



PROJETS SCIENTIFIQUES EN ÉQUIPE

---

# Étude cognitive : comment le cerveau encode le calcul ?

---

Méthodes et protocoles

Léa DAVENET, Léa LEFER & Joséphine SCHELLE

*Promotion 138*

Comprendre comment le calcul peut être encodé dans le cerveau est un sujet d'étude sur lequel l'intérêt porté n'a fait que croître depuis quarante ans. A travers notre projet, nous nous sommes plus particulièrement intéressées à l'étude des stratégies de calcul mental qui peuvent être utilisées pour effectuer des opérations. Les questions que nous nous sommes posées à partir de la problématique générale sont :

- Les performances en calcul mental dépendent-elles du profil académique/ professionnel des personnes ?
- Observe-t-on, comme le suggèrent nos recherches bibliographiques effectuées en amont, une utilisation significative de stratégies pour faire du calcul mental ?
- Quelles sont les stratégies de calcul mental utilisées selon les opérations proposées ?
- Quelle est la répartition de ces stratégies au sein de la population selon les opérations proposées ?
- Est-il possible de savoir quelle stratégie de calcul mental un sujet utilise sans avoir recours à une restitution orale post-tâche ?

Ces différentes questions ont permis de définir les différentes étapes de notre projet.

# Table des matières

<b>1</b>	<b>Des performances en calcul mental déterminées par le profil académique/professionnel ?</b>	<b>3</b>
1.1	Acquisition des données . . . . .	3
1.2	Traitement des données . . . . .	6
1.2.1	Étude de l'influence du type d'opération et de l'ordre des opérands sur le temps de réponse . . . . .	6
1.2.2	Étude de l'influence de l'âge sur le temps de réponse et sur le taux de succès . . . . .	7
1.2.3	Étude de l'influence du niveau d'étude sur le temps de réponse et sur le taux de succès . . . . .	8
<b>2</b>	<b>Quelles sont les stratégies de calcul mental utilisées et comment les identifier ?</b>	<b>10</b>
2.1	Acquisition des données . . . . .	10
2.2	Traitement des données . . . . .	21
2.2.1	Vérification de l'importance des stratégies pour le calcul mental	
	Identification et répartition de ces stratégies . . . . .	21
2.2.2	Connaître la stratégie de calcul mental utilisée sans avoir recours à la restitution orale post-tâche . . . . .	23

# Chapitre 1

## Des performances en calcul mental déterminées par le profil académique/professionnel ?

### 1.1 Acquisition des données

Avant de faire une étude plus approfondie sur les performances en calcul mental pour des calculs généraux, nous avons voulu évaluer les performances concernant les briques de base du calcul mental : les additions, soustractions et multiplications de deux chiffres. Pour ce faire, nous avons créé un questionnaire comportant 21 calculs mentaux élémentaires.

#### Matériel

Le questionnaire a été édité et diffusé via la plateforme de sondage LimeSurvey. L'utilisation de cette plateforme a nécessité une version "pro" dont le coût était de 29 €/mois. Le questionnaire a été laissé ouvert pendant deux semaines.

#### Sujets

177 sujets d'âges et d'horizons variés se sont portés volontaires. La participation au questionnaire a été proposée à toute la communauté de l'ESPCI, ainsi qu'à nos proches et amis extérieurs à l'ESPCI par envoi d'un lien menant au questionnaire par mail.

## Choix des tâches de calcul

Les opérations proposées sont de trois types (additions, multiplications et soustractions). Pour chaque catégorie, sept opérations sont proposées (nombre d'opérations utilisées dans l'étude de Baranyai et *al.* [1]).

Pour le choix des opérations, nous nous sommes basées sur des tables d'opérations de deux nombres à un seul chiffre les plus couramment utilisées dans les études cognitives. Les opérations proposées sont les suivantes :

**Additions :**  $8 + 9$ ,  $2 + 3$ ,  $4 + 6$ ,  $2 + 7$ ,  $5 + 8$ ,  $7 + 9$ ,  $6 + 7$ .

**Multiplications :**  $3 \times 4$ ,  $8 \times 9$ ,  $7 \times 8$ ,  $6 \times 9$ ,  $5 \times 7$ ,  $2 \times 9$ ,  $4 \times 8$ .

**Soustractions :**  $9 - 3$ ,  $7 - 5$ ,  $8 - 2$ ,  $4 - 3$ ,  $9 - 7$ ,  $7 - 4$ ,  $8 - 6$ .

Pour étudier l'impact de l'ordre des opérandes sur le temps de calcul, nous avons créé trois questionnaires différents :

**Questionnaire 1** Le premier terme (*resp.* facteur) est le plus petit des deux pour les 7 additions (*resp.* les 7 multiplications).

**Questionnaire 2** Le premier terme (*resp.* facteur) est le plus grand des deux pour les 7 additions (*resp.* les 7 multiplications).

**Questionnaire 3** Dans 4 additions sur 7, le premier terme est le plus petit des deux ; dans les 3 cas restants, le premier terme est le plus grand. Dans 3 multiplications sur 7, le premier terme est le plus petit des deux ; dans les 4 cas restants, le premier terme est le plus grand.

Pour les soustractions, nous avons imposé que le premier terme soit le plus grand des deux dans tous les cas, afin de n'avoir que des résultats positifs.

Les participants ont été répartis dans les groupes de façon aléatoire et sans avoir connaissance de l'existence des ces trois questionnaires distincts.

## Méthode/Déroulé du questionnaire

Le questionnaire s'articule de la façon suivante :

1. Présentation du projet et explication du déroulement du questionnaire
2. *Questions démographiques*
  - Âge ?
  - Niveau d'études ?
  - Dernier diplôme obtenu ?
  - Profession actuelle ou dernière profession ?
  - Présence d'une activité sur le temps libre impliquant le calcul mental ?
  - Présence de troubles de l'apprentissage ?

— Sources de déconcentration pendant le remplissage du questionnaire ?  
Aucune limite de temps n'était fixée pour répondre à ces questions.

3. *Série de 7 additions*

Le sujet doit indiquer sa réponse dans un encadré. Il ne reçoit aucune indication sur la véracité ou non de son résultat. Une minute est laissée pour répondre à chaque question ; au-delà, le questionnaire passe à la question suivante.

4. Affichage du score pour la série d'additions

5. *Visionnage d'une vidéo* de 3 minutes environ présentant le fonctionnement du cerveau : [\[Comment ça marche\] Le cerveau et les neurones](https://www.youtube.com/watch?v=hXFBWxuZIdo) (<https://www.youtube.com/watch?v=hXFBWxuZIdo>)

6. *Série de 7 multiplications*

Le sujet doit indiquer sa réponse dans un encadré. Il ne reçoit aucune indication sur la véracité ou non de son résultat. Une minute est laissée pour répondre à chaque question ; au-delà, le questionnaire passe à la question suivante.

7. Affichage du score pour la série de multiplications

8. *Visionnage d'une vidéo* de 3 minutes environ présentant la mémoire de travail : [Mémoire de travail \(Français\)](https://www.youtube.com/watch?v=fhuLdj6LaMM) (<https://www.youtube.com/watch?v=fhuLdj6LaMM>)

9. *Série de 7 soustractions*

Le sujet doit indiquer sa réponse dans un encadré. Il ne reçoit aucune indication sur la véracité ou non de son résultat. Une minute est laissée pour répondre à chaque question ; au-delà, le questionnaire passe à la question suivante.

10. Affichage du score pour la série de soustractions

11. Message de remerciement

## Données obtenues

Grâce aux fonctionnalités de LimeSurvey, nous récupérons le pourcentage d'exactitude et les temps de réponse de chaque sujet pour chaque question.

Il était difficile, dans le contexte de Covid-19, de faire des études sur différents groupes de sujets pour approfondir l'étude des performances en fonction du profil académique/professionnel, voire d'étendre cette étude pour voir si les stratégies utilisées dépendent du profil académique/professionnel. Nous nous sommes donc concentrées par la suite sur les autres problématiques et avons restreint notre choix de sujet à la communauté PCène.

## 1.2 Traitement des données

Trois types de traitements des données acquises ont été effectués selon le paramètre dont on voulait tester l'influence.

### 1.2.1 Étude de l'influence du type d'opération et de l'ordre des opérandes sur le temps de réponse

#### Données traitées

Nous nous sommes servis des colonnes suivantes du tableau de données fourni par LimeSurvey :

- ID de la réponse (pour identifier le sujet)
- Dernière page (pour savoir si le sujet est allé au bout du questionnaire)
- Numéro du questionnaire
- Résultats de chaque sujet aux 21 opérations
- Scores de chaque sujet sur chaque type d'opération (additions, multiplications et soustractions)
- Temps de réponse de chaque sujet sur chaque opération

#### Matériel/logiciel de traitement

Tableur Google Sheet

#### Méthode

- Les participants ont été regroupés par questionnaire (1, 2 ou 3).
- Nous avons calculé la **durée médiane de calcul** de chaque groupe pour chaque opération de calcul mental. Nous avons également additionné les durées médianes de chaque groupe par type d'opération (additions, multiplications et soustractions).
- Nous avons fait tracer le **temps de réponse total** (*i.e.* la somme des durées médianes) aux trois types d'opération en fonction du groupe<sup>1</sup> sous forme d'un diagramme en colonnes. Ceci doit permettre de déterminer quel type d'opération demande le plus (ou le moins) de temps de calcul et quel ordre des opérandes (pour les additions et multiplications) permet de calculer le plus rapidement.
- Pour étudier l'impact de la grandeur des termes choisis sur la durée de calcul d'une **soustraction**, nous avons fait tracer la **durée médiane de réponse** en fonction de la **somme des deux termes** et en fonction du **résultat**, sous

---

1. Pour les soustractions, les résultats des trois groupes ont été mis en commun.

forme de nuages de points.

Pour les **additions** et les **multiplications**, nous avons fait tracer le **temps de réponse total** pour un type d'opération donné en fonction du **résultat** (additions et multiplications), de la **somme des deux facteurs** (multiplications) et du **plus petit terme** (additions), et ce pour chaque groupe pris séparément et pour la moyenne des trois groupes.

