

Licence Informatique

La Rochelle Université

Extensions du framework GALACTIC



Jules Bricou

2022

Utilisation du portrait d'Évariste Galois réalisé par M. Yann Gautreau
aimablement autorisée dans un cadre universitaire

Table des matières

1	L'entreprise	4
2	La plateforme GALACTIC	5
3	Le Stage	7
3.1	Le Sujet Du Stage	7
3.2	La commande Galactic	7
4	Installation de tous les outils nécessaire pour la création de la commande.	7
4.1	Installation de MSYS2	8
4.2	Installation de Virtual Environment & Galactic	8
5	Comparaison des différents moteurs de squelette disponibles.	9
5.1	Cheetah3	9
5.2	Jinja2	10
5.3	Template	10
5.4	Mako	10
5.5	Comparaison finale:	11
6	Comparaison des différent langages pour la création de la commande \$galactic.	12
6.1	CLICK	12
6.2	Typer	12
6.3	Cleo	12
6.4	Comparaison finale:	13
7	Installation de Poetry, Jinja, et Cleo.	13
8	Conclusion	13
9	Annexes	14
9.1	Sources	14
9.1.1	Présentation de l'entreprise	14
9.1.2	Présentation de la Plateforme GALACTIC	14
9.1.3	Les moteurs de squelettes	14
9.1.4	Les interfaces de lignes de commande	14
9.1.5	Poetry	15

Liste des figures

Remerciements

Tout d'abord, je souhaite remercier Monsieur Yacine GHAMRI-DOUDANE, directeur du laboratoire L3i pour m'avoir accepté au sein de son laboratoire pour mon stage.

Je souhaite remercier mon maître de stage Monsieur Christophe DEMKO, enseignant-chercheur à l'université de La Rochelle, pour toute l'aide qu'il m'a prodigué pendant ma recherche de stage, ainsi que durant le stage. Cela m'a permis de mieux comprendre la plateforme sur laquelle j'allais devoir travailler ainsi que les incompréhensions que j'ai pu avoir concernant le sujet de mon stage.

Je remercie aussi ma tutrice, Madame Karell BERTET, enseignant-chercheur à l'université de La Rochelle, pour ses explications concernant le fonctionnement d'un projet de recherche, mais aussi pour ses explications concernant la plateforme et l'algorithme NextPriorityConcept utilisé pour le projet.

Je souhaite de par la même occasion remercier les enseignants-chercheurs et les doctorants travaillant dans l'équipe du projet Galactic pour leurs explications concernant la plateforme pendant les réunions.

Enfin, je remercie mes camarades Alexandre BUFFARD, Kevin VOISIN, Martin ELINGER et Eléonore THONNEAU qui ont effectués leur stage en même temps que moi pour leur aide et leur soutien tout le long de celui-ci, ainsi que pour le travail commun effectué sur la présentation d'entreprise ainsi que la présentation de la plateforme.

Résumé en français

Dans le cadre de stage de fin de troisième année de Licence Informatique, à l'Université de La Rochelle, Pôle Scientifique, j'ai eu l'opportunité de pouvoir réaliser un stage en milieu de recherches durant 6 semaines. Intéressé par le sujet, j'ai décidé de réaliser mon stage dans le laboratoire de recherches Laboratoire Informatique, Image et Interaction (L3i) de l'Université de La Rochelle, de part mon intérêt dans le domaine informatique et ma curiosité quand aux activités effectués dans un domaine de recherche. Le L3i m'a accueilli du 2 mai au 27 juin 2022, et j'ai intégré un groupe de stagiaires travaillant sur la création de différentes extensions pour la plateforme Galactic.

Ce rapport de stage contient les recherches exécutées pour pouvoir déterminer les outils nécessaires à la création d'une ligne de commande permettant de créer un nouveau squelette d'extension de Galactic, ainsi que les installations et processus ayant été nécessaire durant le commencement de ce stage. Ce rapport est incomplet, car écrit durant la période de stage, avant la fin de celui-ci. Une partie du travail n'est donc pas encore réalisée.

Ces recherches incluent notamment une comparaison entre les moteurs de squelettes Cheetah3, Jinja2, Template, et Mako, ainsi que la comparaison entre les trois interfaces de lignes de commande (CLI): CLICK, Typer et Cléo, et la conclusion des deux comparaisons.

