

# Comparaison des langages ELM et JavaScript

## I. Introduction

Dans le cadre de nos projets en Écosystèmes et Langages de Programmation, nous avons eu l'opportunité de travailler avec les langages de programmation ELM et JavaScript. Pour notre projet de TCTurtle, ELM a été utilisé pour créer un programme permettant de visualiser des commandes de dessins entrées par l'utilisateur. Pour notre deuxième projet, JustOne, un jeu coopératif, nous avons utilisé JavaScript. Nous allons ainsi comparer ces deux langages de programmation dans le contexte de ces projets.

## II. Système de Typage

ELM est un langage de programmation à typage statique, avec un système de types qui nous a été très utile pour éviter les erreurs d'exécution. Le compilateur de ELM analyse chaque ligne de code pour détecter les erreurs de typage. Cela nous a permis de vérifier toutes les erreurs lors de la compilation. Au contraire, JavaScript est un langage à typage dynamique, les types de données sont ainsi déterminés à l'exécution, par conséquent, nos erreurs de types n'apparaissent qu'à l'exécution. Cela a rendu le développement plus rapide au début, car il n'est pas nécessaire de déclarer explicitement les types, mais cette flexibilité a souvent entraîné des erreurs parfois difficiles à détecter.

## III. Langage Fonctionnel et Language multi-paradigm

ELM est un langage de programmation fonctionnel, ce qui signifie que nos programmes ne sont constitués que de fonctions et que les données sont immuables. Nos fonctions prennent des données en paramètres et retournent de nouvelles données. De plus, ces fonctions ne modifient pas les variables qui ne leur appartiennent pas. En revanche, en JavaScript, on trouve de multiples paradigmes car le langage supporte plusieurs styles de programmation : pour notre projet, nous avons surtout utilisé la programmation impérative, manipulant directement nos données avec des boucles qui n'utilisent pas nécessairement la récursivité.

## IV. Conclusion

ELM et JavaScript offrent tous les deux des avantages et des difficultés. Le système de typage de ELM, ainsi que son aspect fonctionnel nous ont permis de créer un programme fiable, mais nécessitent un temps d'apprentissage plus important et qui a ralenti le développement du projet. A l'inverse, la flexibilité de JavaScript nous a permis d'avoir un temps de développement beaucoup plus rapide, mais plus de temps à été consacré au débogage.