## 1.2.2 Étude de l'influence de l'âge sur le temps de réponse et sur le taux de succès

### Données traitées

Nous nous sommes servis des colonnes suivantes du tableau de données fourni par LimeSurvey :

- ID de la réponse (pour identifier le sujet)
- Dernière page (pour savoir si le sujet est allé au bout du questionnaire)
- Catégorie d'âge
- Résultats de chaque sujet aux 21 opérations
- Scores de chaque sujet sur chaque type d'opération (additions, multiplications et soustractions)
- Temps de réponse de chaque sujet sur chaque opération

### Matériel/logiciel de traitement

Tableur Google Sheet

### Méthode

- Les participants ont été regroupés par **catégorie d'âge** (11-15 ans, 16-25 ans, 26-49 ans, 50-69 ans).
- On a calculé la **moyenne des résultats** et la **médiane du temps de réponse** à chaque opération de chaque catégorie d'âge.
- Connaissant la moyenne des résultats, nous avons pu en déduire le **taux de succès** de chaque catégorie d'âge, pour chaque type d'opération et pour toutes les opérations confondues. Nous avons également calculé la **somme des temps de réponse médians** de chaque catégorie d'âge, pour chaque type d'opération et pour toutes les opérations confondues.
- Sous forme d'un diagramme à colonnes, nous avons fait tracer le **temps de réponse total** (*i.e.* la somme des temps médians pour un type d'opération donné) en fonction de la **catégorie d'âge**, pour les additions, multiplications, soustractions et pour toutes les opérations confondues. Nous avons fait de même en remplaçant le temps de réponse total par le **taux de succès**.

- Sous forme d'un diagramme à colonnes, nous avons enfin fait tracer le **temps de réponse médian** à chaque opération en fonction de la **catégorie d'âge**.

### 1.2.3 Étude de l'influence du niveau d'étude sur le temps de réponse et sur le taux de succès

#### Données traitées

Nous nous sommes servis des colonnes suivantes du tableau de données fourni par LimeSurvey :

- ID de la réponse (pour identifier le sujet)
- Dernière page (pour savoir si le sujet est allé au bout du questionnaire)
- Niveau d'étude
- Diplômes
- Profession
- Résultats de chaque sujet aux 21 opérations
- Scores de chaque sujet sur chaque type d'opération (additions, multiplications et soustractions)
- Temps de réponse de chaque sujet sur chaque opération

#### Matériel/logiciel de traitement

Tableur Google Sheet

#### Méthode

- Les participants ont été regroupés par **niveau d'étude** (Bac ou moins, Bac+1, Bac+2, Bac+3, Bac+4, Bac+5, Bac+6/+7, Bac+8 ou plus).
- Nous avons calculé la **moyenne des résultats** et le **temps de réponse médian** à chaque opération de chaque catégorie.
- Connaissant la moyenne des résultats, nous avons pu en déduire le **taux de succès** de chaque catégorie, pour chaque type d'opération et pour toutes les opérations confondues. Nous avons également calculé la **somme des temps de réponse médians** de chaque catégorie, pour chaque type d'opération et pour toutes les opérations confondues.
- Nous avons alors fait tracer sous forme d'un nuage de points le **temps de calcul total** (toutes opérations confondues) et le **taux de succès** sur l'ensemble des opérations en fonction du **niveau d'étude**.
- Nous avons ensuite calculé l'**efficacité** (définie comme le rapport du **taux de succès sur le temps de calcul**) sur chaque type d'opération et sur toutes les opérations confondues de chaque catégorie. Nous avons alors fait tracer l'efficacité sur l'ensemble des calculs en fonction du **niveau d'étude**.

- Enfin, nous avons fait tracer le temps de calcul sur chaque type d'opération ainsi que le temps de calcul médian sur chaque opération en fonction du niveau d'étude.

# Chapitre 2

## Quelles sont les stratégies de calcul mental utilisées et comment les identifier ?

### 2.1 Acquisition des données

#### Matériel et logiciels

- Diaporama de 48 diapositives présentant 7 tâches de calcul réalisé et diffusé via Microsoft PowerPoint 2016.  
Couleur d'arrière-plan des diapositives : gris clair.  
Police : Arial.  
Taille de la police : 15,5 pour la diapositive de présentation, 28 et en gras pour les titres, 18 pour les diapositives de consigne, 90 et en gras pour les opérations à résoudre, 40 pour les résultats intermédiaires.  
Le diaporama complet est fourni en **Annexe 1**.
- Un poste d'ordinateur de bureau (433 mm × 243 mm)
- Fiches de suivi (une par sujet) réalisées via Google Docs.  
Un exemple de fiche complétée est fourni en **Annexe 2**.
- Tobii Eye-Tracker 5 (Disponible sur <https://gaming.tobii.com/product/eye-tracker-5>, 229 €)
- Logiciel Tobii Experience v 1.56 pour le pilotage de l'eye-tracker et une première calibration des yeux (Téléchargeable ici : <https://gaming.tobii.com/getstarted/>)
- Logiciel Tobii Ghost v.1.10.1 (Téléchargeable ici : <https://gaming.tobii.com/getstarted/>)

- Logiciel OBS Studio v 26.1.1 pour les captures d'écran et des trajets oculaires (Téléchargeable ici : <https://obsproject.com/fr/download>).

## Sujets

Nous avons travaillé sur une population constituée de 38 sujets volontaires et anonymisés, parmi lesquels figuraient 34 élèves de notre promotion et 4 encadrants de PSE.

## Tâches de calcul mental à résoudre

Les opérations de calcul mental à résoudre sont trois additions, deux multiplications et deux soustractions. Le nombre d'opérations a été fixé à sept conformément à l'étude des stratégies faite par Baranyai et *al.* [1].

Les calculs choisis pour l'expérience sont les suivants :

**Additions :**  $63 + 12$ ,  $59 + 11$ ,  $36 + 27$ .

**Multiplications :**  $16 \times 15$ ,  $99 \times 11$ .

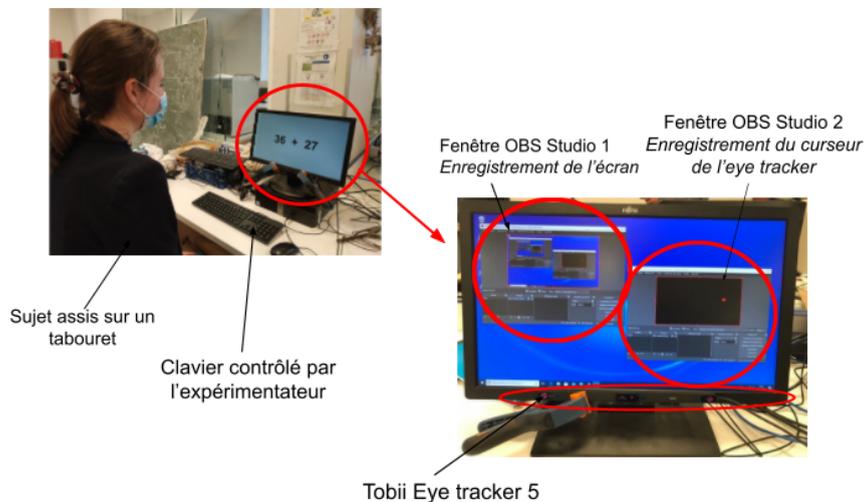
**Soustractions :**  $32 - 9$ ,  $89 - 36$ .

## Protocole

Les expériences ont toutes été réalisées dans la salle de PSE de physique de l'ESPCI.

### Avant chaque série d'expériences

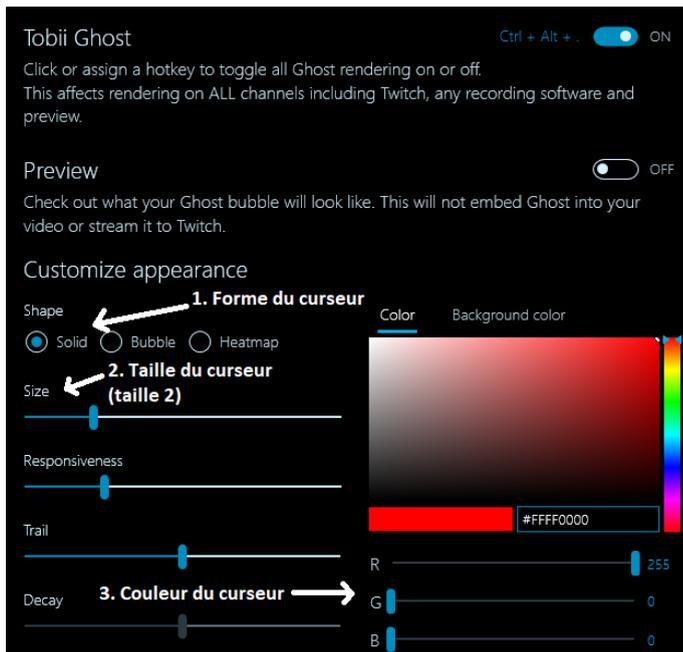
1. *Réalisation du montage expérimental*



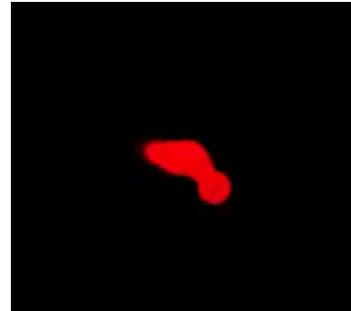
## 2. Configuration des logiciels utilisés pendant l'expérience

### — Configuration du curseur de l'eye-tracker via Tobii Ghost

Une fois la fenêtre Tobii Ghost ouverte, sélectionner “Settings”, puis “Embed Ghost into content”. Il est alors possible de configurer le type de curseur souhaité. Dans le cadre de notre expérience, nous avons fait les réglages suivants par rapport aux paramètres de base :

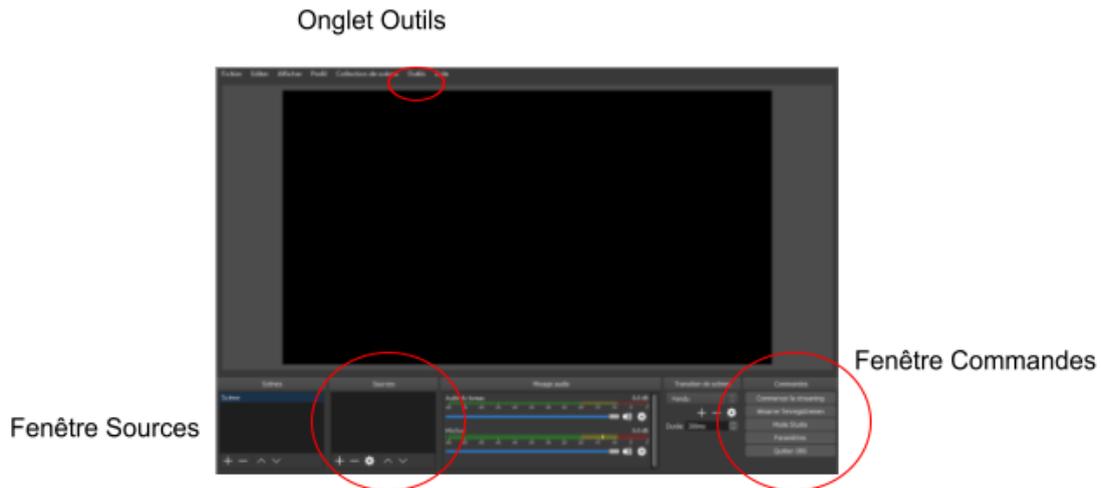


De tels réglages permettent d'obtenir le curseur ci-dessous.



