

**LES APPLICATIONS D'INTERACTION HUMAIN - ORDINATEUR**

(İNSAN - BİLGİSAYAR ETKİLEŞİMİ UYGULAMALARI)

Réalisé par

**Murat YILMAZ**, Ingénieur en Génie Informatique

**Thèse**

Présenté en vue de

l'obtention du diplôme de

**MASTER**

Date de soumission : Septembre 14, 2010

Date de soutenance : Octobre 13, 2010

Directeur : Yrd. Doç. Dr. Burak ARSLAN

Membres de Jury : Doç. Dr. Orhan FEYZİOĞLU

Yrd. Doç. Dr. Tankut ACARMAN

## **PREFACE**

Je voudrais remercier mon directeur de thèse professeur Dr. Burak Arslan pour me guider pendant le travail.

Je voudrais également remercier mes parents et mes amis pour me rendre motivé et encouragé.

Murat YILMAZ

Istanbul, le 14 Septembre 2010

## TABLE DES MATIERES

|   |     |
|---|-----|
| PREFACE .....   | ii  |
| TABLE DES MATIERES.....   | iii |
| LISTE DES NOTATIONS.....  | v   |
| LISTE DES FIGURES.....  | vi  |
| LISTE DES TABLEAUX.....   | ix  |
| RESUME.....   | x   |
| ABSTRACT.....   | xii |
| ÖZET .....  | xiv |
| 1. INTRODUCTION .....   | 1   |
| 1.1. Qu'est ce que L'Interaction Humain-Ordinateur ?.....                   | 1   |
| 1.2. Quelques Exemples d'Interaction Humain-Ordinateur .....                | 2   |
| 1.2.1. Quelques Exemples des Matériels d'Entrée .....                       | 2   |
| 1.2.2. Sketchpad .....  | 4   |
| 1.2.3. Dynabook .....   | 5   |
| 1.2.4. Réalité Augmenté .....   | 6   |
| 1.2.5. Reactable .....  | 7   |
| 1.3. Interaction Cerveau-Ordinateur .....                                   | 8   |
| 1.4. Motivation .....   | 10  |
| 2. LES APPLICATIONS RÉALISÉES DE L'INTERACTION HUMAIN –<br>ORDINATEUR ..... | 12  |
| 2.1. Informations Générales sur PureData .....                              | 12  |
| 2.2. Contrôleur Piano .....   | 18  |
| 2.3. Contrôleur à Blobs en Utilisant Le Filtrage des Couleurs .....         | 26  |
| 2.4. Contrôleur Panneau Tactile avec Arduino .....                          | 31  |
| 3. L'INTERACTION EN UTILISANT LES ONDES CÉRÉBRALES .....                    | 36  |
| 3.1. Électro-Encéphalogramme .....  | 36  |

|      |  |    |
|------|--|----|
| 3.2. | Notre Travail sur Le Base de Données Existante.....  | 40 |
| 4.   | LA CRÉATION ET LES APPLICATIONS RÉALISÉES DE<br>L'INTERFACE CERVEAU – ORDINATEUR .....                 | 46 |
| 4.1. | L'équipement Utilisés .....  | 46 |
| 4.2. | Interaction en Utilisant Les Ondes d'Alpha avec L'Alphabet du<br>Morse.....                            | 49 |
| 4.3. | Est-ce qu'une Interface P300 Auditive est Possible? .....  | 50 |
| 4.4. | Interface de Sélection de Boissons en Utilisant des Stimuli<br>Auditoires et des Stimuli Visuels ..... | 52 |
| 5.   | CRÉATION DE L'INTERFACE CERVEAU – ORDINATEUR EN<br>LIGNE .....   | 62 |
| 5.1. | Le Flux de Travail de L'ICO en Ligne.....  | 62 |
| 5.2. | Réalisation de L'ICO P300 Speller .....  | 65 |
| 5.3. | Réalisation de L'ICO de Sélection de Boissons.....   | 67 |
| 6.   | CONCLUSION.....  | 69 |
|      | BIBLIOGRAPHIE .....  | 72 |
|      | BIOGRAPHIE .....   | 76 |

## **LISTE DES NOTATIONS**

EEG : Electro-Encéphalogramme

IHM : Interface Homme-Machine, Interaction Homme-Machine

ICM : Interface Cerveau-Machine, Interaction Cerveau-Machine

IHO : Interface Homme-Ordinateur, Interaction Homme-Ordinateur

ICO : Interface Cerveau-Ordinateur, Interaction Cerveau-Ordinateur

PE : Potentiel Evoqué

GEM : Graphics Environment for Multimedia

SVM : Support Vector Machine

ICA : Independent Component Analysis

## LISTE DES FIGURES

|   |    |
|---|----|
| Figure 1.1: Carte perforée d'Hollerith .....  | 3  |
| Figure 1.2 : Un gant utilisé pour interaction.....  | 3  |
| Figure 1.3: Utilisation de Sketchpad avec dispositif de pointage.....   | 4  |
| Figure 1.4: Illustration d'une DynaBook. Il ressemble à un Blackberry. Deux fentes sur le côté droit près de l'écran sont marqués 'Files', tandis qu'un slot stylet est sur le côté droit en bas..... | 5  |
| Figure 1.5: Une scène de l'application 'Metro Paris' pour les téléphones portables avec camera .....  | 6  |
| Figure 1.6: Reactable utilisé par plusieurs personnes .....   | 7  |
| Figure 1.7: Diagramme de framework reactIVision et exemple de quatre marqueurs de la reactIVision d'ensemble "amobea" .....   | 8  |
| Figure 1.8: signaux EEG : l'un contient la composante P300 et l'autre n'en contient pas.....  | 10 |
| Figure 2.1: Exemple 'Hello World' .....   | 13 |
| Figure 2.2: Patch nombres premiers .....  | 15 |
| Figure 2.3: L'affichage des nombres premiers obtenus .....  | 15 |
| Figure 2.4: Les signaux de 'osc~' et de 'phasor~' .....   | 16 |
| Figure 2.5: Patch sonore .....  | 17 |
| Figure 2.6: Patch sonore avec contrôle de notes .....   | 18 |
| Figure 2.7: Utilisation de la bibliothèque de GEM dans PureData pour récupérer des données à partir d'une webcam .....  | 19 |
| Figure 2.8: Explication de fonctionnement de l'objet 'pix_data' .....   | 19 |
| Figure 2.9: Création de l'objet 'karecik' en PureData.....  | 20 |
| Figure 2.10: Prise de données de 8 points de l'objet 'karecik' .....  | 22 |
| Figure 2.11: L'affichage des objets 'karecik', 'kareyazi' et 'faders' superposés sur le flux vidéo de la webcam.....  | 23 |

|   |    |
|---|----|
| Figure 2.12: Explication de l'utilisation de Contrôleur Piano.....  | 24 |
| Figure 2.13: Contrôleur Piano connecté au logiciel Reason sur MidiYoke .....  | 25 |
| Figure 2.14: Configuration du système, webcam pour l'entrée visuelle, fond blanc, des objets colorés, PureData pour la création de la rétroaction auditive.....   | 26 |
| Figure 2.15: Blobs comme des sphères blanches indiquant le centre de gravité de l'objet coloré. (a) objet unique, le blob est sur l'objet. (b) Le blob est entre deux objets de couleur bleue .....                 | 27 |
| Figure 2.16: Blobs trouvés pour les objets rouges, verts et bleus. (a) Filtrage des couleurs et les valeurs ajustées. (b) Intervention du mouvement de la main sur le canal rouge, position de blob trompeuse ..... | 28 |
| Figure 2.17: L'exemple de délai avec commande de rétroaction (à gauche), la modification de délai pour obtenir l'effet 'Beat Repeat' (à droite) .....   | 30 |
| Figure 2.18: Léon Theremin avec son instrument .....  | 31 |
| Figure 2.19: Matériel Arduino et panneau tactile à 4 fils .....   | 31 |
| Figure 2.20: Patch PureData qui sert à lire les données envoyés par Arduino sur le port série .....   | 32 |
| Figure 2.21: Patch PureData qui sert à créer des échelles musicales .....   | 34 |
| Figure 3.1: Enregistrement EEG .....  | 36 |
| Figure 3.2 : Les ondes alpha .....  | 38 |
| Figure 3.3 : Les ondes beta .....   | 39 |
| Figure 3.4 : Les ondes delta .....  | 40 |
| Figure 3.5 : La matrice de caractères, utilisée en paradigme 'oddball' .....  | 41 |
| Figure 3.6 : Schéma qui illustre les désignations d'électrodes et les numéros des canaux utilisés dans cette expérience .....   | 42 |
| Figure 3.7 : Signaux représentant la moyenne de réponses pour chaque caractère.....   | 44 |
| Figure 4.1 : Figure représentant l'enregistrement des données des sujets à l'aide de 2 ordinateurs, un pour la production de stimuli et l'autre pour l'enregistrement EEG.....                                      | 46 |
| Figure 4.2 : Interface graphique pour création des stimuli .....  | 47 |
| Figure 4.3 : Enregistrement d'EEG et bande alpha de ce signal. On peut voir la différence autour de 3500ème échantillon ou les yeux du sujet sont ouverts.....  | 49 |

|   |    |
|---|----|
| Figure 4.4 : La trace verte est la moyenne des signaux attendus P300, la trace bleue est la moyenne des signaux attendus non-p300 .....   | 51 |
| Figure 4.5: Stimuli visuels montrant 5 images, le cercle jaune clignote pendant 500 ms aléatoirement sur chaque image .....   | 53 |
| Figure 4.6 : Exemple de signal EEG contenant P300 visualisé entre 000-800 ms (trace bleue), <i>zéro-meaned</i> et la somme de niveaux de détail 6, 7 et 8 de la décomposition en ondelettes (trace verte).....  | 55 |
| Figure 4.7 : Moyennes P300 et non-P300 des sessions; auditive (à gauche), auditive + visuelle (au milieu) et seulement visuelle (à droite), des canaux; Pz (en haut) et mastoïde (en bas) pour le sujet 1 ..... | 56 |
| Figure 4.8 : Moyennes P300 et non-P300 des sessions; auditive (à gauche), auditive + visuelle (au milieu) et seulement visuelle (à droite), des canaux; Pz (en haut) et mastoïde (en bas) pour le sujet 2 ..... | 56 |
| Figure 4.9 : Moyennes P300 et non-P300 des sessions; auditive (à gauche), auditive + visuelle (au milieu) et seulement visuelle (à droite), des canaux; Pz (en haut) et mastoïde (en bas) pour le sujet 3 ..... | 57 |
| Figure 4.10 : Exemple de moyennes des réponses de 5 sessions P300 (trace bleue) et 4 non-réponses des mêmes sessions non-P300.....  | 57 |
| Figure 5.1: Système ICO à 2-Ordinateurs, BIOPAC MP35 et amplificateur .....   | 63 |
| Figure 5.2: Schéma détaillé de flux de travail de l'ICO .....   | 64 |
| Figure 5.3: Création de stimuli différents avec l'interface utilisateur.....  | 65 |
| Figure 5.4: Ordinateur numéro 1, affichant les stimuli.....   | 66 |
| Figure 5.5: Enregistrement EEG d'utilisateur en session auditif (les yeux fermés).....  | 67 |



## **LISTE DES TABLEAUX**

|  |    |
|--|----|
| Tableau 2.1: Valeurs d'entrée et de sortie pour le patch 'scaler' et leurs valeurs de fréquences pour les octaves, première et sixième ..... | 34 |
| Tableau 4.1: Valeurs de VP pour différents types de stimuli de différents niveaux des moyennes pour le sujet 1.....                          | 59 |
| Tableau 4.2: Pourcentages des classifications vraies pour 3 sujets .....   | 60 |
| Tableau 4.3: Pourcentages des classifications vraies pour la totalité des sujets .....   | 60 |

## **RESUME**

L'interaction Humain-Ordinateur (IHO) est l'un des sujets les plus importants d'aujourd'hui dans tous les aspects de notre vie quotidienne. L'aube de l'informatique personnelle dans les années 1980 fut le début d'une main mise de la technologie en évolution rapide sur la vie quotidienne. Le monde mécanique des machines à écrire, des unités de traitement de texte consacrées et les machines à calculer avec des manivelles a été rapidement laissés dans un sillage des micro-processeurs. IHO a évolué immensément depuis 20 ans. Il existe diverses manières d'interagir avec les ordinateurs d'après plusieurs champs de recherche. Le terme 'utilisabilité' est un concept essentiel dans IHO qu'il avait fait l'objet de nombreuses recherches. Aujourd'hui IHO devrait être autant 'utile' qu'utilisable.

Depuis cartes perforées jusqu'aux interfaces clavier-souris-écran, il est impossible de ne pas penser à l'avenir d'IHO. Aujourd'hui, il y a beaucoup de recherches sur les systèmes d'interaction basé sur la détection de mouvement, des objets tangibles, reconnaissance de gestes etc. Les humains peuvent communiquer verbalement, par écrit, avec mimiques et gestes etc. mais il manque aux humains la possibilité transmettre l'information directement à partir du cerveau. Même s'il n'est pas possible aujourd'hui de créer une interface pour transmettre directement les pensées, il existe des moyens d'obtenir une IHO en utilisant les seules pensées.

Quand un homme a une activité cérébrale, elle peut être mesurée en plaçant des électrodes sur le cuir chevelu. C'est ce qu'on appelle l'électro-encéphalogramme (EEG). Les recherches effectuées depuis 70 ans ont montré que certaines caractéristiques distinctes existent sur les enregistrements EEG qui pourraient être utilisés pour obtenir une interaction avec l'ordinateur. L'interface Cerveau-Ordinateur (ICO) est un domaine de recherche important qui vise la communication volontaire entre l'homme et les

ordinateurs utilisant uniquement des pensées, cela est possible par le traitement des données d'électroencéphalographie (EEG). Dans cette recherche, on travaille principalement sur ICO P300, qui est une interface basée sur potentiels évoqués (PE). Un ICO P300 peut être construit, en étant basé sur le paradigme 'oddball'. Ce paradigme est d'abord utilisé par LA Farwell et E. Donchin pour construire un ICO. Une matrice de 6 x 6 symboles a été présentée à des sujets. Ils se sont concentrés sur un symbole dans la matrice. Les colonnes et les lignes ont été flashées successivement et les PE P300 étaient présent dans les données EEG lorsque le symbole a été flashé. P300 est un PE spécifique qui représente une déviation positive, 300 ms après que le stimulus soit présenté, dans le signal EEG.

Cette étude début par la description de travaux qu'on a fait sur les systèmes d'interaction basés sur la vidéo en temps réel. Un utilisateur peut réaliser l'interaction en employant une webcam et en conséquence des rétroactions auditives et visuelles sont créées par l'ordinateur. Pour la partie concernant L'ICO, on a tout d'abord étudié quelques exemples des ICO existantes pour avoir une expérience sur des données EEG en utilisant des bases de données déjà créées. Ensuite, on décide de créer une interface avec deux types de rétroactions, sonores et visuelles, pour voir si les stimuli auditifs peuvent aussi être utilisés dans une ICO. Les méthodes qu'on utilise établissent une ICO sans avoir besoin des données d'entraînement, ni des sujets entraînés. Un sujet est censé choisir un élément à l'aide des stimuli auditifs, visuels et auditifs + visuels. L'étude a porté sur le choix d'une boisson parmi 5 proposées. La classification des données P300 est basée sur la décomposition en ondelettes et la comparaison de moyennes de racines carrées de certaines valeurs choisies qui décrivent les potentiels P300. Ensuite, on discute les résultats de stimuli auditifs et visuels et on les compare pour construire une ICO auditive, pour des sujets qui ont une déficience visuelle.

Enfin, lorsque l'on est en mesure de classifier les potentiels évoqués à l'aide des méthodes de traitement hors ligne sur les données EEG, on approche la question d'un ICO qui peut être utilisé en ligne. On explique le système en détail qui permet d'enregistrer et de traiter les données EEG et d'interpréter les résultats pour l'utilisateur en ligne.

## **ABSTRACT**

Human Computer Interaction (HCI) is one of the most important subjects today in every aspect of our daily lives. The dawn of personal computing in the early 1980's was the beginning of a rapidly evolving technology stranglehold on everyday life. The mechanical world of typewriters, dedicated word processors and adding machines with cranks were quickly left in a wake of microchips. HCI has evolved immensely in late 20 years. Today it is not the only option to use keyboard and mouse, there are very specific ways of interactions based on field of researches. The term usability which is one essential concept in HCI is studied and well developed, today a HCI is expected to be also useful as useable.

From the Punch-Cards to keyboard-mouse-monitor interfaces, it is impossible not to think about HCI in the future. Today, there are lots of researches on real-time interaction systems based on movement detection, tangible objects, gesture recognition etc. Humans can communicate verbally, written, with mimics and gestures etc. and HCI is trying to achieve these levels of communication. But humans lack of transmitting information directly from the brain. Although it is not possible today to create an interface to interact directly with a human's thoughts directly, there are ways to obtain an interaction using ones solely thoughts. When a human has brain activity going on, it can be measured by putting electrodes on the scalp. This is called electroencephalogramme (EEG). Researches made since 70's showed that some distinctive EEG features exist on the recordings and these could be used to obtain an interaction with a computer. Brain computer interface (BCI) is an important research area which aims voluntary communication between human and computers using solely thoughts, this is possible by processing electroencephalography (EEG) data.

In this research, firstly we described some of our works on real-time video based interaction systems. In these works, a user can achieve interaction by using a webcam and as a result auditory and visual feedbacks are created. In BCI part of the project we worked mostly on P300 BCIs which is a type of event related potentials (ERP). A P300 BCI can be built based on the oddball paradigm. It is first used by L. A. Farwell and E. Donchin to build a BCI, a 6 x 6 symbols matrix was presented to subjects and while they focused on one symbol in the matrix, columns and rows were flashed and P300 event related potential (ERP) was present in the EEG data when the focused symbol was flashed. P300 is a specific type of ERP which represents a positive deflection in the EEG signal 300 ms after the stimulus is presented.

In the proceeding sections we aim to build a BCI. First we have a look at some examples of existing BCIs, and have experience on EEG data using existing databases. Then we decide to create an interface with both auditory and visual feedback to see if auditory stimuli can also be used in a BCI. Our BCI should be simple in terms of both collection and classification of EEG data using event related potentials towards a real time BCI system that could be used on line. Our methods should establish a BCI with no need of training data nor trained subjects. As the interface, a subject is wanted to choose a 'need' of his/her using auditory, visual and auditory + visual stimuli. Classification of P300 data is based on wavelet decomposition and comparing root mean squares of certain values that describe P300 potentials. Then we discuss results of auditory and visual stimuli and compare the results toward an auditory BCI for subjects who are also visually impaired.

Finally, being able to classify event related potentials using offline processing methods on the EEG data, we approach to the question of a brain computer interface which can be used on line, and give a detailed explanation of our system that allows recording and processing of EEG data and also interpreting results to the user.

## ÖZET

İnsan-Bilgisayar Etkileşimi (IBE) bugün, günlük hayatımızın içine kadar girmiş en önemli araştırma konularından biridir. 1980'lerin başında kişisel bilgisayar ile günlük hayatımızı tamamen değiştirecek gelişmeler yaşandı. Daktiloların mekanik dünyası yazı işlemcileri, hesap makineleri gibi kavram ve aygıtlar, mikroçiplere doğru hızlı bir evrim geçirdi. IBE de geçtiğimiz 20 yıl içinde çok gelişti ve değişim gösterdi. Çok yaygın olarak kullanılan fare ve klavyeler dışında da çeşitli araştırma sahalarının ilgi alanlarına göre çok çeşitli insan bilgisayar arayüzleri (IBA) oluşturuldu. IBE'de en önemli faktörlerden biri olan kullanılabilirlik, oldukça geliştirilmiş olmasıyla birlikte, gün geçtikçe IBA yararlılık konusuna da önem vermeye başladı.

Punch-kartlardan klavye-fare-monitör üçlülerine geçişi düşündüğümüzde, gelecekte etkileşimin nasıl olacağını düşünmemek elde değil. Bugün dahi, araştırmacılar, hareket algılama, tutulabilir objeler, vücut tanıma gibi bir çok teknik kullanarak yeni IBE'ler gerçekleştiriyorlar. İnsanlar, konuşarak, yazılı olarak, vücut hareketleriyle ve daha bir çok yolla iletişim kurabiliyor, IBE de gün geçtikçe insanın iletişim yeteneklerinin daha fazlasına cevap verebilecek şekilde geliyor. Ancak insanlar, beyinlerindeki bilgileri bir başka insana veya makineye direkt olarak aktaramıyor. Bugün, bunu gerçekleştirecek bir IBE oluşturmak imkansız gibi gözükse de, sadece düşünceleri kullanarak makinelerle iletişim kurabilmenin yolları bulunmuş durumda. Bir insanda beyinsel aktivite olduğu sürece, ki beyinsel aktivite olmaması insanın yaşamadığını gösterir, bu aktivite kafatasına koyulan elektrotlar aracılığı ile ölçülebilmektedir. Bu ölçüme elektro ensefalografi (EEG) adı verilir. 70'li yıllardan beri süregelen araştırmalar gösteriyor ki EEG sinyallerinde zaman zaman ayırd edici özellikler bulunabiliyor, ve bu özellikler sayesinde çeşitli beyin bilgisayar arayüzleri (BBA) gerçekleştirilebiliyor.

Beyin bilgisayar etkileşimi, insanlar ve makineler arasında sadece düşünceleri kullanarak istendiği zaman komut veya mesaj gönderip geri bildirim alabilme amacıyla ilerleyen bir araştırma dalıdır. Bu konsept, genellikle EEG dalgalarının işlenmesiyle gerçekleşmektedir.

Bu çalışmada, öncelikle gerçek zamanlı video işleme yöntemlerini kullanarak çeşitli insan bilgisayar arayüzleri gerçekleştirdik. Bu çalışmalarda kullanıcının sadece webcam kullanarak bilgisayardan sesli ve görsel geribildirim alabilmesini sağladık. Çalışmanın BBA kısmında ise özellikle P300 BBA'lar üzerine yoğunlaştık. P300, bir uyarılmış potansiyel örneği olup, bir çok insanda benzer özellik gösteren bir dalga çeşididir. P300 kullanılarak Oddball paradigmasına dayalı ilk BBA L. A. Farwell and E. Donchin tarafından kullanılmıştır. Bu çalışmada kullanıcılara 6 x 6 bir sembol matrisi üzerinde çeşitli satır ve sütunlar rasgele bir şekilde aydınlatılıp söndürülüyordu. Kullanıcının yoğunlaştığı sembolü içinde bulunduran satır veya sütunlardan biri aydınlatıldıktan sonra EEG sinyalinde uyarılmış potansiyel görülebiliyordu. Bu potansiyel uyarıcının sunulumundan yaklaşık 300ms sonra ortaya çıkıyor ve pozitif yönde bir değişim gösteriyordu.

Bu araştırmamızda hedefimiz bir BBA oluşturmak oldu. Öncelikle çeşitli IBE'leri gözlemledik, daha sonra BBA ve EEG hakkında tecrübe kazanmak için hazırda bulunan bazı datasetleri kullanarak P300 ayıklama algoritmaları oluşturduk. Daha sonra sadece görsel değil, işitsel uyarıların da kullanıcıya sunulacağı bir sistem geliştirdik ve çeşitli kullanıcılardan aldığımız datalar ile işitsel bir BBA oluşturma konusunun üzerine gittik. P300 sınıflandırması wavelet dekompozisyonu ve P300 bileşeninin görülmesi beklendiği alanda bazı değerlerin karekök ortalama (RMS) değerlerinin karşılaştırmasıyla yapıldı. Karmaşık ve bir çok kanal EEG datası gerektiren yöntemlerden çok, tek kanal EEG ile gerçek zamanlı bir BBA elde edebilecek şekilde çalışmalarımıza devam ettik. Çalışmalarımızda gördük ki işitsel BBA oluşturarak, hareket engelli aynı zamanda görme engelli kişilerin de BBA kullanarak iletişime geçmesi için gerekli sistemler oluşturulabilir. Son olarak, veri toplama, veri işleme ve sonuçları kullanıcıya bildirme işlemlerini gerçek zamanlı olarak meydana getirebilen, bir BBA işleyişini ve gerçekleştirilmesini detaylı bir şekilde inceledik.

## **1. INTRODUCTION**

### **1.1. Qu'est ce que L'Interaction Humain-Ordinateur ?**

L'un des aspects les plus importants des êtres humains est la capacité de communiquer. La communication humaine est la plus complexe et la plus développée de toutes les formes de communication. Considérant que les humains peuvent communiquer entre eux de plusieurs manières, comme parler, écrire, regarder, toucher, etc., l'homme devrait être en mesure de communiquer avec les machines aussi. Dans de nombreux domaines de la vie quotidienne, les humains interagissent avec les machines, appareils et ordinateurs. Aujourd'hui, la façon la plus courante d'exploiter un ordinateur est d'utiliser un clavier, une souris et un écran. Ils sont utilisés comme les moyens d'entrée et de sortie des données, mais évidemment ce n'est pas le seul moyen de communiquer avec des ordinateurs.

L'évolution de l'IHO continue toujours dans les aspects de facilité d'utilisation et d'efficacité. Aujourd'hui, un logiciel éditeur de texte est facilement contrôlé avec un clavier par un utilisateur capable, alors qu'il serait très difficile d'utiliser le même logiciel avec des cartes perforées, comme les systèmes anciens. Naturellement, l'écriture avec un clavier n'est encore pas le moyen le plus rapide pour communiquer avec les ordinateurs. L'IHO est un domaine de recherche dans le but d'améliorer la facilité d'utilisation et l'efficacité de la communication entre les humains et les ordinateurs en termes de matériel et de logiciel.

