RAPPORT

MATHON Tristan
RASOAMANANA Nathanaël

Année 2023-2024

Remerciements

Nous tenons à exprimer notre profonde gratitude envers nos encadrants Martin Bodin et Emmanuel Beffara pour leur bienveillance et leur accompagnement tout au long du projet. Leurs conseils avisés ont été d'une aide précieuse.

Nous remercions également notre responsable de l'UE, Damien Pellier, pour ses orientations et ses précieux conseils lors de nos réunions.

Nos remerciements vont également à Maude Chorier pour sa contribution essentielle à l'obtention des bureaux et son accueil chaleureux à l'IMAG.

Enfin, nous sommes reconnaissants envers toute l'équipe du LIG pour leur assistance lors de la réunion d'information et de sécurité.

Cadre du projet

Dans le cadre de notre master, nous participons à l'UE Travaux Encadrés de Recherche (TER), qui nous permet d'acquérir une expérience en recherche. Parmi les sujets proposés, nous avons choisi "Implémentation et évaluation d'une interface pour l'enseignement de la preuve mathématique", car il combine nos intérêts pour l'informatique et les mathématiques.

Les assistants à la preuve, tels que Coq, sont des logiciels qui vérifient la cohérence logique des démonstrations mathématiques. Bien qu'ils soient généralement destinés à des utilisateurs experts, le projet dans lequel nous travaillons permet de les rendre accessibles aux étudiants en intégrant des éléments graphiques, notamment les tableaux de variation, couramment utilisés dans l'enseignement.

Nos travaux visent à combler le fossé entre les méthodes traditionnelles de la preuve et les exigences formelles des assistants à la preuve, en enrichissant l'apprentissage grâce à des représentations graphiques. Nous allons créer une interface permettant la création interactive de tableaux de variation.

L'analyse des besoins

Une maquette de présentation nous a été présentée par nos encadrants dès notre première réunion (cf annexe 1). Cette maquette nous donne une vision globale et bien illustrée de ce qu'il nous est attendu de faire et des réflexions qui ont déjà été faites.

Nos tuteurs attendaient de nous qu'on crée une interface graphique ciblant principalement les étudiants en première années d'étude supérieur dans le domaine des mathématiques mais aussi leurs professeurs. Cette interface doit correspondre à un tableau de variation, celui-ci doit être dynamique et ergonomique afin de faciliter au mieux son utilisation. Ce défi permet aux étudiants d'étudier les tableaux de variation qu'ils connaissent déjà, tout en s'appropriant a la fois le principe d'assistant à la preuve mathématique. Ainsi nous devons réaliser une interface qui est accessible à l'étudiant et aussi aux professeurs.

Néanmoins, nous avons eu quelques confusions. Etant donné qu'on est sur un projet touchant un domaine qui nous était jusqu'ici complètement inconnu : l'assistance à la preuve mathématique, nous avons semblé faire face aux limites de nos capacités. Cependant, nos encadrants nous ont assuré qu'on n'allait pas manipuler d'assistant à la preuve. Notre focalisation était principalement sur la

création de l'interface graphique. Mais cela ne nous a pas empêché de faire l'état de l'art, de creuser et d'en manipuler pour appréhender et enrichir notre projet.

A la fin de cette phase, on devait rédiger un cahier des charges, un cahier de recettes et un faire un plan de développement.

La conception

- **Modélisation de l'interface**

Nous avons utilisé des outils tels que Paint, Figma, ainsi que du papier et un stylo pour concevoir plusieurs versions de l'interface. Ces itérations successives visaient à converger vers une version définitive qui intégrerait les éléments les plus appropriés pour assurer un développement ergonomique en aval .

Au début, nous avons créé des versions sans prendre en compte les contraintes du projet (cf annexe 2). Progressivement, nous avons affiné notre modélisation en prenant en compte les exigences, les méthodes de développement, les délais, et les limitations spécifiques. (cf annexe 3)

- **Aperçus des outils et technologies de développement**

Aucun langage de programmation ne nous a été imposé pour ce projet. Cependant, étant donné que nous nous inspirons de jsCoq, nous avons convenu avec les encadrants d'utiliser JavaScript, ce qui entraîne l'utilisation de html et de css.