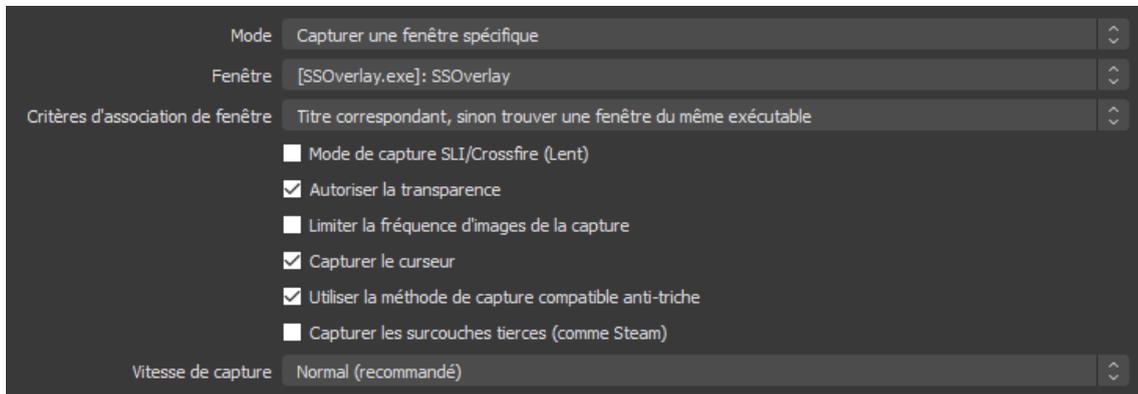
### — Configuration du logiciel OBS Studio

Dans le cadre de notre expérience, nous avons besoin d'enregistrer deux vidéos via OBS Studio : l'une du curseur de l'eye-tracker présenté ci-dessus, et l'autre de l'écran de l'ordinateur sans le curseur. On ouvre donc avant chaque série d'expériences deux fenêtres OBS Studio comme celle présentée ci-dessous. Ces deux fenêtres sont configurées séparément via l'onglet “Paramètres” dans la fenêtre “Commandes”, la fenêtre “Sources”, et l'onglet “Outils”.



CONFIGURATION DE L'ENREGISTREMENT DU CURSEUR :

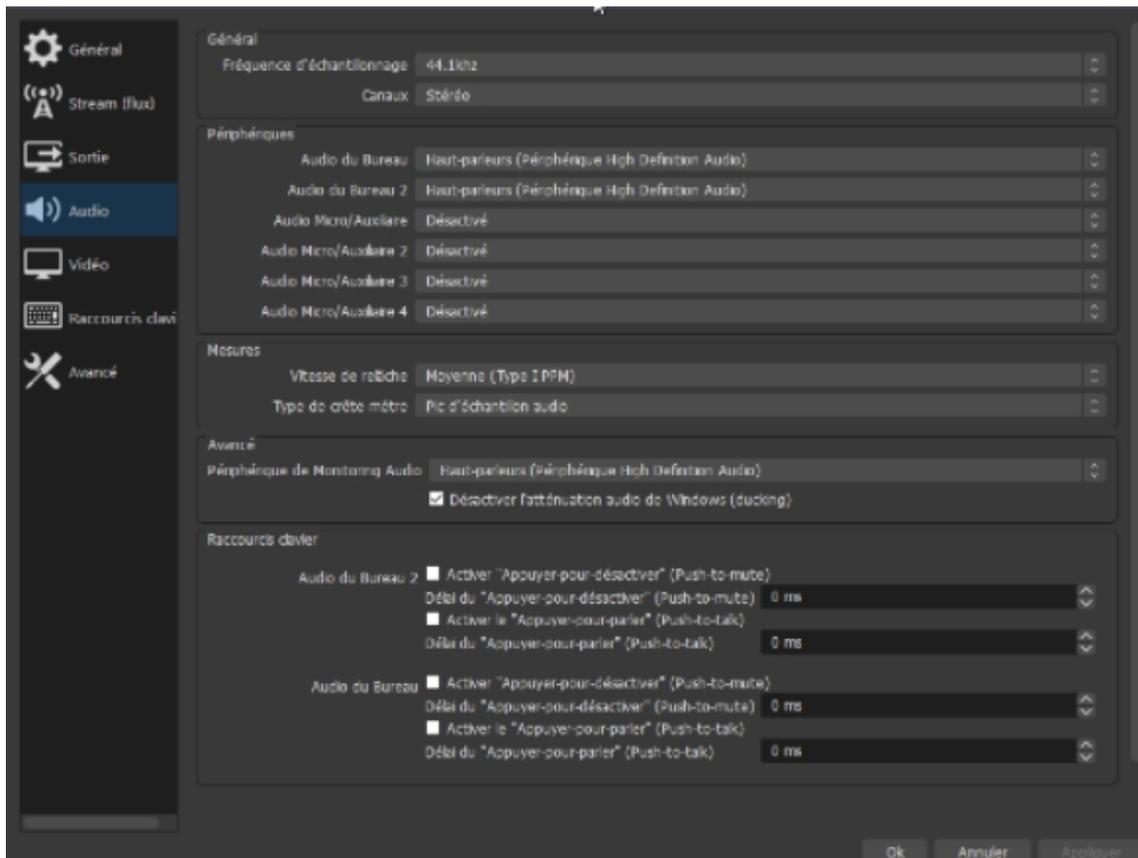
*Fenêtre "Sources"* : Ajouter "Capture de jeu", double cliquer sur "Capture de jeu" et sélectionner les paramètres suivants dans la fenêtre qui s'ouvre alors.



*Onglet "Paramètres"* : La fenêtre correspondante est la suivante.

- Dans l'onglet "Sortie", le format de l'enregistrement doit être mp4.
- Dans l'onglet "Audio", désactiver tous les périphériques.
- Dans l'onglet "Vidéo", le débit d'images doit être de 30 fps, et la résolution de sortie de 1280 × 720.
- Dans l'onglet "Raccourcis clavier", ajouter un raccourci commun pour "Démarrer l'enregistrement" et "Arrêter l'enregistrement". Nous avons choisi : Ctrl + Maj + R.
- Dans l'onglet "Avancé", indiquer le format du nom de fichier comme

*ghost\_%CCYY - %MM - %DD%hh - %mm - %ss.*



Onglet "Outils" : Sélectionner "Minuterie des sorties", et choisir "arrêter l'enregistrement après 15 minutes". Ainsi, même si on oublie d'arrêter l'enregistrement manuellement, cela sera fait automatiquement.

#### CONFIGURATION DE L'ENREGISTREMENT DE L'ÉCRAN :

Fenêtre "Sources" : Ajouter "Capture d'écran", double-cliquer sur "Capture d'écran" et sélectionner les mêmes paramètres que précédemment à l'exception de "Capturer le curseur".

#### Ongles "Paramètres" :

- Dans l'onglet "Sortie", le format de l'enregistrement doit être mp4.
- Dans l'onglet "Audio", désactiver tous les périphériques.
- Dans l'onglet "Vidéo", le débit d'images doit être de 30 fps, et la résolution de sortie de 1280 × 720.
- Dans l'onglet "Raccourcis clavier", ajouter un raccourci commun pour "Démarrer l'enregistrement" et "Arrêter l'enregistrement". Nous avons choisi :

Ctrl + Maj + R.

- Dans l'onglet "Avancé", indiquer le format du nom de fichier comme `screen_%CCYY-%MM-%DD%hh-%mm-%ss`.

Onglet "Outils" : Sélectionner "Minuterie des sorties", et choisir "arrêter l'enregistrement après 15 minutes". Ainsi, même si on oublie d'arrêter l'enregistrement manuellement, cela sera fait automatiquement.

**ATTENTION** : Il faut réduire les deux fenêtres ouvertes dans OBS Studio avant de commencer tout enregistrement, sinon une seule vidéo est enregistrée !

### Pendant chaque expérience

Chaque expérience dure une dizaine de minutes en moyenne. Le sujet est assis sur un tabouret en face de l'écran d'ordinateur. Il n'a pas le contrôle du clavier, qui est entièrement géré par l'expérimentateur. Une expérience se déroule en plusieurs étapes :

1. *Première calibration du regard du sujet*

Cette étape de calibration est entièrement conçue par les développeurs de Tobii et est faite via le logiciel Tobii Expérience. Durant cette étape, le sujet ne doit pas quitter l'écran des yeux. Des points bleus apparaissent les uns après les autres à différents endroits sur l'écran. Le sujet doit fixer ces points bleus jusqu'à ce qu'ils explosent. Cela permet ainsi de faire en sorte que l'eye-tracker s'adapte au regard du sujet, et qu'il le suive le mieux possible dans la suite de l'expérience. Une fois la calibration effectuée, le réglage de l'eye-tracker ainsi fait est sauvegardé pour le sujet en question en créant un profil utilisateur pour celui-ci. Ainsi, il suffit d'ouvrir le profil correspondant au sujet interrogé sur Tobii Expérience avant de lancer une expérience, pour qu'on soit sûr que le réglage de l'eye-tracker soit le plus adapté possible.

Une fois la calibration effectuée pour le sujet, on ferme Tobii Expérience.

Pour les étapes suivantes, l'expérimentateur diffuse, via Microsoft PowerPoint, un diaporama en plein écran sur l'écran de l'ordinateur dont le contenu va être détaillé pas à pas. Les enregistrements des deux vidéos (celles du curseur de l'eye-tracker et celle de l'écran) sont lancés dès le lancement du diaporama.

2. *Lancement des enregistrements, explication du but du projet ainsi que du déroulement de l'expérience par l'expérimentateur*

Les diapositives diffusées sont les suivantes : Le temps donné pour cette étape est libre. Le sujet est invité à lire les explications et à poser des questions.