Abstract in English

Within the framework of an internship at the end of my third year of Licence Informatique, at the Université de La Rochelle, Pôle Scientifique, I had the opportunity to be able to apply for an internship in

researches culture for a period of 6 weeks. Interested by the subject, I ended up settling on an internship in the research laboratory Laboratoire Informatique, Image et Interaction (L3i) of the Université de La Rochelle, from the interest I took in the IT sector as well as my curiosity at the activities performed in researches culture. The L3i welcomed me from the 2nd of May to the 27th of June, 2022, and I was included in a group of interns working on the creation of different extensions for the Galactic platform.

This internship report contains the researches done so as to define each tool necessary for the creation of the command line that will allow the user to create a new extension of Galactic using a template, as well as the installation process necessary at the internship's start. This report is incomplete, as it was done while still in the internship period, before this one ends. Part of the work thus has yet to be made.

These researches includes in particular a comparison between the different template engines Cheetah3, Jinja2, Template, and Mako, as well as the comparison between three Command Line Interfaces (*CLI*): Click, Typer and Cleo, and the conclusion of these two comparisons.

1 L'entreprise



La présentation de l'entreprise est issue du site de l'entreprise (<https://l3i.univ-larochelle.fr/Présentation-317>) et il s'agit d'un travail fait en commun avec Éléonore THONNEAU, Kévin VOISIN, Alexandre BUFFARD, Martin EHLINGER et moi-même.

Le Laboratoire Informatique, Image et Interaction (L3i) a été créé en 1993, près de 100 personnes travaillent au sein du laboratoire. Il s'agit d'un laboratoire de recherche de Sciences du Numérique de l'Université de La Rochelle, il associe les chercheurs en Informatique de l'IUT ainsi que les chercheurs du Pôle Sciences de l'Université de La Rochelle. Il compte 33 chercheurs et chercheuses, 35 doctorants et doctorantes, 4 personnes permanentes de soutien à la recherche et 25 personnes sur projets et est dirigé par Monsieur Yacine GHAMRI-DOUDANE.

Le laboratoire L3i fait parti de réseaux de recherches régionaux (Fédération CNRS MIRES, ERT " Interactivité Numérique "), nationaux (GDR I3 et GDR IRIS) et internationaux (IAPR) dans les secteurs de visibilité de son action scientifique, autour d'un projet stratégique lié à la gestion intelligente et interactive des contenus numériques. Cela est renforcé grâce à une politique de volontariste de participation ou de pilotages de projets de recherches labélisés (ANR, PCRD, ...), au sein desquels le laboratoire occupe couramment une position de leadership.

Le Laboratoire est également membre de l'Alliance Big data lancée en 2013 pour favoriser le développement en France de nouveaux services et projets dans ce domaine.

Ses travaux sont menés en partenariat avec une dizaine de centres de recherches nationaux (dont l'INRIA, institut spécialisé). Le L3i entretient par ailleurs des liens privilégiés avec de nombreux centres de recherche à travers le monde. Il est également engagé dans près d'une vingtaine de partenariats industriels sur l'ensemble du territoire français.

Le laboratoire du L3i est divisé en 3 équipes:

