



Année Universitaire 2025-2026

Cahier des charges

CAHIER DES CHARGES DU TER

Présenté par

Jeanne-Esther Illouz 12504150

Gatien Caillet 12116760

Tuteur

Benoît Lemaire

Professeur

Damien Pellier

TABLE DES MATIÈRES

TABLE DES MATIÈRES	1
TABLE DES FIGURES	2
I. CONTEXTE	1
Origine du projet	1
II. OBJECTIFS DU PROJET	2
Précisions sur le modèle	2
En entrée du modèle	2
Traitements du modèle	2
En sortie du modèle	2
III. DESCRIPTION DES BESOINS ET CONTRAINTES	3
Interface graphique	3
Import de fichiers	3
Clarté	3
IV. PLAN DE DÉVELOPPEMENT	4
Organisation	4

TABLE DES FIGURES

Figure 1 : Diagramme de Gantt du 26 janvier au 4 mai avec une journée de travail par semaine_____5

Figure 2 : Diagramme de Gantt du 4 mai au 15 juin avec 5 jours de travail par semaine_____5

I. CONTEXTE

Dans le cadre du TER de notre M1 MIASHS-IC, nous souhaitons trouver un sujet qui liait les sciences cognitives et l'informatique. Nous avons donc contacté B. Lemaire qui nous a présenté ses recherches. Parmi les sujets sur lesquels il travaille, celui qui a retenu notre attention porte sur l'apprentissage arithmétique.

Origine du projet

Dans ce projet, nous partirons de zéro au niveau informatique, cependant la recherche sur l'apprentissage arithmétique sur laquelle nous nous baserons a déjà été menée et nous aurons des données sur lesquelles nous pourrions nous appuyer.

L'expérience à partir de laquelle nous travaillerons tout au long du TER a pour but de mettre en lumière les processus qui se mettent en place au cours de l'apprentissage arithmétique. Étant donné que mener une expérience sur les enfants est compliqué et peut mener à différents biais, l'étude se base sur le paradigme d'arithmétique alphabétique développé par Logan et Klapp (1991). Ce paradigme consiste à présenter des calculs ressemblant, par exemple, à $A + 2 = C$. Ce type de stimulus permet de recréer une situation d'apprentissage arithmétique chez les adultes avec une tâche qui n'est pas habituelle. Ainsi, les participant.e.s de l'étude peuvent être des adultes et pas des enfants. Dans ce type de tâche, on appelle la lettre de départ (A dans notre exemple) l'augend, et le chiffre l'addend.

Le déroulement de l'expérience consiste à afficher des équations de ce type, et les participant.e.s doivent indiquer si elle est fausse ou juste en cliquant sur des touches du clavier. Dans notre exemple, l'équation est bien correcte car C est bien 2 rangs après A dans l'alphabet.

Comme dit précédemment, le but est de voir comment se fait l'apprentissage arithmétique. On sait qu'il existe deux stratégies : compter dans l'alphabet pour trouver le résultat (pour notre exemple, faire A , B , C) ou bien récupérer le résultat en mémoire. Étant donné qu'ils sont face à une tâche inhabituelle, les participant.e.s commenceront tous.les l'expérience avec la première stratégie ; le but est de voir si cette stratégie évolue et comment. Les résultats obtenus nous indiquent qu'il y a globalement 2 profils de participant.e.s : ceux qui vont continuer de compter, et ceux qui vont progressivement récupérer en mémoire au fur et à mesure qu'ils rencontrent les stimulus.

II. OBJECTIFS DU PROJET

Le but principal du projet est de créer une application Javascript qui modélise les processus de l'apprentissage arithmétique humain. L'application est destinée à des chercheurs.ses qui connaissent bien ce sujet. Ils savent utiliser excel pour manipuler les données des expériences, mais ne comprennent pas le code Javascript. C'est pourquoi une interface graphique sera nécessaire.