Nous avons évalué diverses bibliothèques (Vue.js, jQuery, React et Angular) en créant un tableau de variation avec chacune d'elles. Vue.js nous a particulièrement intéressés en raison de sa manière flexible d'interagir avec les données. Cependant, en fonction de l'évolution du projet, du temps de prise en main des bibliothèques et de leur utilité que nous leur avons consacrée, nous avons finalement décidé de ne pas utiliser de bibliothèque pour la manipulation du tableau. Néanmoins, jQuery sera partiellement utilisé, mais en dehors du contexte du tableau de variation.

- **Construction de l'API**

Bien que notre tâche principale soit de nous occuper de l'interface graphique, il était nécessaire d'établir une connexion avec l'assistant de preuve. Cependant, ce logiciel qui nous intéresse utilise le langage OCaml (Coq). Nos encadrants se sont donc chargés de développer un programme simulateur, tandis que nous devions définir les fonctions nécessaires pour le bon fonctionnement du tableau de variation.

Ensemble, nous avons réfléchi au fonctionnement de cette API, notamment les paramètres, l'environnement ,les nécessités et le traitement des informations.

Une fois le module construit en Ocaml, il a été compilé en JavaScript pour pouvoir être importé et utilisé par l'interface. Cette étape, bien que non prévue initialement, est devenue cruciale pour le projet.

- **Modèle du produit final**

Nous avons élaboré un modèle final en intégrant cette API. Ce modèle représente le produit final, bien que certaines fonctionnalités et utilisations aient évolué au cours du développement.

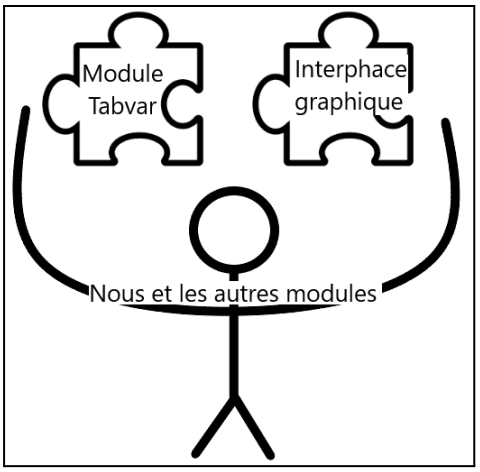


Schéma 1 : Représentation imagée des modules et de leurs liens

Le développement

- **Planification**

1. Nous avons d'abord produit le visuel du tableau de variation en utilisant uniquement du HTML.

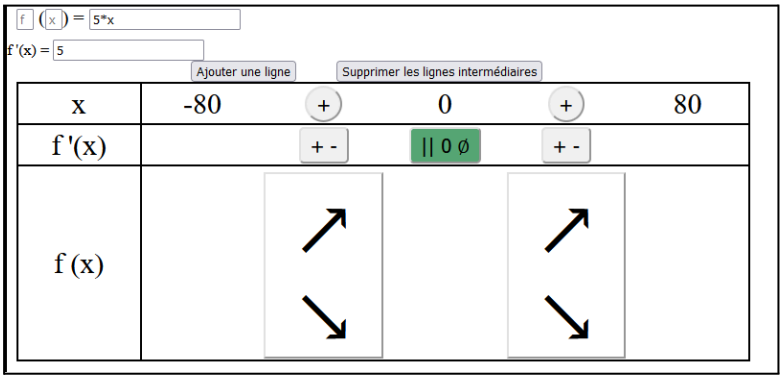


image 1 : Dernière version du tableau de variation

2. Nous avons rendu ce tableau interactif pour l'utilisateur en ajoutant un script JavaScript permettant d'entrer des données, d'ajouter des lignes et des colonnes, ainsi que de supprimer et modifier des données.

3. Nous avons ajusté l'interface pour qu'elle corresponde à notre modèle de produit final, divisant l'interface en quatre parties et découpant le script en quatre modules principaux de gestion :
    - Gestion de l'interaction avec le tableau de variation.
    - Gestion de la vérification des données du tableau et des entrées de fonctions.
    - Gestion de la zone textuelle.
    - Gestion de la zone dédiée à l'aide sur les fonctions saisies dans la zone textuelle (la partie aide).

Nous avons divisé l'interface en quatre parties afin de visuellement représenter nos quatre modules principaux de gestion interne. La partie de gauche est principalement dédiée aux entrées de l'utilisateur et la partie de droite aux retours de l'interface.