L'interaction humain-ordinateur est un terme utilisé pour désigner la compréhension et la conception des différentes relations entre les êtres humaines et les ordinateurs. Dans la fin des années 70, la préoccupation principale de l'IHO a été de faciliter l'utilisation des ordinateurs. Depuis lors, L'IHO a connu une évolution impressionnante dans le



développement et l'application de toutes sortes de méthodes de conception et d'évaluation pour assurer que les technologies soient faciles à apprendre et facile à utiliser [1]. Ainsi, par exemple, il a produit un volume considérable de données perspicaces ou des preuves sur ce qui est utilisable ou non. Plus récemment, L'IHO a commencé à développer des techniques pour inventer des choses qui ne sont pas seulement 'utilisables', mais aussi 'utile'.

Dans les années précédentes, avec la capacité et la vitesse accrue des systèmes informatiques, des interfaces graphiques ont aussi évolués à bien des égards. Aujourd'hui, les textes, les images, les représentations 3D, et les vidéos sont largement utilisés dans la plupart des applications de la vie quotidienne, comme dans les navigateurs web, les jeux informatiques etc. Dans la partie suivante, on étudie quelques exemples intéressants d'IHO qui ont été source d'inspiration pour les moyens classiques d'interactions d'aujourd'hui et qui continuent à changer la manière dont les humains interagissent avec les machines.

## **1.2. Quelques Exemples d'Interaction Humain-Ordinateur**

### **1.2.1. Quelques Exemples des Matériels d'Entrée**

Comme nous l'avons mentionné précédemment, les claviers et les souris sont les périphériques d'entrée de données les plus couramment utilisés aujourd'hui. Mais IHO a continué à se développer en différents points de vue concernant les besoins de chaque domaine de recherche. Voici quelques exemples.

Une carte perforée (carte Hollerith ou de la carte IBM), est un morceau de papier en carton qui contient des informations numériques représentée par la présence ou l'absence de trous dans des positions prédéfinies. Ils ont été utilisés à travers le 20<sup>ème</sup> siècle dans les machines d'enregistrements unitaires pour la saisie, le traitement et le stockage des données. Premiers ordinateurs numériques sont utilisés par des cartes perforées, souvent préparés avec des machines perforatrices, comme moyen principal pour des programmes informatiques et l'entrée des données.

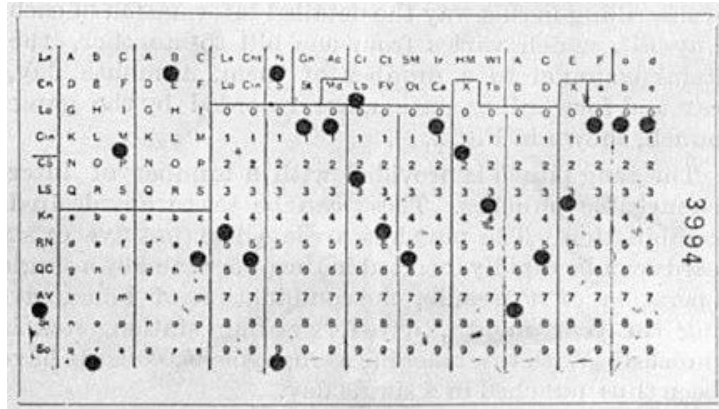


Figure 1.1: Carte perforée d'Hollerith [2].

Un gant de fil est un périphérique d'entrée à gants, pour les environnements de réalité virtuelle. Différentes technologies de capteurs sont utilisées pour la saisie des données physiques telles que la flexion des doigts. Souvent, un suivi de mouvement, comme un dispositif de repérage magnétique ou inertielle dispositif de surveillance, est joint à capturer la position globale des données de rotation / de la boîte à gants. Ces mouvements sont ensuite interprétés par le logiciel qui accompagne le gant, de sorte que tout mouvement, on peut dire un certain nombre de choses. Les gestes peuvent être classés en informations utiles, telles que de reconnaître la langue des signes ou d'autres fonctions symboliques. Quelques gants de hauts de gammes peuvent également fournir des informations 'haptiques', qui simule le sens du toucher. Cela permet un gant filaire également être utilisé comme un périphérique de sortie.



Figure 1.2: Un gant utilisé pour interaction [3].

### 1.2.2. Sketchpad

Sketchpad est un logiciel informatique révolutionnaire créé par Ivan Sutherland en 1963 dans le cadre de sa thèse de doctorat, pour lequel il reçoit le prix Turing en 1988. Il a contribué à changer la façon dont les gens interagissent avec les ordinateurs. Sketchpad est considéré comme l'ancêtre du moderne dessin assisté par ordinateur (CAD) ainsi que d'une percée majeure dans le développement de l'infographie en général. L'interface graphique a été dérivé de Sketchpad ainsi que la programmation orientée objet moderne. Ivan Sutherland a démontré que l'infographie pourrait être utilisée à des fins artistiques et techniques en plus, il a présenté une nouvelle méthode de l'interaction homme-ordinateur.

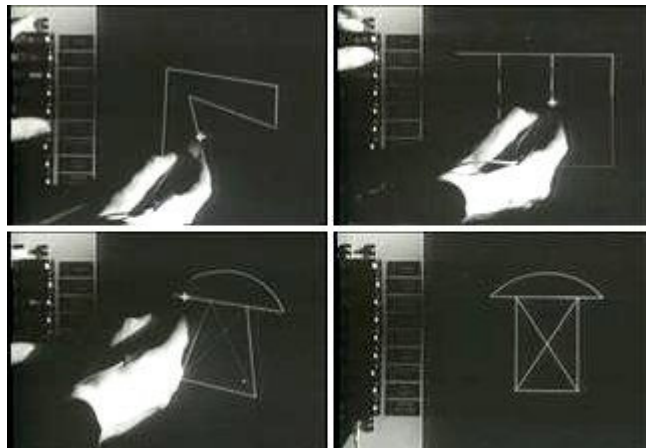


Figure 1.3: Utilisation de Sketchpad avec dispositif de pointage [4].

L'interface de manipulation directe maintenant omniprésente, où les objets visibles à l'écran sont directement manipulés avec un dispositif de pointage, a d'abord été démontrée avec Sketchpad [5]. Sketchpad permet la manipulation d'objets en utilisant un crayon optique, y compris saisir les objets, les déplacer, changer de taille, et utiliser des contraintes. Il contient les germes d'une myriade d'idées d'interface important.

### 1.2.3. Dynabook

Le concept de l'ordinateur portable remonte à l'époque Alan Kay au Xerox PARC. En 1968, lorsque le concept d'un ordinateur nécessitait une salle blanche, les cartes perforées et un montant grand d'électricité, il avait imaginé le Dynabook, un ordinateur portable pesant moins de deux livres [6].

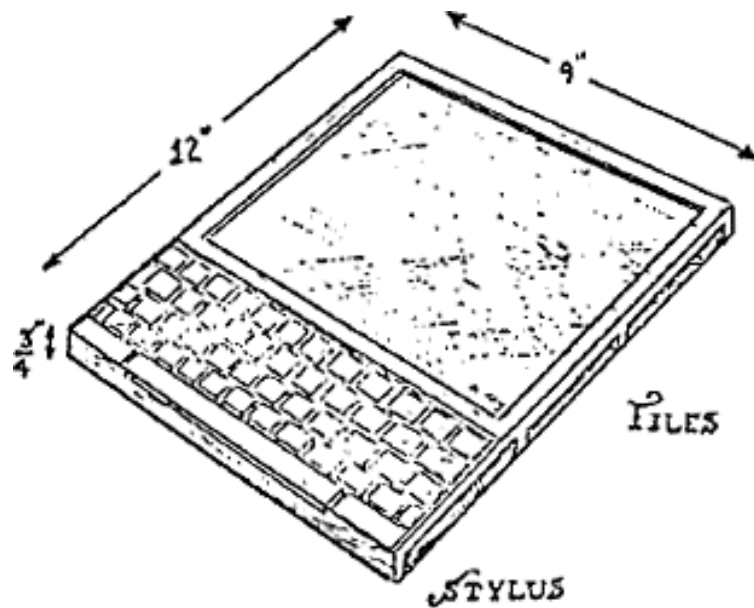


Figure 1.4: Illustration d'un DynaBook. Il ressemble à un Blackberry. Deux fentes sur le côté droit près de l'écran sont marqués 'Files', tandis qu'un slot stylet est sur le côté droit en bas [7].

Le concept de Kay était un ordinateur de 9 "x 12" x 3 / 4 " avec un clavier fixe, de sorte qu'il n'était ni un Tablet PC, ni un ordinateur portable. Le Dynabook aurait un stylet afin que l'utilisateur puisse interagir avec l'écran, ce qui aurait au moins 1 million de pixels. Le concept Dynabook, décrit ce qui est maintenant connu sous le nom d'un ordinateur portable ou (dans certains de ses autres incarnations) un Tablet PC ou ordinateur ardoise avec la vie de la batterie presque éternelle et de logiciels visant à donner aux enfants l'accès aux médias numériques. Les adultes peuvent également utiliser un Dynabook, mais le public cible était les enfants [8].

### 1.2.4. Réalité Augmenté

La notion de réalité augmentée (RA) désigne les systèmes qui rendent possible la superposition d'un modèle virtuel 3D ou 2D à la perception humaine en temps réel. Ce système peut aussi bien s'appliquer à la perception visuelle (superposition d'image virtuelle aux images réelles) qu'aux perceptions proprioceptives comme les perceptions tactiles ou auditives [9]. Réalité augmentée est une variante des environnements virtuels (EV), ou la réalité virtuelle comme il est plus communément appelé. La réalité virtuelle immerge l'utilisateur dans un environnement complètement synthétique. Bien immergé, l'utilisateur ne peut pas voir le monde réel autour de lui. En revanche, RA permet à l'utilisateur de voir le monde réel, avec des objets virtuels ou superposés, formés autour du monde réel. Par conséquent, RA suppléments réalité, plutôt que de le remplacer complètement. Idéalement, il semble à l'utilisateur que les objets réels et virtuels coexistent dans le même espace [10].

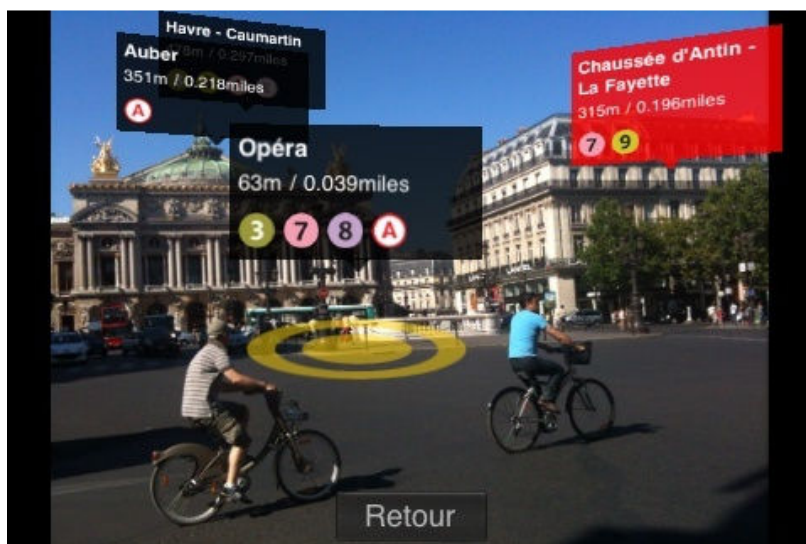


Figure 1.5: Une scène de l'application 'Metro Paris' pour les téléphones portables avec camera [11].

La réalité augmentée explore l'application de l'imagerie générée par ordinateur dans les flux des vidéos comme un moyen d'élargir le monde réel. La recherche avancée inclut l'utilisation de visiocasques et rétines virtuelles, et la construction d'environnements contrôlés contenant un certain nombre de capteurs et d'actionneurs. Mais aussi,

aujourd'hui, la réalité augmentée est utilisée dans de nombreux appareils de poche avec des capteurs vidéo et il ya de nombreux exemples d'applications, comme l'application du métro de Paris, et RA jeux de compagnie virtuel [12].

### 1.2.5. Reactable

Reactable est un exemple intéressant de l'IHM, c'est un instrument de musique électronique avec une interface utilisateur composé d'une table corporelle, qui a été développé au sein du groupe musical de la technologie de l'Universitat Pompeu Fabra de Barcelone, en Espagne par Sergi Jordà, Marcos Alonso, Martin Kaltenbrunner et Günter Geiger. Reactable, est une table ronde translucide, utilisé dans un endroit sombre, et apparaît comme un écran rétro-éclairé. En plaçant des blocs appelés tangibles sur la table, un synthétiseur virtuel modulaire est exploité, créant de la musique ou des effets sonores et 'feedback visuel' est généré en fonction des signaux audio. [13].

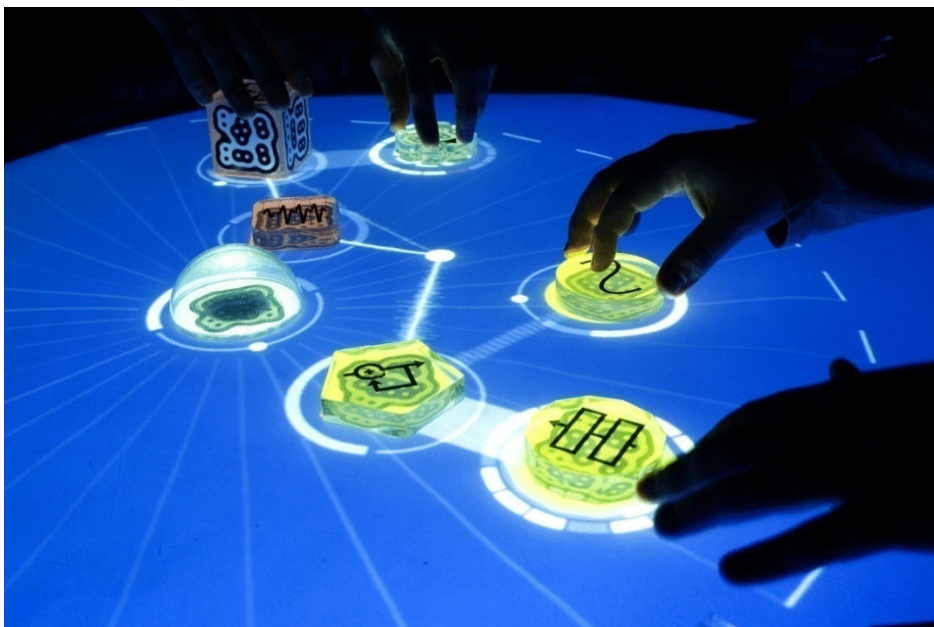


Figure 1.6: Reactable utilisé par plusieurs personnes.

Reactable utilise un logiciel 'reactIVision' comme un moteur de vision, qui est un 'framework' de vision de performance. ReactIVision est utilisé pour le suivi rapide et robuste des marqueurs repères dans un flux vidéo en temps réel. Marqueurs fiduciaires sont des symboles graphiques spécialement conçus, qui permettent l'identification et l'emplacement des objets physiques avec ces symboles ci-joint. Ainsi, les objets tangibles de ces symboles attachés deviennent les données d'entrée pour le système et le son et les éclairages sont obtenus comme sortie.

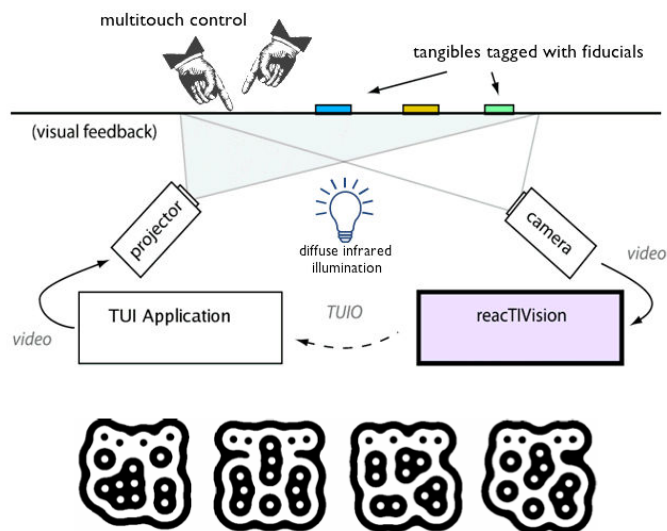


Figure 1.7: Diagramme de framework reactIVision et exemple de quatre marqueurs de la reactIVision d'ensemble "amobea" [14].

### 1.3. Interaction Cerveau-Ordinateur

Une interface cerveau-ordinateur décrit un système de communication en temps réel pour permettre aux utilisateurs d'envoyer des messages ou des commandes volontairement sans utiliser les voies de sortie normal du cerveau. Les messages sont produits en analysant les traits distinctifs dans les signaux EEG. ICO permet aux sujets souffrant de graves maladies neurologiques ou musculaires de communiquer avec des ordinateurs ou d'autres êtres humains [15].

Il est impossible de directement transmettre des pensées, des idées, des émotions entre les humains, mais ceux-ci sont traduits en les déclarations écrites, verbaux, des dessins ou des gestes. Quand on pense sur ce concept, une ICO qui implique une communication directe entre le cerveau et l'ordinateur de l'utilisateur serait à l'interface intuitive ultime qui a également attiré de nombreux films de science [16]. Bien que, loin de l'interprétation directe de la pensée à l'informatique, aujourd'hui, il est possible d'avoir une interaction en utilisant la détection de l'activité électrique du cerveau.

ICO est un sujet qui a été étudié et développé depuis les années 70s, mais il n'est pas très souvent que l'on voit les systèmes ICO utilisé sur la vie quotidienne. L'inconvénient principal des systèmes d'ICO est le taux de transfert d'information pauvre. Quand un dactylographe qualifiée peut écrire plus de 100 mots par minute, les ICO les plus récentes permettent d'envoyer seulement quelques mots de données par minute.

Il existe plusieurs façons de parvenir à une ICO, une des techniques les plus souvent utilisé est le paradigme 'oddball'. La caractéristique la plus distinctive de ce type d'interaction est la composante P300 de PE (Potentiel évoqué). Beaucoup de formes d'onde PE sont caractérisés par une lettre suivie d'un numéro. La lettre indique si la vague est essentiellement positive ou négative au cours de la fenêtre de temps de l'enquête, et les numéros décrivent la latence en signaux. Le paradigme 'oddball' est d'abord utilisé par [17] L.A. Farwell et E. Donchin pour construire une ICO, un matrice de 6 x 6 symboles a été présenté à des sujets et ils se sont concentrés sur un symbole dans la matrice, les colonnes et les lignes ont été flashé (éclairé) et P300 était présent dans les données EEG lorsque le symbole a été flashé. P300 est un PE spécifique qui représente une déviation positive dans le signal EEG qui se produit 300 ms après le moment où le stimulus est présenté. De nombreuses études ont été faites pour augmenter les performances de paradigme Speller P300 et plusieurs méthodes de classification, comme les machines à vecteurs de support, analyse en composantes indépendantes ou l'analyse discriminante linéaire, fourni des résultats efficaces.



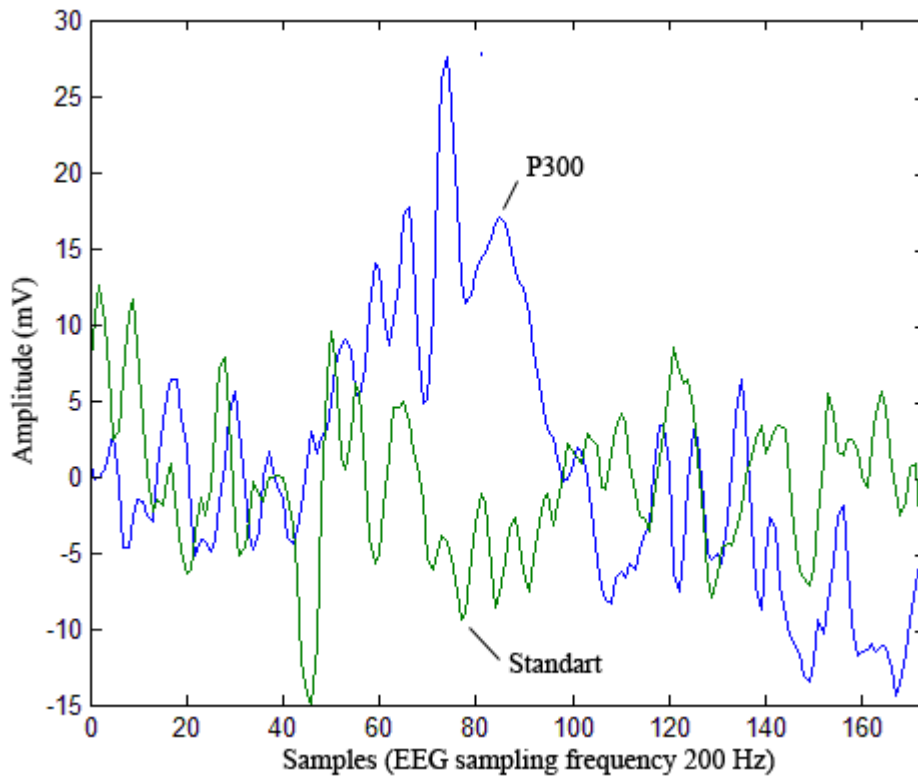


Figure 1.8: Deux signaux EEG : l'un contient la composante P300 et l'autre n'en contient pas.

#### 1.4. Motivation

Les études sur l'interaction homme-machine a été un sujet intéressant et très développé depuis 50 dernières années. Il existe beaucoup de projets qui visent interfaces de plus en plus utilisable et utile entre l'homme et la machine. Systèmes d'interaction en temps réel qui sont en cours d'amélioration dans tous les points de vue sont inévitables pour la facilité d'utilisation et d'efficacité et peuvent créer des environnements créatifs.

Mesure de signaux EEG a également permis à l'homme d'envoyer des messages via des ordinateurs ou des appareils électroniques en utilisant uniquement des pensées. Une interface où les idées dans le cerveau sont directement représentées aux autres êtres humains ou des machines, la communication par la pensée, est une idée qui est imaginé dans de nombreux films de science fiction. Aujourd'hui, une interface qui donne une représentation directe de la pensée semble être loin, mais il existe des moyens de communiquer par la pensée. Même si il n'est pas possible de penser à un mot, par

exemple le mot 'moutons', et présenter le texte sur l'écran ou, voir une image des moutons, il est possible de préciser le mot 'moutons' ou de choisir un l'image des 'mouton' parmi les autres est possible via les ICO.

Dans ce projet, nous allons essayer de créer et montrer plusieurs exemples et des applications de l'IHO. Tout d'abord, nous présentons nos travaux sur les IHO utilisant vidéo en temps réel, et expliquer comment il est appliqué pour créer un environnement d'interaction créant les retours sonores et visuelles pour les utilisateurs. Puis, notre objectif sera de comprendre les concepts d'ondes EEG distinctifs, et les utiliser pour construire un environnement d'interaction entre le cerveau et l'ordinateur. Pour cela, nous examinerons certains de nos travaux sur les bases de données existantes ICO et d'expliquer nos méthodes et nos expériences vers une ICO et de les comparer aux recherches précédentes. Enfin, notre objectif sera la construction d'une interface Cerveau - Ordinateur qui peut être utilisé en ligne.

## **2. LES APPLICATIONS RÉALISÉES DE L'INTERACTION HUMAIN – ORDINATEUR**

Dans ce chapitre, on expliquera quelques méthodes nouvelles sur interaction humain-ordinateur. On a travaillé sur l'environnement 'PureData' qui est un environnement de programmation visuelle qui est un exemple intéressant de l'IHO lui-même [18]. Dans la section suivante, on donne une explication brève de PureData et montre quelques exemples de programmation visuelle pour expliquer comment il fonctionne. Puis, on aborde sur les travaux sur IHO en utilisant la vidéo en temps réel pour générer des signaux auditifs. On essaie de trouver de nouveaux moyens d'interaction avec le thème 'production de la musique' comme un exemple, bien que ces techniques puissent être utilisées dans d'autres manières de traitement des données.

### **2.1. Informations Générales sur PureData**

PureData (ou abrégé pd) est un environnement de programmation à base graphique pour la création et traitement de l'audio et multimédia en temps réel. Autour de PureData, on peut travailler sur des sujets programmation, mathématiques, audionumérique, physique, arts et plusieurs branches d'intérêt modification temps réel car PureData est un logiciel qui peut être utilisé en interaction avec le monde réel. Puredata a l'origine le logiciel 'Patcher' qui a été premièrement écrit par Miller Puckette en 1988. Patcher a été réécrit par David Zicarelli sous le nouveau nom de Max/MSP. Après, la conception de Patcher a été repris par Miller Puckette pour faire un logiciel nouveau dans le but d'explorer les idées et raffiner Max et le rendre transportable à des fins audio-visuelles en temps réel. Un point très important en utilisation PureData est qu'il offre 2 modes de fonctionnement. 'Edit Mode' (mode pour éditer), 'Run Mode' (mode pour marcher). On fait la programmation qui consiste

en assemblage, connexion et organisation des objets en ‘Edit Mode’. On utilise le document, en ‘Run Mode’ qui permet l’utilisation en temps réel.

Voici un exemple très simple de programmation en PureData. On se met en ‘Edit Mode’ en le choisissant dans le menu Edit de patch. Maintenant on peut choisir et mettre les objets que l’on veut utiliser dans le patch. Pour cet exemple on met un ‘bang’, un ‘message’ et un ‘object’ dans le patch, puis, on peut écrire ‘hello world’ dans le ‘message’ et ‘print’ dans ‘object’. Finalement on réalise les liaisons en utilisant les entrées et sorties qui sont des petits rectangles en noirs en haut et bas des objets. Ces entrées et sorties s’appellent ‘inlet’ et ‘outlet’. Le patch est prêt à utiliser quand on se met en ‘Run Mode’. La différence de ‘Run Mode’ est que l’utilisateur peut cliquer sur les messages, les objets IGU (Interface graphique utilisateur) comme ‘bang’ et ‘toggle’, et il peut modifier les nombres en utilisant le souris. Ici on clique sur ‘bang’ qui crée un déclenchement et faire le message ‘hello world’ passer à ‘object’ a la quelle elle est liée. On peut observer le résultat sur la console PureData.