Précisions sur le modèle

En entrée du modèle

Le modèle aura en entrée une liste de problèmes de type $A + 2 = C$, mais aussi des paramètres comme le temps d'encodage, α : le temps de calcul minimum entre chaque lettre (de D vers E puis de E vers F, etc.), le nombre de fois où ils ont fait ce même calcul (qui sera initialisé à 0), le temps de comparaison (décider si l'équation est vraie ou fausse), et le temps pour cliquer sur la bonne réponse (en appuyant sur des touches du clavier). Il y a aussi le paramètre η qui est basé sur la moyenne des temps de réponse. Ce temps de réponse comprend : le temps d'encodage (240 ms), le temps de commande moteur (300 ms) et le temps de comparaison (200 ms). Pour calculer η , on prend la moyenne moins tous les processus sous-jacents mentionnés précédemment. Enfin le taux d'erreur est pris en compte (généralement fixé à 5%).

Traitements du modèle

Le modèle prend la décision entre les deux stratégies possibles, à savoir compter pour trouver le résultat ou récupérer le résultat en mémoire. Cette décision se fait en fonction du stimulus et des paramètres. Par exemple, elle peut changer en fonction du nombre de fois où le modèle a rencontré l'équation.

En sortie du modèle

En sortie, le modèle donne le temps qu'il met à répondre. Les chercheurs.ses pourront ainsi calculer le Root Mean Squared Error (RMSE) entre le temps de réponse humain et celui du modèle pour en tirer des conclusions sur les nouveaux paramètres.

III. DESCRIPTION DES BESOINS ET CONTRAINTES

L'application sur laquelle nous travaillerons sera codée dans le langage Javascript, car cela nous permet d'avoir une meilleure aisance, à la fois pour le code et pour l'interface graphique. Elle est destinée aux chercheurs.ses et devra leur permettre de modifier les différents paramètres mentionnés précédemment en entrée du modèle. Pour permettre cela, il faut prendre en compte différents besoins.

Interface graphique

Le premier et le plus important est que l'endroit avec lequel les chercheurs.ses devront interagir devra être clair et accessible. Cela est dû au fait que les chercheurs.ses en psychologie ou sciences cognitives sont souvent peu familiarisé.e.s avec du code Javascript, et que l'application serait difficilement utilisable pour elles.eux s'ils.elles devaient modifier directement le code.

Pour ce faire, il est nécessaire d'avoir recours à une interface graphique. Etant donné que nous coderons en Javascript, nous utiliserons les langages HTML et CSS. Pour ce qui est de la mise en page, nous utiliserons la librairie Bootstrap, version 5.3.

Nous aurons donc besoin d'implanter une interface graphique pour la modification des paramètres, mais également pour que le modèle fournisse des graphiques des résultats, sur lesquels les chercheurs.ses pourront s'appuyer pour tirer des conclusions. Pour ce faire, nous ferons appel à la librairie D3.js.

Tests

Pour ce qui est des tests unitaires, nous avons comme objectif d'avoir une couverture de 80%. Nous écrirons les tests après chaque implémentation d'une nouvelle fonction. Dans ce but, nous utiliserons Jest.

Import de fichiers

Le second besoin qui devra être rempli est la possibilité d'importer des fichiers sous forme de tableau excel. Etant donné que les utilisateurs.rices de l'application sont habitué.e.s à manipuler ce type de fichiers, ce sera à elles.eux de les mettre en forme au préalable selon les besoins.

Clarté

Le fonctionnement de notre code devra être assez clair et documenté, car dans le contexte de la création de cette application il est possible qu'elle ait par la suite besoin de modifications ou d'améliorations. Il sera donc utile que des personnes de l'équipe de recherche puissent récupérer et améliorer facilement le code.

IV. PLAN DE DÉVELOPPEMENT

Pour ce projet, nous avons décidé de développer selon le modèle incrémental : nous ajouterons les fonctionnalités une par une, en s'assurant qu'une fonctionnalité marche avant de passer à la suivante.

Notre organisation se basera sur les diagrammes de Gantt qui suivent. Le premier concerne la période du 26 janvier au 4 mai, pendant laquelle nous travaillerons une journée par semaine. Le second diagramme décrit les étapes à réaliser du 4 mai au 15 juin, période durant laquelle nous avancerons sur le projet 5 jours par semaine.