### Pilote

#### BLAPLA PSE

Bienvenue sur le projet **Blapla** qui a pour but de comprendre comment le cerveau encode le calcul.

Cette expérience vise à évaluer votre rapidité autant que vos réponses sur des opérations de base i.e. additions, soustractions et multiplications de deux chiffres.

Trois groupes de questions se succéderont à l'écran. Le premier groupe comporte deux additions. Le deuxième regroupe deux multiplications. Le dernier mélange deux soustractions.

Bien entendu, vous résoudrez ces calculs **sans aide et sans calculatrice**.

Cette expérience étant basée sur l'eye-tracking, veuillez à bien regarder l'écran tout au long des calculs. Vous pouvez cependant cligner des yeux.

Voilà vous savez tout, à vous de jouer maintenant !

L'expérimentateur ajoute à ceci le fait qu'on observe les stratégies qui vont être utilisées pour résoudre les différentes opérations. Il précise également les deux temps qui suivront chaque tâche de calcul :

- Une diapositive présentera tous les nombres impliqués dans les différentes stratégies possibles pour résoudre l'opération en question. Le sujet est invité à "chercher et regarder de manière attentive les nombres impliqués dans la décomposition dont il s'est servi".
- Le sujet sera ensuite invité à restituer oralement son résultat et la décomposition qu'il a utilisée.

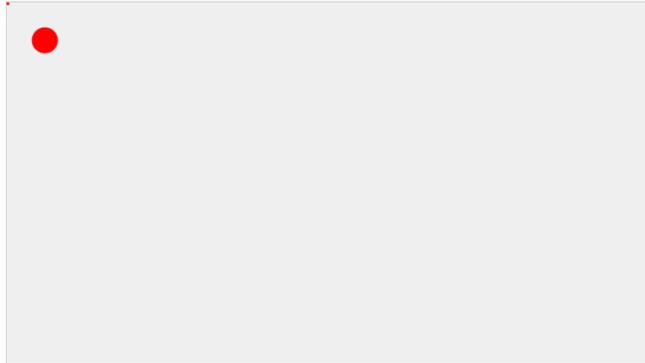
3. *Deuxième calibration du regard du sujet* Cette deuxième calibration est cette fois nécessaire au traitement des données par la suite. Elle permet en effet de pouvoir superposer les deux vidéos enregistrées (celle du curseur et celle de l'écran) pour savoir à quel endroit de l'écran correspond la position du curseur qui suit le regard du sujet à chaque instant. Une diapositive de consigne est d'abord présentée (temps libre pour la lecture des consignes par le sujet).

#### Etape 1 : Calibration à 12 points

Fixer le point rouge sur chacune de ses positions (1 seconde par position).  
Les points défilent automatiquement, inutile d'appuyer sur la touche pour passer à la slide suivante entre deux points.

Passer à la slide suivante pour commencer.

Douze diapositives comme celle présentée ci-dessous s'enchaînent ensuite. Chacune de ces diapositives fait apparaître un disque rouge placé à nouvel endroit de la diapositive. Le sujet fixe à chaque fois le point. Le diaporama est ici chronométré, chaque point n'apparaît que 3 secondes à l'écran.



#### 4. *Résolution des tâches de calcul mental*

Le sujet entre alors dans le vif de l'expérience et se trouve confronté à 7 tâches de calcul mental. La première est un exemple, non analysé, qui lui permet de bien comprendre le fonctionnement de cette étape de l'expérience et de poser les éventuelles dernières questions.

L'étape de résolution des tâches de calcul mental est divisée en trois parties : une partie addition, une partie multiplication et pour finir une partie soustraction. Au début de chaque étape, une diapositive de consigne comme celle ci-dessous est présentée au sujet. Le temps qui lui est donné pour lire ces consignes est libre.

**Etape 2 : Additions**

Vous allez devoir résoudre deux additions. Pour chaque addition :

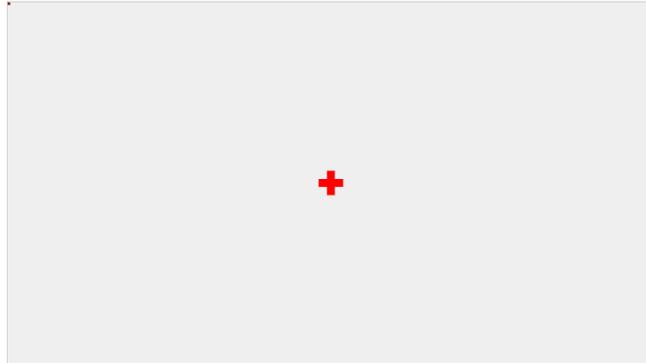
- Fixer la croix au centre de l'écran. L'addition apparaît 1 seconde plus tard.
- Résoudre mentalement l'addition sur la première slide.
- Passer à la slide suivante.
- Observer les nombres sur la deuxième slide.
- Dès que l'addition réapparaît, énoncer le résultat de l'addition et expliquer la décomposition du calcul.

Passer à la slide suivante pour passer à l'exemple.

L'exemple est donné dans la partie additions, juste après la diapositive de consigne. Nous nous appuyons sur cet exemple dans ce document pour présenter le protocole commun à toutes les tâches de calcul que le sujet aura à résoudre durant l'expérience.

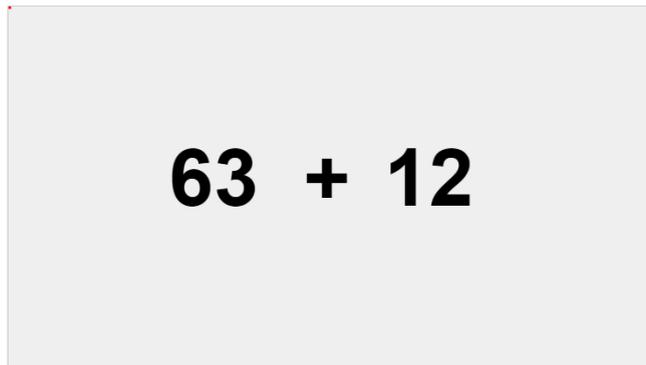
#### TROISIÈME CALIBRATION DU REGARD DU SUJET

Cette étape a pour but de vérifier que l'eye-tracker suit toujours bien le regard du sujet avant de commencer chaque tâche de calcul mental. Le sujet fixe une croix rouge qui apparaît au centre de l'écran pendant 3 secondes.



#### RÉSOLUTION DE LA TÂCHE DE CALCUL MENTAL

L'opération à résoudre apparaît à l'écran, le temps laissé au sujet pour la résoudre est libre. Il signale à l'expérimentateur lorsqu'il a le résultat, et celui-ci passe à la diapositive suivante. Le sujet garde son résultat en tête.



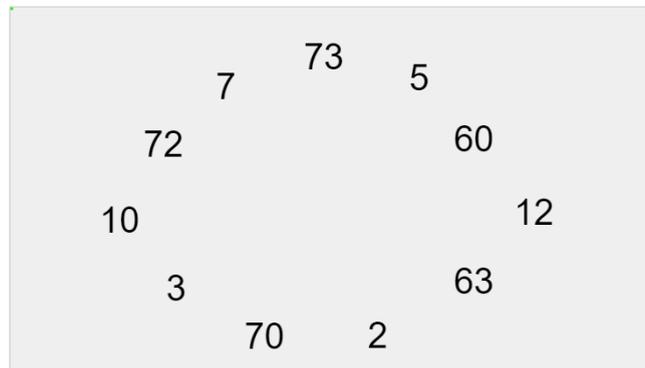
#### FIXATION DES RÉSULTATS INTERMÉDIAIRES INTERVENANT DANS LA STRATÉGIE UTILISÉE

Les résultats intermédiaires intervenant dans les différentes stratégies possibles pour résoudre l'opération sont présentés en cercle/ellipse sur une diapositive. Ils sont régulièrement espacés. Cette diapositive reste affichée 15 secondes. Le sujet cherche et fixe plus attentivement les résultats intermédiaires qui interviennent dans la décomposition qu'il a utilisée.

**N.b.** : Les résultats intermédiaires pour chaque tâche de calcul ont été choisis en s'appuyant sur les stratégies identifiées pour chaque type d'opérations (additions, multiplications et soustractions) dans l'étude de Baranyai et *al.* [1]. Nous avons effectué chaque opération en appliquant chacune des stratégies et avons relevé les résultats intermédiaires impliqués.

*Exemple pour  $63 + 12$ , en utilisant la stratégie "splitting" :*

$63 + 12 = (60+10) + (3+2) = 70 + 5$ . On relève donc 60, 10, 3, 2, 70 et 5.



RESTITUTION ORALE DE LA STRATÉGIE UTILISÉE ET DU RÉSULTAT OBTENU

L'opération apparaît une nouvelle fois à l'écran. Le sujet est invité à donner le résultat de cette opération, et à détailler la décomposition du calcul qu'il a faite pour l'obtenir. Le temps donné pour cette restitution est libre.



Durant cette restitution, l'expérimentateur inscrit les résultats et la décomposition du calcul sur une feuille de suivi dont le modèle est donné ci-dessous. Le modèle est le même pour chacun des sujets.

**FICHE DE SUIVI**  
Expérience pilote

Identifiant du participant : \_\_\_\_\_ Date : \_\_\_\_\_  
Profession du participant : \_\_\_\_\_ Test utilisé : \_\_\_\_\_  
Consigne: \_\_\_\_\_

Avant de procéder au test, vérifier les points suivants :

- o La personne est assise.
- o L'eye-tracker détecte son regard.
- o Tobii Ghost est lancé et les paramètres du curseur sont correctement réglés.
- o Deux interfaces d'OBS Studio sont lancées et les paramètres d'enregistrement sont correctement réglés.
- o Le diaporama du test est diffusé en mode plein écran.
- o Les deux enregistrements sont lancés en utilisant la combinaison *Ctrl + Alt + R*.

Résultats du participant :

Opération	Résultat	Stratégie utilisée
59+11		
36+27		
16×15		
99×11		
32-9		
89-36		

Observations de l'expérimentateur : \_\_\_\_\_

Une fois l'exemple terminé, les 6 tâches de calcul mental (2 additions, 2 multiplications et 2 soustractions) s'enchaînent exactement selon le même protocole que l'exemple.

