mémoire, plus il y aura de cases mémoires contenant plus de  $L$  quanta, plus le résultat de la fonction  $\rho$ , donc la récompense, sera importante. Toutefois, si le but est de retenir au moins une information pendant une durée la plus longue possible, la récompense sera plus élevée lorsqu'au moins une information est encodée avec un maximum de quanta.

Tous les éléments de la MDP définis, il est maintenant possible de trouver une politique optimale permettant de maximiser la récompense accumulée. Etant donné le nombre important d'états possibles (combinaisons possibles de quantas entre les cases mémoires), les auteurs de [42] ont réduit le problème et analysé la politique optimale en essayant de lui trouver une forme utilisable dans un cas plus général. Les observations ont permis d'établir que cette politique peut être représentée suivant le principe psychologique du nom d'axiome de choix de Luce [28]. Il consiste à exprimer la probabilité de choisir une action parmi d'autres en fonction de la "force de perception" du stimulus. Le principe postule que lorsqu'une personne est confrontée à un choix parmi un certain nombre de possibilités, qui lui apportent des récompenses différentes, la probabilité de sélectionner une option est proportionnelle à la valeur de la récompense. La probabilité  $P(a)$  de choisir une action  $a$  est donnée par :

$$P(a) = \frac{v(a)^\sigma}{\sum_{x \in A} v(x)^\sigma},$$

où  $v(x)$  est la valeur de l'action  $x$  et  $\sigma$  la sensibilité de l'agent décisionnel pour l'action donnée. En faisant varier  $\sigma$ , le comportement du preneur de décision se trouve modifié. Par exemple, pour  $\sigma = 0$ , l'agent devient insensible aux signaux, rendant le choix neutre, ou encore pour  $\sigma = \infty$ , seul le choix qui offre la plus grande récompense sera choisi. On peut ainsi ajuster le comportement à différentes situations.

### 3.1.2 Système de contrôle

Le framework MATCHS permet d'adapter la quantité d'informations nécessaire afin de pouvoir accomplir la tâche en fonction des capacités cognitives de l'utilisateur. Pour cela, MATCHS est composé d'un système de contrôle (Figure 3.2) qui va ajuster cette quantité d'informations selon les résultats de la tâche effectuée par l'utilisateur.

L'élément principal de MATCHS est l'espace de paramètres de la mémoire (MPS). Il représente l'ensemble des paramètres qui définissent le comportement de la mémoire de travail de l'utilisateur. Selon le modèle de Suchow (décrit en 3.1.1), MPS est un espace 2D, avec pour dimensions  $Q$ , le nombre de quanta disponible, et  $\delta_t$ , l'intervalle de temps (en ms) entre chaque itération. Le seuil  $L$  est fixé de manière heuristique à 7 [41].

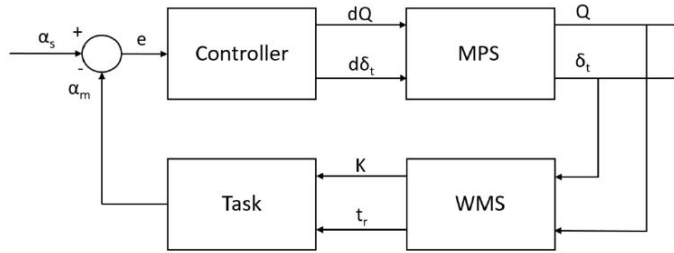


FIGURE 3.2 – Système de contrôle

Le framework permet de trouver, pour une tâche donnée, une estimation des paramètres de la mémoire de travail de l'utilisateur. Le but principal de MATCHS est donc, en fonction de la performance de l'utilisateur pour la tâche mesurée par l'erreur  $e$ , de mettre à jour l'estimation de ses paramètres en ajustant (incrémentant ou décrémentant)  $Q$  et  $\delta_t$  d'une quantité  $dQ$  et  $d\delta_t$  respectivement. On doit donc pouvoir être capable, pour une tâche, de mesurer la performance de l'utilisateur  $\alpha_m$ , qui correspond à la proportion d'informations oubliée.

Les valeurs estimées de  $Q$  et  $\delta_t$  sont ensuite injectées dans le simulateur de mémoire de travail (WMS), que l'on décrira dans la partie suivante. On récupère ainsi une mesure  $t_r$ , qui correspond au temps pendant lequel l'utilisateur est capable de retenir l'information avant d'en oublier  $\alpha_s\%$ . Cette valeur peut être utilisée pour adapter, en fonction de l'application, l'interface de la tâche. Il est également possible d'adapter l'interface en modifiant le nombre  $K$  d'éléments que l'utilisateur doit garder en mémoire. Par exemple, si la performance de l'utilisateur est inférieure à celle désirée, on peut considérer que le temps d'attente était trop long, et ainsi le diminuer, ou considérer que l'utilisateur n'a pas les ressources nécessaires pour conserver  $K$  éléments dans sa mémoire, et qu'il faut plutôt ajuster le nombre d'éléments affichés.

### 3.1.3 Simulateur

Le simulateur de mémoire de travail (SMT) utilise la politique du MDP décrit précédemment pour simuler l'oubli dans la mémoire de travail d'une personne. Il permet par la suite d'ajuster la tâche pour mieux correspondre aux capacités cognitives de l'utilisateur

Pour utiliser ce simulateur, il faut définir une tâche qui dépend fortement des capacités de mémoire de travail, afin de fournir un contexte pertinent. Dans notre cas, l'utilisa-

teur doit retenir autant d'informations que possible pendant un certain temps. Cela correspond à un cas particulier de l'axiome de choix de Luce, où  $\sigma = 1$ , et les valeurs des actions  $v(a)$  sont représentées par le nombre de quanta  $n_i$  dans chaque case mémoire  $b_i$ . En effet, avec  $\sigma = 1$  la probabilité de choisir une action dépend uniquement de la valeur de cette action. On fait donc bien le parallèle avec la politique du MDP, où la probabilité de sélection d'une case de mémoire dépend du nombre de quanta qu'elle contient.

Afin de simuler l'évolution de l'oubli dans la mémoire de travail, le SMT a besoin de 5 paramètres :

- $Q$  : Le nombre de quantas disponibles dans la mémoire de travail de l'utilisateur.
- $K$  : Le nombre d'éléments présentés lors de la tâche mémoire.
- $\delta_t$  : Le laps de temps entre chaque itération de la politique.
- $T$  : Le temps total de la simulation (en ms).
- $L$  : Le seuil de stabilité, qui détermine le nombre de quantas minimum qu'une case mémoire doit contenir avant de perdre l'information qu'elle contient.

Il faut également préciser au système l'état initial  $s_0$ , qui correspond aux cases mémoires de l'utilisateur après la présentation des stimuli. Dans notre cas, on considère que les  $Q$  quanta disponibles sont distribués uniformément entre les  $K$  cases mémoires, de sorte à ce que chaque case mémoire contienne au moins  $L$  quanta. Si le nombre total de quantas n'est pas suffisant pour remplir cette condition, on distribue aléatoirement  $L$  quanta à un maximum de cases mémoires, et le reste, s'il y en a, est attribué de manière aléatoire entre les cases restantes.

Durant le temps de la simulation  $T$ , la politique de l'agent décisionnel va désigner, à chaque itération  $\delta_t$ , une case mémoire dont le nombre de quanta va augmenter au détriment d'une autre sélectionnée aléatoirement. Le nombre de quanta peut également ne subir aucune modification si la case choisie est la même que celle désignée aléatoirement pour être dégradée. A la fin de ce temps, le ratio de cases mémoire contenant plus de  $L$  quanta sur le nombre total d'éléments  $K$  représente la probabilité qu'un élément donné soit encore en mémoire.

Après simulation, on obtient donc l'évolution pendant  $T$  de la probabilité de se rappeler des éléments. Étant donné la nature stochastique du processus, chaque simulation est différente, mais suit une évolution comparable. Pour obtenir une courbe la plus proche de la réalité, il est donc nécessaire de simuler plusieurs fois et de moyenner les résultats. On a donc effectué cette opération avec plusieurs valeurs du nombre de simulation pour sélectionner celle qui donne un compromis raisonnable entre régularité

de la courbe et temps d'exécution. On peut voir en Figure 3.3 un exemple de courbe d'oubli  $r(t)$  pour des nombres de simulations différents. Cette courbe nous permet de déterminer la probabilité de se rappeler des éléments après un certain temps compris entre 0 et  $T$ . D'après nos observations, nous avons décidé que 60 simulations étaient suffisantes pour obtenir un résultat correct.

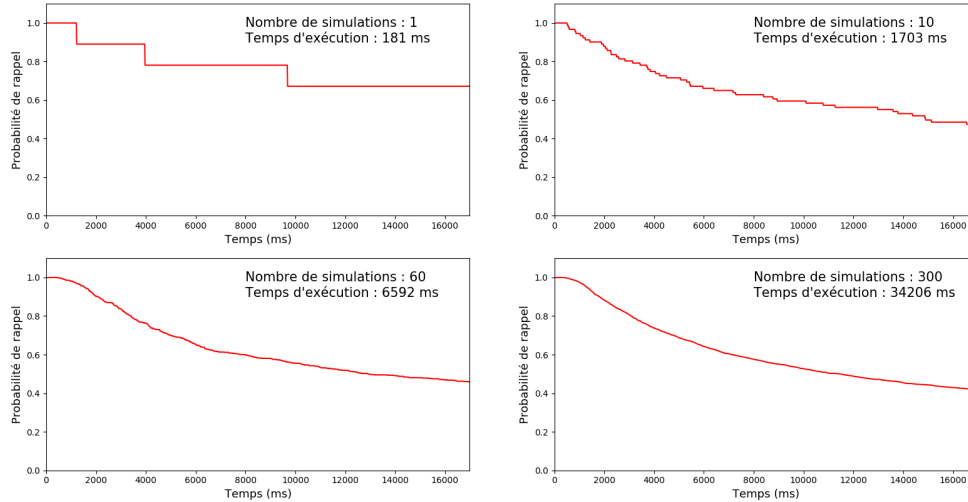


FIGURE 3.3 – Courbes d'oubli  $r(t)$  en fonction du nombre de simulations.  
Paramètres :  $Q=112$ ,  $K=8$ ,  $\delta_t = 13\text{ms}$ ,  $T=17\text{ s}$ ,  $L=7$

A l'aide de la courbe obtenue, on peut déterminer le temps de rétention  $t_r$  au bout duquel un pourcentage donné d'informations est encore en mémoire. Ainsi, si l'on souhaite qu'un maximum de  $\alpha_s\%$  (proportion d'informations oubliée) soit autorisé à être oublié par l'utilisateur, il faut trouver la valeur de  $t$  correspondant à  $(1-\alpha_s)$  sur la courbe  $r(t)$ .

Considérons la courbe pour 60 simulations de la Figure 3.3. Si l'on souhaite que l'utilisateur se rappelle de 60% des informations présentées, on doit trouver la valeur de  $t = t_r$  telle que  $r(t) = 1 - \alpha_s = 0.6$  (40% des informations oubliées et 60% d'informations encore en mémoire). Dans ce cas, le temps  $t_r$  sera d'environ 6000 ms. Si  $\alpha_s = 0.6$ ,  $t_r$  sera égal à 13000 ms.