La première partie (haut gauche) est dédiée à tout ce qui concerne l'interaction directe avec le tableau de variation ainsi que la saisie de la fonction principale et de la dérivée. La deuxième partie (haut droite) est destinée à la vérification et la cohérence des données dans le tableau ou agissant avec le tableau. La troisième partie (bas gauche) est dédiée à l'accès à des fonctions de l'API en dehors du tableau de variation. La dernière partie nous sert d'affichage d'aide sur les fonctions des différentes commandes existant dans l'API qu'on peut éventuellement utiliser.

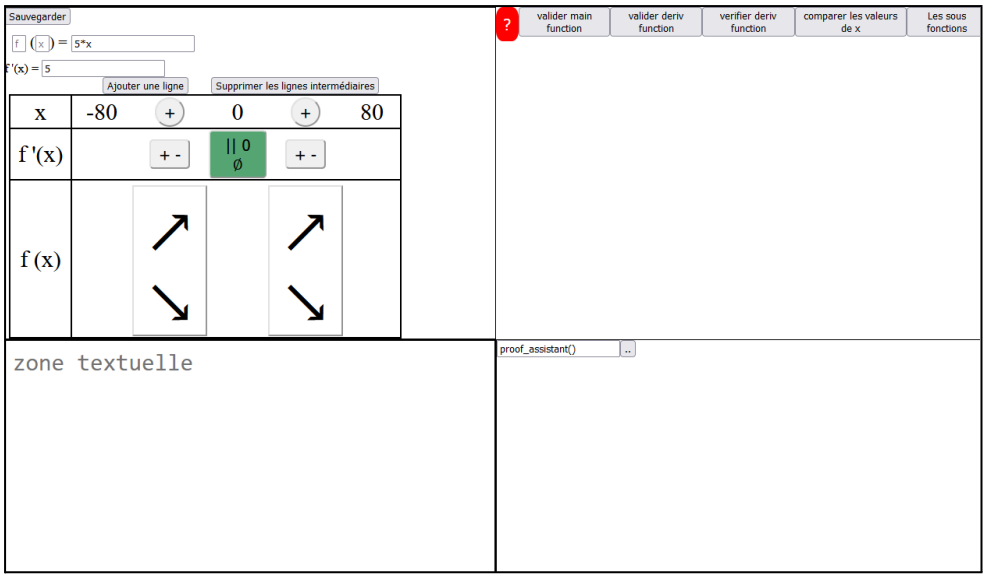


Image 2 : Dernière version de l'interface graphique avec les 4 éléments.

- **Organisation**

Le développement a exigé une collaboration sur l'ensemble des modules de l'interface, cependant, pour les modules nécessitant l'API, nous avons opté pour un travail simultané. L'un était dédié à la vérification des données du tableau et des entrées de fonctions, tandis que l'autre se focalisait sur la gestion de la zone textuelle, permettant la saisie manuelle d'une fonction du module tabvar.

- **Difficultés**

Les deuxième et troisième phases du développement, durant lesquelles nous avons rendu le tableau interactif, ont pris beaucoup de temps. Nous avons dû effectuer de nombreux ajustements et nous avons réalisé qu'il n'était pas si simple de concrétiser nos idées initiales lors de la conception. Certains problèmes qui n'avaient pas été pris en compte dès le départ ont ressurgi, tels que la manière d'entrer et de modifier les données, notamment les formules, la gestion de l'espace car le tableau est dynamique, l'emplacement de certains boutons comme l'ajout de ligne, ainsi que le recueil des données du tableau. Et d'autres détails qui ont pris du temps de développement et que nous avons longuement discuté avec nos tuteurs.
Nous avons constaté que les cellules du tableau contenaient beaucoup d'autres choses, en dehors des valeurs cibles.

- **Changement d'utilisation de certaines zones de l'interface**

Initialement, le quatrième module de gestion devait recevoir les retours des fonctions saisies dans la zone textuelle. Nous avons jugé plus pertinent de regrouper l'entrée et le retour dans une même zone, permettant à l'utilisateur d'accéder facilement à une aide lorsqu'il utilise une fonction. Cette dernière est utilisée par l'utilisateur en soumettant le nom de la fonction dans la barre de recherche dont il veut savoir des informations, et cela lui retourne la documentation interne de la fonction.