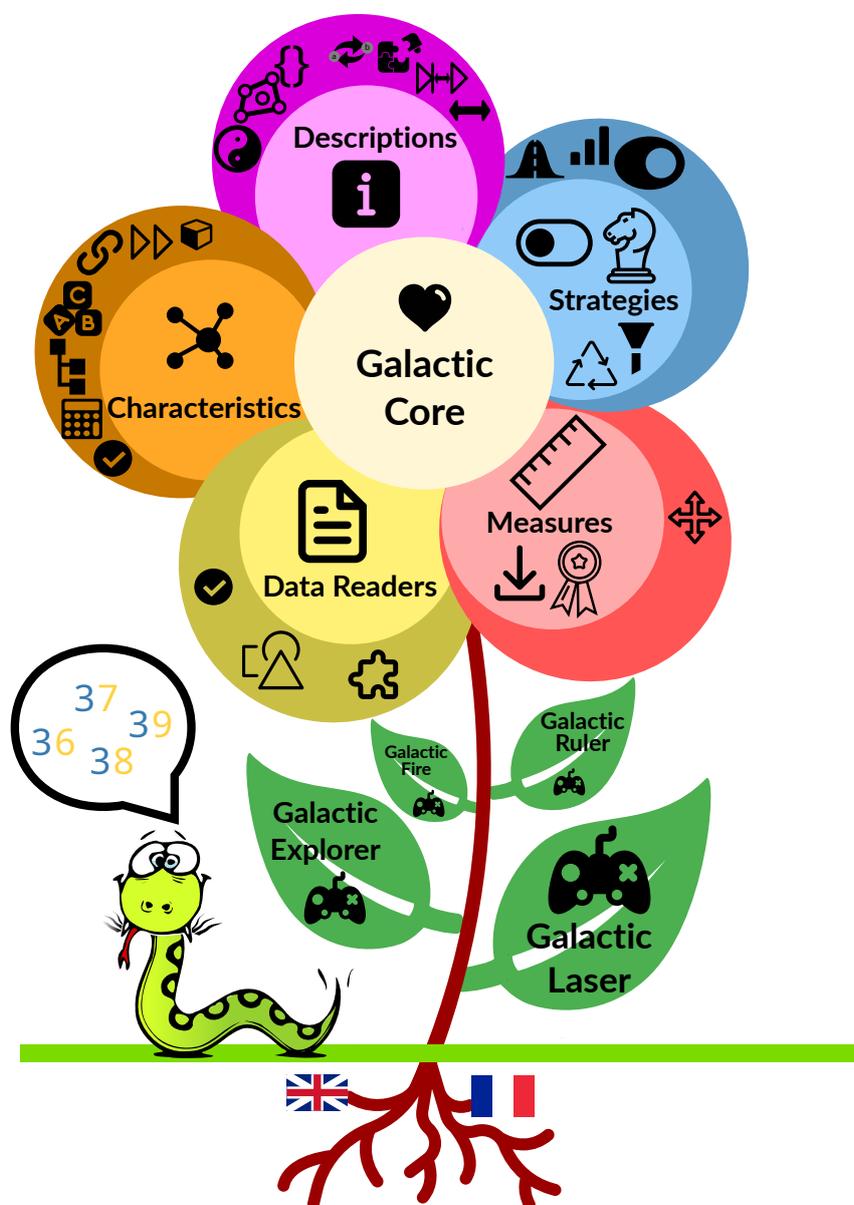
- Modèles et Connaissances
- Images et Contenus
- Dynamique des systèmes et Adaptativité

Mon stage se déroule au sein de l'équipe Modèles et Connaissances et plus précisément autour du projet GALACTIC de cette équipe.

2 La plateforme GALACTIC

La présentation de la plateforme GALACTIC est issue du diaporama de présentation de l'architecture de la plateforme (<https://galactic.univ-lr.fr/slides/architecture/Galactic-Architecture-slides-20min-complete.pdf>) et il s'agit d'un travail fait en commun avec Éléonore THONNEAU, Kévin VOISIN, Alexandre BUFFARD, Martin EHLINGER et moi-même.

GALACTIC est l'acronyme de **GA**lois **LA**ttices **C**oncept **T**heory **I**mplicational system and **C**losures. L'objectif de la plateforme est pouvoir mettre en place un framework permettant de travailler sur l'Analyse Formelle des Concepts ainsi que la Théorie des treillis. La plateforme GALACTIC est structurée de la façon suivante:



- Un noyau, représenté par le coeur de la fleur, qui contient les opérations de base et structures de données et qui implémente un nouvel algorithme (NextPriorityConcept) inspiré des pattern

structures.

- Des plugins, représentés par les pétales de la fleur:
 - Caractéristiques : Booléen, Numérique, Catégorique, String, Séquentielle, Chain, Triadic. Ils permettent de définir les caractéristiques.
 - Descriptions : Booléen, Logique, Catégorique, String (regex), String (distance), Chain, Séquentielle, Séquentielle (distance), Triadic. Ils définissent les prédicats et les espaces de description utilisés pour représenter et définir les données avec précision.
 - Stratégies : Booléen, Logique, Catégorique, Numérique, String, String (distance), Chain, Séquentielle, Séquentielle (distance), Triadic. Ils définissent la manière utilisée pour explorer les données, il utilise des descriptions pour générer des prédécesseurs pour chaque concept dans le treillis.
 - Mesures : predecessor Cardinality, successor Cardinality, Confidence. Ce sont des paramètres des stratégies de filtrage prédéfinis dans la librairie core.
 - Data Readers : Ils sont utilisés pour lire différents types de fichiers de données. Le moteur de base détecte le type de fichier en utilisant son extension. Les plugins Data Readers existants sont: YAML, JSON, CSV, TOML, INI, TXT, SLF, DAT, CXT.
 - Localization: Ils sont utilisés pour traduire les applications dans d'autres langues.