Il est également possible que l'on ne puisse pas atteindre la précision souhaitée avec le temps  $T$  fourni. En gardant la même courbe en exemple, si l'on souhaite que



l'utilisateur ne se souvienne que de 20% des informations, on ne peut déterminer le temps nécessaire pour atteindre l'objectif. Toutefois, en modifiant le nombre d'éléments  $K$  présenté (ici en augmentant ce nombre), il est possible de trouver une valeur qui corresponde à la précision souhaitée (voir Figure 3.4). Le paramètre  $K$  est renvoyé en sortie du simulateur, et sert également à l'adaptation de l'interface.

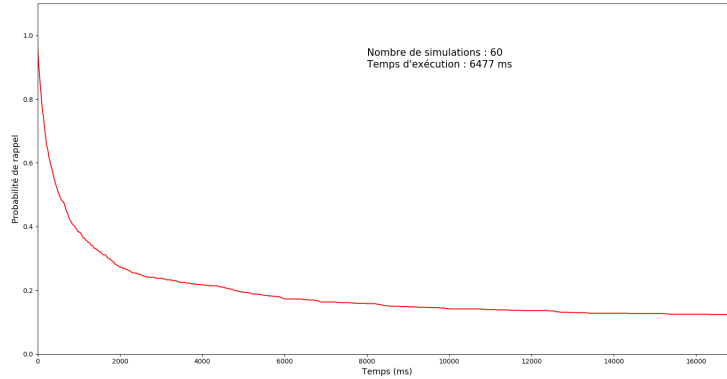


FIGURE 3.4 – Courbe d'oubli  $r(t)$   
 Paramètres :  $Q=112$ ,  $K=15$ ,  $\delta_t = 13\text{ms}$ ,  $T=17\text{ s}$ ,  $L=7$

## 3.2 Limitations du modèle

Le modèle que nous avons décrit permet de modéliser l'oubli dans la mémoire de travail lors d'une tâche, afin d'adapter cette dernière à l'utilisateur. Toutefois, l'utilisation de ce modèle se restreint à un certain type de tâche. On peut déjà remarquer que l'on se trouve dans un cas particulier de l'axiome de choix de Luce, imposant ainsi l'état initial de la distribution de quantas. De plus, le modèle de Suchow est basé sur la mémoire de travail visuelle, qu'on distingue de la mémoire de travail verbale, ce qui implique que le modèle utilisé n'est pas forcément adapté pour des informations verbales.

### 3.2.1 L'encodage et la mémoire de travail

Chaque stimulus que l'on perçoit, que ce soit une couleur, un son, un mot, ou des stimuli plus complexes encore, comme un morceau de musique, un tableau, ou un discours, va parcourir un long chemin avant d'être interprété et qu'on puisse lui donner

du sens. A chaque instant, des milliers de signaux de plusieurs natures entrent par nos différents capteurs sensoriels, et pourtant, seule une infime partie d'entre eux sont traités par notre cerveau, alors que tous les autres sont filtrés ou ignorés, sans même qu'on y pense.

Ce phénomène de tri des informations peut être interprété comme une optimisation d'énergie de notre cerveau. Par exemple, lorsqu'on est assis, il n'est pas forcément nécessaire de ressentir le contact de nos pieds avec nos chaussures ou de notre peau avec nos habits, et il serait inutile de regarder chaque détail de la rue que l'on traverse tous les matins. En effet, le cerveau est souvent décrit comme un système d'optimisation [20] ayant pour but d'utiliser un minimum d'énergie [10] [12] pour atteindre un but.

La première sélection se fait pendant l'encodage des signaux. Afin d'accomplir une tâche, on doit maintenir notre concentration sur le but, les instructions ainsi que les informations qui nous permettent de réaliser la tâche. Il faut également se focaliser sur les éléments qui nous semblent les plus pertinents pour y parvenir. Par exemple si le but est de retrouver un élément rouge parmi des formes colorées, il n'y a aucune raison de se concentrer sur la forme géométrique ; seule l'information de couleur est cruciale à encoder. La nature de la tâche a donc une influence majeure sur la manière d'encoder les informations.

De plus, le coût énergétique pour garder une information en mémoire est proportionnel à la complexité de cette information [18], et étant donné la capacité limitée de la mémoire de travail, l'information est souvent "compressée" ou simplifiée pour ne garder que les caractéristiques utiles à la tâche.

D'après les modèles de Baddeley et de Cowan décrits précédemment, l'attention ou focus attentionnel est en grande partie responsable de cette sélection. Il faut tout de même noter que l'attention est un terme très général, et d'après le modèle de Van Zomeren et Brouwer [45], on distingue trois formes majeures d'attention, à savoir :

- L'attention soutenue : C'est le fait de pouvoir rester concentré volontairement sur une tâche pendant une durée de temps plus ou moins longue.
- L'attention sélective : Cela correspond à la capacité de sélectionner les éléments les plus pertinents, et d'autre part ignorer les éléments distracteurs ou inutiles pour l'accomplissement de la tâche.
- L'attention partagée : Elle permet de distribuer notre concentration afin de traiter simultanément deux ou plus de types d'informations de natures différentes.

Ainsi, si une personne est inattentive lors de l'exécution d'une tâche, il y a de fortes chances qu'elle ne parvienne pas à la mener à bien. L'attention n'est par contre pas constante chez une personne, et peut varier pour diverses raisons, comme la fatigue ou

la motivation à effectuer la tâche [43].

De ce fait, lors d'une tâche de mémoire de travail, si une personne porte plus d'attention à un élément, les chances de s'en rappeler augmentent. Par conséquent, en reprenant le modèle de Suchow, l'allocation des quantas pour chaque élément, donc l'encodage de chaque information, dépend de l'attention qu'on porte à chacun d'entre eux.

Dans notre framework, on considère que les quantas sont équitablement répartis entre chaque case mémoire (si le nombre de quantas mis à disposition par l'utilisateur est suffisant). Cette hypothèse ne peut donc être vérifiée que dans certains cas de figure, où toutes les informations sont indépendantes (qu'on ne puisse pas regrouper), de complexités équivalentes, présentées en même temps pendant un même laps de temps, et où le choix est neutre. Ainsi, si une tâche est séquentielle, et en considérant que l'attention soutenue est limitée dans le temps, il sera difficile de porter la même attention à un nombre d'objets présentés à la suite. Le modèle ne peut donc pas s'appliquer directement sur ce type de tâche, à moins de connaître la répartition de quanta, ou l'attention portée à chaque élément.

### 3.2.2 Mémoire de travail verbale

Dans son modèle, Baddeley fait la distinction entre la boucle phonologique, responsable des informations verbales, et le calepin visio-spatial, des informations visuelles et spatiales. Plusieurs travaux ont démontré par des moyens différents, comme des tests de tâches concurrentes visuelles et verbales [11] ou par observation d'analyse PET (positron emission tomography) [38], que ces deux modules sont à distinguer. Toutefois, la capacité de mémoire de travail général reste affectée si un de ces modules se retrouve touché chez un patient.

La capacité de l'humain à communiquer avec ses pairs dépend de sa compréhension de son interlocuteur. Nous avons une tendance naturelle à adapter notre discours à la personne à qui on parle, en observant ses réactions lorsqu'on essaye de lui transmettre des informations, on par retour direct, si la personne dit qu'elle ne comprend pas.

La mémoire de travail est très sollicitée lors d'une conversation, car on doit retenir un minimum d'informations pour pouvoir suivre le fil de la discussion. On a vu que la mémoire de travail a une capacité de stockage assez limitée en temps et en quantité (4 pour Cowan, 7 pour Miller), et pourtant, il est possible de suivre une histoire et d'en comprendre le sens, même si bien plus de 7 éléments sont à retenir. Certains disent que c'est dû à la capacité de regrouper des informations élémentaires en morceaux (chunks). En linguistique, on explique cela par la manière de construire une phrase.

On peut considérer qu'une phrase, impliquant au moins un sujet, un verbe et un complément, est une chaîne dans laquelle chaque détail enrichi l'information. De plus, des associations faites par la mémoire à long terme ou la mémoire épisodique peuvent intervenir. Par exemple, une phrase comme : "le pompier a éteint le feu" peut être retenue comme une chaîne où le sujet, le pompier, exécute l'action d'éteindre un feu. Comme on associe le pompier au fait d'éteindre un feu, il sera plus aisé de retenir cette phrase que : "le pompier a réparé une voiture".

Ce phénomène peut une fois encore être vu comme une optimisation effectuée par le cerveau. En effet, comme dans une chaîne de Markov, à chaque nouvelle information qui arrive, on essaye de prédire l'information suivante. Par conséquent, des informations complètement nouvelles, des mots de vocabulaire inconnus ou peu utilisés, ou des associations de mots inattendues seront plus coûteux à traiter, et plus difficile à compresser. Ainsi, moins on a de surprise, plus on a de chance de se rappeler de cette information.

Par ailleurs, chaque individu possède ses propres stratégies d'encodage et une capacité de compréhension différente. Celles-ci sont établies tout au long des expériences sociales et à travers les connaissances personnelles de chacun, rendant l'encodage des informations encore plus difficile à prévoir. Malgré ces restrictions, il est possible d'effectuer une prédiction en récoltant des informations sur l'utilisateur qui permettront d'obtenir des indices sur ses connaissances générales et son milieu socio-culturel, même si cela reste une approche généraliste et demande un travail fastidieux.

La mémoire de travail verbale présente donc des particularités à prendre en compte lors de son évaluation. De par son aspect séquentiel et l'encodage par compression propre à chaque individu, on doit se limiter, dans les phases de test, à des cas d'utilisation où l'on ne fait pas appel à la mémoire épisodique ou long terme, tout en étant neutre et indépendant des individualités des utilisateurs.

### **3.3 Extensions du framework**

Nous avons pu voir dans la partie précédente les limites que présente le modèle utilisé pour l'évaluation des capacités cognitives. Nous avons donc exploré et analysé plusieurs théories et idées afin d'élargir le domaine d'utilisation du framework.

#### **3.3.1 Une dégradation temporelle inéluctable**

Dans son modèle, Suchow décrit la dégradation des éléments de mémoire comme étant concurrentielle. En effet, le processus de Moran utilisé pour décrire cette évolution

se résume à une succession de duplication et dégradation d'individus dans des populations, sélectionnés selon les probabilités définies par l'importance de chacune d'entre elles. Ainsi, une information aura plus de chances d'être conservée si elle est mieux fixée, c'est-à-dire si plus de quantas lui sont alloués.

Toutefois, si seul un élément est à retenir, peu importe le nombre de quantas alloués pour conserver cette information, elle sera à chaque itération sélectionnée pour être dupliquée et dégradée. Par conséquent, l'information est théoriquement inaltérable, à moins qu'une autre information n'entre en jeu, ce qui est en contradiction avec une des propriétés fondamentales de la mémoire de travail, qui est sa limitation en temps.

La problématique de la dégradation de la mémoire dans le temps a été souvent abordée, et initiée par Hermann Ebbinghaus [16]. Son étude ne traite pas l'oubli dans la mémoire de travail, mais à plus long terme. Il postule que l'oubli d'une information est inévitable, et que la vitesse de l'oubli dépend de la force de la mémoire, à quel point elle a été fixée, ainsi que la répétition, consciente ou inconsciente, de cette information. Il a formalisé ce concept à l'aide d'une formule mathématique :

$$R = e^{-\frac{t}{F}},$$

avec  $R$  la probabilité de se rappeler correctement de l'information,  $F$  la force de fixation de la mémoire, et  $t$  le temps.