5. *Arrêt des enregistrements et retour sur l'expérience par le sujet*

Après la restitution portant sur la dernière soustraction, le diaporama s'arrête sur cette diapositive :



L'expérimentateur arrête alors les enregistrements. Le sujet est invité à faire des remarques sur le déroulement de l'expérience (notamment le temps laissé pour les différentes tâches, la clarté des consignes, l'intérêt porté à ces tâches et la facilité à expliciter ses décompositions).

## 2.2 Traitement des données

Deux types de traitement des données acquises lors de ces expériences ont été effectués selon la question à laquelle nous voulions répondre.

### 2.2.1 Vérification de l'importance des stratégies pour le calcul mental

#### Identification et répartition de ces stratégies

##### Données traitées

Nous nous servons des décompositions de calcul faites par chaque sujet, pour chaque tâche de calcul mental et que nous avons restituées à l'écrit sur les fiches de suivi.

##### Matériel/logiciels de traitement

- Tableur Google Sheet
- Microsoft Excel 2016

##### Méthode

- À partir de la description des différentes stratégies présentées dans l'étude de Baranyai et *al.* [1], nous avons traduit chacune des décompositions de calcul restituée par les sujets en stratégie. Celles-ci, ainsi que les résultats qu'elles permettaient d'obtenir ont été ensuite regroupées pour chaque sujet pour

chaque tâche dans un tableau récapitulatif.

La traduction des décompositions en stratégies a été faite en suivant ces règles :

**Additions et soustractions** —

- SPLITTING : le sujet décompose chacun des deux nombres de l'opération en sous-somme (*exemple* :  $36 + 27 = (30+20) + (6+7) = 50+13 = 63$ )
- STRINGING : le sujet décompose un seul des deux nombres en sous-somme (*exemple* :  $36 + 27 = 36 + (20+7) = 56+7 = 63$ )
- BRIDGING THROUGH TEN : le sujet décompose un des deux nombres pour compléter le second jusqu'à la prochaine dizaine (*exemple* :  $36 + 27 = (36+24) + 3 = 60+3 = 63$ )
- COMPENSATION : le sujet passe un des deux nombres à la dizaine la plus proche et compense ensuite cela (*exemple* :  $36 + 27 = 40 + (27-4) = 40+23 = 63$ )
- BALANCING : le sujet ajoute ou soustrait un chiffre pour mener un des deux nombres à la prochaine dizaine, puis compense ce qu'il a ajouté/soustrait (*exemple* :  $36 + 27 = (36+4) + (27-4) = 40+23 = 63$ )

**Multiplications** —

- DOUBLING AND HALVING : le sujet double un des deux nombres puis prend la moitié du produit obtenu (*exemple* :  $15 \times 16 = 30 \times \frac{16}{2} = 30 \times 8 = 240$ )
- SPLITTING : le sujet décompose un des deux nombres en sous-somme (*exemple* :  $15 \times 16 = 10 \times 16 + 5 \times 16 = 160 + 80 = 240$ )
- COMPENSATION : le sujet multiplie un des deux nombres par un nombre qui l'arrange puis compense ensuite cela (*exemple* :  $15 \times 16 = 15 \times 20 - 15 \times 4 = 300 - 60 = 240$ )
- FACTORIZATION : le sujet décompose un des deux nombres en sous-multiplication (*exemple* :  $15 \times 16 = 3 \times (5 \times 16) = 3 \times 80 = 240$ )

Les stratégies qui regroupaient plusieurs des stratégies décrites ci-dessus ont été regroupées sous le nom "Combination of methods". En ce qui concerne les rares stratégies non répertoriées parmi celles décrites plus haut, nous les avons regroupées sous l'appellation "Other strategies".

Quand le sujet réalisait le calcul écrit dans sa tête (soit une absence de stratégie finalement), on parlait de "written algorithm". Enfin, quand un sujet ne savait pas restituer sa stratégie, ou disait "avoir vu écrit le résultat dans sa tête", nous appelions cela "no strategy" (nous avons préféré distinguer cela de "written algorithm", mais nous aurions pu les rassembler).

- Pour chacune des 6 tâches de calcul analysées, nous avons calculé le pourcentage de sujets utilisant chacune des stratégies décrites ci-dessus. Nous avons ensuite fait de même pour chaque type d'opérations (addition, soustraction et multiplication).

**N.b.** : Quand un sujet n'obtenait pas le bon résultat en utilisant une stratégie, nous n'avons pas pris en compte la donnée pour ce sujet et pour cette tâche dans l'analyse de la répartition des stratégies au sein de la population des sujets.

## 2.2.2 Connaître la stratégie de calcul mental utilisée sans avoir recours à la restitution orale post-tâche

On a cherché à savoir si on pouvait connaître la stratégie utilisée par un sujet sans avoir à lui demander de détailler la décomposition qu'il avait effectuée, en analysant uniquement le parcours de son regard sur l'écran :

- d'une part, pendant qu'il résolvait la tâche de calcul mental ;
- d'autre part, après la tâche, lorsqu'on lui présentait les résultats intermédiaires intervenant dans différentes stratégies qui peuvent être utilisées pour cette tâche.

Nous avons effectué plusieurs niveaux d'analyse.

Traitement des deux vidéos obtenues pour chaque sujet grâce au code  
Matlab Pipeline\_1<sup>1</sup>

### Données traitées

- Enregistrement vidéo de l'écran de l'ordinateur pendant l'expérience pour chaque sujet
- Enregistrement vidéo du curseur de l'eye-tracker pendant l'expérience pour chaque sujet

Un exemple de vidéos obtenues pour chacun de ces deux types de vidéos est fourni en annexe.

### Matériel/logiciel de traitement

Matlab (version R2020b ; utilisation de l'app "Signal Analyser")

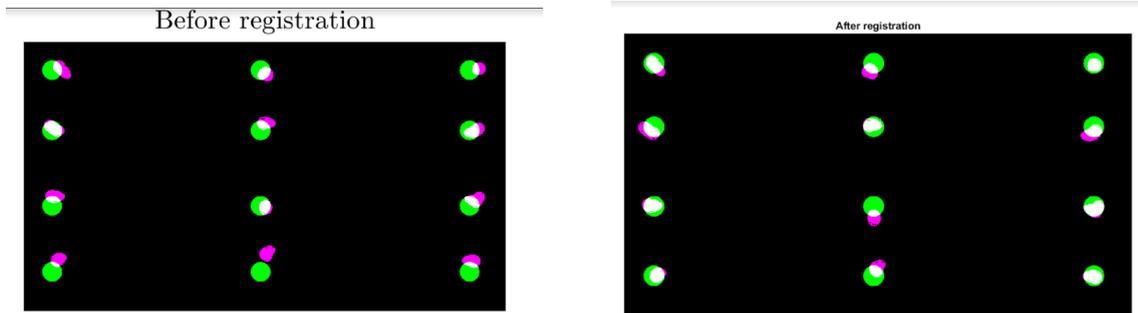
---

1. Ce code figure parmi les annexes à ce document.

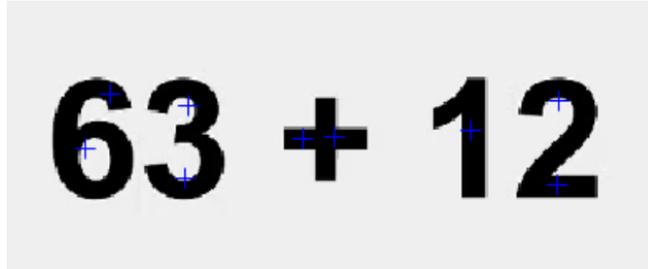
## Méthode

Le schéma global de l'analyse menée par le code est le suivant :

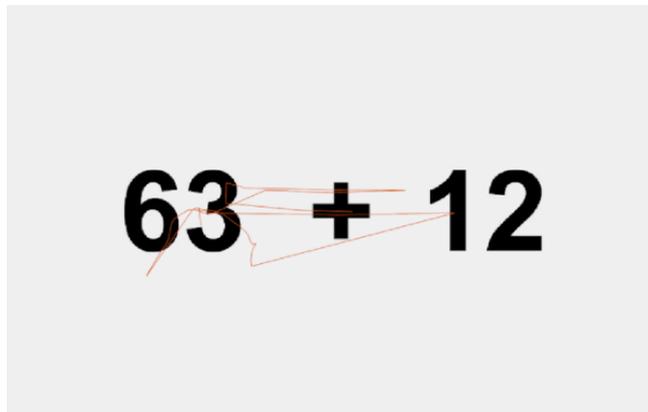
- **Extraction des propriétés des deux vidéos** (durée de chacune des deux vidéos, dimensions verticale et dimension horizontale des images, fréquence d'échantillonnage de chacune des deux vidéos : 30 images par seconde pour nos expériences).
  
- Première lecture des deux vidéos pour **détecter les instants des transitions, i.e. les instants où on change de diapositive**. Le code détecte en fait le changement de couleur (changement d'intensité) d'un petit carré placé dans le coin en haut à gauche de chacune des diapositives. La même couleur a été choisie pour les diapositives correspondant au même type de tâche de l'expériences (carré bleu pour les diapositives de consigne et de présentation, carré rouge pour les diapositives de calibration à douze points, carré rouge foncé pour les diapositives de calibration avec la croix, carré rouge pour les diapositives présentant l'opération pour la première fois, carré vert pour les diapositives de résultats intermédiaires et carré bleu pour les diapositives présentant l'opération pour la seconde fois).
  