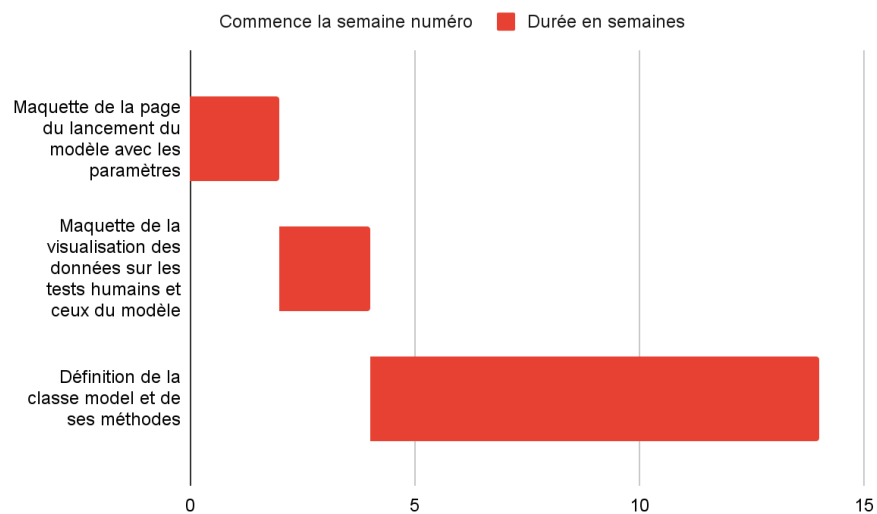


Figure 1 : Diagramme de Gantt du 26 janvier au 4 mai avec une journée de travail par semaine

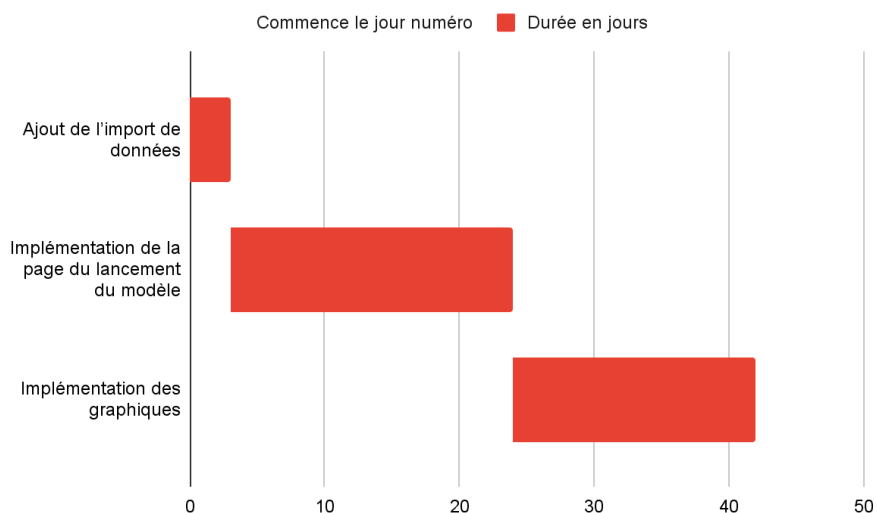


Figure 2 : Diagramme de Gantt du 4 mai au 15 juin avec 5 jours de travail par semaine

BIBLIOGRAPHIE

- Logan, G. D., & Klapp, S. T. (1991). Automatizing alphabet arithmetic: I. Is extended practice necessary to produce automaticity? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 17(2), 179–195. <https://doi.org/10.1037/0278-7393.17.2.179>
- Stéphanie Rousset Chouteau. Apprentissage de l'addition : comptage ou récupération en mémoire? Approches expérimentale et computationnelle. Psychologie. Université Grenoble Alpes [2020-..], 2024. Français. ⟨NNT : 2024GRALS027⟩. ⟨tel-05020405⟩