Dans ses travaux, Ebbinghaus étudie l'oubli sur des périodes assez longues (jusqu'à plusieurs jours), en prenant en compte l'effet de répétition sur le renforcement de la mémoire. On peut tout de même remarquer que la même dynamique s'applique pour des durées différentes, plus ou moins longues selon la force de la mémoire, ce qui laisse penser que la dégradation suit une évolution similaire.

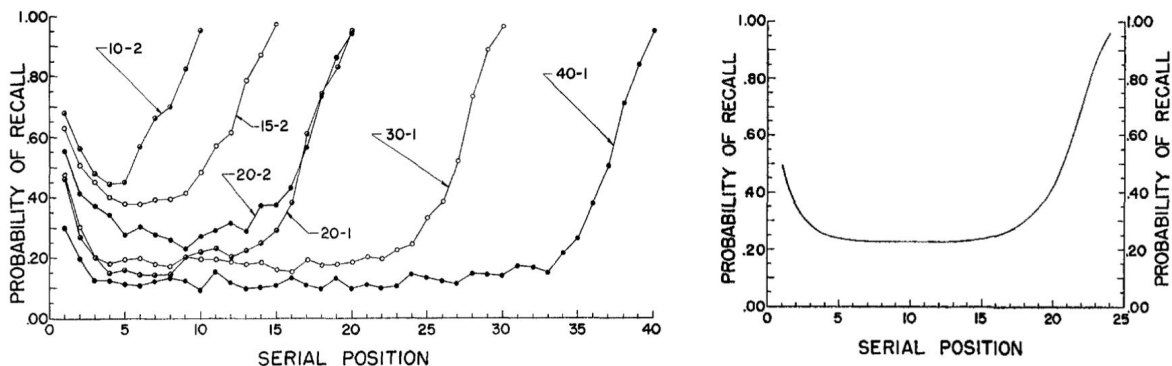
Pour faire le parallèle avec le modèle que l'on utilise, on peut dire que la force de fixation correspond à la quantité de quanta alloués pour stocker l'information. Les répétitions permettant de renforcer cette information en mémoire correspondent donc à un ajout de quanta destinés au stockage de cette même information. Pour finir, on postule que chaque information commence à se dégrader dès le moment où elle a été encodée, et donc que le nombre de quanta qui lui est alloué diminue, en suivant la formule d'Ebbinghaus.

Une autre hypothèse consiste à supposer que chaque mémoire, même seule, peut se dégrader, mais en suivant une évolution stochastique. En effet, si la dégradation entre les mémoires suit une évolution stochastique, il est plus cohérent de penser que la dégradation de chaque élément n'est pas totalement déterministe

### 3.3.2 Modélisation de l'encodage

Comme il a été décrit précédemment, l'encodage des stimuli est un procédé complexe et très variable. Des informations visuelles et verbales ne sont pas encodées de la même manière, et il en est de même pour des informations présentées en même temps ou de manière séquentielle. De plus, comme l'attention est en grande partie responsable de la sélection de l'importance des informations à traiter, la stratégie d'encodage dépend grandement du but de la tâche.

Comme on a pu le voir, notre cas d'utilisation se limite aux informations présentées simultanément, ce qui ne pose pas de problème pour l'encodage, que l'on considère équitable entre chaque élément. Nous avons donc choisi de modéliser l'encodage dans un cas séquentiel, plus complexe. Pour cela, nous nous sommes appuyés sur les travaux de Murdock [32], qui a soulevé le "serial-position effect", ou effet de position sérielle. D'après ses résultats, lorsque l'on présente à une personne une liste d'éléments qu'elle doit par la suite répéter, dans un ordre libre, cette dernière a tendance à mieux se rappeler des premiers et derniers éléments, formant ainsi une courbe en "U".



(a) Courbes de rappel pour différentes longueur de listes selon la position sérielle.

(b) Courbe idéale pour une liste de 24 mots.

FIGURE 3.5 – Courbes de rappel pour des séquences de mots<sup>4</sup>.

Nous avons donc récupéré les données de l'expérience réalisée par Roberts [36] similaire à celle réalisée par Murdock, mises à disposition sur [1], afin d'avoir une base correspondant à des tests de mémoire en séquence. Le procédé consiste à énoncer une

4. Source : Bennet B Murdock Jr. The serial position effect of free recall. *Journal of experimental psychology*, 64(5) :482, 1962.

liste de mots aléatoires avec un taux de présentation constant, puis après un temps défini, le sujet doit enfin se rappeler d'un maximum de mots de la liste, dans l'ordre qu'il choisit. Afin d'étudier l'impact de la longueur de liste et du taux de présentation, des listes de longueurs différentes (10,20,30 et 40 mots) ainsi que deux taux de présentation (0,5 sec/mot et 1 sec/mot) ont été utilisés. Les sujets ne connaissent pas à l'avance les modalités d'énonciation.

Chaque mot, ou élément, requiert un nombre de quanta pour ne pas être oublié. À chaque nouvel élément énoncé, on va allouer une quantité d'énergie cognitive pour la conserver. On rappelle que la quantité d'énergie cognitive est limitée pour une personne et donc le nombre d'élément possible à retenir aussi (en moyenne  $7 \pm 2$ ), d'où le choix du nombre de mots minimal dans les listes utilisées par Roberts. Puisque la courbe de rappel moyenne est toujours de la même forme, on peut considérer que la stratégie d'encodage de la plupart des personnes est similaire, et doit correspondre à la stratégie optimale afin de conserver un maximum d'éléments.

Sachant que les sujets ne connaissent pas le nombre de mots énoncés, ils ne peuvent pas allouer le même nombre de quanta à chaque information, à moins de se retrouver à cours d'énergie à distribuer avant la fin de la liste, et oublier les derniers éléments, ce qui ne correspond pas aux observations de Murock. De ce fait, on définit un coefficient variable  $r$ , qui désigne la proportion de quanta, parmi ceux disponibles, à allouer au stockage d'une nouvelle information.

Afin de déterminer les valeurs possible de  $r$ , on s'appuie sur trois paramètres (pour une personne moyenne) : le nombre maximal d'informations que l'on peut conserver dans la mémoire de travail, le nombre de quanta  $Q$  disponible, ainsi que le seuil  $L$  à partir duquel une information est perdue. On considère que l'allocation s'effectue par paquets de  $r_{paquet}$  quanta, avec  $r_{paquet} \geq L$ , limitant les valeurs possibles de  $r$  aux multiples de  $r_{paquet}$ . On postule également qu'une personne n'allouera pas toute son énergie cognitive disponible  $Q$  pour un élément, et qu'on peut définir le maximum de  $r$  tel que  $r \leq Q$ .

Tous ces éléments permettent une première modélisation du phénomène d'encodage. Tout comme la dégradation, on peut modéliser l'encodage par un processus de décision markovien (MDP), où l'ensemble d'état est également représenté par les différents états possibles de la mémoire. L'ensemble des actions est défini par les différents choix possibles de quantité de paquets  $p$  de quantas à distribuer pour l'information, ce qui revient à choisir une valeur de  $r$ .

On représente chaque état comme suit :  $s = [q, n_1, \dots, n_k]$ , avec  $q$  le nombre de quantas restants à distribuer, et  $n_i$  le nombre de quanta attribué pour chacune des  $K$  cases mémoires  $b_i$ . Le nombre total  $Q$  disponible pour la tâche est donc tel que  $Q = q + \sum_{i=1}^K n_i$ . Comme les informations sont en séquence, la taille de chaque état  $s$  est égale au nombre

d'éléments de la liste, et l'état initial est constant, et égal à :  $s_0 = [Q, 0, \dots, 0]$ .

Une fois encore, la fonction de récompense dépend du but de la tâche. Dans le cas des données de Roberts, le but est de retenir un maximum d'éléments à la fin de la séquence, on va donc maximiser la récompense lorsqu'un maximum de cases mémoires possèdent plus de  $L$  quanta. La récompense est donc proportionnelle au nombre de cases mémoires possédant plus de  $L$  quanta. On note par contre, que si l'on souhaitait retenir uniquement un élément, la récompense serait maximale tant qu'au moins un élément possède plus de  $L$  quantas, changeant totalement la fonction de récompense.



# Chapitre 4

## Expériences

Dans cette partie, l'ensemble des expériences et tests du framework et de ses extensions sont présentés.

### 4.1 Cas d'utilisation 1 : MATCH<sup>2</sup>S

Afin de tester la validité de notre framework, nous avons conçu un jeu basé sur l'expérience neuropsychologique utilisée par [40]. En utilisant le système de contrôle décrit en 3.1.2, la difficulté doit s'adapter au cours du jeu, pour correspondre aux capacités de mémoire de travail visuelle de l'utilisateur.

#### 4.1.1 Description du jeu

L'ensemble du jeu, le simulateur et le contrôleur ont été codés sous python v2.7. La bibliothèque graphique utilisée pour mettre en place l'interface est PySide v1.2.4. La manipulation et les calculs matriciels ont été réalisés à l'aide la bibliothèque numpy v1.14.3.

#### **Interface graphique et gameplay**

L'interface graphique est composée d'un menu principal qui permet d'accéder au tutoriel, aux réglages ainsi qu'au jeu en lui-même. Elle a été développée de sorte à être très simple à prendre en main.

Le tutoriel a été conçu afin de faciliter l'expérience utilisateur. Les différents éléments du jeu y sont présentés explicitement, et un tour de jeu y est décrit. Le tutoriel est essentiel car il faut s'assurer que le participant a complètement appréhendé un tour de

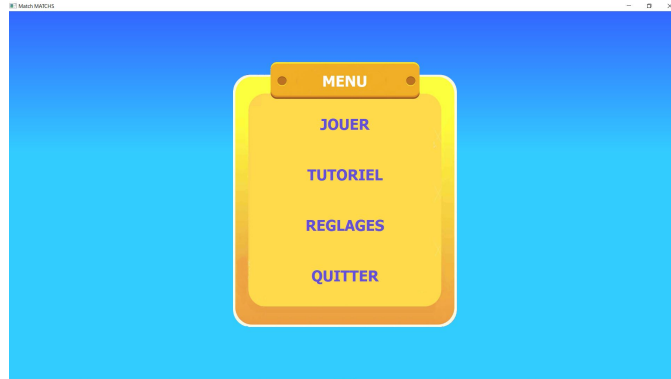


FIGURE 4.1 – Menu principal du jeu

jeu et que les consignes sont claires pour lui, afin d'éviter de biaiser les résultats. Ainsi, les ressources cognitives utilisées pendant le jeu sont réservées exclusivement à la tâche de mémoire.



FIGURE 4.2 – Captures d'écran du tutoriel

Le menu des réglages permet de modifier les paramètres du simulateur avant le début du jeu. Il permet également l'affichage en cours de jeu de l'évolution des différents

paramètres mis à jour par le système de contrôle. Il a été fait pour vérifier et faciliter les calibrages du système de contrôle au cours de l'établissement du protocole expérimental.