- **Amélioration manuelle de la calibration de l'eye-tracker** à l'aide des douze points. Sur toute la durée identifiée comme étant celle de la calibration à douze points, le code trouve et donne les coordonnées des points. Les coordonnées du curseur lors de l'affichage de chaque point sur le diaporama sont également donnés par le code. Celui-ci superpose alors les 12 positions des points et les 12 positions correspondantes du curseur sur une même image. On clique alors sur 4 positions différentes de points, puis sur les 4 positions correspondantes du curseur dans le même ordre. Le code implémente alors une transformation affine qui permet de superposer au mieux la position du curseur et la position du point pour chacun des 4 points rouges. Cette transformation affine est également appliquée aux autres points rouges. On recommence l'opération en choisissant à chaque fois 4 positions de points parmi les 12 jusqu'à obtenir la meilleure superposition possible de toutes les positions de curseur et de points.  
Un exemple de ce qu'on obtient avant et après cette calibration est donné ci-dessous :



- **Application de la transformation affine** définie lors de la calibration à l'ensemble de l'enregistrement du curseur sur toute la durée de l'expérience. Pour cela, le code doit relire l'enregistrement vidéo du curseur une seconde fois.
- L'analyse des données et l'affichage des résultats vont alors **être décomposés en trois parties** : **"task"** (lorsque le sujet résout l'opération pour chaque tâche de calcul mental), **"post"** (lorsque le sujet regarde les résultats intermédiaires après chaque tâche de calcul mental), et **"cross"** (la calibration avant chaque tâche de calcul mental). Le code commence par **repérer sur toute la durée des enregistrements les intervalles de temps** correspondant à "task", "post", et "cross". À partir de ce moment, l'analyse se concentre uniquement sur ces intervalles de temps.
- **Acquisition des résultats pour "task"**  
Pour chaque tâche de calcul mental, on obtient les résultats suivants :
  - *"Targets"* : Après avoir identifié les changements de diapositive et les différents moments de l'expérience, notre code identifie les chiffres et les opérateurs présents sur la diapositive, en se basant sur les différences de couleur des groupes de pixels. Il assimile ainsi les chiffres et les opérateurs à des motifs plus simples : des croix. Il affiche une superposition de la diapositive pour la tâche analysée et des croix identifiées. Ce résultat est un moyen pour nous de contrôler que notre code analyse bien la bonne partie de la vidéo et qu'il le fait correctement : la position des croix doit correspondre à celle des chiffres et des opérateurs.



- "*Simple trajectory*" : Les positions du curseur pendant toute la durée de la tâche sont obtenues via le code et tracées par dessus la diapositive.



- "*Simple heatmap*" : Notre code génère également une carte de chaleur en estimant les densités de probabilité de regarder à un endroit précis de l'écran par noyau gaussien. Cela nous permet de savoir en un coup d'œil quelle partie de l'opération le sujet a le plus regardé (en nombre d'occurrences du curseur aux différents endroits de l'écran).

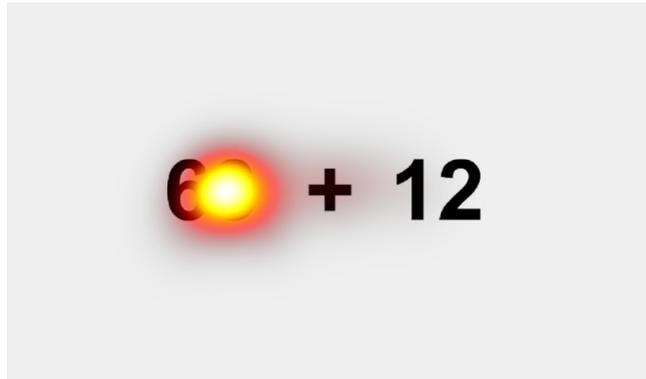
Le filtre passe-bas gaussien utilisé est d'espérance 500 et d'écart-type 50. La gamme de couleur utilisée pour créer ces heatmaps est le tableau de couleurs "hot" de Matlab.

La lecture de la heatmap se fait ainsi :

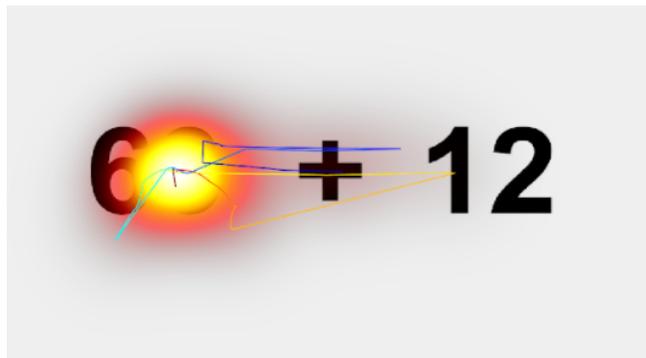
- Absence de taches sur l'écran : La fréquence à laquelle le sujet a regardé cet endroit est inférieure à la fréquence d'échantillonnage des vidéos.
- Taches visibles sur l'écran : Le sujet regarde d'autant moins fréquemment à cet endroit que la tache est rouge pâle ; il regarde d'autant plus souvent à cet endroit que la tache est blanche et éclatante.

**N.b.** : La taille des pixels de l'écran limite la résolution de la heatmap. Il faut donc veiller à espacer et agrandir suffisamment les digits et les opérateurs lorsque le diaporama est créé, et ne pas choisir une taille de

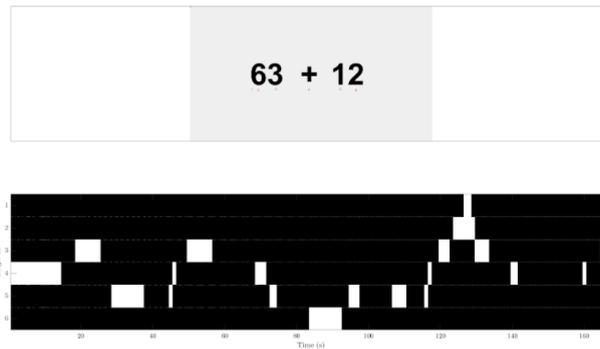
curseur de l'eye-tracker trop importante lors de la configuration de celui-ci.



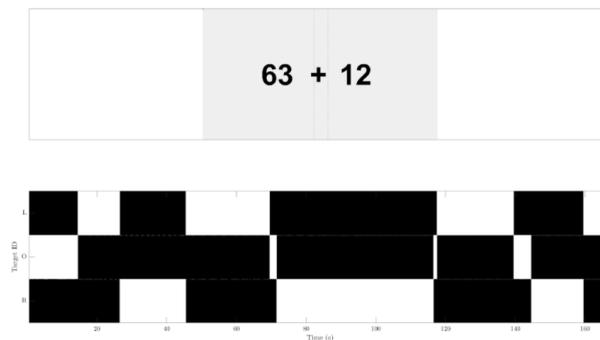
- *"Trajectory and heatmap"* : Notre code génère également une superposition de la heatmap et du tracé de la trajectoire du regard du sujet.



- *"Word Analysis"* : Comme expliqué précédemment, les taches de la heatmap n'offrent pas une résolution optimale et pour une analyse plus quantitative et précise, nous avons cherché à retranscrire le trajet oculaire sous forme d'un mot. Pour cela, nous avons attribué à chaque digit et à l'opérateur un chiffre. A chaque instant, le code détermine de quel élément le regard du sujet est le plus proche et retranscrit cette information en ajoutant le chiffre correspondant au mot formé. Pour cela, le code s'appuie sur les positions des croix définies à l'étape "Targets". Il regarde ensuite, à chaque instant, si le curseur est à une distance inférieure à 40 pixels d'une de ces croix ; si oui, il considère alors que le sujet regarde le "chiffre" lié à cette croix (soit un des digits à l'écran ou l'opérateur). On obtient alors une figure comme celle présentée ci-dessous, qui traduit le temps passé par le sujet sur chaque mot et à quels instants de l'expérience.



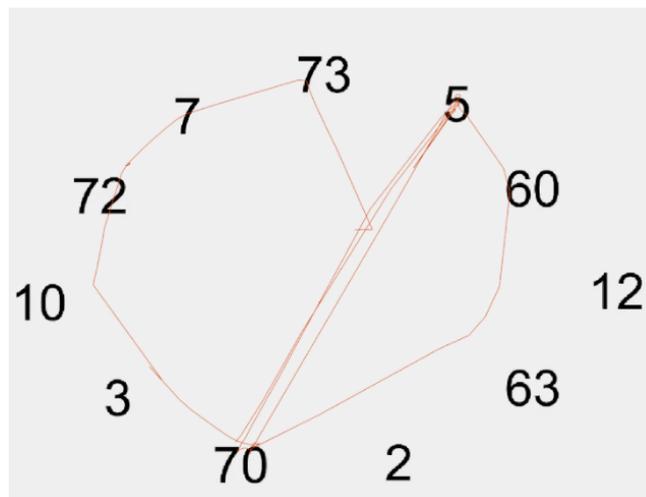
- *"Trinary Word Analysis"* : Le principe est le même que précédemment, sauf que les lettres du mot sont à présent respectivement la partie gauche, la partie centrale et la partie droite de la diapositive affichée à ce moment. Le découpage est fait de sorte à ce que la partie centrale ne prenne en compte que la bande dans laquelle se trouve l'opérateur. Le code repère ensuite la position du curseur à chaque instant de l'expérience.



- *"Task Object"* : Le code enregistre enfin un fichier de données Matlab résumant tous les résultats acquis pendant l'analyse de "task" pour la tâche de calcul mental en question, sous la forme de tableaux de données (positions du curseur, couleurs de la heatmap, positions des digits et de l'opérateur...).
- **Acquisition des résultats pour "post"**  
On obtient quasiment les mêmes types de résultat que pour "task", selon la même méthodologie que celle qui vient d'être décrite. Les résultats en commun avec ceux de "task", ainsi que des exemples d'affichage obtenus sont présentés ci-dessous :
  - *"Targets"*



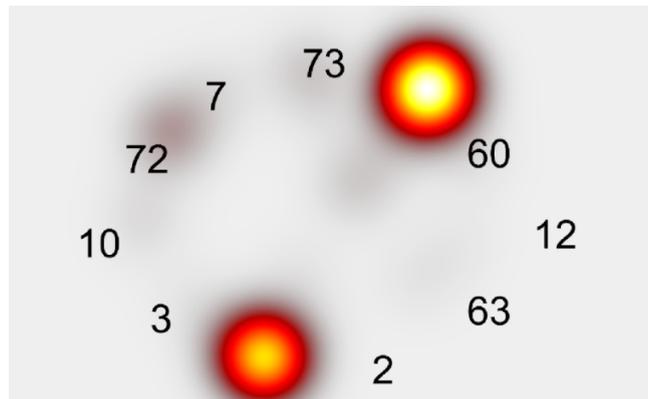
— *"Simple Trajectory"*



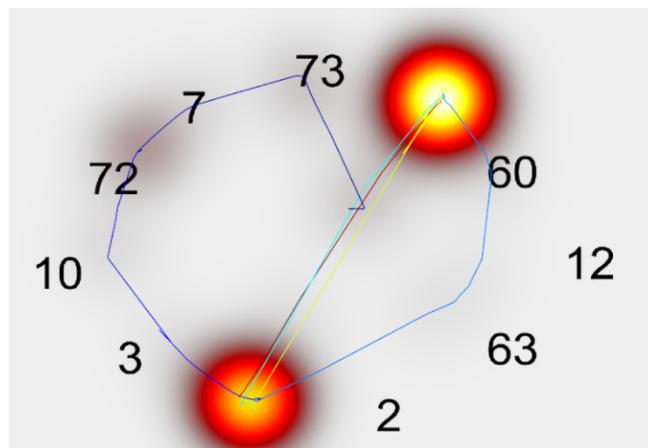
— *"Simple Heatmap"* : Pour "post", comme pour "task", il faut veiller à espacer suffisamment les digits à l'écran pour avoir une bonne résolution sur la heatmap, sans pour autant trop les espacer et augmenter le temps que le sujet pourrait mettre à passer d'un nombre à l'autre.