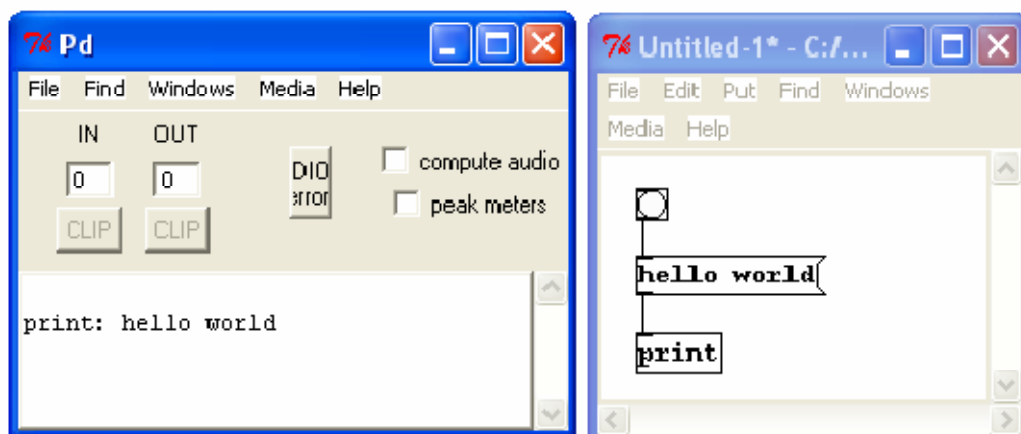


Figure 2.1: Exemple ‘Hello World’.

Maintenant, on va essayer d’implémenter un algorithme simple pour trouver des nombres, juste pour mieux comprendre le concept de travail de PureData.

L'algorithme qu'on va utiliser pour trouver des nombres premiers :

```

int max, i, j, y ;
input max ;
  for (i = 2 à max)
    {
      for (j = 2 à (max-1))
        {
          y = i % j ;
          si y = 0 arret;
        }
      si j = (max - 1) print i;
    }

```

Cet algorithme accepte un nombre 'max' comme entrée pour trouver tous les nombres premiers qui sont plus petits que 'max'. Afin de trouver les nombres premiers, l'algorithme vérifie si le nombre 'i' est divisible par 'j', si oui il arrête de tester et passe au nombre suivant qui sera 'i + 1', sinon alors il n'arrête pas jusqu'au j obtient la valeur 'max-1'. Alors si 'j' est égale à 'max - 1', il comprend que le moduleau 'i % j' n'est jamais 0 alors 'i' n'est pas divisible par les nombres 2, 3, ..., i -1, donc 'i' est un nombre premier et il l'affiche à l'écran.

Si on examine l'implémentation de cette algorithme en PureData, pour réaliser les boucles 'for' on utilise un objet qui s'appelle 'until' qui crée des bangs successives de nombre qu'il reçoit à 'inlet' gauche, et il arrête s'il reçoit un bang à 'inlet' gauche. Sur la figure on voit les implémentations des parties de l'algorithme. L'utilisation de ce patch se fait en 'Run Mode'. On entre la valeur de nombre 'max' a la boite de nombre en haut et sur la console PureData on obtient l'affichage de tous les nombres premiers plus petits que 'max'.

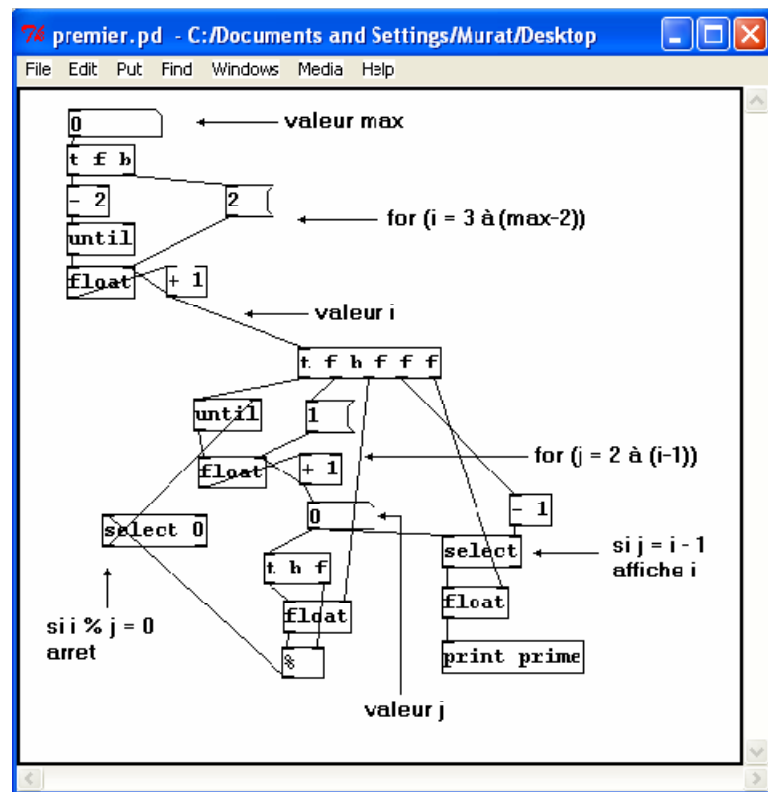


Figure 2.2: Patch nombres premiers.

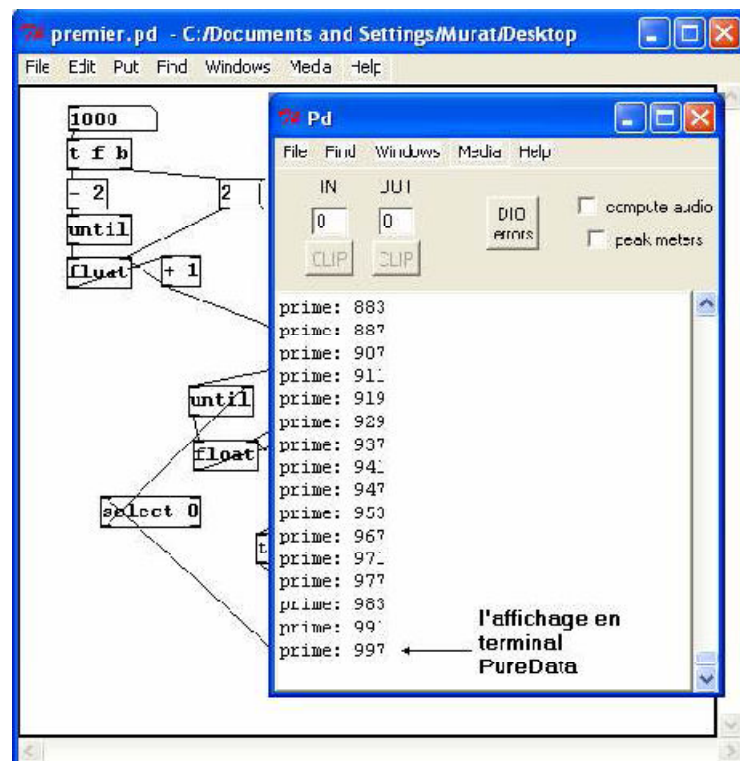


Figure 2.3: L'affichage des nombres premiers obtenus.

Jusqu'à cette partie de ce chapitre, l'utilisation et programmation des algorithmes classiques en PureData ont été expliqués. Pour être capable de créer l'environnement musical, on doit travailler sur des outils de traitement de signal de PureData. La grande partie sera au propos des objets 'tildes' de PureData. Ce sont des 'objects' qui ont une relation aux signaux et leur nom finissent avec le caractère tilde (~). L'objet qu'on va utiliser dans ce patch est 'osc~' qui est un générateur des oscillations sinusoïdaux. Cet objet possède deux 'inlet' pour régler la fréquence et la phase et un 'outlet' où le signal sinusoïdal qui oscille entre les valeurs '-1' et '1' est obtenu. PureData offre un autre oscillateur qui fonctionne de même façon qui s'appelle 'phasor~'. L'oscillation obtenue par 'phasor~' est de type scie et l'oscillation est entre '0' et '1'. On va visualiser et étudier les signaux générés par ces deux types d'oscillateurs en utilisant les 'arrays' dans un patch.

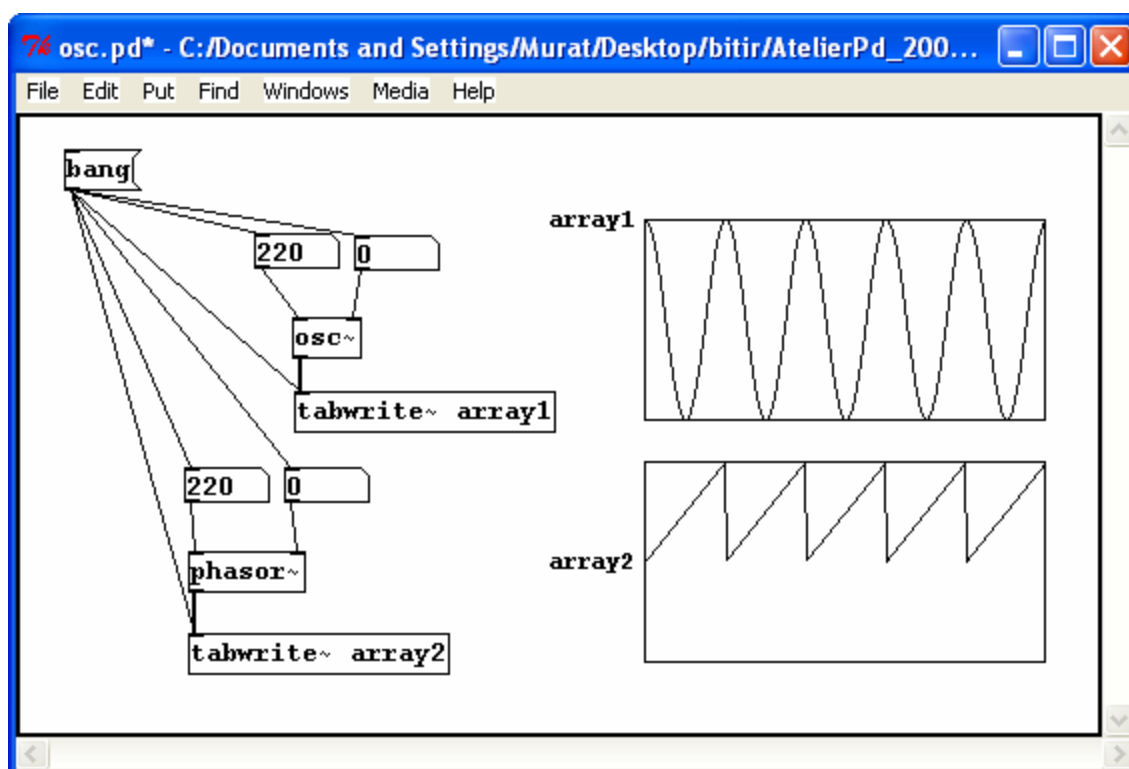


Figure 2.4: Les signaux de 'osc~' et de 'phasor~'.

L'objet 'tabwrite~' dessine le signal qu'il obtient quand il arrive un 'bang'. Dans le patch, le message 'bang' est envoyé aux nombres qui fixent les fréquences et phases des signaux, puis l'objet 'tabwrite~' prend le nom d'array comme argument. Pour convertir les signaux de PureData aux signaux sonores on utilise l'objet 'dac~' qui fonctionne comme un convertisseur digitale/analogue. On va construire un patch qui sonne en fonction de la fréquence d'oscillation de l'objet 'osc~'. Afin de contrôler l'amplitude de signal de sortie audio, on va utiliser un objet de 'math audio' que PureData offre, '\*~', qui calcule la multiplication d'un signal par un nombre. Si ce nombre est '1', 'osc~' oscille de '-1' à '1', si le nombre est '0.5', 'osc~' oscille de '-0.5' à '0.5'. Ces objets de 'math audio' sont utiles pour les calculs de calibration ou modification des signaux.

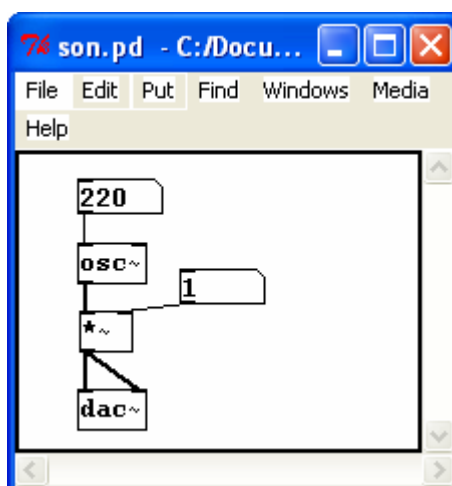


Figure 2.5: Patch sonore.

Le patch en Figure 2.5 illustre l'utilisation de 'osc~' comme un générateur de son. On peut faire des petites modifications pour mieux utilisation. L'objet 'mtof' nous permet de créer des fréquences qui correspondent aux notes musicales. L'objet 'mtof' fonctionne comme un convertisseur de midi à fréquence. Par exemple la fréquence 440 Hz est égale à 69 comme valeur de midi. Les notes musicales utilisés aujourd'hui sont marqués de type: A, A#, B, C, C#, D, D#, E, F, F#, G, G#. 'A' signifie la fréquence 440 Hz par convention. Une octave musicale contient ces 12 notes. Si la fréquence pour une note qui est en n<sup>ème</sup> octave est 'f', la fréquence de ce note en n+1<sup>ème</sup> octave est 'f\*2'. D'où les fréquences 55, 110, 220, 440, 880... génèrent la note musicale 'A' dans



différents octaves. Après ces informations générales sur la théorie de musique, on peut construire un patch qui génère les sons de notes musicales conventionnels d'aujourd'hui.

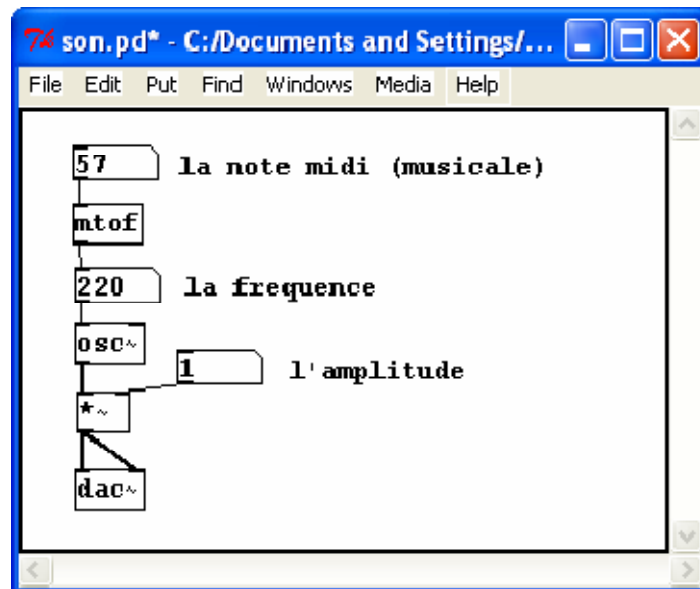


Figure 2.6: Patch sonore avec contrôle de notes.

## 2.2. Contrôleur Piano

Dans cette partie du travail, on utilise PureData pour construire une interface homme-machine qui permet à un utilisateur de jouer des notes de musique en utilisant une webcam. On utilise utiliser la version 0.40.3 Pd-extended et GEM (Graphics Environment for Multimedia) [19] Ver: 0.91.1 qui est une bibliothèque pour le traitement vidéo dans PureData. Un Webcam A4tech PC caméra a été utilisée dans ce projet, l'objet 'pix\_video' dans la bibliothèque du GEM est utilisé pour récupérer le flux vidéo de la caméra connectée à l'ordinateur. GEM est un logiciel efficace pour l'utilisation du traitement vidéo en temps réel. Il fournit des éléments de base du traitement de la vidéo comme un seuil, le masquage, le filtrage, etc.

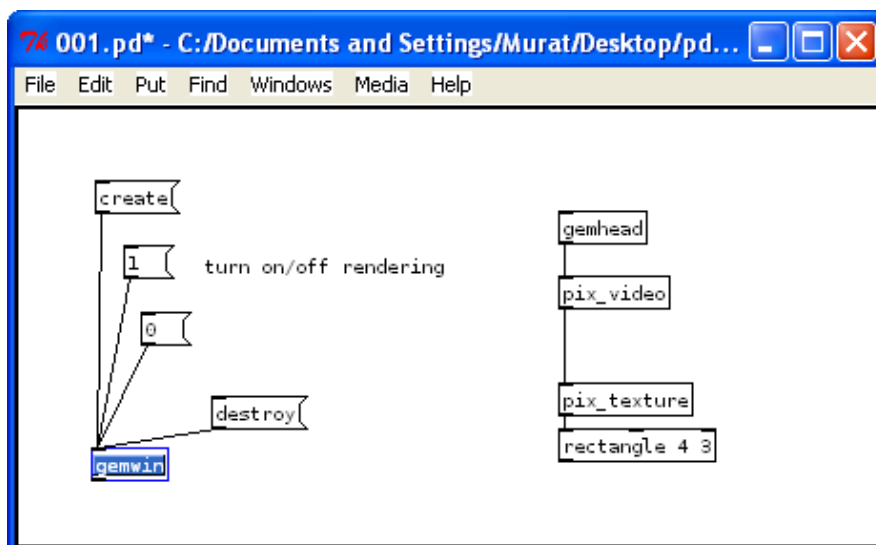


Figure 2.7: Utilisation de la bibliothèque de GEM dans PureData pour récupérer des données à partir d'une webcam.

Dans la figure 2.7, l'exemple le plus simple d'utiliser GEM pour obtenir un flux de vidéo est affiché. L'objet 'gemwin' crée la fenêtre dans laquelle la vidéo sera affichée et le message '1' commence le flux de données. L'objet 'gemhead' est nécessaire pour chaque cas dans la fenêtre de GEM. Les connexions 'gemhead-> pix\_video-> pix\_texture-> rectangle 4 3' indiquent que l'entrée vidéo sera utilisée comme la texture d'un rectangle de taille 4x3 unités.

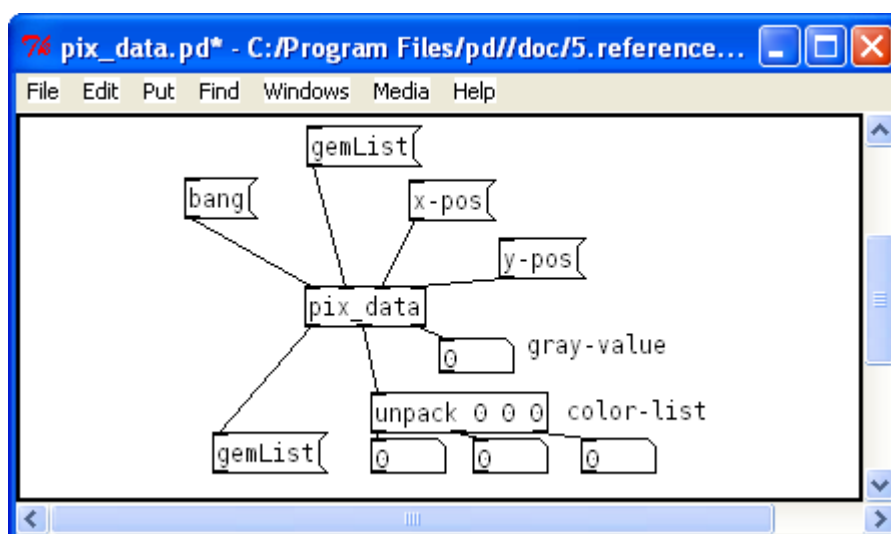


Figure 2.8: Explication de fonctionnement de l'objet 'pix\_data'.

La figure 2.8 montre le fonctionnement de l'objet 'pix\_data'. En utilisant l'objet 'pix\_data', on peut obtenir certaines informations d'un pixel dans la fenêtre GEM. Cet objet sort les valeurs de 'rouge, vert, bleu' et aussi la valeur de gris d'un pixel. Les coordonnées de ce pixel sont sélectionnées par les valeurs d'entrées 'x-pos' et 'y-pos'. Dans une fenêtre GEM les coordonnées des pixels ont des valeurs entre '0,0' (bas-gauche) et '1,1' (haut-droit). Par exemple dans une fenêtre de '500x500' pixels, si on veut obtenir des informations sur le pixel (200,150), les valeurs 'x-pos' et 'y-pos' doivent être (0.4 , 0.3).

### *Création de Performance Musicale*

#### *karecik.pd:*

Dans ce projet, l'objectif premier était de créer des objets carrés, qui devrait être visible par l'utilisateur. Ces carrés permettraient le déclenchement des notes de musique en détectant les changements dans les valeurs des pixels sur lesquels ils se trouvent. Pour cet objectif, on crée un patch appelé 'karecik' qui construit un carré dans n'importe quelle partie de la fenêtre, dans n'importe quelle couleur, la taille et la transparence.

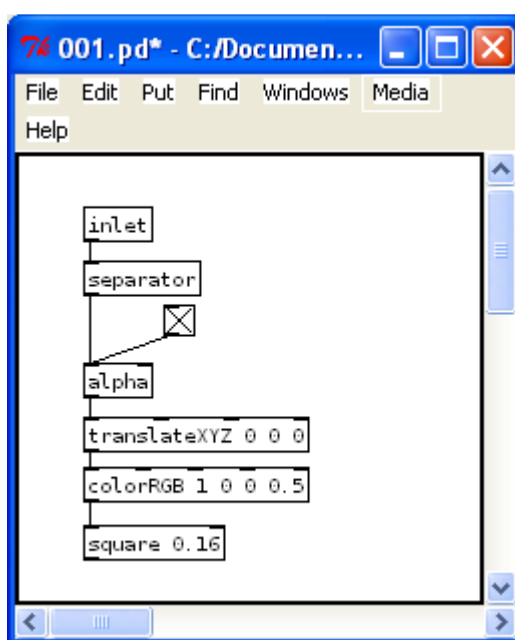


Figure 2.9: Création de l'objet 'karecik' en PureData.

Dans la Figure 2.9, on peut voir le patch pour la création d'un carré de la taille de '0,16' en couleurs (1,0,0) et la transparence de '0,5'. L'objet 'separator' permet de créer le carré sur l'écran séparément de l'entrée vidéo. L'objet 'alpha' définit la fonctionnalité de transparence. L'objet 'translateXYZ' est utilisé pour déplacement du carré qu'on a créé sur une partie quelconque de l'écran. Le carré crée dans l'exemple est créé avec 'translateXYZ 0 0 0' ce qui signifie le centre de l'écran. Par exemple, dans une fenêtre GEM de 500x500 pixels, un carré créé avec l'objet 'translateXYZ 4 4 0' se trouve en haut à droite de la fenêtre. Cela signifie que, une unité de cette traduction est égale à 62,5 pixels. C'est aussi la même unité dans l'argument de création 'square 0.16' qui aboutit à une longueur de  $0.16 \times 62.5 = 10$  pixels. Cette valeur représente la longueur entre le point central et un côté du carré donc à la fin on obtient un carré de 20x20 pixels. L'objet 'colorRGB' prend 4 arguments, rouge, vert, bleu et la valeur alpha. Ces valeurs devraient varier entre 0 et 1, ils définissent les valeurs de rouge, vert bleu et transparence du carré créé.

Avec les possibilités que nous offre GEM, on peut facilement détecter un changement d'une valeur de rouge, vert, bleu ou gris d'un pixel donné. Cependant, on veut détecter un changement des valeurs dans le carré qu'on a créé. Alors on analyse 8 points; 4 coins et 4 points milieu des côtés du carré pour un changement de valeur de gris. Les meilleurs résultats dans ce projet sont obtenus avec un fond blanc et d'objets sombres comme des gants noirs. Déclenchement d'une note de musique est basée sur la valeur gris lit de l'objet 'pix\_data'. D'un fond blanc, cette valeur est d'environ de '0,8' à '0,9' alors que, il génère de '0,1' à '0,2' au même pixel, lors de sa présentation des gants noirs.

Dans la figure, on peut voir une partie de l'objet 'karecik' en travail. 8 objets 'pix\_data' écrit les résultats sur les objets 'nombres' qui affichent ces valeurs et ces chiffres sont comparés à une certaine valeur (qui est de '0.4' dans notre cas). Si l'une des valeurs sont inférieures à '0.4', l'objet '<' génère 1 en sortie et '8or' patch qu'on a créé génère résultat '1' aussi. Ainsi, les 8 points du carré sont vérifiés pour être touché par le gant noir de l'utilisateur.