Conclusion

- **Ce qu'on a appris de ce TER**

Ce projet a été une véritable aventure, pleine de découvertes et de défis. Nous nous sommes plongés dans le monde complexe des assistants à la preuve. En développant cette interface, nous avons affiné nos compétences en langages web comme JavaScript, HTML et CSS. La création de notre propre API a été une grande première pour nous. Malgré quelques changements de bureaux, nous avons su nous adapter et avancer de manière autonome et régulière. Nous avons également appliqué les méthodes de génie logiciel et de gestion de projet pour structurer notre travail. En fin de compte, ce projet nous a permis de gagner en expérience et de

dépasser nos limites, tout en renforçant notre engagement envers l'excellence et la collaboration.
Ce travail nous a également appris qu'il ne faut pas s'attarder sur les petits détails qui consomment beaucoup de temps. Il est préférable de les laisser pour plus tard et de se concentrer sur des aspects plus importants, qui prennent généralement moins de temps et ont un impact plus visible.

- **Les objectifs atteints et non atteints**

Nous avons finis ce TER avec certains objectifs atteints que se soit du côté interface graphique et interaction et du côté vérification du tableau de variation.
Du côté de l'interface graphique nous sommes en mesure de présenter une interface interactive avec de multiple fonctionnalités tel que l'ajout de valeur, l'ajout de sous fonctions, la variation de la fonction… Ce qui permet déjà d'interagir facilement et sans contrainte avec le tableau de variation. En plus de cela nous proposons une interface textuel afin de rentrer les commandes à la main et aussi une partie "Aide" qui permet à l'utilisateur de consulter la documentation interne pour chaque fonction.
Pour ce qui est de la vérification et la validation du tableau de variation nous avons pour le moment la capacité d'évaluer les fonctions, la dérivé, et les valeurs.

Nous n'avons pas pu présenter un prototype fonctionnel aux futurs utilisateurs de cette interface. Cela est dû au fait que nous avons commencé à travailler sur l'interface graphique lorsque nous avons débuté le TER à plein temps, et à ce moment-là, les étudiants étaient en pleine période de révision pour leurs partiels.

- **Les limites**

Lors de ce travail nous avons rencontré de nombreuses limites liées à son utilisation et ainsi qu'à son développement.
Les limites liées à son utilisation sont dû au fait que nous n'avons pas fait de questionnaire qualitatif aux principaux concernés, donc nous ne savons pas exactement ce qui est attendu et donc cela créer des doutes et certaine fausse route pour ce qui de l'interface graphique. Voici quelque limites soulevées

- L'attribution de nom aux sous fonctions afin de les réutiliser plus tard.
- La comparaison des valeurs se fait que deux à deux et successivement, la possibilité de comparer les valeurs extrême entre elles afin de regarder qu'elle est exactement la ou les valeurs qui ne vont pas à l'intérieur de ces bornes (si nous partons du principe que les bornes sont justes).
- Une valeur de x peut être en fonction d'une autre variable si celle ci est définie au par avant, ce qui n'est pas le cas pour le moment.
- La possibilité de réécrire quelque chose dans les zone de texte dans le tableau de variation après la suppression de tout le texte dans cette dernière.

- La possibilité de récupérer tout ce qu'a fait l'utilisateur dans le tableau de variation et de le restituer, pour but que le professeur récupère ce que l'étudiant.e à réaliser et le corrige.
- Le nom prédéfinie des sous fonctions dans le tableau de variation, ces noms n'ont aucune valeur et ne servent a rien a part à différencier les lignes, mais les noms sont récurrents et elles dépendent du nombre de lignes du tableau.

- **Les perspectives**

Pour nos perspectives, il est important de souligner que notre projet s'inscrit dans le cadre d'un projet plus vaste et à long terme, où nous poursuivons le travail déjà entrepris en amont. À terme, notre objectif est de rendre notre API compatible avec différents types d'assistants à la preuve tels que Coq, Lean, et d'autres encore. Nous envisageons également d'améliorer l'interface que nous avons créée, qui reste pour l'instant à un stade prototypique. Pour ce faire, nous prévoyons de la tester auprès d'enseignants, d'étudiants et d'autres utilisateurs potentiels. Notre ambition est de transformer cette interface en un module que l'on pourrait intégrer dans les assistants à la preuve existants, en commençant par jsCoq.
Nous souhaitons aussi avoir une interface graphique capable de se remplir, non grâce à son interaction, mais grâce aux commandes que l'utilisateur pourra entrer dans la zone textuelle. A court terme nous voulons que la vérification du tableau de variation se fasse entièrement, c'est-à- dire que le programme puisse valider l'intégralité du tableau de variation, les signes des sous fonctions, de la dérivé mais aussi les variations de la fonction.