## Étude cognitive : comment le cerveau encode le calcul ?

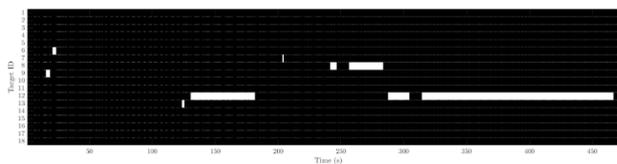
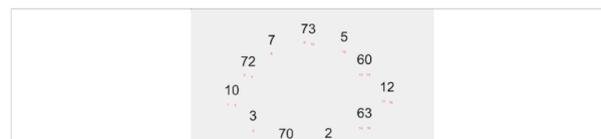
---



— "Trajectory and heatmap"

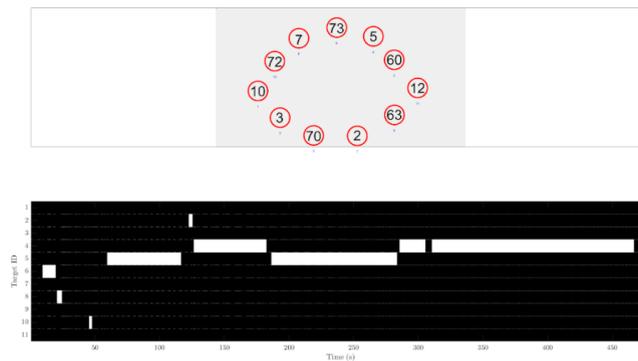


— "Word analysis" : Pour "post", la limite pour savoir si le sujet regarde tel ou tel partie de l'écran n'est plus 40 pixels (car les caractères ne peuvent pas être aussi espacés que pour "task"), mais de 20 pixels.



— *"Post Object"*

Pour “post”, on remplace cependant “Trinary Word Analysis” par “Cluster Word Analysis”. Le principe est le même que pour “Word analysis”, mais les différents caractères identifiés lors de cette étape sont rassemblés en clusters correspondant aux résultats intermédiaires. La limite pour savoir si le sujet regarde tel ou tel cluster est ici fixée à 50 pixels. On obtient le même type de résultat que ci-dessous.



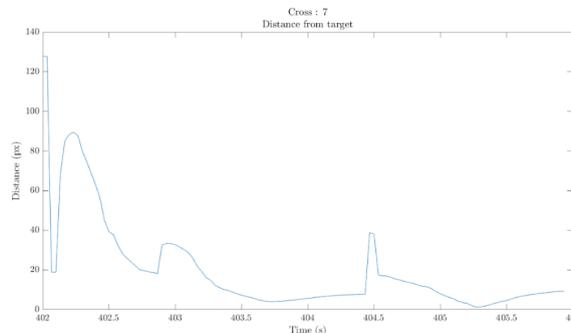
— **Acquisition des résultats pour “cross”**

Pour chaque tâche de calcul mental, les résultats obtenus pour “cross” ont pour but de vérifier tout au long de l’analyse si la transformation affine à l’étape d’amélioration manuelle de la calibration de l’eye-tracker” est appliquée et si elle permet d’avoir une bonne superposition du curseur et de l’écran. Voici les trois résultats obtenus pour cette partie :

— *"Superposition target and observation"* : L’image affichée superpose la diapositive présentant la croix et la position du curseur à ce moment avant et après la transformation affine. Si la transformation affine est bonne, les deux sont censés être superposés puisque le sujet est censé fixer la croix à cette étape.



- "*Distance from target*" : Cet affichage donne la position du curseur de l'eye-tracker par rapport à la croix au cours du temps pendant les quelques secondes que durent la partie "cross". Là encore, c'est un moyen pour nous de contrôler que les positions du regard sur l'écran données par le code sont cohérentes.



- "*Cross Object*" : Comme pour "post" et "task", il s'agit d'un fichier de données Matlab résumant tous les résultats acquis pendant l'analyse de "cross" pour la tâche de calcul mental analysée.

Approche probabiliste 1\_ Clustering des ratios de temps passé sur chaque partie de l'écran (ratios L/O/R) pendant la tâche

### Données traitées

- Les fichiers de données Matlab "Task Object" pour 35 de nos sujets (et plus précisément les données liées à "Trinary Word Analysis" dans ces fichiers)
- Stratégies établies pour chaque sujet, pour chaque tâche de calcul

### Matériel/logiciels de traitement

- Matlab (version R2020b; utilisation de l'app "Signal Analyser")
- Microsoft Excel 2016

### Méthode (pour chacune des 6 tâches de calcul mental de l'expérience)

- **Traitement** des fichiers "Task Object" pour la tâche de calcul mental en question via le code Matlab "Trinary Word Analysis" fourni en annexe. Ce code utilise la fonction "regexpdir", également fournie en annexe, qui permet de sélectionner tous les fichiers "task" de tous les sujets, pour la tâche souhaitée, dans le dossier regroupant toutes les données acquises via le code Matlab

“Pipeline\_1”.

Il accède en particulier aux données liées à “Trinary Word Analysis” et calcule pour chaque sujet le temps passé sur la partie gauche de l’écran, le temps passé sur la partie droite de l’écran et le temps passé au centre de l’écran.

- On trace ensuite pour chaque sujet les **diagrammes circulaires représentant ces ratios L/O/R** calculés.
- On **classe ces diagrammes en 6 catégories (LOR, LRO, OLR, ORL, RLO, ROL)** de la façon suivante : si le sujet regarde davantage la partie gauche, que les parties droite et centrale, et qu’il regarde davantage la partie centrale que la partie droite, alors on classe le diagramme correspondant dans la catégorie LOR. Ainsi de suite, pour les autres catégories.
- Pour chaque catégorie, on **attribue à chaque diagramme la stratégie** correspondante au sujet dans le tableau récapitulatif des stratégies établies au début de la phase de traitement des données.
- On compte le nombre de stratégies différentes dans chaque catégorie, et on **calcule la probabilité d’avoir chacune des stratégies selon la catégorie** pour savoir si notre clustering basé sur les ratios L/O/R est comparable à celui basé sur les stratégies utilisées.

## Approche probabiliste 2\_ Clustering des heatmaps post-tâches

### Données traitées

- Les images “simple heatmap” acquises pour “post” pour 35 de nos sujets réparties selon les tâches de calcul mental
- Stratégies établies pour chaque sujet, pour chaque tâche de calcul

### Matériel/Logiciel de traitement

Microsoft Excel 2016

### Méthode pour chacune des 6 tâches de calcul mental de l’expérience)

- On fixe les règles qui vont nous permettre de classer les heatmaps. On reprend chacune des stratégies définies dans l’étude de Baranyai et *al.* [1] pour

chaque type d'opérations, et on note ce qu'on estime être les "résultats intermédiaires" pour ces stratégies, selon la tâche de calcul mental considérée. Nous définissons une catégorie par stratégie et nous classons les heatmaps dans une catégorie si une tâche au moins rouge vif est présente sur chacun des résultats intermédiaires définis pour la stratégie correspondante.

Voici les règles de détermination des résultats intermédiaires selon les stratégies que nous nous sommes fixées pour les trois type d'opérations :

#### **Additions et soustractions** —

- SPLITTING : les résultats intermédiaires sont les deux derniers nombres sommés (*exemple* :  $36 + 27 = (30+20) + (6+7) = 50+13 \rightarrow 50$  et  $13$  sont les résultats intermédiaires)
- STRINGING : les résultats intermédiaires sont les deux derniers nombres sommés (*exemple* :  $36 + 27 = 36 + (20+7) = 56+7 \rightarrow 56$  et  $7$  sont les résultats intermédiaires)
- BRIDGING THROUGH TEN : le résultat intermédiaire est la dizaine obtenue (*exemple* :  $36 + 27 = (36+24) + 3 = 60+3 \rightarrow 60$  est le résultat intermédiaire)
- COMPENSATION : le résultat intermédiaire est le nombre par lequel on compense (*exemple* :  $36 + 27 = 40 + (27-4) = 40 + 23 \rightarrow 4$  est le résultat intermédiaire)
- BALANCING : les résultats intermédiaires sont les deux derniers nombres sommés (*exemple* :  $36 + 27 = (36+4) + (27-4) = 40 + 23 \rightarrow 40$  et  $23$  sont les résultats intermédiaires)

#### **Multiplications** —

- DOUBLING AND HALVING : le résultat intermédiaire est le facteur issu du doublement (*exemple* :  $15 \times 16 = 30 \times \frac{16}{2} = 30 \times 8 \rightarrow 30$  est le résultat intermédiaire)
- SPLITTING : les résultats intermédiaires sont les deux derniers nombres sommés/soustraites (*exemple* :  $15 \times 16 = 10 \times 16 + 5 \times 16 = 160 + 80 \rightarrow 160$  et  $80$  sont les résultats intermédiaires)
- COMPENSATION : le résultat intermédiaire est le produit obtenu en modifiant le deuxième nombre avant la compensation (*exemple* :  $15 \times 16 = 15 \times 20 - 15 \times 4 = 300 - 60 \rightarrow 300$  est le résultat intermédiaire)
- FACTORIZATION : le résultat intermédiaire est le résultat de la première multiplication faite après avoir factorisé (*exemple* :  $15 \times 16 = 3 \times (5 \times 16) = 3 \times 80 \rightarrow 80$  est le résultat intermédiaire)

**N.b.** : Nous n'avons pas créé de catégories propres à "written algorithm", "no strategy", et "other strategies".

- On **classe les heatmaps** correspondant à la tâche de calcul mental étudiée dans les catégories définies ci-dessus.
- Pour chaque catégorie, on **attribue à chaque heatmap la stratégie** correspondant au sujet dans le tableau récapitulatif des stratégies établies au début de la phase de traitement des données.
- On compte le nombre de stratégies différentes dans chaque catégorie, et on **calcule la probabilité d'avoir chacune des stratégies selon la catégorie**. Ceci nous permet de savoir si notre clustering basé sur les résultats intermédiaires (sur lesquels figurent une tache au moins une tache rouge vif) est comparable à celui basé sur les stratégies utilisées.