- Des applications, représentées par les feuilles de la fleur, qui sont développées afin d'utiliser la librairie de la plateforme. Ce sont des interfaces pour l'utilisateur:
 - GALACTIC Laser : pour construire les treillis et explorer les données.
 - GALACTIC Explorer : pour explorer de façon interactive les treillis construits.
 - GALACTIC Ruler : pour extraire les règles d'implication.
 - GALACTIC Fire : pour exécuter un système de règles.

- La bulle au dessus du serpent indique les différentes versions de python avec lesquelles la plateforme GALACTIC est compatible.

- Au niveau des racines on peut voir les différentes langues disponibles sur la plateforme.

3 Le Stage

3.1 Le Sujet Du Stage

Mon stage s'est déroulé au sein du Laboratoire Informatique, Images et Interaction (L3i) de l'Université de La Rochelle, au sein de l'équipe Modèles et Connaissances. J'ai travaillé sur le projet de la plateforme GALACTIC présentée auparavant. Plusieurs plugins ont déjà été créés pour la plateforme GALACTIC afin de pouvoir travailler sur différents types de données telles que les données numériques, les données logiques, les données séquentielles, les données catégoriques ou encore les données de type `String`.

Le travail qui m'a été demandé est de créer avec Python une interface en ligne de commande permettant de créer des squelettes pour les extensions du framework GALACTIC.

Il m'a fallu donc tout d'abord déterminer quel moteur de squelette utiliser pour la création des squelettes, ainsi que choisir quel type de Command Line Interface permettrait de créer une interface par terminal, avant de commencer à coder la commande sous Python.

3.2 La commande Galactic

Le but de cette commande est de pouvoir créer, à partir d'un modèle de base à remplir, le squelette d'une extension pour le framework galactic. Pour cela, lorsque la commande

```
>_ | $ galactic
```

est entrée par le biais d'un terminal, elle doit pouvoir soit remplir le modèle automatiquement grâce aux options ajoutées dans la commande entrée, soit pouvoir demander après exécution de la commande à pouvoir remplir chaque partie du modèle par le biais de questions répondue à travers le terminal.

4 Installation de tous les outils nécessaire pour la création de la commande.

Avant même de pouvoir commencer à travailler sur le sujet de mon stage, il m'a fallu installer des outils pour pouvoir travailler. Le travail suivant a été réalisé sur l'OS Windows 10. La plupart des personnes utilisant normalement Linux, je partais déjà avec un certain désavantage: Linux est plus souple que Windows quand il s'agit de coder.

4.1 Installation de MSYS2

Msys2 est un outil de développement de software en open-source permettant de créer des Exécutables Portables (PE) permettant de créer du code de type Unix sous Windows.

- On va sur msys2.org et on télécharge l'installeur x86_64.
- On suit les instructions sur la page pour pouvoir télécharger correctement l'environnement Msys2.
- On active depuis le menu de démarrage MSYS2 MinGW x64, ouvrant une fenêtre de terminal.

Une fois MSYS2 téléchargé et ouvert, on va mettre à jour et installer quelques addons.

- `$ pacman -Suy`
permet de mettre à jour l'environnement de MSYS2.
- `$ pacman -S msys/gcc mingw-w64-x86_64-toolchain base-devel`
permet de mettre en place un environnement de base.
- `$ pacman -S mingw-w64-x86_64-python3 mingw-w64-x86_64-python3-pip`
permet d'installer Python et Pip.
- `$ pacman -S mingw-w64-x86_64-graphviz`
installe GraphViz (<http://www.graphviz.org/>).



Warning

Il est crucial d'utiliser MSYS2 MinGW X64 pour les étapes suivantes. Je me suis trompé durant l'installation en utilisant x86 à la place, et j'ai dû recommencer toute les installations en basculant sur x64.