Ressources

Natural Number Game (NNG)
Lien : Natural Number Game
Description : Un jeu interactif et éducatif utilisant l'assistant de preuve Lean pour enseigner les concepts des nombres naturels à travers des puzzles mathématiques.

jsCoq
Lien : jsCoq
Description : Une interface web pour l'assistant de preuve Coq, permettant de développer et de vérifier des preuves mathématiques directement dans le navigateur, sans nécessiter d'installation locale.

Vue.js
Lien : Vue.js
Description : Un framework JavaScript progressif pour construire des interfaces utilisateur. Il est conçu pour être adoptable de manière incrémentale et se concentre sur la vue dans le modèle MVVM.

jQuery
Lien : jQuery
Description : Une bibliothèque JavaScript rapide, petite et riche en fonctionnalités. Elle simplifie la manipulation du document HTML, la gestion des événements, l'animation et les interactions Ajax pour un développement web plus rapide et plus simple.

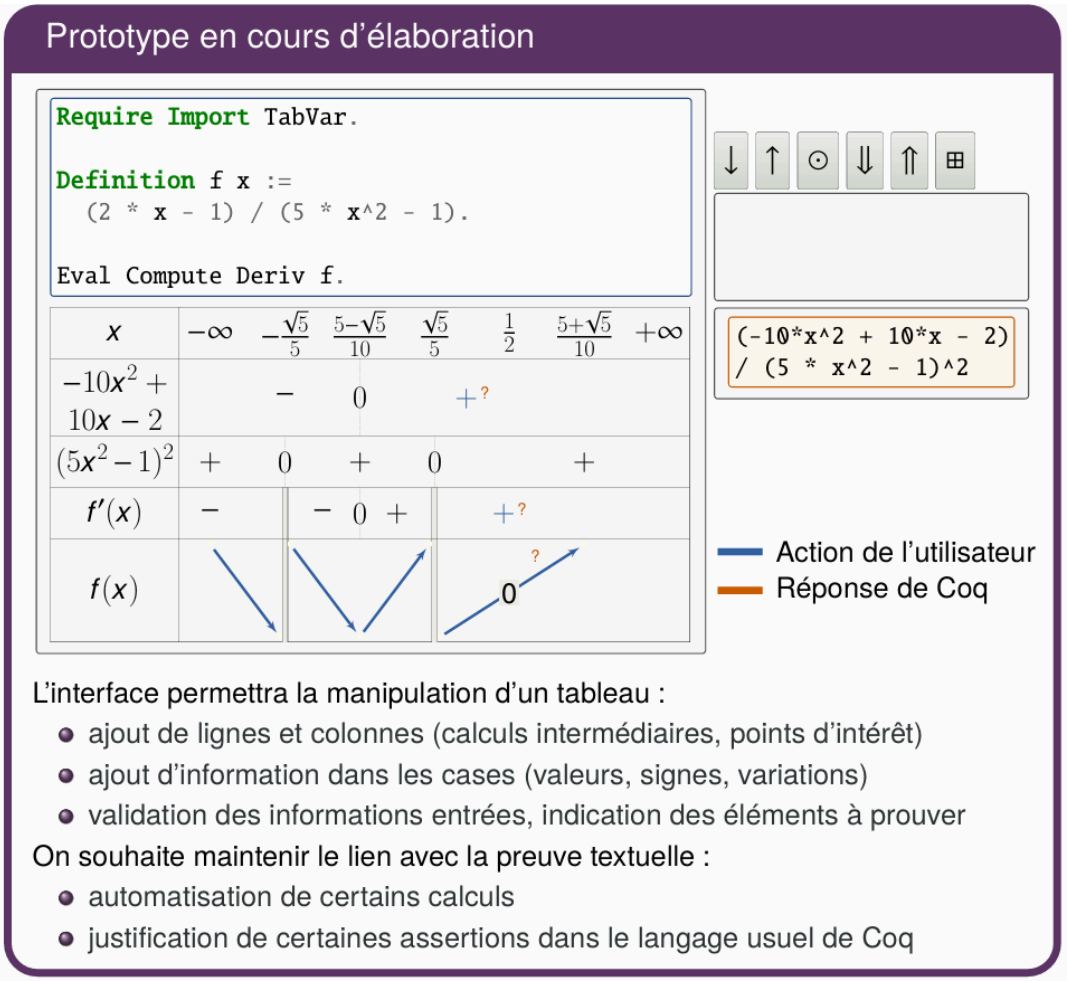
React
Lien : React
Description : Une bibliothèque JavaScript pour construire des interfaces utilisateur, développée par Facebook. Elle permet de créer des composants réutilisables qui gèrent leur propre état et produisent des interfaces dynamiques et performantes.