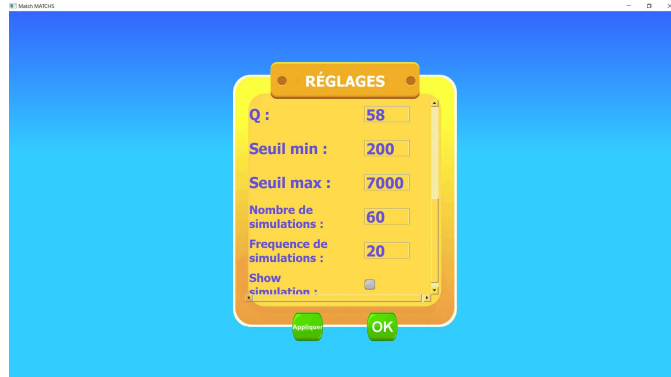


FIGURE 4.3 – Menu des réglages

Le jeu Match<sup>2s</sup> est un jeu de mémoire. A chaque tour,  $N$  carrés de couleurs différentes sont dévoilés pendant 500 ms, puis cachés. Après un temps d'attente  $t_{wait}$ , une fenêtre apparaît indiquant la couleur du carré à retrouver parmi ceux précédemment dévoilés. Un tour de jeu est illustré en Fig 4.4.

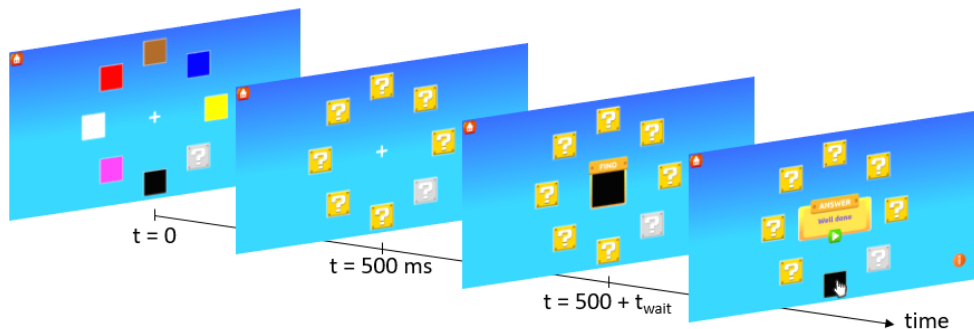


FIGURE 4.4 – Illustration d'un tour de jeu de Match<sup>2s</sup>

### Intégration du système de contrôle

Afin d'adapter la difficulté aux capacités cognitives de l'utilisateur, on utilise le système de contrôle décrit en 3.1.2. Le nombre de carrés  $N$  et le temps d'attente  $t_{wait}$  sont

les deux paramètres de l'interface à ajuster.

Le système de contrôle utilise un contrôleur proportionnel simple de gain  $G$ . Il permet de modifier dynamiquement l'estimation du nombre de quanta  $Q$  de l'utilisateur, en fonction de sa performance  $\alpha_m$ , d'une quantité  $dQ$ , telle que  $dQ = Ge$ , avec  $e$  le taux d'erreur. La performance souhaitée  $\alpha_s$  est à spécifier en début de partie, ce qui implique que MATCHS force l'utilisateur à atteindre cette performance en modifiant l'interface.

On choisit de fixer  $\delta_t$ , le temps entre chaque itération de la politique, limitant l'espace de paramètres aux valeurs de  $Q$ .

Afin d'avoir une estimation significative de la performance de l'utilisateur, on ne peut se contenter d'un seul tour de jeu. Il est nécessaire de la trouver en calculant une moyenne sur un certain nombre de tours. Plus le nombre de tours utilisé est grand, plus l'estimation de la performance sera précise. Par exemple si on utilise la moyenne sur 5 tours, les seules valeurs possibles pour la performance sont limitées à :  $[0,0.2,0.4,0.6,0.8,1]$ . Toutefois, pour que le joueur n'ait pas à jouer une centaine de tours avant de voir l'interface s'adapter, on décide de limiter cette valeur à un nombre de tour  $d$ , à estimer pendant les premiers tests du jeu.

Le simulateur, décrit en 3.1.3, prend en entrée la nouvelle valeur de  $Q$  ainsi que  $\delta_t$  pour déterminer la courbe d'oubli  $r(t)$  estimée de l'utilisateur. On note qu'il est possible, dans le jeu Match<sup>2</sup>s, que le joueur choisisse un carré au hasard, et que cette possibilité doit être prise en compte. Pour cela, on utilise la courbe d'oubli  $R(t)$ , qui est fonction de  $r(t)$ , telle que :

$$R(t) = \frac{(1 - r(t))}{N} + r(t)$$

On détermine ensuite sur cette courbe d'oubli le temps d'attente qui correspond à la performance souhaitée. Toutefois, il est possible que l'utilisateur soit bien plus performant que l'estimation du simulateur, entraînant des temps d'attente assez long. Nous avons donc décidé de limiter le temps d'attente  $t_{wait}$  à un maximum de  $T = t_{max}$  pour rendre le gameplay plus dynamique et éviter l'ennui de l'utilisateur, qui entraîne en général une baisse de l'attention. Si le temps d'attente estimé par le simulateur dépasse  $t_{max}$ , on ajoute un carré à l'interface, ce qui augmente la difficulté. La simulation est par la suite relancée, résultant en une nouvelle courbe d'oubli, avec un temps  $t_r$  estimé théoriquement inférieur à celui déterminé durant la simulation précédente. En effet, s'il faut un temps  $t_r$  pour une personne pour oublier  $K$  éléments, il est fort probable qu'il oubliera  $K + 1$  éléments en moins de temps.

On procède de manière similaire pour les temps trop courts, en limitant le temps minimal à  $t_{min}$ , ce qui implique que si le simulateur trouve un temps  $t_r < t_{min}$ , on diminue le nombre de carrés à retenir d'une unité.

## Traitement des données

Au cours du jeu, il est primordial de récolter les données de jeu de chaque utilisateur, et de s'assurer qu'elle ne soient pas perdues en cas de problème. Nous avons donc créé une classe `fileManager`, qui permet d'enregistrer les données nécessaires, puis de les copier sur un compte Dropbox.

Avant de commencer à jouer, l'utilisateur doit renseigner certaines informations comme on peut le voir sur la figure ci-dessous.

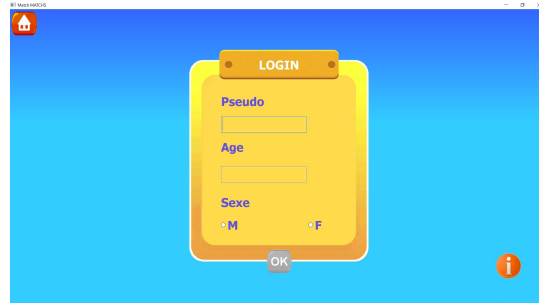


FIGURE 4.5 – Fenêtre de login

Un dossier est créé pour cet utilisateur, dans lequel deux fichiers de tableur au format csv sont créés. Le premier contient les données utilisateur et les données de jeu. Ils regroupent les clics et leur timestamp qui permettent de retracer les événements importants de la partie, la performance du joueur, ainsi que les paramètres  $K$  et  $t_{wait}$  de chaque tour de jeu.

Le deuxième fichier est réservé aux données de simulation. Il contient les données des différentes courbes d'oubli simulées tout au long de la partie, les paramètres qui définissent chacune d'entre elles et la configuration du contrôleur. On peut voir en figure 4.6 un exemple de chacun de ces fichiers.

À chaque tour de jeu pour le premier fichier, et à chaque itération du simulateur pour le deuxième, ces fichiers sont mis à jour. Cela garantit que si la partie s'arrête avant la fin, on conserve quand même les données jusqu'à ce moment là.

À la fin de la partie ou lorsque l'on quitte le jeu, on utilise l'API Dropbox v2 de python pour uploader une copie des fichiers ainsi créés. On apporte ainsi une sécurité aux

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	pseudo	age	gender	K	t <sub>wait</sub>	display time	guess time	guess	gx	gy	answer	ax	ay	correct
2	ben	40 m		4	2140	500	840	6	478	463	6	478	463	1
3	ben	40 m		4	2140	500	175	3	1169	749	3	1169	749	1
4	ben	40 m		4	2140	500	831	1	1169	176	1	1169	176	1
5	ben	40 m		6	1000	500	615	3	1169	749	5	596	749	0
6	ben	40 m		6	4075	500	638	4	883	868	7	596	176	0
7	ben	40 m		7	575	500	6499	2	1288	463	1	1169	176	0
8	ben	40 m		7	575	500	1206	3	1169	749	6	478	463	0
9	ben	40 m		7	575	500	293	0	883	58	4	883	868	0
10	ben	40 m		7	575	500	113	1	1169	176	1	1169	176	1
11	ben	40 m		7	575	500	253	5	596	749	6	478	463	0
12	ben	40 m		7	575	500	489	6	478	463	5	596	749	0
13	ben	40 m		7	575	500	402	4	883	868	5	596	749	0
14	ben	40 m		7	575	500	1068	6	478	463	6	478	463	1
15	ben	40 m		7	575	500	768	5	596	749	4	883	868	0
16	ben	40 m		7	575	500	272	7	596	176	7	596	176	1

(a) Extrait d'un fichier de données de jeu.

	A	B	C	D	E	F	G	H	I	J
1	t <sub>sim</sub>	prob <sub>sim</sub>	Q	dt	t	n	Accp	Accs	G	error
2	0	1	68	10	575	7	0	0.7	0.3	0
3	10	1	68	10	575	7	0	0.7	0.3	0
4	20	1	68	10	575	7	0	0.7	0.3	0
5	30	1	68	10	575	7	0	0.7	0.3	0
6	40	1	68	10	575	7	0	0.7	0.3	0
7	50	0.99387755	68	10	575	7	0	0.7	0.3	0
8	60	0.98979592	68	10	575	7	0	0.7	0.3	0
9	70	0.98367347	68	10	575	7	0	0.7	0.3	0
10	80	0.98163265	68	10	575	7	0	0.7	0.3	0
11	90	0.98163265	68	10	575	7	0	0.7	0.3	0

(b) Extrait d'un fichier de données de simulation.

FIGURE 4.6 – Exemples de fichiers de données.

fichiers, dans le cas où l'ordinateur utilisé pour le jeu subirait une panne ou un autre quelconque problème. Cela permet également de pouvoir accéder aux fichiers depuis n'importe quel endroit avec une connexion internet, et faciliter l'échange, l'analyse et la manipulation des données par les membres de l'équipe.

#### 4.1.2 Protocole et campagne de test

Pendant et après le développement du jeu, il est nécessaire d'effectuer des tests afin de valider le modèle, la convergence du système, l'influence des paramètres ou encore la jouabilité. Pour cela nous avons réalisé deux évaluations différentes. La première a été effectuée avec un automate, qui simule un joueur virtuel jouant au jeu. La deuxième a été faite avec des groupes de personnes volontaires jouant au jeu.

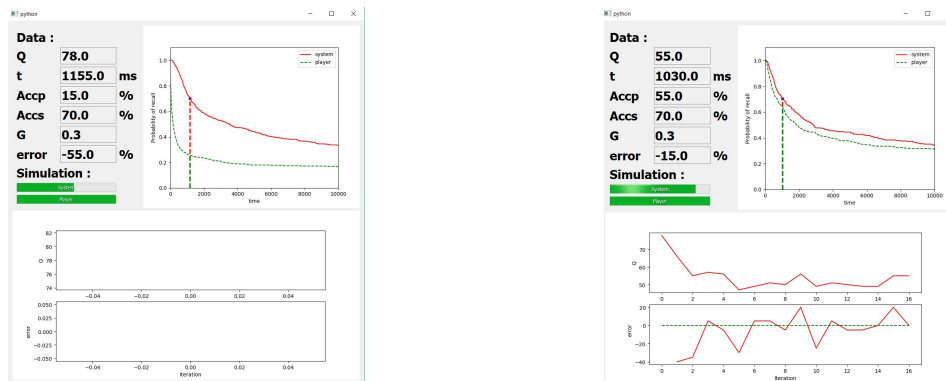
#### Tests avec le simulateur de joueur

L'automate que nous avons implémenté simule le comportement d'un joueur humain dont la dégradation de la mémoire de travail suit le même modèle que celui implémenté

dans le simulateur de mémoire. On caractérise ce joueur par trois paramètres fixes, à savoir  $Q_p$ , le nombre de quantas disponible,  $\delta_p$  le temps entre chaque itération de la politique, ainsi que  $L_p$ , le seuil de quanta à partir duquel une information est oubliée. Ces valeurs permettent d'établir un courbe de rappel  $r(t)$  qui définit, à chaque tour de Match<sup>2</sup>s, le comportement de l'automate selon le nombre de carrés  $K$  présentés.