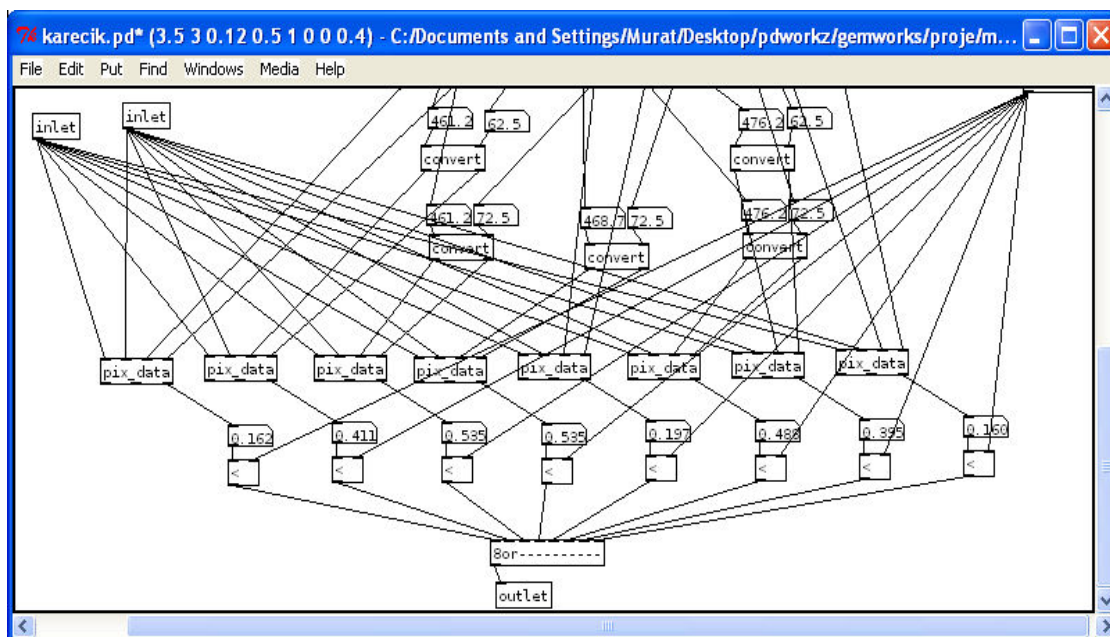


Figure 2.10: Prise de données de 8 points de l'objet 'karecik'.

### ***Faders.pd:***

Cet objet est utilisé comme un fader (potence linéaire) pour contrôler n'importe quel paramètre d'un synthétiseur musical. Le fonctionnement est similaire à l'objet 'karecik' mais cette fois 15 points, qui sont sur une ligne verticale, sont examinés pour les changements des valeurs de gris. Dans ce projet, cet objet est utilisé pour contrôler le volume principal. Cet objet a une longueur de 150 pixels et génère des valeurs allant de 0 à 140 (avec des intervalles de 10). Un autre petit carré est utilisé pour donner une rétroaction visuelle à l'utilisateur pour montrer l'endroit où le volume est. Si on appelle la valeur générée par 'faders' comme 'x', le lieu de la petite place se traduit à la verticale avec la valeur de  $(x-70) * 0,016$ .

### ***Kareyazi.pd:***

Cet objet s'accorde avec les mêmes propriétés 'karecik' mais aussi il peut afficher un mot désiré. Dans ce projet il ya des objets 'kareyazi' qui affichent, *up*, *down* (pour la sélection d'octave), *saw*, *sine*, *pulse* et *square* (pour la sélection de forme d'onde de synthé)

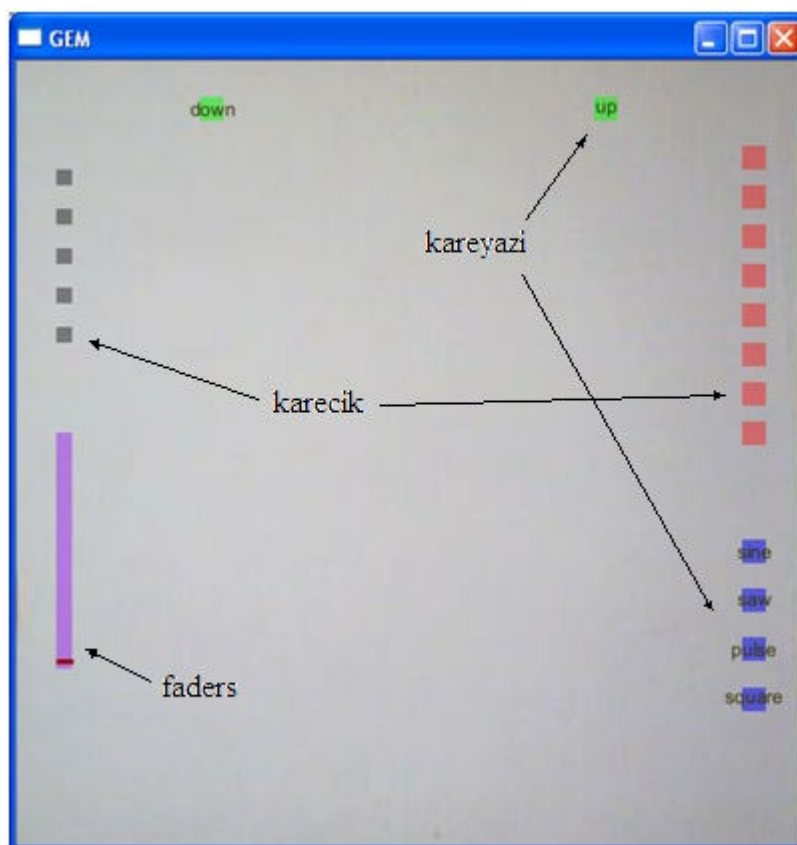


Figure 2.11: L'affichage des objets 'karecik', 'kareyazi' et 'faders' superposés sur le flux vidéo de la webcam.

### **Utilisation**

On commence le patch et choisit 'Audio ON' sur l'onglet média de PureData. Lorsque les messages 'start' et '1' sont envoyés à l'objet 'gemwin', le feedback visuel est créé. Les meilleurs résultats sont obtenus avec un fond blanc et des gants noirs. La fenêtre GEM qu'on obtient aura une interface comme ceci:

Les carrés rouges sur le côté droit de l'écran représentent les notes blanches comme 'do, ré, mi ...' et les points noirs sur le côté gauche de l'écran représentent les notes noires comme 'do #, ré #, fa #, sol # et la #'. Les carrés verts sur le dessus sont utilisés pour changer l'octave de notes de haut en bas. Les boîtes bleues sont utilisées pour choisir la forme d'onde du son synthétisé et le rectangle mauve sur le côté en bas à gauche est utilisé pour contrôler le volume de sortie.

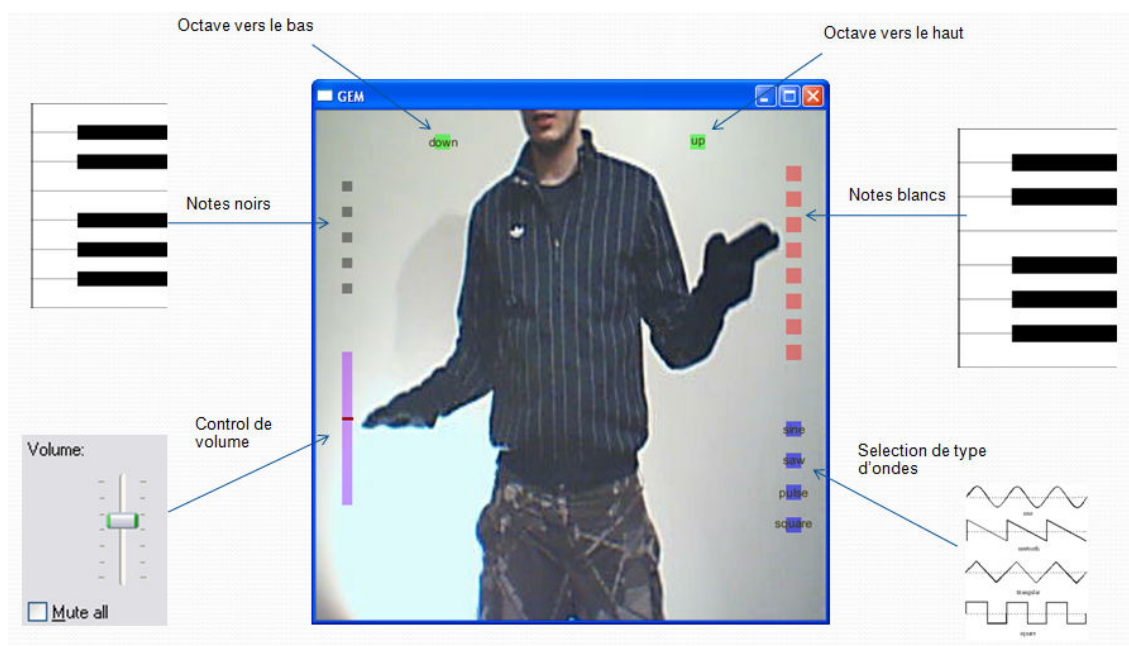


Figure 2.12: Explication de l'utilisation de Contrôleur Piano.

L'utilisateur se trouve près de 1,5 - 2 mètres loin de la caméra et utilise ses mains pour faire fonctionner le système. De cette façon, on obtient une interaction en temps réel qui permet le traitement vidéo en temps réel qui génère des feedback audio à l'utilisateur. Comme dans l'exemple 'faders', un retour visuel est aussi représenté à l'utilisateur. Cet exemple est également utilisé pour contrôler d'autres appareils numériques. Par exemple, en reliant les données de notes déclenchés à la sortie MIDI (Musical Instrument Digital Interface) [20] de PureData, on peut aussi contrôler les synthétiseurs de logiciels de production music 'Reason' [21] en utilisant l'outil MidiYoke [22] qui sert à faire des liaisons entre différentes logiciels qui utilisent MIDI.



Figure 2.13: Contrôleur Piano connecté au logiciel Reason sur MidiYoke.

Dans ce projet, seulement la valeur de gris de pixels sont utilisés pour détecter les mouvements de l'utilisateur, tandis que de nouveaux objets autres que 'karecik', 'faders' et 'kareyazi' pour comprendre changements en valeurs de rouge, vert et bleu des pixels pourrait être construit afin d'obtenir de meilleures performances et la convivialité. Dans la section suivante, on va également utiliser la bibliothèque de GEM PureData avec une webcam mais cette fois on utilise trois objets de différentes couleurs pour contrôler plusieurs paramètres musicaux sur un plan XY.



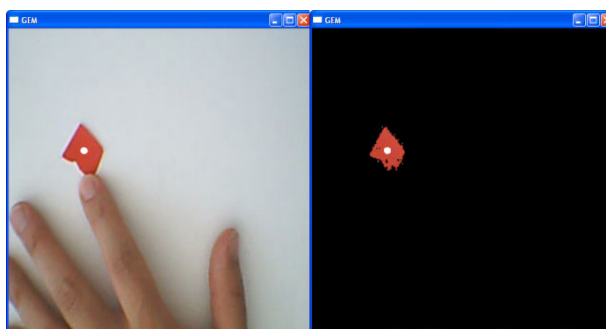
### 2.3. Contrôleur à Blobs en Utilisant Le Filtrage des Couleurs

Dans cette partie du projet, on présente un contrôleur vidéo en temps réel qu'on utilise pour le traitement audio en vue des résultats créatifs et utilisable. Une webcam est utilisée pour suivre la position des objets de différentes couleurs. Seuillage couleur sur un espace de couleur RVB (Rouge, Vert et Bleu) est utilisé pour distinguer les objets par leurs couleurs et la 'détection blob' est utilisée pour le suivi de mouvement de ces objets de couleur différente. Webcam est placée à 20 cm sur une feuille blanche blanc et commentaires audio PureData est programmé que 6 paramètres peuvent être commandés séparément par 3 morceaux de papier coloré de taille environ 1cm<sup>2</sup>.

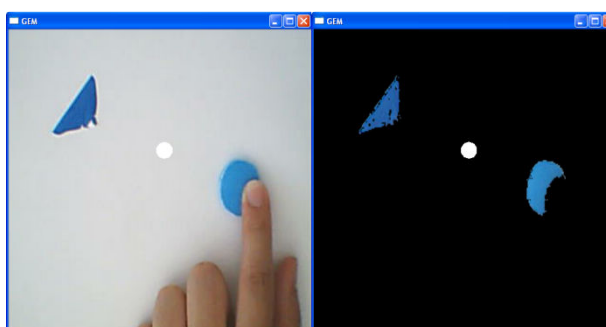


Figure 2.14: Configuration du système, webcam pour l'entrée visuelle, fond blanc, des objets colorés, PureData pour la création de la rétroaction auditive.

On utilise l'algorithme de détection de blob avec l'objet 'pix\_blob' de GEM sur 3 canaux différents 'Rouge', 'Vert' et 'Bleu'. Cet algorithme de détection de blob analyse l'image et renvoie les coordonnées d'une seule goutte sur une image, la position des taches est calculé en fonction de la gravité de centre de valeurs 'alpha' d'une image. Les valeurs 'alpha' sont calculées comme la moyenne des valeurs de 'rouge' 'vert' et 'bleu'. La figure 2 montre des taches trouvées sur un objet rouge et deux objets de couleur bleue.



(a)

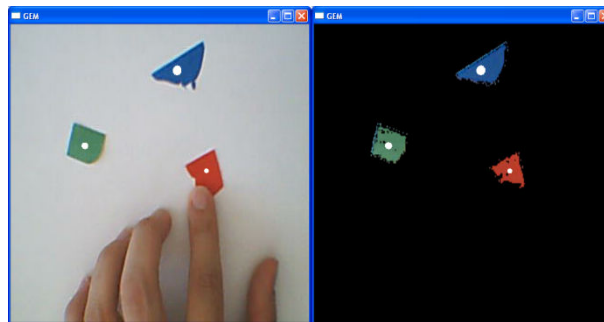


(b)

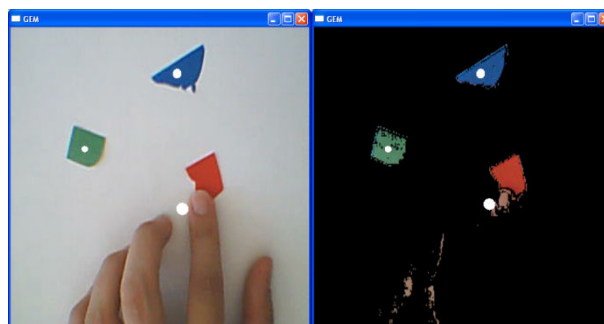
Figure 2.15: Blobs comme des sphères blanches indiquant le centre de gravité de l'objet coloré. (a) objet unique, le blob est sur l'objet. (b) Le blob est entre deux objets de couleur bleue.

Pixels d'une image RVB dans GEM peuvent avoir des valeurs entre  $[0\ 0\ 0]$  (noir) et  $[1\ 1\ 1]$  (blanc), alors, sur un fond noir, la position d'un objet coloré de nous donnerait la bonne position pour un blob alors que, sur un fond blanc, si on essaie de trouver des grottes à l'aide d'un objet rouge, on ne peut pas avoir des résultats précis, dans un blob fait les coordonnées ne changerait guère. C'est parce que tous les pixels blancs auraient une valeur alpha de '1' et une petite zone rouge avec une valeur moyenne d'alpha de '0,333' n'affecterait pas le centre de gravité. De plus, un autre problème, c'est que lorsqu'on a les informations de pixels d'une zone, il ne suffit pas pour obtenir les informations du canal rouge pour détecter rouge, parce que tous les pixels blancs ont la valeur '1' comme valeur du canal rouge. Donc, si on veut obtenir seulement les pixels rouges d'une image, il faut vérifier les valeurs des trois canaux et de distinguer seulement  $[1\ 0\ 0]$  les valeurs en rouge. Pour des situations réelles, si on veut avoir des zones principalement rouges d'une image, on utilise des pixels où les valeurs de 'R' (rouge) sont comprises entre '0,75' et '1', et 'V' (vert) et les valeurs 'B' (bleue) entre '0' et '0,4' en raison des problèmes de l'éclairage et de bruit du signal. Cela se fait par

objet 'pix\_alpha' de GEM qui fixe des valeurs alpha des pixels ayant des valeurs entre  $[0,75 \ 0 \ 0]$  et  $[1 \ 0,4 \ 0,4]$  à '1' et les autres à zéro. Cette opération de filtrage sépare l'objet rouge de fond et de la main de l'utilisateur. En utilisant l'objet 'pix\_alpha' avec les valeurs de  $[0,75 \ 0 \ 0]$  et  $[1 \ 0,4 \ 0,4]$  pour l'extraction de rouge, de  $[0 \ 0,75 \ 0]$  et  $[0,4 \ 1 \ 0,4]$  pour l'extraction de vert et de  $[0 \ 0 \ 0,75]$  et  $[0,4 \ 0,4 \ 1]$  pour l'extraction de bleu (ces valeurs peuvent changer en vertu des conditions d'éclairage différentes), on obtient des 3 blobs séparés et afficher une sphère blanche comme un blob sur l'image pour en fonction des coordonnées, la taille de la sphère est ajusté proportionnellement à la taille de chaque objet. Si les valeurs 'RVB' qui sont filtrés par objet 'pix\_alpha' ne sont pas correctement ajustés, la main de l'utilisateur peuvent interférer avec les canaux résultant positions trompeuse de taches.



(a)



(b)

Figure 2.16: Blobs trouvés pour les objets rouges, verts et bleus. (a) Filtrage des couleurs et des valeurs ajustées. (b) Intervention du mouvement de la main sur le canal rouge, position de blob trompeuse.

Afin de surmonter les problèmes d'interférences et d'avoir un meilleur filtrage des couleurs, on a développé un système d'étalonnage qui obtient les valeurs RVB des objets et des filtres usagés toutes les autres couleurs. Ainsi, l'algorithme de détection de blob fonctionne correctement sous différentes conditions de l'éclairage. Par exemple, si l'objet rouge a une valeur de RVB (0,75, 0,35, 0,24) pendant l'instant de l'étalonnage, mesurée avec l'objet 'pix\_blob', le blob est créé en fonction de filtrage de l'espace de couleur entre (0,65, 0,25, 0,14) et (0,85, 0,45, 0,34).

Jusqu'à cette partie du travail, on obtient une contribution de 6 valeurs (x et y les valeurs des positions des 3 blobs) que, dans cette section, on va utiliser pour le contrôle de traitement des signaux audio. Cette partie de l'étude est très ouverte et peut compter sur la créativité de personne, mais on donne un effet DSP et une technique de synthèse sonore à titre d'exemples.

Un des effets les plus populaires sur la musique d'aujourd'hui est appelé 'Beat Repeat' qui est surtout utilisée par les compositeurs de musique électronique, et surtout dans le spectacle vivant. Un effet de 'Beat Repeat' boucle une petite partie du signal de façon continue en fonction de la BPM (battements par minute) de la musique. Par exemple, sur un morceau de musique qui a une signature '4 / 4' temps et 120 bpm, il est de 500 millisecondes de temps entre deux battements, notre effet battu le répète en boucle une partie de 500 millisecondes lorsqu'il est activé. Comme on peut avoir 2 paramètres à contrôler avec un objet, par les valeurs de l'axe 'x' (horizontal), on assigne le temps de la boucle, pour un son BPM 120 on peut avoir '62.5ms, 125ms, 250ms, 500ms, 1000ms...' Alors les répétitions seraient harmonieuses sur le tempo, et par les valeurs de l'axe 'y' (vertical), on contrôle la fréquence d'un filtre passe-bande. On vérifie aussi la moyenne des valeurs alpha de l'image pour vérifier si l'objet existe dans le cadre, sinon cette valeur est 0 et le signal audio n'est pas affecté, au moment où on montre l'objet de couleur à l'appareil, l'effet 'Beat Repeat' est activé. Il s'agit d'une mise en œuvre d'une vidéo contrôlée, réplique de coût diminué d'un effet populaire d'un appareil commercial des effets tactiles qui s'appelle Korg Kaosspad [23]. Effet 'Beat Repeat' dans PureData peut être obtenue par une modification d'un effet de délai audio qui peut être construit en utilisant les objets '~delread' et '~delwrite'. L'objet '~delwrite' écrit un signal audio à une ligne de retard et l'objet '~delread' lit le signal retardé, si le retard du signal de

'~delread' est réacheminé dans l'objet '~delwrite', on peut avoir une commande de rétroaction qui est une opération en multipliant par une valeur entre '0' et '1'. Pour obtenir un effet 'Beat Repeat', lorsqu'il est activé, on met la valeur de 'feedback' à '1' et coupe l'entrée principale audio en multipliant avec '0', la seule chose qu'on entend, c'est l'effet de boucle, et quand on désactive, on reconnecte l'entrée principale audio en multipliant avec '1' et on met la valeur de 'feedback' à '0'.

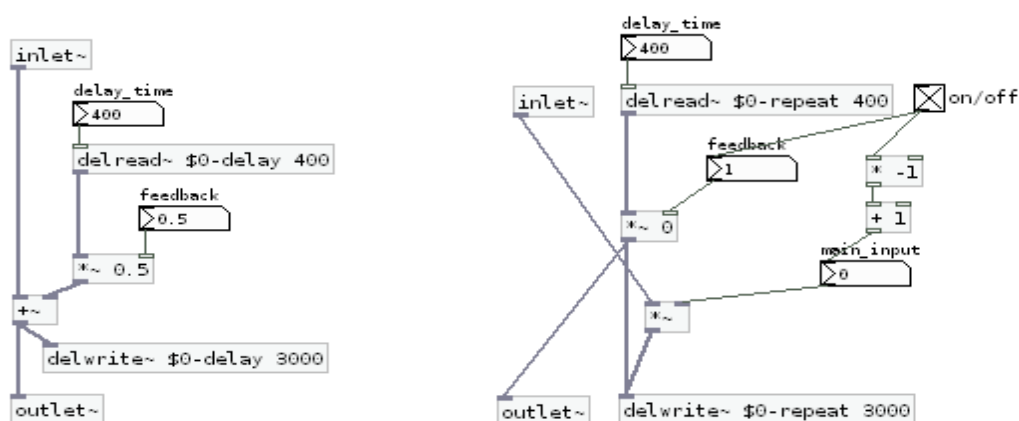


Figure 2.17: L'exemple de délai avec commande de rétroaction (à gauche), la modification de délai pour obtenir l'effet 'Beat Repeat' (à droite).

Comme un exemple de synthèse du son, avec un seul objet on a attribué les valeurs de l'axe 'x' à la fréquence d'une onde de sinus et ordonnée les valeurs de la modulation de cette fréquence avec un LFO (oscillateur basse fréquence) d'avoir des sons similaires à Theremin. Le Theremin (également Termenvox, Aetherophone) est un instrument de musique électronique joué sans contact en agitant ses mains. Une main agitant devant une antenne verticale contrôle la hauteur tandis qu'une autre main flottant au-dessus d'une antenne anneau horizontal de contrôler le volume. La façon surprenante de jouer cet instrument et éthérée des sons qu'il produit se prête à l'autre nom de l'aetherophon. Inventé en 1920 par Léon Theremin (Lev Termen Serguéévitch), le Theremin est le premier instrument de musique entièrement électronique pour atteindre une audience grande et toujours être utilisé dans la musique d'aujourd'hui.



Figure 2.18: Léon Theremin avec son instrument [24].

## 2.4. Contrôleur Panneau Tactile avec Arduino

Dans cette section, on travaille sur un autre type d'interaction homme-machine utilisant PureData. Cette fois, on utilise Arduino [25] qui est un dispositif utilisé comme entrée analogique et numérique - dispositif de sortie pour les ordinateurs. On utilise également une touche - 'écran pour saisir l'information et créer des retour audio - visuel avec PureData.

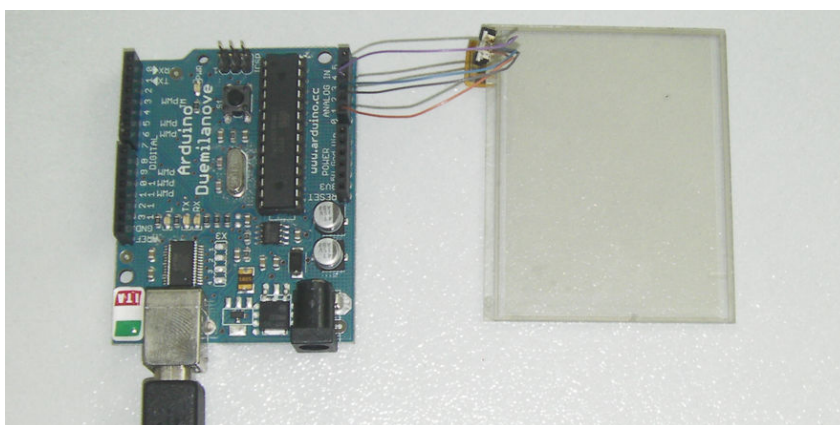


Figure 2.19: Matériel Arduino et panneau tactile à 4 fils.

Arduino est connecté à l'ordinateur via USB et est programmé en utilisant son propre logiciel. Plus d'informations sur l'installation et la configuration peuvent être trouvés sur <http://www.arduino.cc/>. Dans ce projet, on utilise un Arduino Duemilanove et un panneau tactile résistif de 4 fils.

Les fils du panneau de commande tactile sont connectés aux entrées analogiques Arduino de '0', '1', '2' et '3'. Broches analogiques Arduino peut être utilisé à la fois les entrées et sorties. On utilise cette fonction pour lire deux informations axe (X-Y) sur le panneau tactile. Pour lire l'axe 'x' de contact avec le panneau, la broche analogique '0' est défini comme sortie GND (0 V) et la broche analogique '2' est défini comme sortie +5 V. Broches analogiques '1' et '3' sont fixés comme entrées et la valeur X est lue à partir de broche '1'. Pour lire l'axe 'y' de contact avec l'écran, la broche analogique '1' est défini comme sortie GND et la broche analogique '3' est défini comme sortie +5 V. Broches analogiques '2' et '4' sont définis comme des entrées et la valeur X est lue à partir de broche '0'. Cet opération est répétée dans un cycle toutes les 200ms. La réponse d'Arduino est envoyée au port série qu'Arduino est connecté et il peut être lu à partir du logiciel Arduino, mais ce qu'on doit faire est d'être en mesure d'utiliser ces valeurs dans PureData.