Angular
Lien : Angular
Description : Un framework de développement d'applications web basé sur TypeScript, maintenu par Google. Angular est conçu pour faciliter la création de SPAs (Single Page Applications) robustes et maintenables.

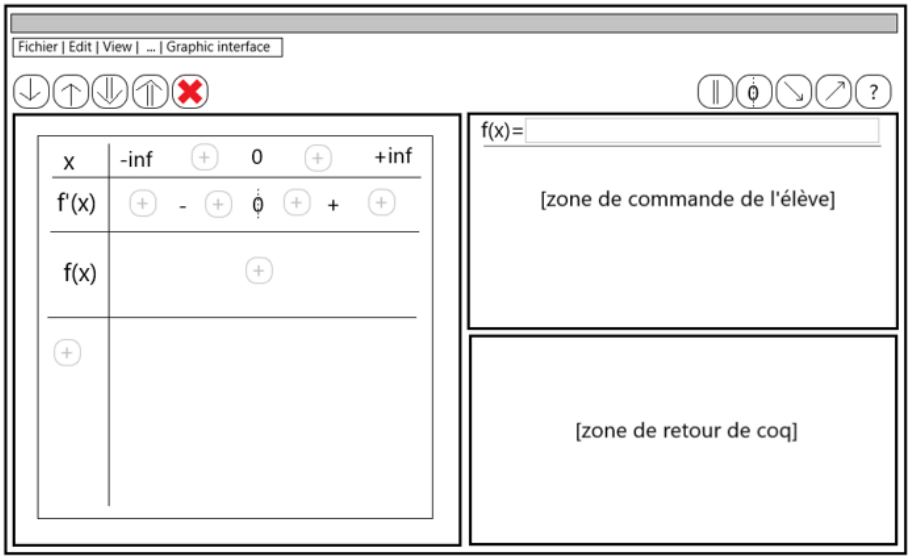
Figma
Lien : Figma
Description : Un outil de conception d'interface utilisateur collaboratif basé sur le web. Il permet aux designers et aux équipes de travailler ensemble en temps réel sur des prototypes interactifs et des maquettes visuelles.

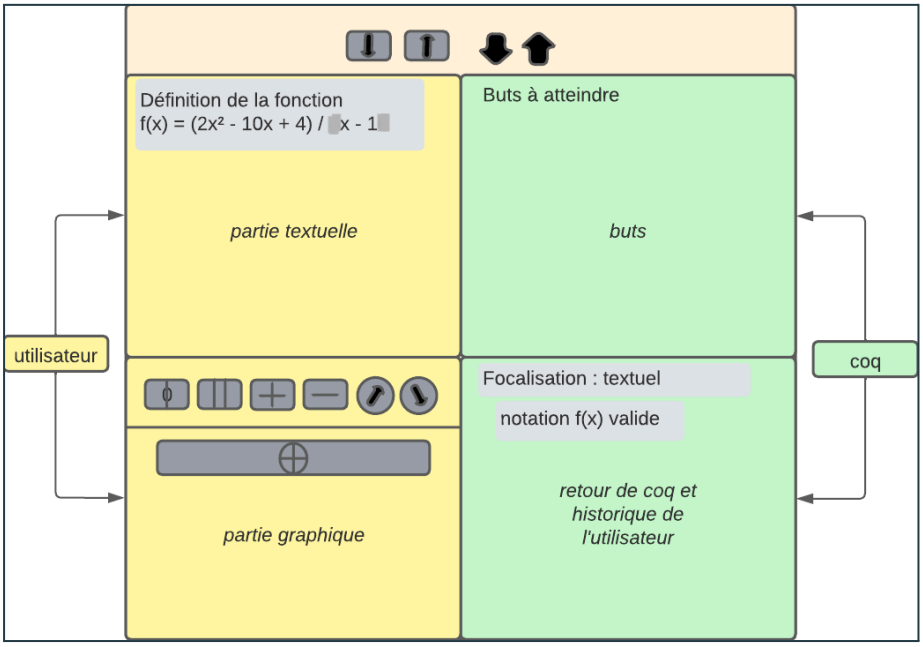
Annexes

Annexe 1 : Maquette de présentation



Annexe 2 : Les premières maquettes





Annexe 3 : Maquette avancée