Ainsi, l'automate contient deux courbes de rappel, une première correspondante à l'estimation des paramètres du joueurs, inconnus du système, et une seconde permettant de simuler le comportement du joueur. Le but du système est de retrouver, en partant de son estimation initiale, les paramètres qui décrivent le joueur virtuel.

Lorsqu'on lance l'automate, le système propose une configuration de jeu, à savoir un nombre de carrés  $K$  et un temps  $t_{wait}$ . Cela sert de valeurs d'entrée au simulateur de joueur qui va estimer une probabilité  $p_s$  de réussir la tâche. On effectue un tirage qui suit une loi de Bernoulli de probabilité  $p_s$ , dont le résultat désignera l'issue de ce tour de jeu. Le système compare ensuite la performance du joueur virtuel pendant  $d$  tours, à celle désirée, qui permettra d'ajuster son estimation  $Q_e$ . Un tour de simulation de l'automate correspond donc à  $d$  tours de jeu. On peut voir en figure 4.7 l'interface de l'automate à différents tours de simulation.



(a) Tour 1 de la simulation :  $Q_e = 78$  quanta. (b) Tour 16 de la simulation :  $Q_e = 55$  quanta.

FIGURE 4.7 – Interface de l'automate : Estimation du nombre de quanta  $Q_e$  d'un joueur virtuel possédant  $Q_p = 50$  quanta.

Cet automate va nous permettre de calibrer certains paramètres, à savoir le nombre de tours de simulation nécessaires pour que le système converge vers une estimation correcte du nombre de quanta de l'utilisateur et le gain  $G$  du contrôleur proportionnel, ainsi qu'observer le comportement du système dans différentes situations et déterminer ses limites.

## Tests avec des humains

Les tests sur les humains sont divisés en deux parties. Une première campagne de test a eu pour but de préparer et définir le protocole à utiliser lors de la phase de tests finale, qui correspond à la deuxième campagne de test.

Durant cette partie, nous avons demandé aux personnes du laboratoire BROCA et du CRI de tester le jeu, et nous donner leur avis en tant qu'utilisateurs et professionnels. Nous avons ainsi validé avec les neuropsychologues du laboratoire que le jeu était pertinent et cohérent avec les tests de mémoire de travail. Cela nous a permis également de tester l'ensemble de l'interface et du gameplay, et y apporter des améliorations et fixer certains paramètres d'initialisation.

Plusieurs modifications graphiques ont été effectuées lors de cette première campagne, concernant essentiellement les couleurs et taille d'écriture, le choix des couleurs des carrés, et des corrections et ajouts dans les instructions.

Les paramètres de l'expérience ont également été fixés durant cette phase. Tout d'abord, nous avons déterminé le nombre de tours convenable entre chaque adaptation. Il fallait faire le compromis entre précision, et donc vitesse de convergence, et dynamisme du jeu, reflété par l'ennui ou l'investissement du joueur. De même, nous avons fixé le nombre total de tours par partie en fonction du retour sur l'état de fatigue des joueurs.

Une fois ces premiers réglages effectués et les paramètres choisis, nous avons défini le protocole de test, puis entamé la deuxième campagne. Nous avons recruté 20 participants volontaires (9 femmes), âgés entre 18 et 40 ans ( $26.05 \pm 5.25$ ), pour jouer à Match<sup>2</sup>s. Toutes ces personnes sont diplômés ou en cours d'éducation supérieure, et ne présentent pas de déficience cognitive. Chaque participant a joué 125 tours de jeu, dont les 5 premiers étaient des tours d'entraînement afin de se familiariser avec le jeu. Durant les 120 tours suivants, à chaque  $d = 20$  tours, le simulateur de mémoire, avec une estimation initiale de  $Q_e = 68$  quanta, met à jour cette estimation en fonction des performances du joueur durant les 20 derniers tours. Les paramètres  $\delta_t$  et  $L$  ont été fixés selon les résultats de [40] en prenant les paramètres qui ont permis d'obtenir la prédiction la plus proche des données des sujets de l'expérience :  $\delta_t = 10\text{ms}$  et  $L = 7$ . Le nombre de carrés présentés est initialisé à 7, ce qui correspond à une tâche cognitivement complexe.



## 4.2 Cas d'utilisation 2 : Extension au cas séquentiel

Cette expérience consiste à simuler l'encodage d'éléments en séquence, selon le modèle décrit en 3.3.2 et le tester sur la base de données disponible. Le but est de valider le modèle pour l'utiliser dans un deuxième jeu de mémoire présenté par la suite.

### 4.2.1 Simulation de l'encodage en séquence

Nous avons présenté en 3.3.2 un modèle d'encodage en séquence basé sur un processus de décision markovien. Pour trouver la politique d'encodage optimale, nous avons utilisé le QLearning [29], qui est une technique d'apprentissage par renforcement. Son but est de trouver une politique optimale que doit suivre un agent pour obtenir une récompense maximale à long terme. Cette politique est définie par un ensemble d'états  $S$  et d'actions  $A$ .

Ici, comme la tâche est de retenir un maximum d'éléments, la récompense est maximale lorsque l'on choisit une action  $a \in A$  qui nous amène à un état  $s \in S$  contenant un maximum d'éléments. L'algorithme permet d'obtenir une fonction de valeur action-état  $Q(s, a)$  représentée sous forme d'une matrice. Cette matrice est initialisée aléatoirement, puis l'agent parcourt l'espace d'états en choisissant à chaque itération une action et observe la récompense et le nouvel état. La matrice  $Q$  est alors mise à jour selon la règle suivante :

$$Q(s, a) = Q(s, a) + \alpha[R(s, a) + \gamma \max_{a'} Q(s', a') - Q(s, a)], \quad (4.1)$$

avec  $R(s, a)$  la récompense reçue pour la transition de l'état  $s$  à l'état  $s'$  en choisissant l'action  $a$ .  $\alpha$  est un paramètre qui correspond au taux d'apprentissage, et détermine l'importance de l'influence des nouvelles informations par rapport aux précédentes sur la nouvelle valeur de  $Q(s, a)$  pendant l'apprentissage.  $\gamma$  est le facteur d'actualisation. Cela correspond à l'importance qu'on donne aux récompenses futures.

Pour implémenter cet algorithme, il faut construire la matrice contenant tous les états possibles et la matrice de récompense qui indique la récompense à attribuer à chaque action possible pour chaque état. La construction de ces matrices peut être très coûteuse en temps selon la taille du vecteur représentant un état  $s$  et le nombre d'actions  $a$  possibles.

Comme on l'a observé en 3.3.2, l'allure des courbes de rappel forment une fonction en "U", observées avec plusieurs tailles de liste différentes. On peut donc supposer que

la même stratégie d’encodage est utilisée, indépendamment du nombre d’éléments. On a donc décidé de se limiter à 4 éléments, en fixant un nombre de quanta assez faible pour ne pas permettre de pouvoir distribuer un maximum de quanta à chaque élément. On retrouve le schéma utilisé par Murdock en proposant des listes contenant un nombre de mots supérieur à la capacité de stockage maximale d’une personne moyenne.

Nous avons également modélisé le phénomène de rehearsal (ou répétition), à chaque arrivée d’un élément, par un renforcement des éléments encore présents en mémoire, en y ajoutant des quanta aux cases mémoire correspondantes.

Après la construction des matrices  $S$  et  $R$ , nous avons implémenté l’algorithme de QLearning tel que décrit précédemment, pour obtenir la matrice  $Q$ . On note qu’à chaque passage d’un état à l’autre, on applique une dégradation à chaque case mémoire allouée individuellement, comme décrit en 3.3.1, puis on applique la dégradation entre les éléments, comme décrit 3.1.1. Nous avons donc observé puis analysé la stratégie utilisée par l’agent optimal avec 4 éléments, en identifiant la logique utilisée pour distribuer les quantas, afin de la reproduire à plus grande échelle.

Une foiscelle-ci définie, nous avons pris les données de l’expérience de Roberts, qui représentent, pour chaque joueur, la position des éléments retrouvés pour chaque liste qu’on lui a énoncée. La base contient des données de 12 personnes ayant participé à 24 sessions de 20 listes de mots, ce qui fait un total de 5760 tours. Elles sont séparées en deux vitesses de présentation et deux nature d’énonciation (visuelle et auditive). Toutes les données des sujets de l’expérience ont été regroupées en fonction de la taille de liste, de la nature de l’énonciation et de la vitesse d’énonciation. Nous avons alors simulé l’encodage en suivant la stratégie que l’on a défini, puis la dégradation, pour obtenir en sortie les courbes de rappel pour chaque taille de liste, selon la position sérielle, à comparer avec celles de la base de données.

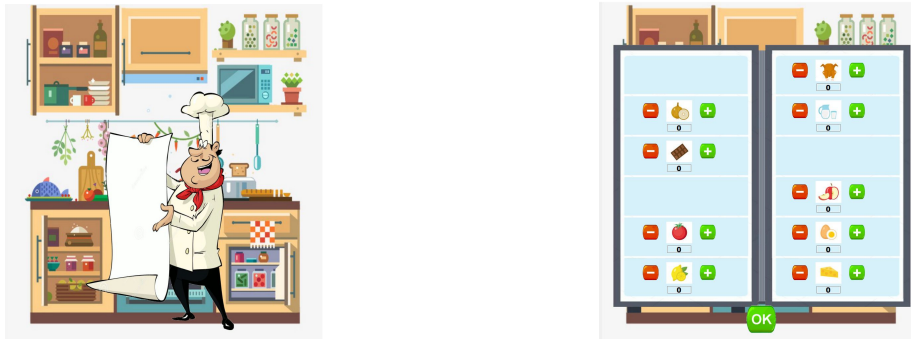
## 4.2.2 Jeu de mémoire de travail verbale

En parallèle du développement du modèle en séquence, nous avons imaginé un jeu qui pourrait correspondre à une tâche de mémoire en séquence. Une fois encore, il faut se mettre dans des conditions où la tâche concerne uniquement la mémoire de travail. Pour être plus près d’une tâche de la vie quotidienne, nous avons décidé d’utiliser cette fois des informations verbales. En effet, il faut se rappeler qu’on doit être capable d’utiliser notre modèle pour assister des personnes en difficulté cognitive dans leur vie quotidienne, où les stimuli verbaux sont plus importants pour leur communication.

Toutefois, comme on a pu le voir en 3.2.2, l’encodage verbal est différent du visuel,

et cela laisse prévoir des problèmes lors de l'utilisation du modèle qu'on utilise. On essaie tout de même de procéder de telle sorte que les éléments ne soient pas contenus dans des phrases, pour être sûrs de minimiser l'action de la mémoire épisodique.