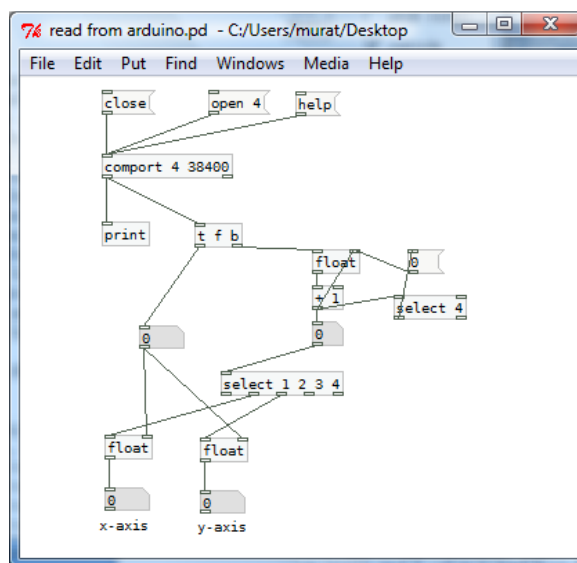


Figure 2.20: Patch PureData qui sert à lire les données envoyés par Arduino sur le port série.

Pour ce faire, on utilise un objet de PureData appelé ‘comport’, qui peut attraper les informations sur le port série sélectionné dans un taux de transmission donné. Être en mesure d'obtenir les valeurs X et Y dans PureData, on peut appliquer l'écran tactile pour tous les paramètres qu'on avait créé en PureData. Comme des exemples ‘Beat Repeat’ et ‘Theremin’ qui ont été expliquées dans le sujet précédent.

L'un des sujets les plus importants dans la création d'IHM, c'est que, il devrait être utilisable et utile pour l'utilisateur. Lors de la création d'une telle interface qui a pour but de créer de la musique, on doit prendre l'avis de certaines propriétés de musique aussi. Par exemple, dans un exemple de synthèse contrôlée par le panneau tactile ou traitement de vidéo (comme dans l'exemple précédent), l'utilisateur est le plus susceptible de contrôler les notes de musique en faisant glisser son doigt sur un coup de hache. On peut tout simplement définir les coordonnées d'un axe à toutes les notes de musique, mais il sera probablement à jouer de quelques notes indésirables aussi. Parce que dans des pièces musicales, en général une échelle est utilisée. En musique, une échelle est une séquence de notes de musique dans l'ordre croissant ou décroissant qui fournit du matériel pour ou est utilisé pour représenter commodément tout ou partie d'une œuvre musicale, y compris la mélodie et / ou de l'harmonie [26]. Il existe une variété de gammes musicales, mais les plus communes sont les échelles majeures et mineures. Par exemple, une échelle en do majeur est composée des notes; C, D, E, F, G, A et B, c'est également dans ces notes blanches d'un piano. Considérant que, à l'échelle ut mineur contient les notes; C, D, D#, F, G, G#, A#.

Quand une pièce musicale est basée sur une échelle, en général, toutes les notes dans l'échelle des sons harmoniques à la pièce dans l'ensemble, mais les notes qui ne sont pas sur l'échelle de son faux. Dans notre exemple d'IHO vidéo en temps réel basé sur filtrage des couleurs, et de panneau tactile, pour surmonter les rétroactions sonores indésirables, on a créé un patch qui s'appelle ‘scaler’ dans PureData de manière à rendre l'interaction plus utile et efficace pour l'utilisateur.



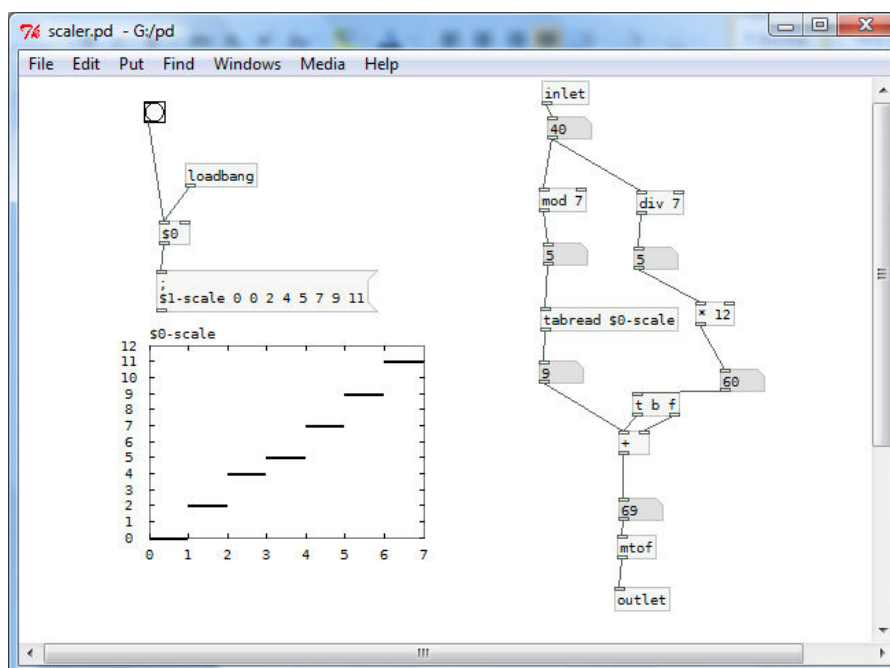


Figure 2.21: Patch PureData qui sert à créer des échelles musicales.

Le patch 'scaler' possède une entrée et une sortie, il est utilisé pour convertir les nombres qui seront plus tard converties en fréquences pour générer des sons. Par exemple, on veut créer les sons dans l'échelle 'C-majeur'. Notez qu'il ya 12 notes au total, mais dans l'échelle, il devrait être de 7 notes. Le nombre '0' représente la note plus bas 'C' en PureData. Ainsi, 'scaler' sorties les bons chiffres pour l'échelle 'C-majeur' en utilisant le tableau appelé '\$0-scale'. On peut voir dans le tableau des résultats pour l'entrée et de sortie pour le patch 'scaler' et leurs valeurs de fréquences pour les octaves, première (0-6) et sixième (35-41).

Tableau 2.1: Valeurs d'entrée et de sortie pour le patch 'scaler' et leurs valeurs de fréquences pour les octaves, première et sixième.

|           |         |         |         |         |         |       |         |
|-----------|---------|---------|---------|---------|---------|-------|---------|
| Entrée    | 0       | 1       | 2       | 3       | 4       | 5     | 6       |
| Sortie    | 0       | 2       | 4       | 5       | 7       | 9     | 11      |
| fréquence | 8.175   | 9.177   | 10.3    | 10.913  | 12.24   | 13.75 | 15.433  |
| Entrée    | 35      | 36      | 37      | 38      | 39      | 40    | 41      |
| Sortie    | 60      | 62      | 64      | 65      | 67      | 69    | 71      |
| fréquence | 261.626 | 293.665 | 329.628 | 349.228 | 391.995 | 440   | 493.883 |

Dans ce chapitre, on a expliqué des applications sur l'interaction en temps réel en utilisant le traitement vidéo ou des dispositifs tactiles. Dans les parties suivantes, on va aborder l'interaction humain-ordinateur dans une nouvelle dimension, la question de l'interaction en utilisant les ondes cérébrales. On aura une compréhension sur plusieurs ICO et essayer d'appliquer de nouvelles méthodes d'interaction et d'expliquer comment on construit un système pour permettre à un utilisateur d'interagir avec les ordinateurs utilisant l'enregistrement EEG.

meilleurs résultats comme dans cet exemple, beaucoup de sommets autour de 300 ms peuvent être déroutant et difficile à classifier.

Nous avons essayé plusieurs techniques de classification fondée sur cette décomposition. L'ensemble de données fournit des 85 ensembles d'apprentissage et 100 ensembles d'analyses pour sujet A. Nous avons atteint près de 50 % le taux correct en utilisant un canal de données EEG (Cz) en utilisant la similitude de signal basés sur la zone entre les deux signaux. Si la zone entre deux signaux est faible, nous demandons deux signaux similaires. Par exemple, deux signaux de longueur 10 avec des valeurs:

1<sup>er</sup> signal: 4 5 6 3 2 -1 -3 -4 -2 5

2<sup>ème</sup> signal: 2 4 3 0 -2 -3 -2 -5 -2 2

Leur valeur de similitude:

$$|4-2| + |5-4| + |6-3| + |3-0| + |2-(-2)| + |-1-(-3)| + |-3-(-2)| + |-4-(-5)| + |-2-(-2)| + |5-2| = 20$$

Utilisant 85 séries de données pour la formation, nous avons essayé de créer des modèles P300 en choisissant le plus semblable modèles P300 et de classer l'installation d'essai en utilisant ces modèles.

Nous avons aussi essayé de former des machines à vecteurs de support en utilisant les données de formation, mais nous ne pouvions pas obtenir des résultats satisfaisants. La cause de ce qui est probablement nous avons besoin d'extraire des formes d'onde P300 non seulement en utilisant la décomposition en ondelettes, mais aussi une séparation basée sur des sources de signaux en utilisant comme analyse en composantes indépendantes.

### 3. L'INTERACTION EN UTILISANT LES ONDES CÉRÉBRALES

#### 3.1. Électro - Encéphalogramme

Le cerveau est enfermé par le crâne, os du crâne immédiatement couvert et protège les surfaces cerveau. Une mince couche de peau, appelé le cuir chevelu, couvre la plupart du crâne. La plus grande partie du cerveau juste en dessous de l'os du crâne est le cortex cérébral. Le cortex cérébral est composé de cellules nerveuses (les neurones), dont beaucoup sont fonctionnellement reliés les uns aux autres, et reliés à d'autres parties du cerveau. L'activité électrique sous la forme d'influx nerveux envoyés et reçus depuis et vers les neurones corticaux est toujours présente, même pendant le sommeil. Dans un sens biologique du terme (ainsi que sur le plan médical ou juridique), l'absence de l'activité électrique du cortex cérébral humain signifie la mort.

L'information sensorielle est transmise de la périphérie vers basse centres dans le cerveau, puis l'information est envoyée à différentes régions du cortex cérébral. Depuis le cortex cérébral est juste sous le crâne, d'électrodes placées sur le cuir chevelu au-dessus des différentes régions du cerveau permet de détecter l'activité électrique associée au fonctionnement des neurones. L'enregistrement de l'activité du cerveau obtenues en utilisant des électrodes est appelé électro encéphalogramme ou EEG.



Figure 3.1: Enregistrement EEG [27].

Le type de cellules du cerveau qui est principalement responsable pour le stockage et la transmission des informations est le neurone. Toutes les communications de neurones impliquent le flux d'ions différents à l'entrée et la sortie de chaque neurone, tels que le sodium, le potassium, du chlore ou de calcium. Étant donné que ces ions ont une charge électrique, toutes les communications de neurones implique des changements dans le potentiel électrique à l'intérieur par rapport à l'extérieur du neurone qui peuvent être détectées à distance grâce à une variété de moyens.

Une électrode EEG principalement détecte l'activité électrique dans la région du cerveau juste en dessous. Néanmoins, les électrodes reçoivent l'activité de milliers de neurones. En effet, d'un millimètre carré de cortex a plus de 100.000 neurones. Étant donné que chaque région du cortex cérébral d'une personne alerte et reçoit occupés, l'intégration et l'envoi de nombreuses impulsions, cette activité est détectée dans l'EEG. Ce n'est que lorsque l'entrée d'une région est synchronisée avec l'activité électrique se produisant en même temps que l'on commence à distinguer, des formes d'onde périodiques dans un EEG.

En 1929, un médecin autrichien, Hans Berger a découvert que des électrodes placées sur le cuir chevelu peuvent détecter des modèles différents de l'activité électrique. Après avoir vérifié que les enregistrements ont été en effet l'enregistrement à partir du cerveau, et ne sont pas des artefacts du muscle ou du cuir chevelu, les scientifiques ont commencé à étudier ces ondes cérébrales. Aujourd'hui, l'EEG est encore un enregistrement médicalement utile pour le fonctionnement du cerveau. Dans la recherche médicale et de base, la corrélation des ondes cérébrales notamment avec des phases de sommeil, des états émotionnels, des profils psychologiques et les types d'activités mentales est en cours.

Quatre simples rythmes périodiques enregistrés sur l'EEG sont alpha, bêta, delta et thêta. Ces rythmes sont identifiés par la fréquence (Hz ou cycles / sec) et l'amplitude. Les amplitudes enregistrées par des électrodes du cuir chevelu sont de l'ordre de microvolts (mV ou  $1 / 1,000,000$  d'un volt).

### *Les Ondes Alpha*

Les quatre rythmes de base ont été associés à des états différents. Alpha est la gamme de fréquence de 8 Hz à 12 Hz. Hans Berger a nommé le premier EEG activité rythmique, il voit, l'onde 'alpha'. C'est l'activité dans la gamme 8-12 Hz vu dans les régions postérieures de la tête des deux côtés, est plus élevé en amplitude sur le côté dominant. En général, le rythme alpha est la forme d'onde en vue EEG d'un adulte qui est éveillé mais détendu, les yeux fermés.

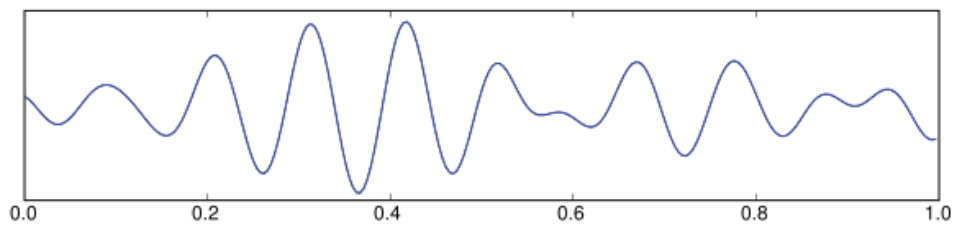


Figure 3.2: Les ondes alpha [28].

Chaque région du cerveau a un rythme alpha caractéristique des ondes alpha, mais de la plus grande amplitude sont enregistrées à partir des régions occipitales et pariétales du cortex cérébral. En général, les amplitudes des ondes alpha diminuent lorsque les sujets ouvrent les yeux et sont attentifs aux stimuli externes, bien que certains sujets formés aux techniques de relaxation puissent maintenir alpha fortes amplitudes même avec les yeux ouverts.

Rythme mu est l'activité alpha-gamme que l'on voit sur le cortex sensori-moteur. Il est atténué caractéristique avec le mouvement du bras controlatéral (ou l'imagerie mentale du mouvement du bras controlatéral).

### *Les Ondes Beta*

Beta est la gamme de fréquences de 12 Hz à 30 Hz. On voit généralement sur les deux côtés dans la distribution symétrique et est le plus évident frontalement. Rythmes bêta surviennent chez des personnes qui sont alertes et attentifs aux stimuli externes ou à l'effort mental d'exercice spécifique, paradoxalement, des rythmes bêta également se produisent pendant le sommeil profond, REM (Rapid Eye Movement) le sommeil quand les yeux basculer.

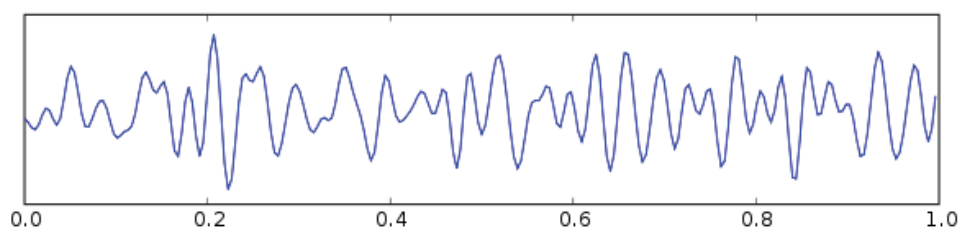


Figure 3.3: Les ondes beta [29].

L'amplitude du rythme bêta tend à être inférieure à celui des rythmes alpha. Cela ne signifie pas qu'il ya moins de l'activité électrique, plutôt que le 'positif' et 'négatif' des activités commencent à contrebalancer sorte que la somme de l'activité électrique est moins. Ainsi, au lieu d'obtenir le modèle ondulatoire de la synchronisation des ondes alpha, désynchronisation ou un bloc alpha se produit. Ainsi, la vague de bêta représente l'excitation du cortex supérieur à un état de vigilance ou de tension. Il peut également être associé à 'rappeler' ou la récupération des souvenirs.

### *Les Ondes Delta and Thêta*

Delta est la gamme de fréquences jusqu'à 4 Hz. Elle tend à être plus élevée en amplitude et le plus lent des vagues. Thêta est la gamme de fréquence de 4 Hz à 7 Hz. Thêta est considéré normalement chez les enfants jeunes. Elle peut être vue dans une somnolence ou excitation chez les adolescents et les adultes, elle peut aussi être vue dans la méditation [30].

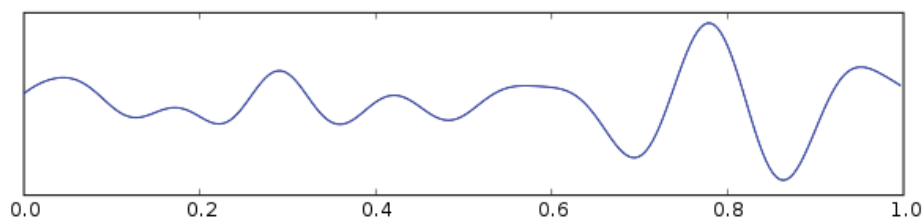


Figure 3.4 : Les ondes delta [31].

Delta et thêta sont tracés EEG de fréquence basse qui augmentent au cours du sommeil chez l'adulte normal. Comme les gens se déplacent de plus léger au plus profond les stades de sommeil (avant le sommeil paradoxal), l'apparition d'ondes alpha diminue et est progressivement remplacé par la fréquence thêta plus basse, puis rythmes delta. Bien que thêta et delta soient généralement plus importants pendant le sommeil, il y a des cas où delta et thêta sont enregistrés par des individus qui sont éveillés. Par exemple, les ondes thêta se produit pour de courts intervalles pendant réponses émotionnelles à des événements ou des situations frustrantes. Ondes delta peut augmenter au cours des activités difficiles nécessitant de la concentration mentale. Généralement la fréquence et l'amplitude du delta et du thêta sont fortement variables chez des individus.

### 3.2. Notre Travail sur Le Base de Données Existante

Avant de travailler sur nos propres données qui seront collectées, nous avons essayé de trouver un algorithme précis pour la détection des P300s, pour cela, nous avons travaillé sur 'Wadsworth BCI Dataset' (P300 potentiels évoqués) de données acquises en utilisant 'P3 BCI2000' de Speller paradigme de la 'BCI Challenge III 2004'.

#### *Explication des données:*

L'utilisateur a été présenté avec une matrice de caractères de taille 6 x 6. La tâche de l'utilisateur a été d'attirer l'attention sur les caractères d'un mot qui a été prescrite par le chercheur (un caractère à la fois). Toutes les lignes et les colonnes de cette matrice ont été successivement et de manière aléatoire intensifié à un taux de 5.7 Hz. Deux des 12



intensifications des lignes ou des colonnes figurant le caractère désiré (une ligne particulière et une colonne en particulier). Les réponses évoquées par ces stimuli rares (la sortie 2 de 12 stimuli qui ne contiennent que le caractère voulu) sont différents de ceux évoqués par les stimuli qui ne contiennent pas le caractère voulu et ils sont semblables aux réponses P300 indiqués précédemment (Farwell et Donchin, 1988, Donchin et al., 2000).



Figure 3.5 : La matrice de caractères, utilisée en paradigme 'oddball'.

Les signaux ont été collectés (bande passante filtrée de 0,1 Hz à 60 Hz et numérisé à 240 Hz) à partir de deux sujets en cinq séances chacun. Chaque session a consisté en un certain nombre de pistes. Dans chaque série, le sujet a attiré l'attention sur une série de caractères. Pour chaque époque de caractères dans la perspective d'affichage des utilisateurs, a été comme suit: la matrice a été affichée pour une période de 2,5 s, et pendant ce temps chaque personnage a la même intensité (la matrice est vide). Par la suite, chaque ligne et colonne dans la matrice a été intensifiée au hasard pour 100 ms (résultant en 12 différents stimuli - 6 lignes et 6 colonnes). Après l'intensification d'une ligne / colonne, la matrice est vide pour 75 ms. Intensifications ligne / colonne ont été en blocs aléatoires dans des blocs de 12. Les séries de 12 intensifications ont été

répétés 15 fois pour chaque époque un caractère (une ligne spécifique / colonne a été intensifiée 15 fois et donc il y avait 180 intensifications total pour chaque époque un caractère). Chaque époque à un caractère suivie d'une période de 2,5 s, et pendant ce temps la matrice était vide. Cette période a informé l'utilisateur que ce personnage a été achevé et de se concentrer sur le caractère suivant dans le mot qui était affiché sur le haut de l'écran (le caractère actuel est indiqué entre parenthèses).

Pour la compétition, les données enregistrées à partir de sujets ont été convertis en 4 fichiers Matlab (\*.mat), une formation (85 caractères) et un essai (100 caractères) pour chacun des deux sujets A et B. Toutes les données sont stockées en simple précision et peut-être besoin d'être converti en double précision (en utilisant la commande double) en fonction de la version de Matlab utilisé. Pour chaque fichier \*.mat, le signal EEG enregistré 64 canaux est organisée dans une matrice de gros 'Signal'. Comme la tâche des sujets étaient pour épeler des mots réels dans chaque série, les époques de caractères ont été brouillées à la formation et des jeux de tests pour empêcher l'identification du bon critère définir des caractères par les participants.

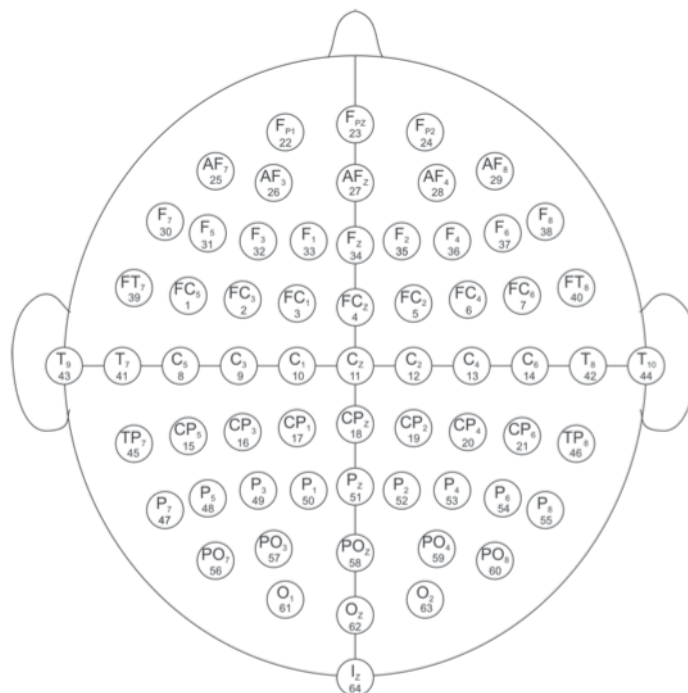


Figure 3.6 : Schéma qui illustre les désignations d'électrodes et les numéros des canaux utilisés dans cette expérience.

Pour chaque échantillon dans la matrice du signal, les événements associés sont codés en utilisant les variables suivantes:

*Flashing*: 1 Lorsque ligne / colonne a été intensifiée, 0 sinon.

*StimulusCode*: 0 si aucune ligne/ colonne est intensifiée (par exemple, la matrice est vide) 1 ... 6 pour les colonnes de l'intensification (1 ... colonne de gauche) 7 ... 12 pour lignes intensifiées (7 ... rangée supérieure les plus-).

*StimulusType* : 0 lorsque aucune ligne / colonne est intensifiée ou intensifiée ligne / colonne ne contient pas de caractères désiré 1 lors de l'intensification de ligne / personnage ne contient que le caractère voulu de cette variable permet d'accéder facilement aux étiquettes dans les ensembles d'apprentissage en ce sens qu'il peut être utilisé pour séparer les réponses qui ne contiennent que le caractère voulu de ceux qui n'ont pas.

*TargetChar*: L'étiquette de caractères corrects pour chaque époque de caractères dans les données de formation.

