Rapport du projet

Par Gabriel Arsenault

Table des matières

[Préparation de l’expérimentation 2](#_Toc1738458680)

[1. Sujet 2](#_Toc400803129)

[2. Objectif 2](#_Toc663225159)

[3. Critères d’évaluation 2](#_Toc367920655)

[Réalisation de l’expérimentation 2](#_Toc279189059)

[4. Expérimentation 2](#_Toc1793400756)

[1.1. Exigences techniques 2](#_Toc1252290240)

[1.2. Déroulement 2](#_Toc281091876)

[1.3. Résultats 2](#_Toc944482940)

[Conclusion 2](#_Toc1753901920)

# Préparation de l’expérimentation

## Sujet

J’ai fait mes recherches sur des projets qui requièrent de la connaissance avancé sur la programmation. Un de ces projets est la conception de ton propre langage de programmation. Créer ton langage de programmation est utile pour répondre à tes besoins, surtout quand tu connais les forces et les faiblesses de ton langage mieux que tout le monde. C’est pourquoi je voudrais essayer de créer mon propre langage de programmation.

## Objectif

Mon projet consiste à créer mon propre langage de programmation qui fonctionnera comme les scripts de Python ou les exécutables de C++, dépendamment des efforts nécessaire pour créer un interpréteur versus créer un compiler.

Pour commencer, je vais créer une librairie standard qui contiendra les fonctions de base pour mon langage, puis je vais coder une application qui présentera ces différentes fonctions.

Le contenu de cette ou ces applications dépend entièrement sur ce que je réussi à implémenter dans ma librairie standard. Une librairie standard de n’importe quel langage de programmation contient de nombreuses composantes sur différents sujets. Voici quelques composantes que je planifie intégrer, en ordre de priorité:

1. Fonctions mathématiques de base (abs, sin, cos, tan, racine, min, max, exposent, tronquer/arrondir, etc.)
2. Fonctions sur les manipulations de strings (longueur du string, remplacer A par B, substring, pad left/right, trim, etc.)
3. Fonctions sur les arrays (array, liste, dictionnaire, etc.)
4. Fonctions sur les inputs (lire la ligne de texte, lire une touche du clavier)
5. Fonctions sur l’interaction du terminal (print, changer la couleur du texte et de l’arrière-plan, cls/clear, etc.)
6. Fonctions audio
7. Fonctions de base de données embarquée dans l’application (SQLite, de la même façon que la librairie SQLite de Python)

Ce n’est pas garanti que je vais toutes les intégrer, mais une bonne portion devrait suffire.

## Critères d’évaluation

1. Je dois apprendre à coder un parser qui interprète les syntaxes du langage.
2. Je dois apprendre à créer un compiler pour convertir mon code en binaires.
3. Je dois apprendre à créer un éditeur avec divers outils pour faciliter la programmation de mon langage.
4. Je dois trouver le langage dont je dois dériver pour créer mon propre langage.
5. Je dois apprendre comment intégrer chaque composante de ma librairie standard.
6. Quand le langage est terminé, je dois présenter ses aptitudes avec des programmes codés par celui-ci.

# Réalisation de l’expérimentation

## Expérimentation

## Exigences techniques

1. ANTLR (ANother Tool for Language Recognition)
   1. Générateur de parser qui permet de traduire le code en binaires.
2. Gradle
   1. Permet de construire une application, incluant mon langage de programmation.

## Déroulement

## Résultats

# Conclusion