Dans le second jeu que l'on a réalisé, on doit assister un chef de cuisine, représenté par un avatar, et retrouver les ingrédients qu'il nous indique. Pour cela, l'avatar énonce à l'aide d'un synthétiseur vocal les ingrédients un à un, chacun précédé par une quantité, avec un taux d'énonciation constant. Un réfrigérateur est alors affiché à l'écran contenant un certain nombre d'éléments. L'utilisateur doit alors retrouver les éléments demandés en bonne quantité, comme demandé par l'avatar. On peut voir des captures d'écran effectuées pendant un tour de jeu en Figure 4.8.



(a) Avatar énonçant les ingrédients (synthèse vocale). (b) Recherche des éléments énoncés dans le réfrigérateur.

FIGURE 4.8 – Interface du jeu de mémoire de travail verbale

Comme dans le premier jeu, on enregistre les données de jeu pour pouvoir les analyser. On a ainsi pu, après avoir mis en place l'interface graphique, tester le jeu et obtenir des premières données. Trois personnes ont testé le jeu, et nous avons commencé par observer ce premier jeu de données, car nous n'avons pas pu, faute de temps, poursuivre avec une campagne de test plus importante.

Ces premières observations ont eu pour but d'observer s'il y a eu une compression pendant l'encodage, comme discuté en 3.2.2. Pour cela nous avons simulé la dégradation des éléments du simulateur de mémoire de MATCHS pour prédire le score, et comparer avec les observations.

# Chapitre 5

## Résultats

### 5.1 Cas d'utilisation 1

Les exécutions du simulateur sous plusieurs conditions ont permis de valider les performances et le comportement du simulateur, permettant de mener la campagne de test et conclure sur l'efficacité du système de contrôle.

#### 5.1.1 Simulateur de joueur virtuel

A l'aide du simulateur de joueur, on a pu tester la convergence du système en considérant que la mémoire de travail d'un joueur est modélisée par le simulateur de mémoire utilisé. En plus de tester la convergence, cela nous permet d'avoir une idée du nombre de tours nécessaires avant que le système ne trouve une estimation du nombre de quanta près de la valeur réelle.

On peut voir en Figure 5.1 l'évolution de la performance du joueur virtuel au jeu Match<sup>2</sup>s, pour 20 itérations du système, correspondant à 400 tours de jeu, en utilisant comme paramètres  $G = 0.25$ ,  $Q_p=70$ ,  $\delta_{tp} = 11$  ms et  $L_p = 7$ . L'estimation initiale du nombre de quanta a été fixée à  $Q = 40$ , et la précision souhaitée à  $\alpha_s = 0.3$ .

Deux graphes y sont représentés. Le premier, en haut, décrit l'évolution de l'estimation du système du nombre de quanta du joueur virtuel, représentée par la courbe continue en rouge, ainsi que le nombre de quanta réel du joueur virtuel, en pointillés verts.

La deuxième courbe montre l'évolution de l'erreur, en rouge au cours de la partie, autour de 0 (ligne constante en pointillés verts). Ces deux graphes nous permettent de constater qu'après 10 itérations du système, l'erreur est déjà stable et autour de 0, et

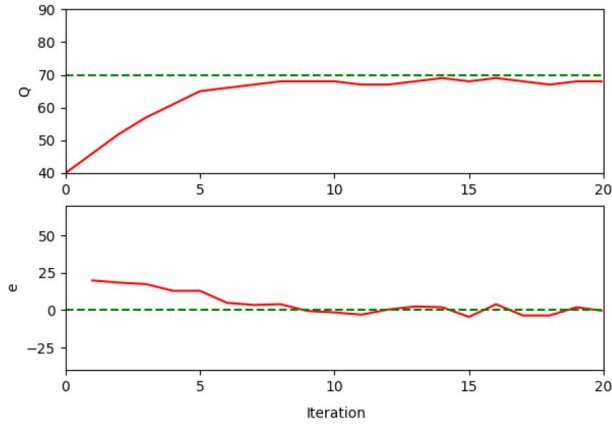


FIGURE 5.1 – Exemple de résultat de simulateur pour 20 itérations.

que la valeur estimée de  $Q$  fluctue autour de  $Q_p$ . Cela démontre la capacité d'adaptation du framework dans une tâche de même nature que le jeu Match<sup>2</sup>s.

En faisant varier les paramètres et en observant le comportement sous certaines conditions, nous avons pu mieux comprendre leur influence sur le système. Nous avons observé que, même en ayant un joueur virtuel avec des valeurs de  $\delta_t$  et  $L$  différentes de celle utilisées par MATCHS, le simulateur de mémoire arrive à forcer le joueur à obtenir la performance souhaitée, simplement en ajustant la valeur de  $Q$ .

On peut voir en Figure 5.2 l'évolution de  $Q$  (en haut, en rouge) et de  $e$  (en bas, en rouge) lorsque le système interagit avec un joueur virtuel modélisé par une configuration des paramètres différentes. Le système est initialisé avec  $Q = 40$ .

On peut voir sur ces graphes qu'une plus grande valeur de  $\delta_{tp}$  peut être compensée par une plus grande valeur de  $Q$ . En effet, comme  $\delta_{tp} > \delta_t$ , la mémoire de travail de joueur virtuel va présenter une dégradation plus lente, ce qui correspond à une personne ayant des capacités cognitives plus importantes.

Le même phénomène se produit lorsque  $L_p < L$ . Une plus petite valeur pour  $L_p$  résulte en un meilleur encodage de l'information, impliquant un comportement similaire à une personne avec de plus grandes capacités cognitives.

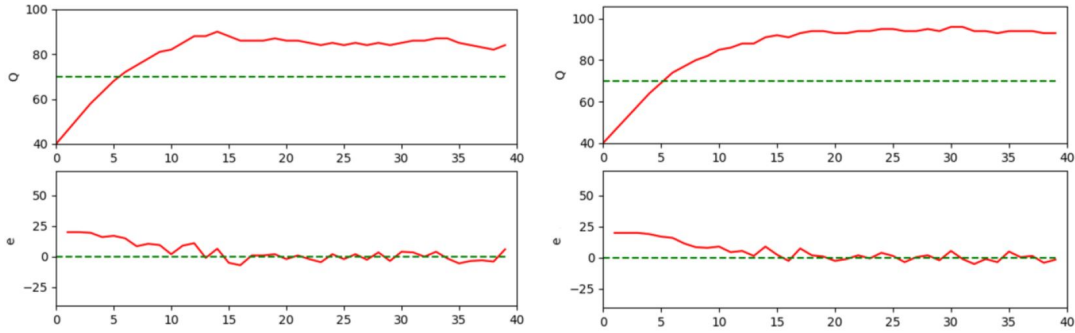


FIGURE 5.2 – A gauche :  $Q_p = 70, \delta_{tp} = 20, Lp = 7$  ;  
A droite :  $Q_p = 70, \delta_{tp} = 10, Lp = 3$

### 5.1.2 Campagnes de test

Après avoir pris en compte les retours des joueurs, et récolté les données de jeu, nous avons analysé les résultats obtenus. Pour évaluer le système, nous avons observé l'évolution de l'erreur de tous les joueurs. Nous avons tracé l'évolution de l'erreur absolue moyenne, qu'on peut voir en Figure 5.3.

Nous y avons appliqué une régression linéaire des moindres carrés (courbe en pointillés rouge), qui suggère fortement que la pente est négative ( $p$ -value =  $1.14 \times 10^{-10}$ ), indiquant que la valeur de l'erreur diminue au cours du temps.

Toutefois, même s'il y a une claire diminution de l'erreur absolue moyenne à chaque itération, on peut remarquer qu'au dernier tour, l'erreur augmente légèrement. Cela peut être dû au fait que le nombre d'itérations de l'algorithme n'a pas été suffisant pour atteindre la stabilité. Le nombre de tours par simulation étant de 20 peut également causer des erreurs d'ajustement une fois près de la valeur idéale, du fait de l'incertitude sur la mesure de la performance. Ce problème pourrait être réglé en optant pour un contrôleur plus complexe, comme un PID, pour amortir plus finement son approche vers la valeur idéale.

Cela signifie également que le joueur, se familiarisant avec le jeu, commence à établir des stratégies. Une fois de plus, on rappelle que le cerveau est en constante recherche d'optimisation d'énergie.

Ce fait est appuyé par une remarque régulière des joueurs en leur demandant leur re-

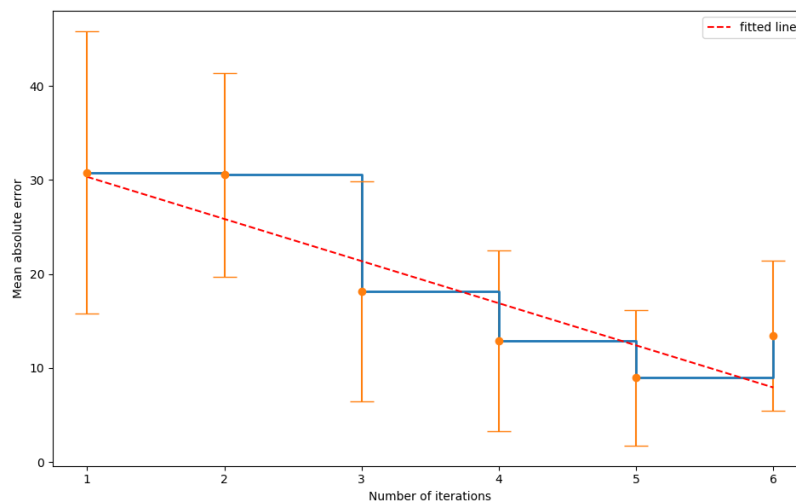


FIGURE 5.3 – Evolution de l’erreur absolue moyenne pendant 6 itérations de jeu.

tour, qui était souvent : "J’ai senti que je devenais meilleur à la fin", ce qui reflète d’une part que MATCHS s’est bien adapté à sa capacité cognitive, mais aussi que le joueur s’est également adapté au jeu.

## 5.2 Cas d’utilisation 2

L’observation de la politique trouvée par QLearning nous a permis d’implémenter une première version de notre modèle d’encodage, qu’on a pu tester sur des données réelles. D’une autre part le développement du second jeu nous a amené à approfondir notre réflexion sur l’encodage dans la mémoire de travail verbale.

### 5.2.1 Encodage et dégradation en séquence

Avant la recherche de la politique optimale par QLearning, nous avons du décider de l’état initial à partir duquel lancer l’algorithme. Lorsqu’un état initial est choisi, l’algorithme va observer à quel tour de jeu il correspond, puis continuer jusqu’à la fin de la liste de 4 éléments, en mettant à jour la matrice  $Q$ . Le tour de jeu peut être déduit en observant la position de la dernière case mémoire contenant au moins un quanta.

Nous étions face à un choix entre une initialisation aléatoire et une initialisation à une valeur particulière. Dans notre cas, il est préférable de choisir l'état initial comme décrit en 3.3.2 §5, à savoir  $s_0 = [Q, 0, 0, 0, 0]$ . Cela nous a également permis de gagner en temps d'exécution pour la construction de la matrice d'états possibles  $S$ , la matrice de récompense  $R$  et donc le calcul de la matrice  $Q$ , car beaucoup de cas se retrouvent exclus avec cette initialisation.