### ***Nos études sur les données***

Les données fournies par la concurrence étaient très grandes en termes de calcul et premièrement on a simplifié les données. On a travaillé principalement sur le canal 'Cz' que c'est l'un des canaux où P300 est bien présente. On a travaillé sur la décomposition en ondelettes [32] des signaux à des caractéristiques distinctes des ondes P300, après le numéro de plusieurs essais, nous avons vu que la somme des détails de transformation en ondelettes au niveau 8,7 et 6 de type 'db4' des ondelettes fournissent les meilleures différences visibles de P300. Considérant 15 essais pour chaque caractère, on examine le signal dans la fenêtre de temps 0-700 ms pour chaque stimulus (clignotant). Pour chaque ligne et chaque colonne, on calcule la moyenne de 15 essais. En conséquence, si nous traçons ces moyennes en 36 parcelles (comme une matrice de 6 x 6), on peut voir les réponses pour chaque caractère dans la matrice (voir image suivante):

A B C D E F  
 G H I J K L  
 M N O P Q R  
 S T U V W X  
 Y Z 1 2 3 4  
 5 6 7 8 9 \_

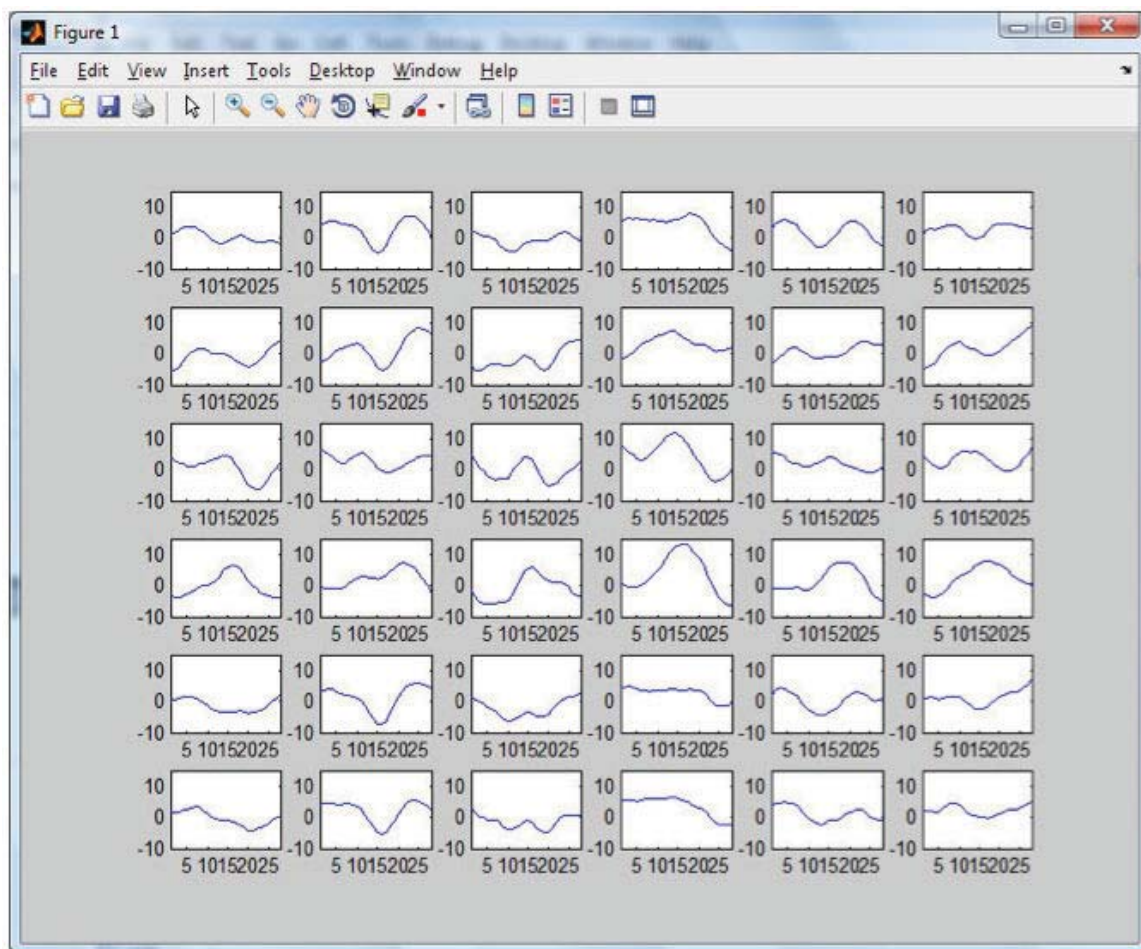


Figure 3.7: Signaux représentant la moyenne de réponses pour chaque caractère.

L'image ci-dessus est un exemple des moyennes pour toutes les lettres dans la formation mis en 4<sup>ème</sup> lettre qui est le caractère 'V'. Les 36 signaux sont des moyennes des sommes d'ondelettes décompositions détail 8, 7 et 6, l'affichage 0 - 700 ms après chaque stimulus. Le caractère 'V' est placé sur la 4<sup>ème</sup> ligne et 4<sup>ème</sup> colonne de la matrice, comme on peut le voir dans l'image, le signal de 4<sup>ème</sup> rang, 4<sup>ème</sup> colonne possède un 'saut' significatif à environ 350 ms. Bien, pas tout procès donne les

meilleurs résultats comme dans cet exemple, beaucoup de sommets autour de 300 ms peuvent être déroutant et difficile à classifier.

Nous avons essayé plusieurs techniques de classification fondée sur cette décomposition. L'ensemble de données fournit des 85 ensembles d'apprentissage et 100 ensembles d'analyses pour sujet A. Nous avons atteint près de 50 % le taux correct en utilisant un canal de données EEG (Cz) en utilisant la similitude de signal basés sur la zone entre les deux signaux. Si la zone entre deux signaux est faible, nous demandons deux signaux similaires. Par exemple, deux signaux de longueur 10 avec des valeurs:

1<sup>er</sup> signal: 4 5 6 3 2 -1 -3 -4 -2 5

2<sup>ème</sup> signal: 2 4 3 0 -2 -3 -2 -5 -2 2

Leur valeur de similitude:

$$|4-2| + |5-4| + |6-3| + |3-0| + |2-(-2)| + |-1-(-3)| + |-3-(-2)| + |-4-(-5)| + |-2-(-2)| + |5-2| = 20$$

Utilisant 85 séries de données pour la formation, nous avons essayé de créer des modèles P300 en choisissant le plus semblable modèles P300 et de classer l'installation d'essai en utilisant ces modèles.

Nous avons aussi essayé de former des machines à vecteurs de support en utilisant les données de formation, mais nous ne pouvions pas obtenir des résultats satisfaisants. La cause de ce qui est probablement nous avons besoin d'extraire des formes d'onde P300 non seulement en utilisant la décomposition en ondelettes, mais aussi une séparation basée sur des sources de signaux en utilisant comme analyse en composantes indépendantes.

## 4. LA CRÉATION ET LES APPLICATIONS RÉALISÉES DE L'INTERACTION CERVEAU – ORDINATEUR

### 4.1. L'équipement Utilisés

Un 'BIOPAC MP35' est utilisé pour enregistrer des signaux EEG. On utilise 'Matlab' pour enregistrer et traiter le signal. Enregistrement de l'EEG est fait directement sur Matlab utilisant 'BIOPAC API'. Nous pouvons déterminer le temps d'échantillonnage et le temps d'enregistrement par les contrôles sur interface graphique pour générer des stimuli. L'enregistrement est démarré en même temps exact où les essais de stimulation pour chaque lettre commencent.

Le système a besoin de deux ordinateurs pour fonctionner. L'un d'eux est connecté à BIOPAC MP35 pour l'acquisition des données EEG. L'autre ordinateur sera utilisé pour créer des stimuli (auditoire ou visuel). Un canal de la carte de son sur deuxième ordinateur est connecté à un amplificateur pour obtenir la tension suffisante pour déclencher l'unité BIOPAC. Déclenchement est utilisé pour surmonter des problèmes de 'timing'. Quand un ensemble de stimulus est sur le point de commencer, BIOPAC est déclenchée et commence à enregistrer les données en provenance des électrodes.



Figure 4.1: Figure représentant l'enregistrement des données des sujets à l'aide de 2 ordinateurs, un pour la production de stimuli et l'autre pour l'enregistrement EEG.

### *Générateur de Stimulation*

On utilise Psychtoolbox [33] de Matlab pour sa précision dans le temps et la visualisation. On a établi une matrice de 6 x 6 chiffres et de lettres, et avec un panneau de commande graphique, on peut déterminer toutes les couleurs (couleurs lettre, clignotant couleurs, le fond, etc.) et les horaires (heures clignote en ms, nombre d'essais et dure chaque lettre.).

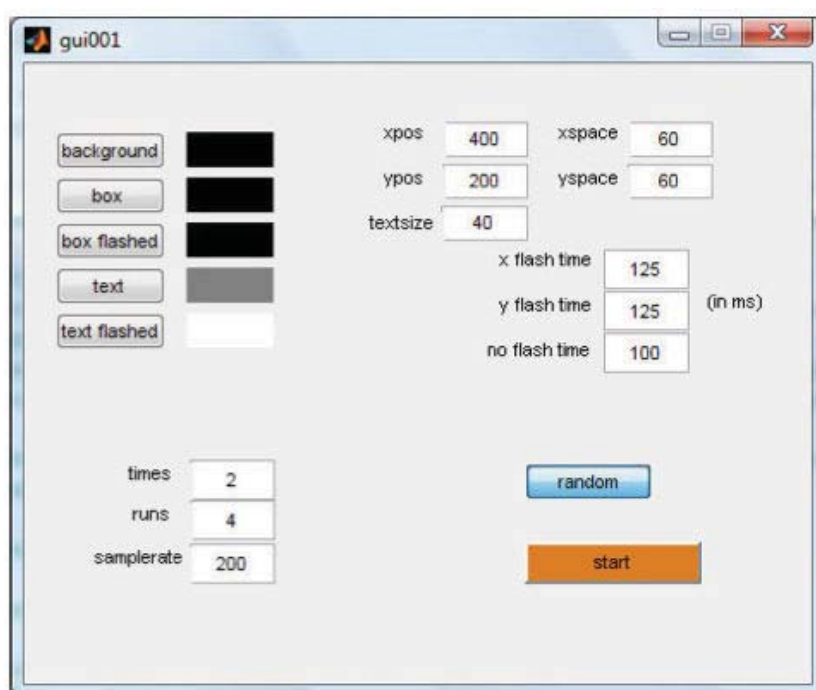


Figure 4.2: Interface graphique pour création des stimuli.

L'image montre l'interface graphique pour l'utilisateur, on peut choisir toutes les couleurs, tailles de texte et d'espacements, de ligne et de colonne fois flash, le nombre de lettres (court) et le nombre de répétitions pour chaque lettre (fois), et d'échantillonnage de l'enregistrement EEG. Le bouton 'Random' sert à créer 12 numéros au hasard, et ces numéros de contrôle à ce que l'ordre des colonnes ou des lignes sera flashé. Lorsque le bouton 'Start' est enfoncée, affiche Psychtoolbox la matrice et les bouffées de lignes et de colonnes à des périodes données et des chiffres.

### *Comment fonctionne le système ?*

Comme mentionné précédemment, on a trouvé l'aide d'ordinateurs 2 pour différentes tâches telles que l'enregistrement EEG et le traitement de stimulation dont le calendrier et la synchronisation sont essentielles. Ordinateur numéro 1 est l'ordinateur que l'utilisateur se penchera sur l'écran, l'interface graphique et de relance sont présentés ici, la 2<sup>ème</sup> est l'ordinateur connecté à BIOPAC MP35 et contrôler l'enregistrement des données EEG. Lorsque le bouton 'start' est appuyé sur l'interface graphique de 1er ordinateur, 'times', 'runs', 'sample rate', 'le temps flash x', 'le temps de flash y' et 'le temps de non-flash" sont envoyés à la 2<sup>ème</sup> ordinateur via le réseau. 2<sup>ème</sup> ordinateur calcule échantillons à enregistrer pour chaque série (chaque lettre).

$$\text{Samples to record} = 6 * (\text{xflashtime} + \text{yflashtime} + 2 * \text{noflashtime}) * \text{samplerate} / 1000$$

Quand un stimulus sur le 1<sup>er</sup> ordinateur commence, le signal déclencheur doit être envoyé à BIOPAC MP35 afin d'établir la synchronisation. On peut utiliser n'importe quelle entrée de BIOPAC MP35 comme entrée de déclenchement. On a envisagé d'utiliser la sortie audio du 1er ordinateur pour déclencher un enregistrement EEG. On utilise Psychsound, qui est une partie de la boîte à outils psychophysique, pour envoyer un signal de déclenchement en raison de ses possibilités à faible latence (près de 5ms). Même si il ya un temps de latence par exemple 'x' ms en sortie du signal audio que l'on peut mesurer, on peut commencer l'enregistrement EEG 'x' ms après, de sorte que ne posera pas de problème. Pour chaque lettre, l'enregistrement de l'EEG est à nouveau déclenché afin de minimiser les erreurs de synchronisation.

Après les stimuli pour une lettre est terminé, 'l'ordre des lignes / colonnes' qui a été créé par hasard au 1er ordinateur est envoyé à la 2<sup>ème</sup> ordinateur via le réseau, de sorte que le deuxième ordinateur peut extraire les réponses P300 et de décider quelle lettre l'utilisateur est concentré sur.



## 4.2. Interaction en Utilisant Les Ondes Alpha avec L'Alphabet du Morse

Avant d'établir une ICO basée sur les PE, on a décidé de tester notre système avec une simple forme d'interaction. On a remarqué que l'amplitude des ondes alpha deviennent plus lorsque l'utilisateur a les yeux fermés et détendue, ils ont tendance à diminuer lorsque les yeux ouverts.

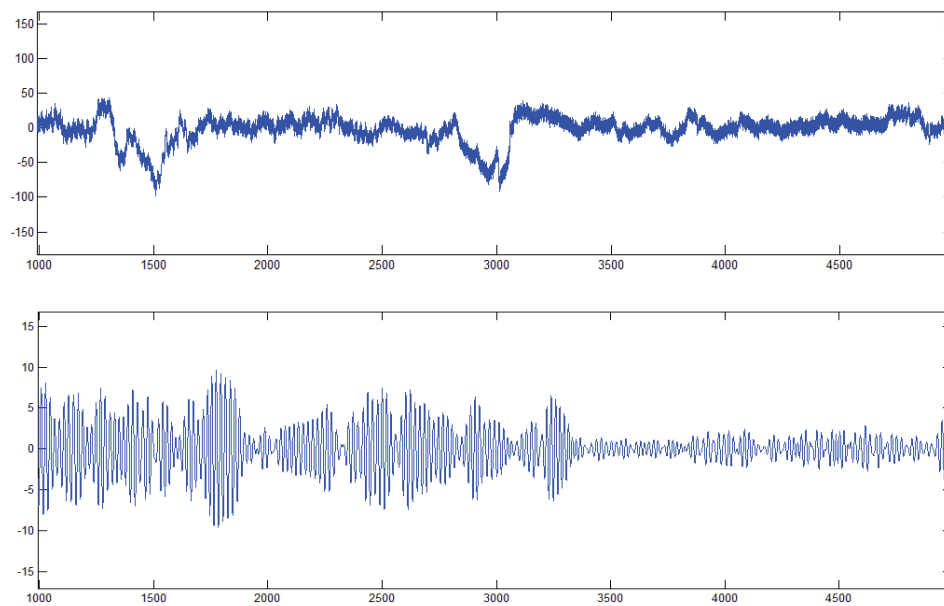


Figure 4.3: Enregistrement d'EEG et bande alpha de ce signal. On peut voir la différence autour de 3500<sup>ème</sup> échantillon ou les yeux du sujet sont ouverts.

On a modifié le code du logiciel d'enregistrement afin que les données d'entrée seront filtré passe-bande entre 8 - 12 Hz pour rester dans la zone des bandes alpha. Alors, pour toutes les 200 ms, moyenne de l'amplitude du signal est calculé. Si cette valeur est supérieure à une valeur fixe (noté 'amp' dans le code), le temps de cette région de forte amplitude est mesurée. Ceci est fait en comptant le nombre de périodes de 200 ms qui ont une valeur moyenne plus élevé que 'amp'. Si cette valeur de comptage (noté 'cntr' dans le code) est comprise entre 4 et 16, cela signifie pour une durée comprise entre 1-4 secondes, les ondes alpha ont des amplitudes élevées, cet événement peut être interprété comme un point '.' en code Morse. Si cette valeur est de plus de 16, cela signifie que pour un temps plus de 4 secondes des ondes alpha ont de forte amplitude, ce qui peut

être interprété comme un tiret '\_' dans le code morse. La partie de code Matlab démontre le processus.

```

if (amp<15)
    amplitudes(1,kth)=0;
    if ((cntr>4)&&(cntr<16))
        dl=dl+1;
        dotline(1,dl)=1;
    end
    if (cntr>16)
        dl=dl+1;
        dotline(1,dl)=2;
    end
    cntr=0;
else
    amplitudes(1,kth)=1;
    cntr=cntr+1;
end

```

Ce code Matlab mesure la durée de ces états alpha dans le signal EEG. La sortie du code entier nous donne deux matrices, appelées 'amplitudes' et 'dotline'. 'Amplitudes' tient le registre d'où les amplitudes de bande alpha étaient élevés, et 'dotline' nous donne un tableau contenant les valeurs 0, 1 et 2. 0 signifie que le niveau était faible amplitude, 1 signifie le point et 2 le tiret dans le code morse.

Version similaire de la technique est d'abord utilisé par Dewan en 1971 [34] et il a montré la possibilité de prendre le contrôle des ondes cérébrales pour envoyer des messages code morse par consciemment oscillant certaines fréquences ondes cérébrales.

### 4.3. Est-ce qu'une Interface P300 Auditive est Possible ?

Les ICM basée sur des stimuli visuels peuvent maintenir la communication chez les patients gravement paralysé, mais seulement si la vision intacte est disponible. Débilantes troubles neurologiques peuvent toutefois conduire à la perte de la vision intacte. Certains patients peuvent avoir des maladies neurologiques ou musculaires, telles que la sclérose latérale amyotrophique (SLA), qui conduit à la déficience motrice que la maladie progresse. Les perspectives les plus horribles des patients est la communication de la soi-disant immobilisé dans l'état (LIS), dans lequel seules

résiduelle de contrôle musculaire volontaire reste, ce qui rend difficile, voire impossible. Dans la complète immobilisé dans l'état (CLIS) à tout contrôle musculaire volontaire est perdu. Les interfaces cerveau-ordinateur systèmes qui se traduisent par l'activité cérébrale en signaux de contrôle des périphériques externes. ICM peut maintenir la communication chez les patients gravement paralysé.

Bien qu'ICM auditoire soit une nouvelle approche, les recherches ont été effectuées montrant que les utilisateurs peuvent apprendre à réagir aux stimuli auditifs, et des résultats stables, peuvent être obtenues à la fois les handicapés et les sujets en bonne santé [35].

Dans la prochaine partie de l'étude, on a essayé d'établir une interface basée sur PE en utilisant stimulus auditif. Comme une première expérience, les données EEG ont été enregistrées de la région mastoïdienne du cerveau. L'utilisateur peut entendre un signal sonore toutes les 400 ms. Ce ton a été choisi au hasard et avait une valeur de fréquence 440 Hz ou 880 Hz. L'utilisateur est invité à compter le nombre de tons 880 Hz. Lorsqu'on a examiné les résultats obtenus dans cette étude, on a compris qu'on a pu obtenir la composante P300 du data EEG avec des stimuli auditifs.

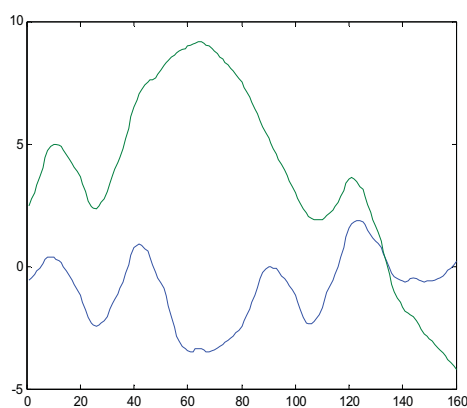


Figure 4.4: La trace verte est la moyenne des signaux attendus P300, la trace bleue est la moyenne des signaux attendus non-P300.

Nous voyons le moyen des signaux P300 (trace verte) et non-P300 (trace bleue) des signaux dans la gamme 0 – 800 ms des données recueillies. Les composants de fréquence des signaux ont été filtrés en utilisant la décomposition en ondelettes pour voir la composante P300 plus clairement.

Sur la base de cette information, on a essayé de développer une interface plus utilisable, et on a créé une interface où un utilisateur peut choisir l'une de cinq boissons qu'il voulait. Dans cette étude, les utilisateurs sont exposés à la fois des stimuli visuels et auditifs.

#### **4.4. Interface de Sélection de Boissons en Utilisant des Stimuli Auditifs et des Stimuli Visuels**

Une ICO est défini comme un système de communication qui permet à un objet pour interagir avec l'environnement par le biais de ses pensées, sans utiliser les voies de sortie du cerveau normal comme des muscles ou des nerfs périphériques [36]. Par conséquent interfaces cérébrales sont censés être utiles sur des sujets handicapés [37], alors notre étude se concentre sur une interface pour aider un sujet de sélectionner un objet de ses besoins à l'aide de stimulation visuelle ou auditive. Notre but est d'employer la moindre quantité de données (un ou deux canaux d'EEG) pour obtenir la meilleure exécution pour l'interaction et ainsi, discuter les résultats du PE utilisant les stimuli auditifs, visuels, auditifs et visuels.

L'ICO devrait fournir un environnement de sélection, on a décidé de produire des stimulations que le sujet peut choisir l'une des 5 verres, il / elle veut. Pour l'interface visuelle, on utilise 5 images abstraites de l'eau, thé, café, soda et de bière (Figure 3.5), et leurs prononciations comme stimuli auditifs. Signaux sonores sont échantillonnés à 16000 Hz et la plus longue des mots prend 500 ms de temps pour jouer. (Signaux audio créés à l'aide d'AT & T texte à moteur de reconnaissance vocale [38]). Stimulus visuel clignote un cercle jaune autour de la boisson images dans un ordre aléatoire pendant 500 ms, et aucune image boisson est flashé à 300 ms après. Playbacks auditives sont aussi présentés tous les 800 ms si les deux stimulations auditives et visuelles ont été

synchronisés. Une session d'enregistrement EEG prend 3,2 secondes de temps et en une seule session d'enregistrement EEG, 1 réponse P300 et 4 non-réponses dans les données EEG P300 sont attendus.



Figure 4.5: Stimuli visuels montrant 5 images, le cercle jaune clignote pendant 500 ms aléatoirement sur chaque image.

### ***L'acquisition de données et de prétraitement***

Biopac MP35 appareil est utilisé pour enregistrer les données EEG à un taux d'échantillonnage de 200 Hz. Deux canaux de données EEG a été acquise, celle de région mastoïdienne et un de lobe pariétal. Pour le canal mastoïdien, les électrodes positives et négatives sont placées à l'arrière des oreilles droite et gauche et électrode de masse est placée au lobe de l'oreille droite, pour le canal pariétal, les électrodes positives et négatives sont placées à 2 cm à droite et à gauche du domaine Pz du droit international système 10 - 20 et électrode de masse est placée au lobe de l'oreille gauche. Les données sont filtrées passe-bande entre 0,5 et 35 Hz et 'notch' filtrés à 50 Hz.

Les données sont recueillies à partir de 3 sujets sains de sexe masculin à l'âge de 24, 25 et 30. Les sujets assis dans une position décontractée, près de l'écran de stimulation, et

d'une oreillette est utilisée pour les stimuli auditifs. Pour chaque sujet, il est attendu de compter le nombre de clignotement (ou le nombre de lecture pour les stimuli auditif) de la boisson désirée. 50 sessions pour les visuels, auditifs pour 50 séances et 50 séances de stimulation à la fois visuelles et auditives ont été enregistrées de chaque sujet. Certaines des données ont été éliminées à cause du bruit élevé et les artefacts. Enfin, pour chaque sujet, pour chaque type de stimuli, On a eu 40 séances de données.

Pour éviter les erreurs lors de la séparation des P300 et non-P300 événements, On a utilisé une telle méthode; L'ordinateur qui crée des stimuli, qui détient le record de type de relance est créée et le nombre de millisecondes après la session a commencé. Par exemple, en raison des retards de calcul, un stimulus qui est nécessaire pour produire 400 ms après le début, peut être produite à 430 ms. Les fonctions de MATLAB qui s'appellent 'tic' et 'toc' ont été utilisés pour registrer les millisecondes où chaque stimulus a été produit.

Après, réponses P300 et non-P300 sont séparés pour chaque matière et chaque type de session stimulus, la réponse qui contient 160 échantillons est rendu *zéro-meaned*. Voici le code Matlab qui le démontre :

```
for j = 1:2
    for i = 1:40
        p3zm(i,j,1:160)=p3(i,j,1:160)-mean(p3(i,j,1:160));
    end
end

for j = 1:2
    for i = 1:160
        nonp3zm(i,j,1:160)=nonp3(i,j,1:160)-mean(nonp3(i,j,1:160));
    end
end
```

### ***Le Traitement et La classification***

Total de données contient 120 séances d'enregistrement EEG pour chaque sujet. Les réponses P300 et non-P300 attendues ont été décomposés en utilisant une fenêtre de 800 ms d'analyse, ce qui entraîne 40 réponses de P300 et 160 réponses de non-P300 pour chaque type de session de relance. Ces signaux contenant de 0 à 800 ms de données après chaque présentation des stimuli étaient *zéro-meaned* et une

transformation en ondelettes est appliquée sur 8 niveaux. On a décidé d'utiliser la somme de niveaux de détails 6, 7 et 8 de la décomposition en ondelettes, ce qui entraîne l'élimination des hautes fréquences et des réponses P300 ont été plus faciles à détecter [39] [40]. Transformée en ondelettes discrète avec Daubechies Type 4 (db4) est utilisé dans le traitement car ce type de transformation convient pour détecter les réponses P300 en raison de sa nature.

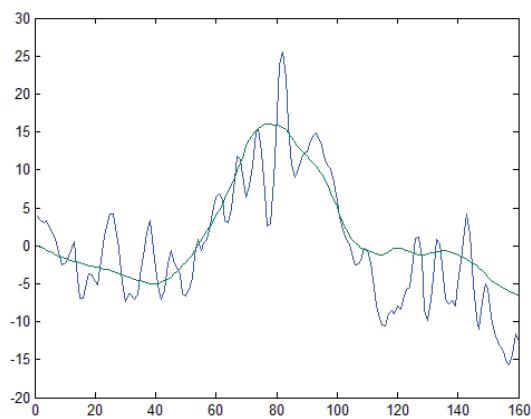


Figure 4.6: Exemple de signal EEG contenant P300 visualisé entre 000-800 ms (trace bleue) , *zéro-meaned* et la somme de niveaux de détail 6, 7 et 8 de la décomposition en ondelettes (trace verte).

La décomposition en ondelettes de data se déroule en utilisant le code ci dessous:

```
for j = 1:2
    for i = 1:40
        signal=p3zm(i,j,:);
        [C,L] = wavedec(signal,8,'db4');
        D6 = wrcoef('d',C,L,'db4',6);
        D7 = wrcoef('d',C,L,'db4',7);
        D8 = wrcoef('d',C,L,'db4',8);
        signal=D8+D6+D7;
        p3w(i,j,:)=signal;
    end
end
```

Après décomposition en ondelettes, la moyenne de tous les données P300 et non-P300 ont été calculé. On a remarqué que pour tous les sujets, dans les données du canal mastoïdien, une déviation positive était présent entre les échantillons 60<sup>ème</sup> au 100<sup>ème</sup> de données qui représente 300 - 500 ms après un stimulus est présenté.