### Traitement des fichiers "Task Object", "Post Object" et "Cross Object" grâce au code Matlab "Pipeline\_2"<sup>2</sup>

Les résultats de l'analyse statistique qui vient d'être décrite montrent que le code "Pipeline\_1" nous donne trop peu de fixations et perd donc un trop grand nombre de saccades au cours de l'expérience. En effet, dans Pipeline\_1, la position du curseur est estimé en prenant le centroïde de la tache rouge qui se déforme au cours de l'expérience. Cette déformation joue le rôle d'un filtre passe-bas et fait qu'on manque des saccades dans l'analyse. Pipeline\_2 tente de corriger cela.

### Données traitées

- Enregistrement vidéo de l'écran de l'ordinateur pendant l'expérience pour chaque sujet.
- Enregistrement vidéo du curseur de l'eye tracker pendant l'expérience pour chaque sujet.
- Tous les fichiers de données Matlab "Task Object", "Post Object" et "Cross Object" acquis lors du traitement des enregistrements vidéos pour les 38 sujets, pour toutes les tâches de calcul mental.

### Matériel/logiciels de traitement

Matlab (version R2020b ; utilisation de l'app "Signal Analyser")

### Méthode

- Le nouvel estimateur de la position du curseur est calculé à chaque instant  $t$ . L'algorithme cherche le pixel rouge le plus éloigné de l'estimateur à l'instant

---

2. Ce code figure parmi les annexes à ce document.

- $t - 1$ . Il considère alors que la direction du regard est la droite qui relie ce nouveau pixel à l'estimateur à  $t - 1$ . Il estime ensuite l'occupation de la tache du curseur (cercle) et place le nouvel estimateur au centre de ce cercle.
- Pour chacune des trois parties "task", "post" et "cross" et pour chaque sujet : Le code utilise les données des fichiers de données Matlab pour savoir quels instants des enregistrements regarder (les instants correspondant respectivement à "task", "post" et "cross") et relit les passages des enregistrements en appliquant le nouvel estimateur. Il stocke alors les nouvelles métadonnées dans de nouveaux fichiers de données Matlab. Les données stockées sont exactement les mêmes que lors du premier traitement.
  - Un filtre de Golay est appliqué à la nouvelle trajectoire du curseur pour ne pas laisser passer trop de hautes fréquences non pertinentes (trop de saccades).
  - Le code effectue à nouveau l'analyse des temps de fixation et des saccades comme décrit précédemment.
  - Le code affiche également les mêmes figures que celles décrites dans la partie "Traitement des deux vidéos obtenues pour chaque sujet grâce au code Matlab Pipeline\_1" pour "task", "post" et "cross" grâce aux nouvelles métadonnées obtenues avec "Pipeline\_2".

## Approche statistique\_ Distribution des temps de fixation et des ratios L/O/R selon les stratégies pendant la tâche de calcul mental

### Données traitées

- Résultats "Task Object" traités grâce au nouveau code "Piepline\_2" pour les 38 sujets, pour toutes les tâches de calcul mental.

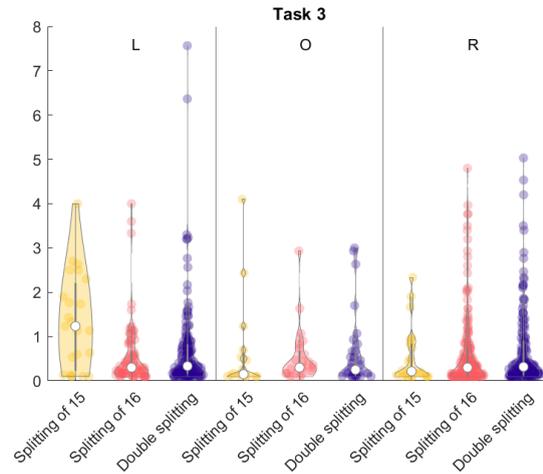
### Matériel/logiciels de traitement

Matlab (version R2020b ; utilisation de la toolbox Statistics and Machine Learning)

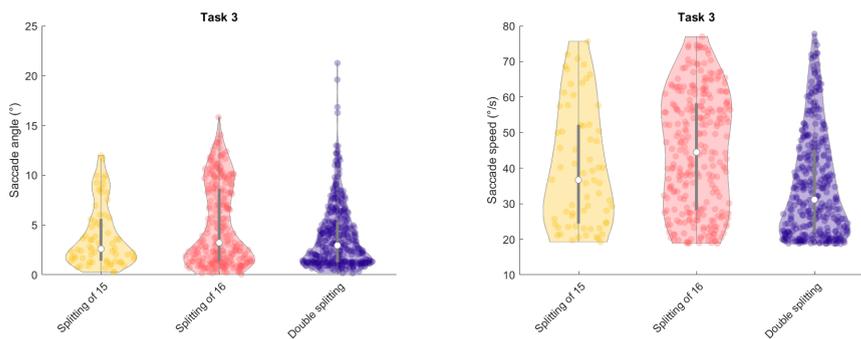
### Méthode

- Les durées des fixations pour chaque tâche et pour chaque sujet sont enregistrées dans l'objet "Task Object". Pour chaque tâche et pour chaque stratégie majoritairement utilisée au sein de cette tâche, on récupère les durées de fixation des sujets ayant utilisé cette stratégie, les fixations étant divisées selon la zone d'intérêt L, O ou R. Les distributions des durées de fixation sur chaque zone d'intérêt, par stratégie et par tâche, sont affichées sous la

forme de diagramme en violon, et les tableaux de données correspondants sont sauvegardés.



- Le même type de diagramme est effectué sur les données des saccades : les distances de saccade sont transformées en angle par relation trigonométrique en prenant en compte la distance du sujet à l'écran de 60 centimètres, ainsi que les vitesses de saccades, qui sont données en degrés par seconde. Les tableaux de données sont sauvegardés.



- Les données de chaque distribution représentée par un violon constituent une colonne des tableaux de données mentionnés ci-dessus. Dans un premier temps, l'hypothèse de distribution normale est testée sur MATLAB en appliquant à chaque colonne le test de normalité de Shapiro-Wilk. Un groupe de distributions est constitué, dans le cas des analyses des fixations, des distributions correspondant à une même tâche, une même zone d'intérêt mais des stratégies différentes. Dans le cas des saccades, un groupe de distributions est constitué des distributions en angle ou en vitesse de saccade d'une

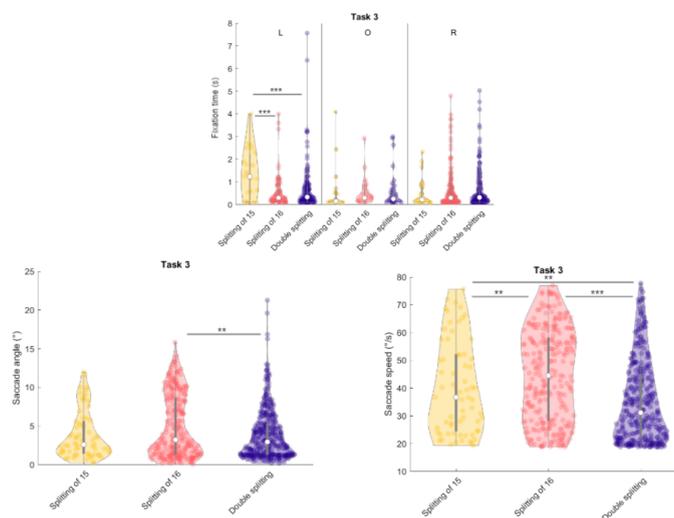
## Étude cognitive : comment le cerveau encode le calcul ?

Statistical difference in distributions with  $\alpha = 0.05$  or  $\alpha = 0.10$

	fixation times on AOI			saccade	
	L	O	R	angle	speed
Task 1	0.01987	0.52877	0.11531	0.00682	0.13856
Task 2	0.60359	0.02509	0.75981	0.81645	0.01406
Task 3	0.01398	0.55201	0.46769	0.01398	0.00000
Task 4	0.98833	0.00151	0.06572	0.10320	0.65577
Task 5	0.09639	0.18696	0.58179	0.00081	0.44855
Task 6	0.80517	0.65452	0.22914	0.15154	0.26211

même tâche mais des stratégies différentes. Pour les données traitées, il y a dans chaque groupe au moins un colonne ne vérifiant pas le test de normalité.

- Par la suite on utilise des tests non paramétriques : test de Wilcoxon pour comparer les moyennes de groupes contenant deux stratégies différentes, et test de Kruskal-Wallis si le groupe contient au moins trois stratégies. On obtient un tableau des valeurs p correspondant à chaque test.
- Dans le cas des groupes à au moins trois stratégies, si le test de Kruskal-Wallis révèle une différence significative d'au moins l'une des stratégies, on applique le test de Wilcoxon à chaque paire de distributions au sein du groupe. On obtient finalement des différences significatives entre certaines stratégies, pour les durées de fixations ou bien pour les angles ou vitesses de saccades, et ce pour cinq des six tâches.



# Bibliographie

- [1] Baranyai et al (2019). Mental calculation stratégies used by pre-service primary school teachers. DOI : [10.21125/edulearn.2019.2167](https://doi.org/10.21125/edulearn.2019.2167)
- [2] Kohlhase, A., Fürsich, M. (2016). Understanding mathematical expressions : an eye-tracking study. In : FM4M/MathUI/thEdu/DP/WIP@ CIKM, pp.42-50
- [3] Krejtz et al. (2018). Eye tracking cognitive load using pupil diameter and micro-saccades with fixed gaze. PLoS ONE 13(9) : e0203629.

# Annexes

Les annexes sont déposées avec ce document sur le site des PSE.

**Annexe 1** Diaporama diffusé pendant l'expérience

**Annexe 2** Modèle de fiche de suivi remplie

**Annexe 3** Exemple d'enregistrement vidéo du curseur de l'eye-tracker

**Annexe 4** Exemple d'enregistrement vidéo de l'écran de l'ordinateur

**Annexe 5** Code Matlab "Pipeline\_1"

**Annexe 6** Code Matlab "Trinary Word Analysis"

**Annexe 7** Fonction Matlab "regexpdir"

**Annexe 7** Code Matlab "Pipeline\_2"

Étude cognitive : comment le cerveau encode le calcul ?

---