Nous avons observé par la suite le comportement de l'agent entraîné sur des séquences en variant le nombre de quanta à disposition. On a remarqué que l'allocation des quanta suivait un certain raisonnement. Au début, l'algorithme alloue une quantité assez élevée au premier élément, puis commence à diminuer le nombre de quanta alloué pour chaque case pour les prochains éléments jusqu'à finir par ne plus pouvoir distribuer au moins  $L$  quanta par case. Les nouvelles informations ne sont même pas encodées par la suite, jusqu'à ce qu'un élément soit perdu de la mémoire, et qu'on récupère les quanta qui lui étaient alloués.

Nous avons donc reproduit la politique observée, dans un simulateur, en prenant en compte la répétition à chaque nouvelle information arrivante comme décrit en 3.3.2. Ensuite, en utilisant les données de l'expérience de Roberts, nous avons effectué les moyennes des performances par groupes. Ces moyennes représentent la courbe de rappel des éléments en fonction de la position sérielle. Le simulateur est configuré de manière à correspondre à un groupe de liste de même taille, de même vitesse d'énonciation, et de même nature d'énonciation. La taille de la liste correspond au nombre de cases mémoires mises à disposition. Un changement de vitesse d'énonciation correspond à un changement du temps de dégradation avant l'arrivée d'un nouvel élément. Si la vitesse correspond à 0.5 s/mot, le temps de dégradation entre les éléments sera plus court qu'à la vitesse de 1 s/mot. La nature de l'énonciation n'a pas été prise en compte, car les résultats sont quasi similaires, et ont une influence non significative sur la performance générale [Roberts]. Toutefois, pour éviter tout biais, seules les informations présentées sous forme visuelle ont été utilisées.

On peut ainsi distinguer deux groupes différenciés par la vitesse d'énonciation des éléments. Chacun de ces groupes se divise en sous-groupes qui diffèrent par la taille des listes utilisées. Nous avons commencé par nous concentrer sur un groupe de vitesse pour rechercher les paramètres  $Q$ ,  $\delta_t$ ,  $r_{paquet}$  et  $L$ , tels que la courbe de rappel moyenne résultante de la simulation effectuée avec le système de simulation de l'encodage et de la dégradation décrit en 3.3 s'approche au mieux des données réelles observées.

Nous avons commencé par fixer des valeurs limites de  $Q$ , le nombre de quanta de



l'utilisateur. Les résultats de Suchow [40] lui ont permis de trouver les meilleurs paramètres pour décrire la moyenne des sujets ayant participé à son expérience, à savoir :  $Q = 58$  quanta,  $\delta_t = 10$  ms et  $L = 7$ . Nous choisissons de nous appuyer sur cette estimation de  $Q$ .

On peut en déduire une valeur approchée du nombre de quanta  $r_p$  que contient un paquet élémentaire pour l'allocation lors de l'encodage. Sachant que le maximum de paquets que l'on pouvait allouer à la fois lors de la recherche de la politique par QLearning était de 4 paquets, on considère que le maximum de quanta pouvant être alloué est  $r_{max} = 4r_p$ . Or on a également établi que  $r \leq Q$ , donc  $r_p \leq Q/4$ . On prend donc  $r_p = 15$  quanta.

$L$  représente le seuil de quanta à partir duquel une information est oubliée. Comme dans l'expérience de Roberts, ce sont des mots et non des couleurs que les sujets doivent se rappeler, on considère qu'il peut être différent pour cette expérience. On le fait donc varier tel que  $1 \leq L \leq r_p$ , et donc  $1 \leq L \leq 15$ . De même, on fait varier  $\delta_t$ , qui correspond au temps entre chaque itération de l'algorithme tel que  $5 \leq \delta_t \leq 100$ .

On peut voir en Figure 5.4 les résultats avec les paramètres qui font obtenir le meilleur score de ressemblance avec les données observées. La Figure 5.4.a) correspond au résultat d'une simulation en utilisant la dégradation exponentielle décrite en 3.3.1 pour décrire la dégradation que subit chaque case mémoire individuellement, et le résultat de la simulation en Figure 5.4.b) utilise une dégradation aléatoire avec une probabilité  $p_{degrade} = 0.05$  de voir un quantum se dégrader dans chacune des cases, à chaque itération de l'algorithme.

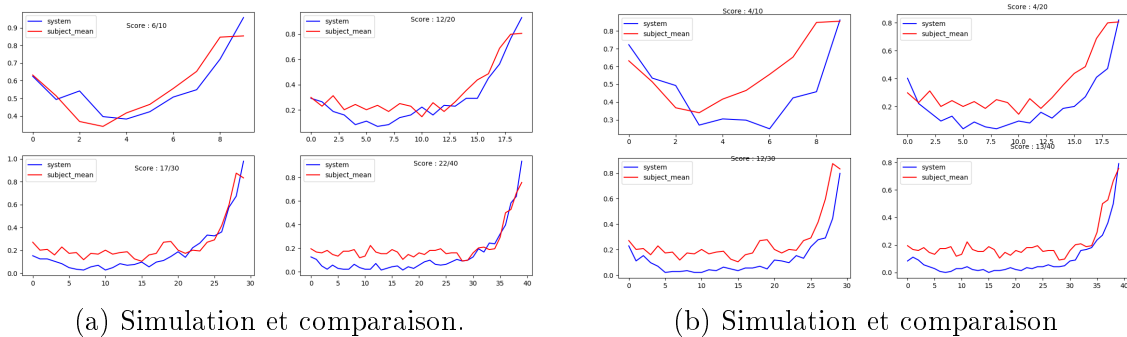


FIGURE 5.4 – Simulation  $v=0.5$  s/mot

Les paramètres pour lesquels les prévisions s'adaptent au mieux donnent des résultats assez satisfaisants. On peut voir que les courbes ont des allures similaires, et que les prédictions sur les performances sont correctes. On remarque également que la dégradation exponentielle a donné un résultat bien plus fidèle aux données que la

dégradation aléatoire.

En fixant les paramètres ainsi trouvés, on regarde le résultat de la simulation en changeant la vitesse de présentation. On ajuste le simulateur en conséquence en changeant le temps entre l’encodage de chaque nouveau mot. On peut voir en Figure 5.5 le résultat des simulations pour une dégradation individuelle des cases mémoire qui suit la loi exponentielle et pour une dégradation aléatoire.

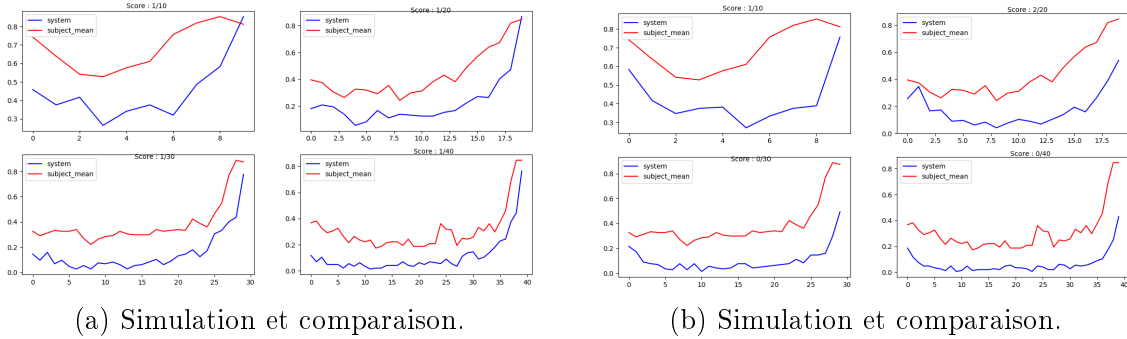


FIGURE 5.5 – Simulation  $v=1$  s/mot

Les résultats pour une vitesse de 1 s/mot n’ont pas été concluants, même si l’allure de la courbe simulée reste proche de celle des données observées.

Pour les deux types de dégradation, les performances simulées par le système sont inférieures à celles des données observées. C’est une des limitations du modèle d’encodage. En effet, dans notre modèle d’encodage, on considère qu’à chaque arrivée d’un nouvel élément, on alloue une quantité  $r$  de quanta divisée en paquets élémentaires de  $r_p$  quanta, telle que  $0 \leq r \leq 4 * r_p$ .

On pourrait penser, que comme la dégradation, elle se fait progressivement tout au long d’une période donnée, ou jusqu’au moment où un nouvel élément arrive. On peut imaginer qu’il existe un paramètre similaire à  $\delta_t$  utilisé pour décrire la dégradation, qui correspond à la fréquence à laquelle on ajoute des quanta. Ainsi, il est très probable que si on attend plus de temps avant que l’information suivante arrive, on lui allouera plus de quantas, et donc elle sera mieux conservée. Ce qui n’est pas le cas avec le modèle d’encodage utilisé, qui distribue un nombre de quanta indépendamment de la durée d’attente avant l’arrivée d’un nouvel élément.

## 5.2.2 Discussion sur le jeu de mémoire de travail verbale

Avant d’implémenter le modèle, nous nous sommes penchés sur la question de compression dans la mémoire verbale.

Nous avons imaginé plusieurs cas possibles qui nous ont amené à penser comment pourraient être encodées les informations verbales dans le cas de notre jeu. Par exemple, si on doit retrouver la liste d'éléments (2 tomates, 2 oignons, 2 poivrons), le nombre d'éléments encodés pourrait être 6 si la liste est encodée de cette manière : (2, tomates, 2, oignons, 2, poivrons), 3 si on considère que la quantité et l'ingrédient sont encodés dans la même case mémoire de la manière suivante : (2 tomates, 2 oignons, 2 poivrons) ou encore 4 si on considère que le chiffre 2 peut être compressé dans une seule case mémoire : (2, tomates, oignons, poivrons) sachant que tous les ingrédients sont en même quantité.

On peut également imaginer la possibilité, que selon le cas, l'encodage va optimiser le nombre de cases mémoires à utiliser en compressant les informations lorsque c'est possible. La compression serait possible pour les quantités, indépendamment des ingrédients, lorsque certains motifs apparaissent, comme des suites incrémentales, des répétitions de chiffres ou d'autres combinaisons facilement identifiables. A contrario, les quantités de la liste (3 tomates, 8 oignons, 5 poivrons) ne peuvent être compressées à l'aide d'un motif facilement identifiable, rendant la tâche de mémorisation plus complexe.

On pourrait également même considérer qu'il y a un effet d'association qui peut être établi entre des ingrédients. Par exemple, on pourrait penser que deux éléments proches, comme "lait" et "fromage", pourraient se renforcer mutuellement par association, et ainsi se retenir plus facilement que "poivron" et "chocolat". Toutefois, nous avons décidé dans un premier temps de négliger cet effet, en choisissant les listes d'ingrédients de manière aléatoire, pour minimiser les possibles associations qui pourraient être plus présentes si on utilise par exemple des listes d'ingrédients de recettes réelles.

Par manque de temps, nous n'avons pas pu organiser une campagne de test et récolter assez de données pour tester ces hypothèses.

# Bilan

La mémoire de travail est un vaste sujet à explorer. Les multiples théories décrivant son fonctionnement témoignent de notre connaissance inaboutie des mécanismes du cerveau impliqués. De multiples travaux et recherches sur l'élaboration de systèmes pouvant mesurer et s'adapter aux capacités cognitives, démontrent qu'il est tout de même possible de s'approcher d'une modélisation de certains aspects de la cognition.