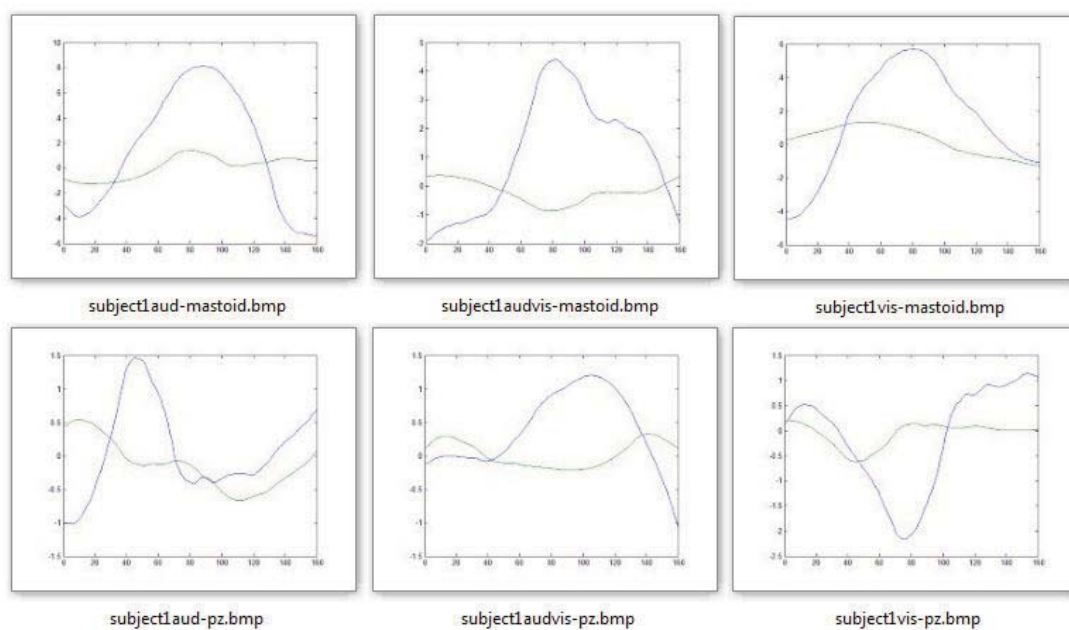


Figure 4.7 : Moyennes P300 et non-P300 des sessions; auditive (à gauche), auditive + visuelle (au milieu) et seulement visuelle (à droite), des canaux; Pz (en haut) et mastoïde (en bas) pour le sujet 1.

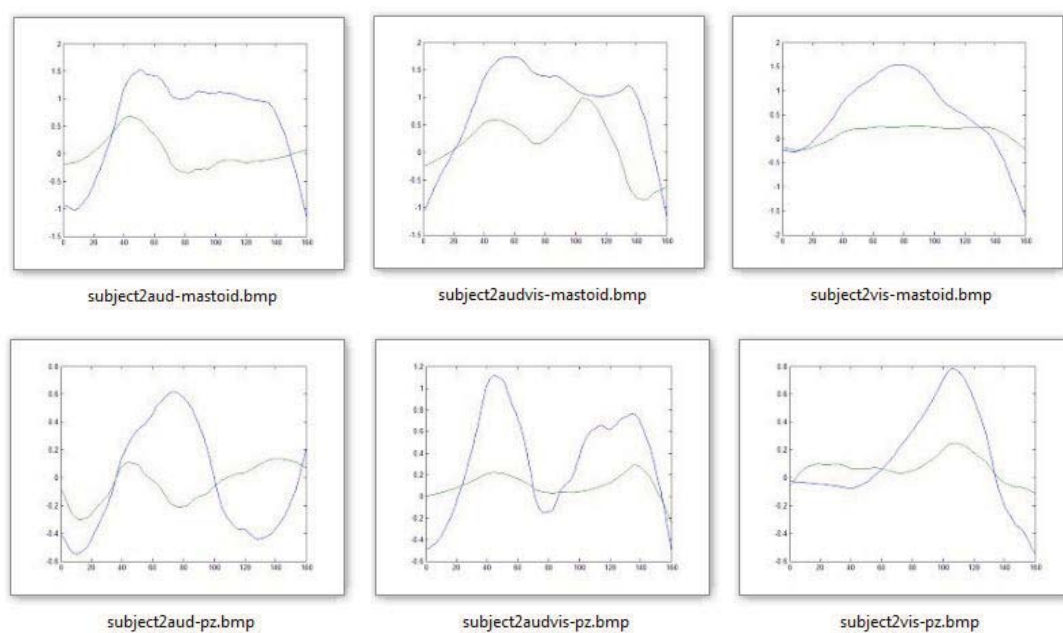


Figure 4.8 : Moyennes P300 et non-P300 des sessions; auditive (à gauche), auditive + visuelle (au milieu) et seulement visuelle (à droite), des canaux; Pz (en haut) et mastoïde (en bas) pour le sujet 2.



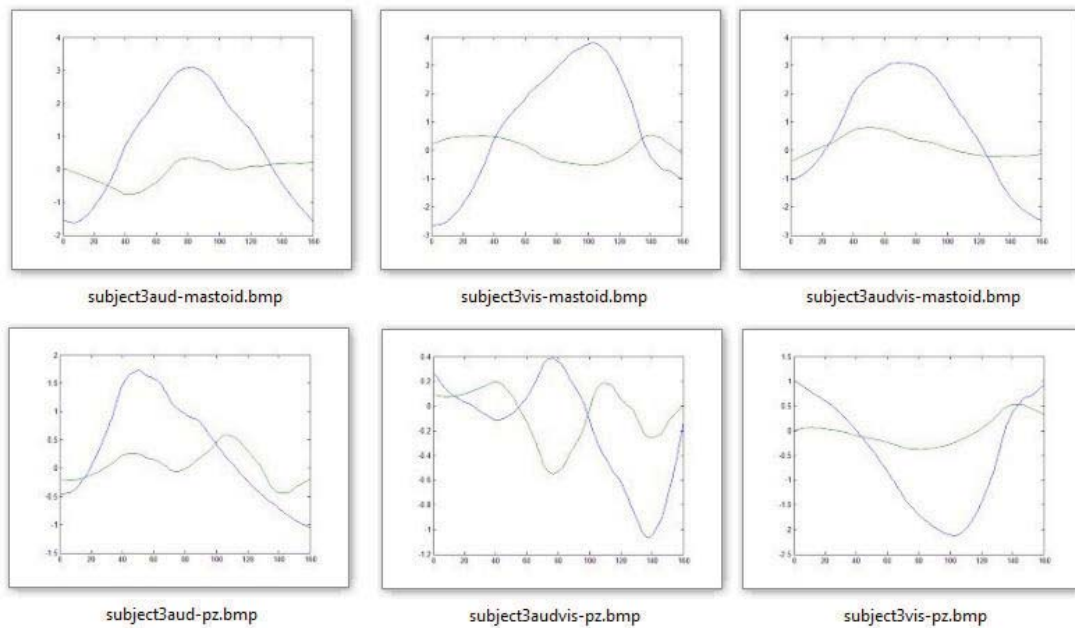


Figure 4.9: Moyennes P300 et non-P300 des sessions; auditive (à gauche), auditive + visuelle (au milieu) et seulement visuelle (à droite), des canaux; Pz (en haut) et mastoïde (en bas) pour le sujet 3.

Après avoir pu voir la différence entre les données P300 et non-P300, on a comparé les moyennes des données P300 et non-P300 aux différents niveaux. Moyennes de 40 séances, 10 séances, 5 sessions ont été calculés. Moyennes des données non-P300 ont été calculés pour chaque boisson, si les moyennes non-P300 sont divisés en 4 parties (4 entre 5 valeurs ne contient pas l'amplitude P300), de cette façon on peut faire une comparaison entre les réponses pour chaque stimulus (1 causant P300 et 4 ne causent pas P300).

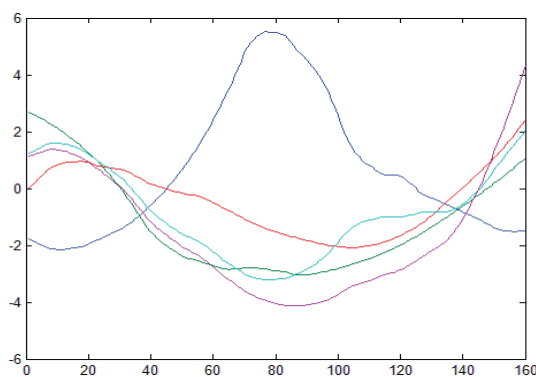


Figure 4.10: Exemple de moyennes des réponses de 5 sessions P300 (trace bleue) et 4 réponses des mêmes sessions non-P300

Pour la classification des données de l'EEG, on a comparé la moyenne quadratique des 60<sup>ème</sup>, 70<sup>ème</sup>, 80<sup>ème</sup>, 90<sup>ème</sup> et 100<sup>ème</sup> échantillons de chaque moyenne. Nous appelons cette valeur 'Vp' pour signaux attendus P300, et 'Vnp1', 'Vnp2', 'Vnp3' et 'Vnp4' pour les signaux attendus non-P300. S représente l'indice de l'échantillon. (Eq. 1)

$$Vp = \sqrt{\frac{(S_{60}^2 + S_{70}^2 + S_{80}^2 + S_{90}^2 + S_{100}^2)}{5}}$$

(1)

Cette valeur est choisie parce qu'elle peut décrire une bonne interprétation s'il y a une déviation positive entre 300 ms et 500 ms des données et nous supposons P300 valeur la plus élevée entre VP, Vnp1, Vnp2, Vnp3 et Vnp4 (Figure 5). Si la valeur la plus élevée entre eux est VP, P300 est classé correctement.

Sur le tableau ci-dessous, on voit ces valeurs pour différents types de stimuli de différentes valeurs moyennes pour le sujet 1. Si la valeur sur le dessus de 5 valeurs est la plus élevée, la classification se fait correctement.

Tableau 4.1: Valeurs de VP pour différents types de stimuli de différents niveaux des moyennes pour le sujet 1.

| subject1 | avg40  | avg10 |        |        |        | avg5   |        |        |        |        |        |        |        |  |
|----------|--------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--|
| auditory |        |       |        |        |        |        |        |        |        |        |        |        |        |  |
| mastoid  | 138,51 | 17,83 | 17,85  | 56,84  | 45,99  | 26,02  | 9,64   | 8,5    | 27,21  | 79,69  | 33,99  | 39,31  | 52,67  |  |
|          | 14,49  | 1,4   | 6,49   | -0,27  | 21,3   | 2,48   | 6,97   | 7,66   | -15,35 | 26,9   | 3,35   | 1,06   | -20,67 |  |
|          | 12,06  | 7,69  | 16,09  | 20,29  | -14,66 | 0,31   | -8,21  | 5,33   | -12,52 | -27,44 | 17,93  | 41,54  | 51,2   |  |
|          | 34,08  | -0,62 | -13,94 | 10,64  | 15,27  | 9,34   | 5,06   | 16,18  | 16,51  | -8,4   | 11,19  | -36,62 | -36,89 |  |
|          | 7,12   | 6,03  | 3,42   | 3,42   | -14,79 | 6,04   | 7      | 16     | -9,67  | 48,98  | -4,36  | 7,29   | 7,32   |  |
|          |        |       |        |        |        |        |        |        |        |        |        |        |        |  |
| cz       | -0,49  | -1,62 | 7,69   | -0,08  | -6,48  | 1,03   | -4,28  | 3,02   | 12,37  | -2,66  | 2,5    | 6,58   | -19,55 |  |
|          | -6,09  | -2,24 | 2,63   | -1,79  | 7,06   | 0,27   | -4,37  | 3,82   | -2,57  | 4,4    | -10,37 | 9,26   | 4,18   |  |
|          | 3,77   | 0,66  | -2,97  | 9,42   | -20,82 | -4,75  | -1,72  | 1,44   | 12,52  | -7,98  | 15,61  | 4,85   | -10,04 |  |
|          | 17,49  | -3,04 | 4,97   | 2,62   | -2,93  | -0,11  | 0,62   | -1,59  | 5,28   | -5     | -9,85  | -22,02 | -11,12 |  |
|          | -34,47 | -1,46 | -0,86  | 7,24   | -17,78 | 1,42   | -3,55  | -4,36  | -7     | 23,83  | 24,34  | -19,61 | -24,44 |  |
|          |        |       |        |        |        |        |        |        |        |        |        |        |        |  |
| aud+vis  |        |       |        |        |        |        |        |        |        |        |        |        |        |  |
| mastoid  | 66,83  | 7,48  | 9,84   | 19,03  | 30,47  | 4,28   | 10,68  | 9,18   | 10,51  | 15,56  | 22,5   | 38,26  | 22,68  |  |
|          | -5,65  | -3,18 | -0,81  | 7,37   | -0,51  | -2,42  | -10,37 | -7,36  | 3,6    | 23,11  | -1,21  | 36,46  | -7,68  |  |
|          | -7,27  | -2,78 | -3,53  | -10,82 | -9,5   | -3,95  | 15,14  | 5,74   | -1,24  | -8,38  | 2,04   | -37,48 | 28,45  |  |
|          | -24,33 | 2,39  | 1,18   | 0,42   | 10,39  | 1,77   | 3,71   | -12,34 | -9,64  | -10,83 | -11,74 | -19,51 | -4,66  |  |
|          | -15,84 | -2,08 | -4,11  | -21,3  | -16,21 | -7,32  | -7,87  | 5,28   | 1,42   | -10,81 | -30,85 | 0,51   | -27,77 |  |
|          | 1      | 4     | 5      | 16     |        |        |        |        |        |        |        |        |        |  |
| cz       | 16,4   | 0,67  | 1,54   | 3,34   | 10,85  | 1,09   | 0,26   | 0,67   | 2,41   | -2,58  | 9,25   | 14,26  | 7,44   |  |
|          | 1,29   | 1,23  | -0,52  | -1,07  | -1,39  | -2,14  | -8,83  | -0,73  | -0,96  | -0,69  | -9,21  | 1,44   | -39,59 |  |
|          | -2,74  | 0,07  | 1,2    | -7,77  | -4,38  | 4,6    | 0,62   | -0,3   | -2,41  | -1,45  | 0,04   | -4,22  | 24,15  |  |
|          | -8,49  | -4,11 | -1,68  | -4,59  | -7,72  | -2,02  | 3,66   | 1,51   | -2,63  | -10,3  | -1,23  | -9,24  | -3,66  |  |
|          | -4,59  | 4,1   | -1,74  | 4,93   | 8,9    | 2,17   | 4,54   | 0,89   | -0,84  | -5,24  | 11,1   | 0,48   | 21,46  |  |
|          |        |       |        |        |        |        |        |        |        |        |        |        |        |  |
| visual   |        |       |        |        |        |        |        |        |        |        |        |        |        |  |
| mastoid  | 100,3  | 11,61 | 18,89  | 23,64  | 46,16  | 9,67   | 13,54  | 17,42  | 20,36  | 17,88  | 29,41  | 60,41  | 31,91  |  |
|          | -0,88  | -5,91 | 3,11   | 2,21   | 3,62   | -0,43  | 4,91   | -11,34 | 14,24  | -5,2   | -8,83  | -1,79  | 0,84   |  |
|          | 2,46   | -3,04 | -2,62  | 9,21   | 15,94  | -11,39 | 2,64   | 17,57  | 10,19  | 9,62   | 18,16  | 9,03   | 16,03  |  |
|          | 20,56  | 3,78  | 12,21  | 4,67   | 8,43   | -13,82 | 6,78   | -9,21  | -21,16 | 9,04   | -26,49 | 28,99  | 12,27  |  |
|          | 36,9   | 4,29  | -10,24 | 4,47   | 8,9    | 7,75   | 1,81   | 3,96   | 0,68   | 9,37   | 35,43  | 2,9    | 5,53   |  |
|          |        |       |        |        |        |        |        |        |        |        |        |        |        |  |
| cz       | -28,62 | -2,02 | -0,75  | -15,88 | -9,97  | -4,15  | 0,1    | 1,02   | -2,52  | -22,16 | -9,6   | -10,54 | -9,4   |  |
|          | 3,83   | 0,92  | 1,49   | -0,77  | 3,8    | 1,31   | 2,77   | -2,2   | -0,06  | -1,92  | -21,55 | -4,49  | 17,44  |  |
|          | -12,25 | 1,16  | 1,92   | -6,14  | 0,3    | 0,52   | -1,84  | 5,18   | -11,15 | 0,37   | 11,93  | 12,09  | 16,54  |  |
|          | -15,24 | 0,46  | -5,6   | -4,81  | 16,99  | -0,02  | 0,13   | 1,48   | -0,53  | -18,24 | 7,05   | -12,04 | -11,06 |  |
|          | 22,28  | 1,29  | -10,06 | -3,52  | 1,19   | 2,34   | 2,44   | 2,36   | -19,59 | 5,95   | -14,09 | 12,65  | 13,44  |  |

Les sujets sont dits à quelle boisson se concentrer au cours des séances d'enregistrement, et on compare les résultats avec leurs choix. Dans le tableau, nous voyons le nombre de classifications vrai pour les 40, 10, 5 moyennes et essais unique, pour chaque sujet dans les sessions des différents stimuli, et deux canaux de données EEG.

Tableau 4.2: Pourcentages des classifications vraies pour 3 sujets

|                  | Moyenne de 40 essais | Moyenne de 10 essais | Moyenne de 5 essais | Essais single |
|------------------|----------------------|----------------------|---------------------|---------------|
| <b>Subject 1</b> |                      |                      |                     |               |
| <b>Auditory</b>  |                      |                      |                     |               |
| Mastoid          | 1/1 (%100)           | 4/4 (%100)           | 6/8 (%75)           | 18/40 (%45)   |
| Pz               | 0/1 (%0)             | 1/4 (%25)            | 0/8 (%0)            | 6/40 (%15)    |
| <b>Aud.+Vis.</b> |                      |                      |                     |               |
| Mastoid          | 1/1 (%100)           | 4/4 (%100)           | 5/8 (%62.5)         | 16/40 (%40)   |
| Pz               | 1/1 (%100)           | 2/4 (%50)            | 2/8 (%25)           | 10/40 (%25)   |
| <b>Visual</b>    |                      |                      |                     |               |
| Mastoid          | 1/1 (%100)           | 2/4 (%50)            | 2/8 (%25)           | 10/40 (%25)   |
| Pz               | 0/1 (%0)             | 0/4 (%0)             | 0/8 (%0)            | 2/40 (%5)     |
| <b>Subject 2</b> |                      |                      |                     |               |
| <b>Auditory</b>  |                      |                      |                     |               |
| Mastoid          | 1/1 (%100)           | 3/4 (%75)            | 5/8 (%62.5)         | 13/40 (%32.5) |
| Pz               | 1/1 (%100)           | 1/4 (%25)            | 1/8 (%12.5)         | 12/40 (%30)   |
| <b>Aud.+Vis.</b> |                      |                      |                     |               |
| Mastoid          | 0/1 (%0)             | 3/4 (%75)            | 5/8 (%62.5)         | 14/40 (%35)   |
| Pz               | 0/1 (%0)             | 1/4 (%25)            | 1/8 (%12.5)         | 10/40 (%25)   |
| <b>Visual</b>    |                      |                      |                     |               |
| Mastoid          | 1/1 (%100)           | 3/4 (%75)            | 2/8 (%25)           | 10/40 (%25)   |
| Pz               | 1/1 (%100)           | 2/4 (%50)            | 2/8 (%25)           | 15/40 (%37.5) |
| <b>Subject 3</b> |                      |                      |                     |               |
| <b>Auditory</b>  |                      |                      |                     |               |
| Mastoid          | 1/1 (%100)           | 3/4 (%75)            | 6/8 (%75)           | 14/40 (%35)   |
| Pz               | 0/1 (%0)             | 1/4 (%25)            | 2/8 (%25)           | 8/40 (%20)    |
| <b>Aud.+Vis.</b> |                      |                      |                     |               |
| Mastoid          | 1/1 (%100)           | 3/4 (%75)            | 3/8 (%37.5)         | 11/40 (%27.5) |
| Pz               | 1/1 (%100)           | 1/4 (%25)            | 1/8 (%12.5)         | 10/40 (%25)   |
| <b>Visual</b>    |                      |                      |                     |               |
| Mastoid          | 1/1 (%100)           | 4/4 (%100)           | 5/8 (%62.5)         | 17/40 (%42.5) |
| Pz               | 0/1 (%0)             | 0/4 (%0)             | 0/8 (%0)            | 5/40 (%12.5)  |

Tableau 4.3: Pourcentages des classifications vraies pour la totalité des sujets

|                  | Moyenne de 40 essais | Moyenne de 10 essais | Moyenne de 5 essais | Essais single |
|------------------|----------------------|----------------------|---------------------|---------------|
| <b>Auditory</b>  |                      |                      |                     |               |
| Mastoid          | 3/3 (%100)           | 10/12 (%83)          | 6/24 (%25)          | 45/120 (%37)  |
| Pz               | 1/3 (%33)            | 3/12 (%25)           | 3/24 (%12)          | 26/120 (%22)  |
| <b>Aud.+Vis.</b> |                      |                      |                     |               |
| Mastoid          | 2/3 (%66)            | 10/12 (%83)          | 13/24 (%54)         | 41/120 (%34)  |
| Pz               | 2/3 (%66)            | 4/12 (%33)           | 4/24 (%16)          | 30/120 (%25)  |
| <b>Visual</b>    |                      |                      |                     |               |
| Mastoid          | 3/3 (%100)           | 9/12 (%75)           | 9/24 (%37)          | 37/120 (%30)  |
| Pz               | 1/3 (%33)            | 2/12 (%16)           | 2/24 (%8)           | 22/120 (%18)  |

Sur les tables, on a voulu montrer le taux de réussite aux différents niveaux des moyennes. En générale, on a obtenu de bons résultats à partir du canal mastoïdien si le numéro de procès est grand. Cette étude est inspectée et traitées hors ligne, mais les méthodes sont choisies pour être simple à calculer si une mise en œuvre d'un système d'ICO utilisé en ligne, serait facile. Le problème dans ce cas est le stimulus auditif a été beaucoup plus lent que les stimuli visuels. Par exemple, un seul essai prends  $800 \times 5 = 3200$  ms de temps, mais seulement les moyennes d'au moins 10 essais peut nous donner des résultats corrects pour que cette ICO peut être utilisable et stable et pour cela, 30 secondes de temps est nécessaire. Mais par rapport aux difficultés d'extraction de traces P300 des stimuli visuels en fonction des données EEG, on a vu que, dans une concentration auditive interaction est acquise plus facile et on peut avoir de meilleurs résultats avec des données en utilisant un seul canal d'EEG.

## **5. CRÉATION DE L'INTERFACE CERVEAU – ORDINATEUR EN LIGNE**

Dans ce chapitre, on donne une explication détaillée de notre système d'enregistrement des données. On va élaborer les aspects de ce système à deux ordinateurs, comme la création de relance, l'enregistrement des données, synchronisation, etc. Et aussi, on comprendra la réalisation et l'utilisation du système d'interaction en ligne.

### **5.1. Le Flux de Travail de L'ICO en Ligne**

L'un des aspects les plus importants de ce travail était de créer l'interaction entre le cerveau et les ordinateurs. Après avoir travaillé sur plusieurs paradigmes et algorithmes sur ICO, dans ce chapitre, on représente le système interaction en ligne. Tout d'abord ce système devrait fournir une réponse fondée sur les utilisateurs des enregistrements d'EEG avec le traitement en ligne. Dans cette ICO P300 que nous présentons, il ya deux paramètres principaux. L'un d'eux est le nombre de séquences, que nous appelons 'runs'. L'autre est le nombre de clignotements pour chaque stimulus, que nous appelons 'times'. Par exemple, avec un ICO P300 Speller, si un utilisateur veut écrire le mot 'bonjour', la valeur de 'runs' doit être 7, et le nombre qui définit combien de fois toutes les lignes et colonnes seront flashé pour la détermination de chaque lettre sera la valeur 'times'.

Nous utilisons le système à 2 - ordinateurs pour enregistrement de données comme expliqué dans le chapitre 3, et cette fois nous la développons pour gérer le traitement en ligne.



Figure 5.1: Système ICO à 2-Ordinateurs, BIOPAC MP35 et amplificateur.

Les valeurs ‘times’ et ‘runs’ sont créées par l'utilisateur dans l'interface que nous avons créé sur l'ordinateur numéro 1. Après que le bouton ‘Démarrer’ est pressé par l'utilisateur, deux ordinateurs vérifie les uns les autres sur le réseau avec une fonction appelée ‘handshake’ (poignée de main). Cette fonction ‘handshake’ est appelée avant chaque transaction sur le réseau entre deux ordinateurs, afin de surmonter les problèmes de réseau tels que les pertes de données. Comme le second ordinateur est correctement connecté au matériel BIOPAC mp35 et si la poignée de main est un succès, les valeurs qui sont créés par l'utilisateur, comme ‘times’, ‘runs’, etc. sont envoyés vers le deuxième ordinateur. Ainsi, le temps d'enregistrement pour chaque séquence est calculé. Avec le signal audio provenant du premier ordinateur, les stimuli et l'enregistrement commence en synchronisation avec le déclenchement de l'unité BIOPAC. Chaque type de stimulus et leurs horaires exacts sont stockés dans le premier ordinateur au cours de la présentation de stimuli dans une séquence. Lorsque cette séquence est terminée, la poignée de main fonction est encore appelée et les valeurs de stimulus et de synchronisation sont également transmises au second ordinateur. À ce stade, le deuxième ordinateur obtient toutes les informations nécessaires pour le traitement des données. Les données sont envoyés à une fonction appelée ‘process’ qui renvoie le résultat, (le stimulus choisi par l'utilisateur) en utilisant l'algorithme qu'il possède. Cette partie du processus du programme est écrit comme une fonction séparée pour permettre travailler sur différents algorithmes a l'avenir. Enfin ce résultat est renvoyé au premier ordinateur et présenté à l'utilisateur.