Nous avons participé au développement de MATCHS, un framework destiné à l'adaptation d'interfaces en fonction des capacités cognitives de l'utilisateur [37]. En utilisant le modèle de dégradation de la mémoire de travail introduit par Suchow, nous avons pu simuler l'oubli d'une personne, lors d'une tâche de mémorisation. Nos résultats pour les tests utilisateurs suggèrent que le framework est capable d'ajuster la difficulté d'une tâche similaire au jeu Match<sup>2</sup>s, en modulant le nombre d'informations et le temps de présentation des éléments.

Le système reste toutefois à améliorer, comme on a pu le voir avec l'adaptation du joueur à la tâche, rendant l'estimation de sa capacité cognitive plus délicate. De plus, les variations de son attention, sa motivation et sa fatigue jouent également un rôle important dans sa performance. Ces paramètres ne sont pas pris en compte par MATCHS, et nous pensons que le système peut encore être amélioré en y ajoutant des modules supplémentaires, capables de mesurer d'autres facteurs influençant le comportement de l'utilisateur.

Nous avons également présenté un premier modèle étendu de MATCHS, permettant d'estimer les performances d'un utilisateur lors d'une tâche de mémoire en séquence. Basé sur des hypothèses sur l'encodage, la dégradation et les renforcements par répétition, ce modèle reste à valider et compléter. Les résultats de tests menés sur les données d'une expérience réalisée par des neuropsychologues laissent penser qu'il est possible d'aboutir à un résultat plus proche du véritable mécanisme d'encodage, mais qu'une réflexion et une validation plus approfondies doivent être apportées.

D'autre part, les premiers tests sur une tâche de mémorisation impliquant la mémoire de travail verbale nous ont permis de mettre en évidence quelques problématiques à résoudre permettant de mieux modéliser le phénomène d'encodage et de dégradation des informations verbales.

# Bibliographie

- [1] Online database. Data archives of computational memory lab at the university of pennsylvania. <<http://memory.psych.upenn.edu/>>. Consulté le 23/08/2018.
- [2] John R. Anderson, Lynne M. Reder, and Christian Lebiere. Working memory : Activation limitations on retrieval. *Cognitive Psychology*, 30(3) :221–256, 1996.
- [3] Kleber De O. Andrade, Guilherme Fernandes, Glauco A.P. Caurin, Adriano A.G. Siqueira, Roseli A.F. Romero, and Rogerio De L. Pereira. Dynamic player modelling in serious games applied to rehabilitation robotics. *Proceedings - 2nd SBR Brazilian Robotics Symposium, 11th LARS Latin American Robotics Symposium and 6th Robocontrol Workshop on Applied Robotics and Automation, SBR LARS Robocontrol 2014 - Part of the Joint Conference on Robotics and Intelligent Systems.*, pages 211–216, 2015.
- [4] A D Baddeley. Working Memory. *Science*, 255(5044) :556–559, 1995.
- [5] A. D. Baddeley, S. Bressi, S. Della Sala, R. Logie, and H. Spinnler. The decline of working memory in alzheimer’s disease : A longitudinal study. *Brain*, 114(6) :2521–2542, 1991.
- [6] Alan Baddeley. The episodic buffer : A new component of working memory? *Trends in Cognitive Sciences*, 4(11) :417–423, 2000.
- [7] Luis M. Bergasa and Jesus Nuevo. Real-time system for monitoring driver vigilance. *IEEE International Symposium on Industrial Electronics*, III(1) :1303–1308, 2005.
- [8] Sara Bunian, Alessandro Canossa, Randy Colvin, and Magy Seif El-nasr. Modeling Individual Differences in Game Behavior Using HMM. *Thirteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-17) Modeling*, pages 158–164, 2017.
- [9] Davide Maria Calandra, Dario Di Mauro, Francesco Cutugno, and Sergio Di Martino. *Navigating wall-sized displays with the gaze : A proposal for cultural heritage*, volume 1621. 2016.

- [10] S. Thomas Christie and Paul Schrater. Cognitive cost as dynamic allocation of energetic resources. *Frontiers in Neuroscience*, 9(JUL) :1–15, 2015.
- [11] Gianna Cocchini, Robert H. Logie, Sergio Della Sala, Sarah E. MacPherson, and Alan D. Baddeley. Concurrent performance of two memory tasks : Evidence for domain-specific working memory systems. *Memory and Cognition*, 30(7) :1086–1095, 2002.
- [12] Guillem Collell and Jordi Fauquet. Brain activity and cognition : a connection from thermodynamics and information theory. *Frontiers in Psychology*, 6(June) :1–11, 2015.
- [13] N. Cowan. Evolving Conceptions of Memory Storage, Selective Attention, and Their Mutual Constraints Within the Human Information-Processing System. *Psychological Bulletin*, 104(2) :163–191, 1988.
- [14] N. Cowan. The magical number 4 in short term memory. A reconsideration of storage capacity. *Behavioral and Brain Sciences*, 24(4) :87–186, 2001.
- [15] Nelson Cowan, Emily M. Elliott, Scott J. Saults, Candice C. Morey, Sam Mattox, Anna Hismjatullina, and Andrew R.A. Conway. On the capacity of attention : Its estimation and its role in working memory and cognitive aptitudes. *Cognitive Psychology*, 51(1) :42–100, 2005.
- [16] Herm Ebbinghaus. Ueber das gedächtnis. 1885.
- [17] Mary Egan, Daniel Bérubé, Geneviève Racine, Carol Leonard, and Elizabeth Rochon. Methods to Enhance Verbal Communication between Individuals with Alzheimer’s Disease and Their Formal and Informal Caregivers : A Systematic Review. *International Journal of Alzheimer’s Disease*, 2010 :1–12, 2010.
- [18] D. Fougny, C. L. Asplund, and R. Marois. What are the units of storage in visual working memory? *Journal of Vision*, 10(12) :27–27, 2010.
- [19] Carmela Germano and Glynda J. Kinsella. Working memory and learning in early Alzheimer’s disease. *Neuropsychology Review*, 15(1) :1–10, 2005.
- [20] Pim Haselager, Jelle van Dijk, and Iris van Rooij. A Lazy Brain? Embodied Embedded Cognition and Cognitive Neuroscience. *Handbook of Cognitive Science*, pages 273–290, 2008.
- [21] J. D. Huntley and R. J. Howard. Working memory in early Alzheimer’s disease : A neuropsychological review. *International Journal of Geriatric Psychiatry*, 25(2) :121–132, 2010.
- [22] Frédéric Imbeault, Bruno Bouchard, and Abdenour Bouzouane. Serious Games in Cognitive Training for Alzheimer ’ s Patients. *IEEE 1st International Conference Serious Games and Applicat for Health (seGAH)*, 2011.

- [23] Pierre Jouvelot. Louise : ECA pour Alzheimer et troubles liés à l'âge. Journées rencontre entre partenaires industriels et chercheurs de l'Institut Mines-Télécom "Santé, autonomie et qualité de vie", March 2016. Poster.
- [24] Michael J Kane and Randall W Engle. The role of prefrontal cortex in working-memory capacity, executive attention, and general fluid intelligence : An individual-differences perspective. *Psychonomic bulletin & review*, 9(4) :637–671, 2002.
- [25] Elizabeth A. Kensinger, Deirdre K. Shearer, Joseph J. Locascio, John H. Growdon, and Suzanne Corkin. Working memory in mild Alzheimer's disease and early Parkinson's disease. *Neuropsychology*, 17(2) :230–239, 2003.
- [26] Anna-Mariya Kirova, Rebecca B. Bays, Sarita Lagalwar, Anna-Mariya Kirova, Rebecca B. Bays, and Sarita Lagalwar. Working Memory and Executive Function Decline across Normal Aging, Mild Cognitive Impairment, and Alzheimer's Disease. *BioMed Research International*, 2015 :1–9, 2015.
- [27] Jia Jiunn Lo, Ya Chen Chan, and Shiou Wen Yeh. Designing an adaptive web-based learning system based on students' cognitive styles identified online. *Computers and Education*, 58(1) :209–222, 2012.
- [28] R Luce. *Individual Choice Behavior : A Theoretical Analysis*. 1959.
- [29] Ms S Manju. An Analysis of Q-Learning Algorithms with Strategies of Reward Function. 3(2) :814–820, 2011.
- [30] G Miller. The Magic Number Seven, Plus or Minus Two : Some Limits on our Capacity for Processing Information. *The Psychological Review*, 63(2) :81–97, 1956.
- [31] P. A P Moran. Random processes in genetics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 1958.
- [32] Bennet B Murdock Jr. The serial position effect of free recall. *Journal of experimental psychology*, 64(5) :482, 1962.
- [33] Neil Peirce, Owen Conlan, and Vincent Wade. Adaptive educational games : Providing non-invasive personalised learning experiences. *Proceedings - 2nd IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning, DIGITEL 2008*, pages 28–35, 2008.
- [34] Martin L Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming*, 1994.
- [35] George E. Raptis, Christina Katsini, Marios Belk, Christos Fidas, George Samaras, and Nikolaos Avouris. Using Eye Gaze Data and Visual Activities to Infer Human Cognitive Styles. *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization - UMAP '17*, pages 164–173, 2017.



- [36] William A Roberts. Free recall of word lists varying in length and rate of presentation : A test of total-time hypotheses. *Journal of Experimental Psychology*, 92(3) :365, 1972.
- [37] Bruno Massoni Sguerra, Amine Benamara, Samuel Benveniste, and Pierre Jovelot. Adaptive Working Memory for Dementia using MATCHS, a MDP Approach to Oblivion Tracking. In *IEEE International Conference on Systems, Man, and Cybernetics*, 2018.
- [38] Edward E. Smith, John Jonides, and Robert a Koeppel. Dissociating Verbal and Spatial Working. *Cerebral Cortex*, 6(July) :11–20, 1996.
- [39] Ben Steichen, Giuseppe Carenini, and Cristina Conati. User-adaptive information visualization : using eye gaze data to infer visualization tasks and user cognitive abilities. *Iui*, pages 317–328, 2013.
- [40] Jordan W Suchow. *Measuring, monitoring, and maintaining memories in a partially observable mind*. PhD thesis, 2014.
- [41] Jordan W. Suchow, David D. Bourgin, and Thomas L. Griffiths. Evolution in Mind : Evolutionary Dynamics, Cognitive Processes, and Bayesian Inference. *Trends in Cognitive Sciences*, 21(7) :522–530, 2017.
- [42] Jordan W Suchow and Thomas L Griffiths. Deciding to Remember : Memory Maintenance as a Markov Decision Process. pages 2063–2068, 1965.
- [43] Daniel Szafrir and Bilge Mutlu. Pay attention ! : designing adaptive agents that monitor and improve user engagement. *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, pages 11–20, 2012.
- [44] Konstantinos Tsiakas, Cheryl Abellanoza, Maher Abujelala, Michalis Papakostas, Tasnim Makada, and Fillia Makedon. Towards Designing a Socially Assistive Robot for Adaptive and Personalized Cognitive Training. *Human Robot Interaction Conference*, (March), 2017.
- [45] Zomeran Van, Adriaan H Zomeran, Wiebo H Brouwer, Wiebo H Brouwer, et al. *Clinical neuropsychology of attention*. Oxford University Press, USA, 1994.
- [46] A. Vinciarelli, M. Pantic, D. Heylen, C. Pelachaud, I. Poggi, F. D’Errico, and M. Schroeder. Bridging the Gap Between Social Animal and\nUnsocial Machine : A Survey of Social Signal\nProcessing. *IEEE Transactions on Affective Computing (to appear)*, 3(1) :1–20, 2011.