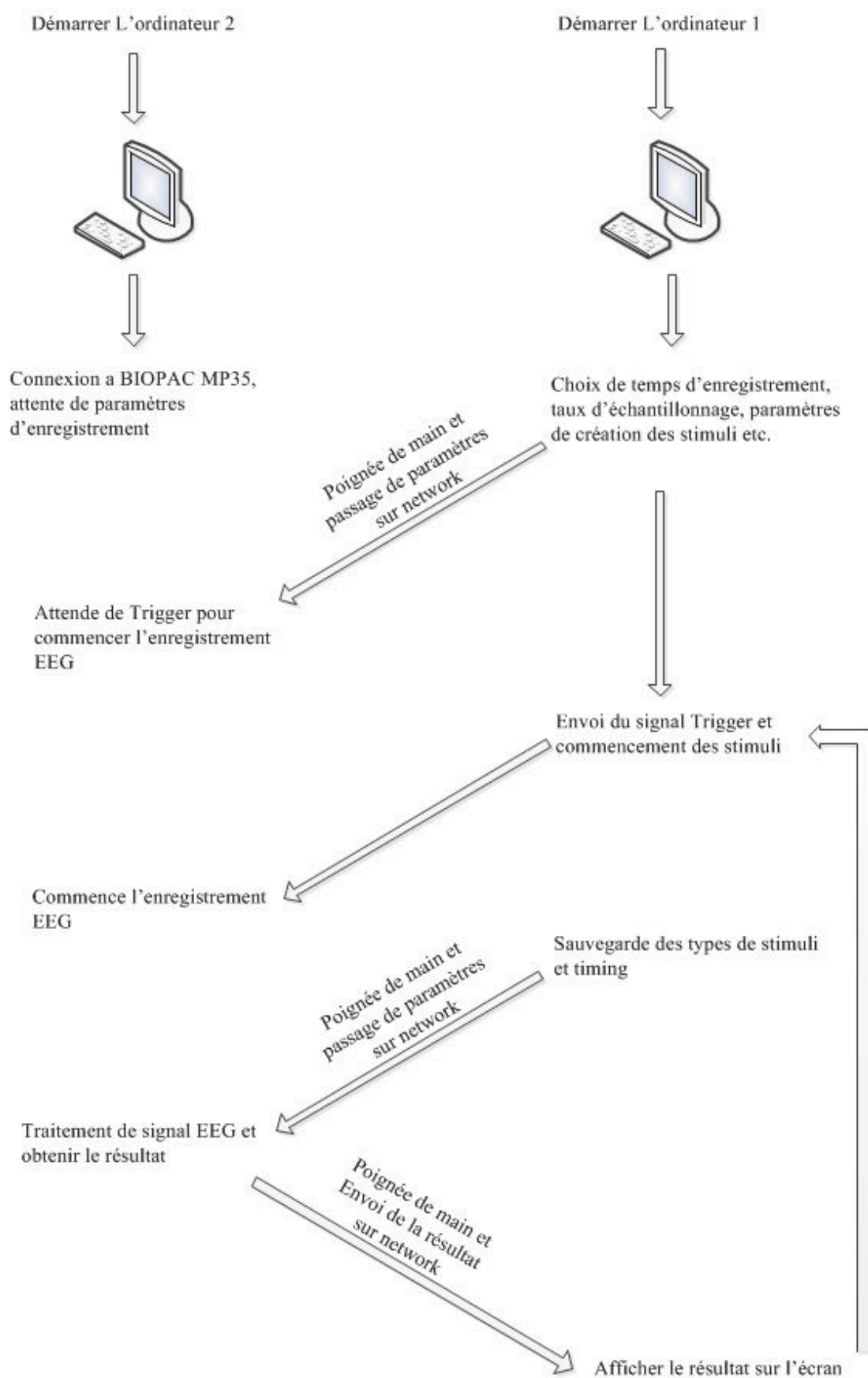


Figure 5.2: Schéma détaillé de flux de travail de l'ICO.



## 5.2. Réalisation de L'ICO P300 Speller

Dans ce partie du travail, on explique la création de stimuli, la façon dont les EEG enregistrés sont traités et comment l'utilisateur sera en mesure d'utiliser le système en temps réel.

Le mécanisme de l'ICO P300 Speller est le même que nous l'avons expliqué dans le 3ème chapitre. Une matrice de 6x6 lettres doit être présentée à l'utilisateur, et les rangées et les colonnes doivent être intensifiées de façon aléatoire. Pour cette partie, nous avons construit une interface à l'utilisateur qui donne la possibilité d'affiner le positionnement de la matrice sur l'écran, la taille des lettres, des couleurs de fond, des lettres, des boîtes où la couleur d'intensification.

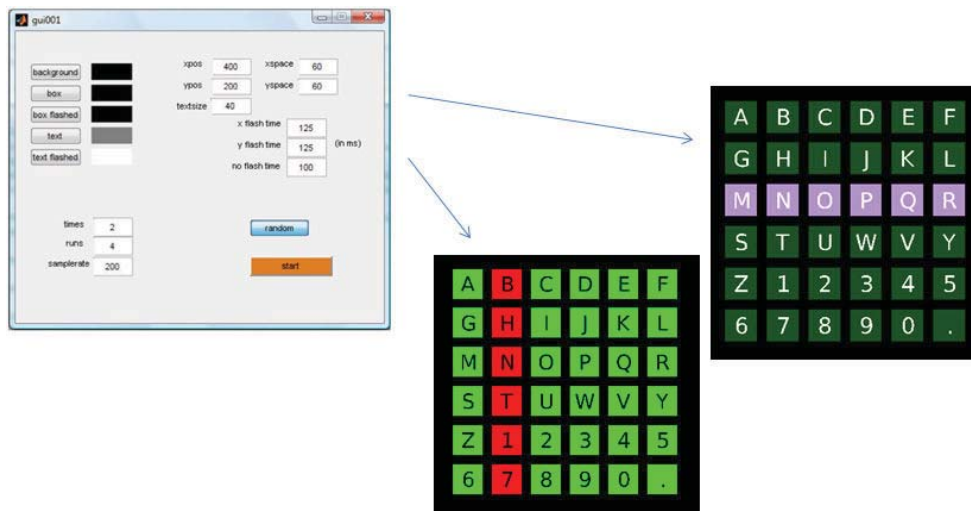


Figure 5.3: Création de stimuli différents avec l'interface utilisateur.

On utilise Psychtoolbox pour créer des stimuli visuels, en raison de sa rapidité et efficacité. Mais le chargement d'une image et l'impression à l'écran de l'ordinateur est un processus très lent et il rend l'interaction presque impossible. Ainsi, les images de 6 lignes et 6 colonnes intensifiées intensifiées doivent être stockées dans la mémoire avant que le processus commence. Ces 12 images et image de fond (non intensifiées) sont créées et pré-chargées dans la mémoire selon les choix de l'utilisateur de l'interface.

Même si on peut créer les stimuli assez vite, le timing reste encore comme un obstacle. Si dans une session d'enregistrement, l'utilisateur a sélectionné les lignes et colonnes à l'intensification de 100 ms de temps et restent non intensifiés pour 100 ms. L'enregistrement EEG commence exactement 1s avant le stimulus, de sorte que le stimulus doit être présenté exactement sur 1000 ms, 1200 ms, 1400 ms ... et si la valeur de 'times' est sélectionné 10, les stimuli dernier doit être présenté sur:  $1000 + (200 * 12) * 10 = 25000$  ms. Mais les valeurs vraies qui sont mesurées dans MATLAB sont différentes. Comme un exemple de présentation des stimuli fois, dans une de sessions de mesure, les stimuli ont été présentés sur: 1000<sup>ème</sup>, 1223<sup>ème</sup>, 1437<sup>ème</sup>, 1650<sup>ème</sup>, 1864<sup>ème</sup>, 2077<sup>ème</sup>... millisecondes et le dernier stimulus a été présenté le 26,4130<sup>ème</sup> seconde. C'est pourquoi ces valeurs doivent être enregistrées et envoyées à 2<sup>ème</sup> ordinateur pour meilleure analyse des réponses P300. Enfin en utilisant ces valeurs enregistrées, le résultat est obtenue en utilisant l'algorithme de sélection de P300 que nous avons expliqué dans la chapitre 4 et le lettre obtenue est affiché a l'écran sur la 1<sup>er</sup> ordinateur, comme c'est l'ordinateur que l'utilisateur interagit.

Le nombre de séquences dans une session d'enregistrement (nombre de lettres souhaité, dans ce projet) a été sélectionné par l'utilisateur avec la valeur s'exécute. A la fin du processus, toutes les lettres trouvées par EEG algorithme de traitement en 2<sup>ème</sup> ordinateur sont combinées et envoyés au 1<sup>er</sup> ordinateur et affichée sur l'écran.

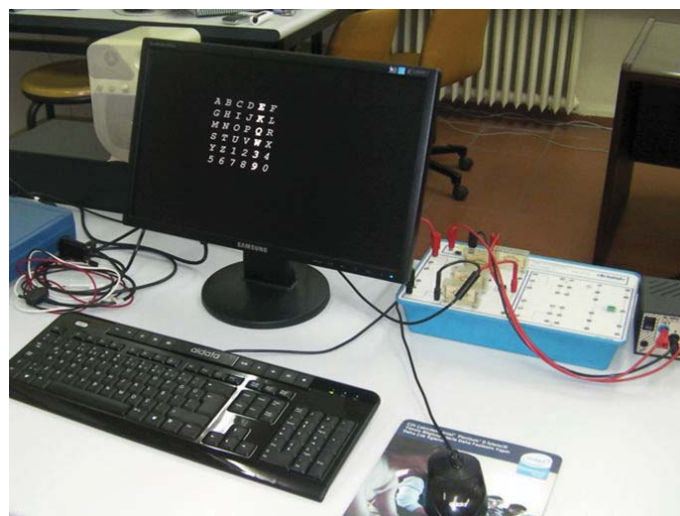


Figure 5.4: Ordinateur numéro 1, affichant les stimuli.

### 5.3. Réalisation de L'ICO de Sélection de Boissons

Le fonctionnement générale de cette partie du projet est similaire au fonctionnement de ICO P300 Speller, mais cette fois, les résultats d'interaction sont présentés à l'utilisateur sur le 1<sup>er</sup> ordinateur à la fin de chaque séquence comme:

‘Vous avez choisi: l'eau’

La création de stimuli visuels n'est pas aussi compliqué que l'exemple précédent, que l'esprit de ce projet était de pouvoir trouver un algorithme simple pour détecter les amplitudes P300 qui serait assez rapide pour traiter les données et retourner le résultat, de manière à créer une interaction. Mais aussi, la création de stimuli auditifs est une partie de ce projet, afin de savoir si les enregistrements EEG de différentes parties du cerveau ont des composantes P300 distinctives pour les stimuli auditifs et visuels.

Comme il est possible d'utiliser ce système avec seulement les stimuli visuels (avec haut-parleurs hors tension), avec les deux stimuli visuels et auditifs, et que des stimuli auditifs (les yeux fermés), les résultats sont également présentés avec la sortie audio.

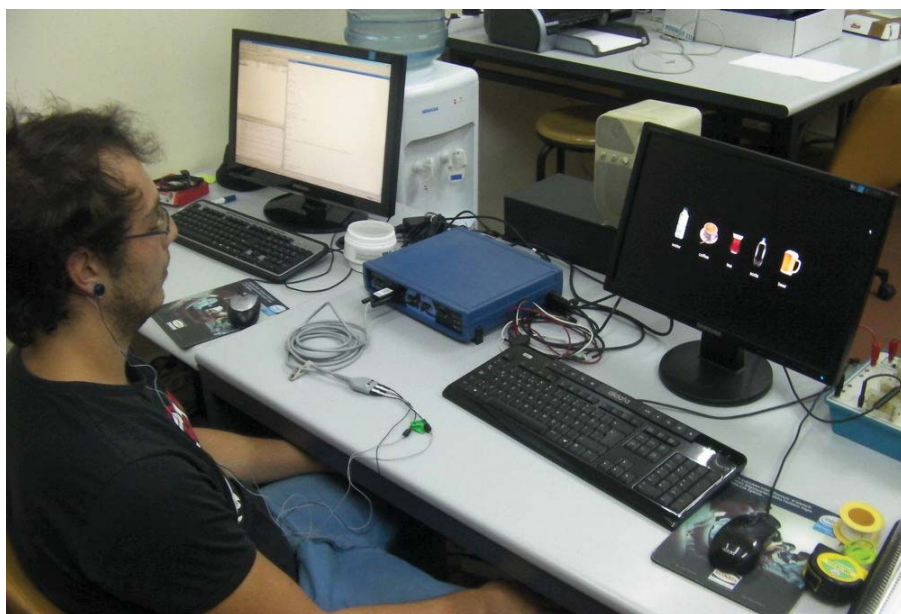


Figure 5.5: Enregistrement EEG d'utilisateur en session auditif (les yeux fermés).

La notion d'interaction en temps réel dans ce projet est basée sur la vitesse de l'algorithme de traitement EEG. Trouver des composants de P300 sur un seul essai de données, est un travail très dur avec un enregistrement EEG seul canal, pour cela, dans la plupart des cas, des algorithmes complexes et consommatrices en temps sont utilisés avec des enregistrements d'EEG en plusieurs canaux. L'objectif principal était de réduire le temps de traitement et d'obtenir une performance optimale sur l'interaction. Dans notre système, la création d'une réponse à l'utilisateur est faite après chaque séquence, parce que la moyenne des données de l'EEG est utilisée, mais le coût du temps est très faible. Dans une session d'enregistrement où la valeur 'temps' est choisie comme 10, le traitement des résultats est faite en 0,031516 secondes, en utilisant un ordinateur avec processeur 3.00GHz et 2Go de RAM.

## 6. CONCLUSION

Au cours de ce travail, on a eu une meilleure connaissance et compréhension des IHO et réalisé à quel point elles sont présentes dans notre vie quotidienne. L'IHO comme sujet, en développement en parallèle de tous les aspects du monde informatique, nous permis de comprendre que l'IHM peut également être source d'inspiration pour les nouvelles technologies.

On a voulu faire l'expérience de quelques-unes des applications des IHM. Dans un premier temps, on a décidé de travailler sur les interactions en utilisant la vidéo en temps réel. Une webcam est un outil abordable qui offre beaucoup de possibilités dans la façon de construire une interaction entre l'homme et la machine. Pour le traitement des données en temps réel, on a choisi de travailler avec l'environnement PureData qui est un outil très utile pour créer des retours audio. La plupart des temps, on a travaillé sur des interactions qui ont abouti à des créations musicales. On a reconnu si une interface est susceptible de permettre à un utilisateur de créer de la musique, alors elle peut facilement être appliquée à d'autres domaines. Un exemple fréquent : L'ordinateur est commandé par un clavier ce qui est similaire à l'utilisation d'un piano. Dans l'exemple de Reactable, qui est construit pour produire de la musique, il est aussi utilisé comme une interface visuelle.

La bibliothèque GEM a été très utile à travailler avec nos projets fonctionnant avec la webcam. Tout d'abord on a construit un contrôleur qui peut activer des notes de musique, changer d'octaves, exploiter le volume global, et sélectionner différents types de sons. Ce projet peut également s'appliquer comme un exemple de réalité augmentée car, les boîtes virtuelles, les potentiomètres qui donnent des rétroactions sonores et visuels sont est présentés sur l'écran.

On a appliqué une autre forme d'interaction basée sur la vidéo en temps réel. Cette fois, on a utilisé des objets tangibles pour créer de la musique. L'interaction est basée sur le filtrage des couleurs. Avec 3 morceaux de papier coloré, on peut avoir le contrôle de 6 paramètres de musique avec PureData. Ces paramètres peuvent être utilisés pour contrôler à peu près tout. Les expériences qu'on a eu montrent que des éléments tangibles comme des morceaux de papier coloré, conduisant l'entrée visuelle à une unité de traitement audio peut former un environnement de création de la musique pour les utilisateurs et établir un nouveau type d'interaction homme-ordinateur. On a utilisé des exemples d'un synthé similaire à Theremin et un dispositif des effets de répétition des battements. On a également travaillé sur Arduino et le fait communiquer avec PureData pour obtenir une interface musicale en fonction de panneau tactile. En fait, à titre d'exemples ICO, avec un logiciel open source comme PureData, on a créé des effets audio et de la synthèse, qui sont pas coûteuses, mais efficaces, par rapport aux appareils véritables commerciaux.

D'autre part, l'objectif était aussi de développer une interface cerveau-ordinateur qui permet aux utilisateurs de communiquer avec des ordinateurs n'utilisant que leurs pensées. Tout d'abord, afin d'avoir une meilleure compréhension des signaux du cerveau, on a créé une interface en temps réel utilisant des ondes alpha pour décoder les lettres du code Morse en contrôlant les ondes alpha avec les états mentaux détendues et normaux en utilisant yeux ouverts et fermés. Après cela, on a travaillé sur ICO basé sur potentiels évoqués. Ayant quelques expériences sur les algorithmes de l'extraction P300 en travaillant sur les bases de données existantes, on a essayé d'appliquer un nouveau système où deux types de stimuli, auditifs et visuels, peuvent être créés. On a essayé de trouver un algorithme de classification qui est 'à faible coût' en termes de temps de traitement, pour être capable de créer un ICO qui marche en ligne comme la phase prochaine des travaux. Les expériences ont montré que, en utilisant la décomposition en ondelettes et la comparaison des valeurs RMS des signaux EEG, on pourrait avoir un taux de succès de classification p300 signaux plus de 80% selon le nombre de répétitions pour une session d'enregistrement.

L'étape suivante a consisté à construire une ligne ICO en utilisant cet algorithme. On a créé un système à 2 ordinateurs, où un utilisateur peut contrôler presque tous les paramètres de la création des stimuli et les enregistrements EEG utilisant l'ordinateur premier, tandis que le deuxième ordinateur est utilisé pour enregistrer et traiter les données EEG pour maximiser la performance. On a séparé le code pour la partie de traitement des signaux EEG de sorte qu'il peut être modifié et amélioré pour différents paradigmes de virtualités évoqué, comme paradigme Speller.

Enfin, on a développé plusieurs exemples de l'interaction homme-ordinateur à la fois en utilisant des capteurs en temps réel et en utilisant des ondes EEG. On a travaillé sur plusieurs algorithmes de classification dans la partie du cerveau de l'ordinateur de travail, mais le but était d'établir un système en ligne et on s'est dirigé dans ce domaine de la recherche en utilisant nombre minimum d'électrodes, pour maximiser la vitesse de la classification et permettre à un utilisateur d'interagir avec le système en utilisant une interface agréable.

## BIBLIOGRAPHIE

- [1] <http://www.iconlogic.com/pdf/HCI.pdf>, (2006).
- [2] [http://en.wikipedia.org/wiki/File:Hollerith\\_punched\\_card.jpg](http://en.wikipedia.org/wiki/File:Hollerith_punched_card.jpg), (2008).
- [3] [http://vrlab.epfl.ch/infrastructure/images\\_infrastructure/cyberglove.jpg](http://vrlab.epfl.ch/infrastructure/images_infrastructure/cyberglove.jpg), (2005).
- [4] <http://www.archive.org/movies/thumbnails.php?identifier=AlanKeyD1987>, (2001).
- [5] Sutherland, I.E. "SketchPad: A Man-Machine Graphical Communication System.", *AFIPS Spring Joint Computer Conference*. **23**. pp. 329-346, (1963).
- [6] Kay, A., "Personal Dynamic Media.", *IEEE Computer*, **10** (3): pp. 31-42, (1977).
- [7] <http://mprove.de/diplom/gui/kay72.html>, (2007).
- [8] Alan C. Key, "A Personal Computer for Children of All Ages.", *Proceedings of the ACM National Conference, Boston Aug.* (1972).
- [9] [http://fr.wikipedia.org/wiki/Réalité\\_augmentée](http://fr.wikipedia.org/wiki/Réalité_augmentée), (2010).
- [10] Ronald T. Azuma, "A Survey of Augmented Reality", *Presence: Teleoperators and Virtual Environments* **6**, 4 355-385, (1997).
- [11] <http://www.presselite.com/iphone/metroparis/>, (2009).



- [12] <http://www.augmentedenvironments.org/lab/category/projects/arf-iphone/>, (2008).
- [13] Sergi Jordà, Günter Geiger, Marcos Alonso, Martin Kaltenbrunner, “The reacTable: Exploring the Synergy between Live Music Performance and Tabletop Tangible Interfaces”, *Music Technology Group, Pompeu Fabra University* (2007).
- [14] <http://reactivision.sourceforge.net/>, (2005).
- [15] B. Z. Allison, “P3 or Not P3: Toward a better P300 BCI,” *Ph.D. Dissertation, Univ. California, San Diego, CA, Apr.* (2003).
- [16] Michael Schmitz, Christoph Endres, Andreas Butz, “A Survey of Human-Computer Interaction Design in Science Fiction Movies”, *The Second International Conference on Intelligent Technologies for Interactive Entertainment* (2008).
- [17] L. A. Farwell and E. Donchin, “Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials.” *Electroenceph. Clin. Neurophysiol.*, vol. 70, pp. 510–523, (1988).
- [18] <http://puredata.org/>, (2003).
- [19] <http://gem.iem.at/>, (2007).
- [20] [http://en.wikipedia.org/wiki/Musical\\_Instrument\\_Digital\\_Interface](http://en.wikipedia.org/wiki/Musical_Instrument_Digital_Interface), (2010).
- [21] <http://www.propellerheads.se/products/reason/>, (2010).
- [22] <http://www.midiox.com/myoke.htm>, (2001).
- [23] <http://www.korg.com/Product.aspx?pd=278>, (2010).

- [24] [http://www.thereminworld.com/pics/Thereminists/Lev/leon\\_theremin\\_young.jpg](http://www.thereminworld.com/pics/Thereminists/Lev/leon_theremin_young.jpg), (1997).
- [25] [www.arduino.cc](http://www.arduino.cc), (2010).
- [26] Benward & Saker “Music: In Theory and Practice”, Vol. I, p.25. Seventh Edition (2003).
- [27] <http://fr.medipedia.be/uploads/editor/epilepsie-eeg.jpg>, (2010).
- [28] [http://en.wikipedia.org/wiki/File:Eeg\\_alpha.svg](http://en.wikipedia.org/wiki/File:Eeg_alpha.svg), (2005).
- [29] [http://en.wikipedia.org/wiki/File:Eeg\\_beta.svg](http://en.wikipedia.org/wiki/File:Eeg_beta.svg), (2005).
- [30] Cahn BR, & Polich J. “Meditation states and traits: EEG, ERP, and neuroimaging studies.”, *Psychological Bulletin*. 132 (2), 180-211. (2006).
- [31] [http://en.wikipedia.org/wiki/File:Eeg\\_delta.svg](http://en.wikipedia.org/wiki/File:Eeg_delta.svg), (2005).
- [32] Daubechies, I. “The wavelet transform, time-frequency localization and signal analysis.”, *IEEE Transactions on Information Theory*, vol. 36, pp. 961–1005. (1990).
- [33] <http://psychtoolbox.org/wikka.php?wakka=HomePage>, (2010).
- [34] Dewan, E. M., “Occipital alpha rhythm, eye position, and lens accommodation.”, *Nature*, 214, 975-977, (1967).
- [35] Nijboer F, Furdea A, Gunst I, et al. “An auditory brain-computer-interface (BCI).”, *Journal of neuroscience* 167:43 – 50, (2008).

- [36] Wolpaw J. R., Birbaumer N., McFarland D. J., Pfurtscheller G. and Vaughan T. M. “Brain-computer interfaces for communication and control.”, *Clin. Neurophysiol*, vol. 113, pp. 767–91, (2002).
- [37] McFarland D J, Sarnacki W A, Vaughan T M and Wolpaw J R. “Brain-computer interface (BCI) operation: signal and noise during early training sessions.”, *Clin. Neurophysiol*, vol. 116, pp. 56–62, (2005).
- [38] <http://www2.research.att.com/~ttsweb/tts/demo.php>, (2010).
- [39] Samar VJ, Bopardikar A, Rao R, Swartz K. “Wavelet analysis of neuroelectric waveforms: a conceptual tutorial.”, *Brain Lang*, vol. 66,pp. 7–60, (1999).
- [40] Donchin E, Spencer KM, Wijesinghe R. “The mental prosthesis: assessing the speed of a P300-based brain–computer interface.”, *IEEE Trans Rehabil Eng*, vol. 8 pp. 174–179, (2000).

## **BIOGRAPHIE**

Murat YILMAZ est né à Bandırma en Turquie en 1985. Il a commencé le collège à Bandırma Anadolu Lisesi. Il a continué ses études à Bandırma Anadolu Lisesi jusqu'à 2002 où il a commencé à étudier génie informatique à l'Université Galatasaray. Il a été diplômé en 2007 du Département de Génie Informatique de l'Université Galatasaray. Il continue, en ce moment, ses études en Mastère de Génie Informatique à l'Université Galatasaray.

### **Articles:**

Murat Yilmaz, Burak Arslan, “**Toward an Auditory P300 BCI**”, *International Symposium on Innovations in Intelligent Systems and Applications, 21-24 June 2010, Kayseri & Cappadocia, TURKEY*, p451, (2010).