Une fois MSYS2 mis à jour les extensions installées, on passe à l'installation de Virtual Environment, et de Galactic.

4.2 Installation de Virtual Environment & Galactic

Virtual Environment, ou Venv, est un outil sur Windows qui permet de créer un environnement virtuel séparant le projet de l'ordinateur pour ne pas polluer l'environnement de la machine avec des installations générales.

- `$ pacman -S mingw-w64-x86_64-python-virtualenv`
installe Venv sur MSYS2.
- `$ virtualenv --system-site-packages --python=python3 projet`
créé l'environnement virtuel "projet".

- `$ source projet/bin/activate`
permet d'entrer dans l'environnement "projet". Pour en sortir, il suffit de taper `$ deactivate`.

On peut ensuite passer à l'installation de Galactic dans l'environnement virtuel.

```
>_ $ python3 -m pip install  
--pre \  
--find-links https://galactic.univ-lr.fr/packages \  
py-galactic[all]
```

- Permet d'installer le package Galactic en version développement.
- `--pre` installe la version instable du projet (*c'est à dire la version toujours en développement*).



Warning

Je n'ai pas pu implémenter certains détails dans l'installation de galactic.

`--upgrade-strategy eager` devait permettre de bien mettre tout à jour, incluant les dépendances de `py.galactic`.

`--no-cache` devait reprendre tout depuis le net, pas le package pré-installé.

Ces détails aurait dû permettre de pouvoir d'installer parfaitement le package galactic.

Une fois Venv et le package Galactic installé, on peut commencer à coder la commande... Une fois que l'on ai recherché et décidé d'un moteur de squelette pour la création de squelettes Galactic, ainsi qu'une interface par ligne de commandes sur terminal.

5 Comparaison des différents moteurs de squelette disponibles.

Nous comparerons ici Cheetah3, Jinja2, Template, et Mako selon différent critères, de telle sorte à ce qu'on puisse extraire de tous le meilleur moteur de squelette à utiliser.

5.1 Cheetah3

Cheetah3 génère des documents de texte de n'importe quelle extension. Il est utilisé pour produire du code en python, java, ou PHP, entre autres. Cheetah sépare le contenu, le code et le design, est hautement modulaire, flexible, à l'architecture réutilisable. Il mélange la flexibilité de Python avec une facilité d'utilisation pour des non-programmeurs. L'interface est orienté objet, accessible depuis le code de Python, et les templates de Cheetah3 permettent de réutiliser le code aisément. Un template peut sous-classer un autre, et réécrire des sections de son code. Les sous-classes sont des classes Pythons, donc peuvent sous-classer des classes Pythons pure, et vice et versa.

- Version de Python requise : 3.6+
- Compatibilité : Windows / Linux / MacOS

- Dernière mise à jour (09/05/22): 20/04/21
- Nombre de commits sur GitHub : 2397
- Nombre de téléchargements : 11 millions

**Note**

Certaines parties de la documentation n'ont pas été mis à jour depuis 2003.

5.2 Jinja2

Jinja2 est un moteur de template rapide et expressif. Il inclue des héritages et inclusions dans les templates, permet de définir et importer des macros dans les templates, a un mode sandbox inclus dans le moteur pour tester les templates. Les templates sont compilés et optimisés en Python just-in-time et stocké, ou peuvent être compilés à l'avance, et, durant le débogage, des exceptions pointent les lignes à corriger.

- Version de Python requise : 3.7+
- Compatibilité : Windows / Linux / MacOS.
- Dernière mise à jour (09/05/22): 28/04/22
- Nombres de commits sur GitHub : 2628
- Nombre de téléchargements : 1 milliard

5.3 Template

Template est un outil de génération de fichiers grâce à des templates de Jinja2 et des variables d'environnement. Il ajoute des filtres pour Jinja ainsi que des fonctions permettant par exemple de convertir vers différentes extensions.

- Version de Python requise : 3.7+
- Compatibilité : Linux
- Dernière Mise à Jour (09/05/22): 15/08/21
- Nombres de commits : 147

**Note**

Template est un add-on de Jinja2.

5.4 Mako

Mako est une bibliothèque de templates en Python. La syntaxe et l'API de Mako emprunte des idées des autres, tel que Django, Cheetah, Myghty et Genshi. C'est par concept un langage Python de type « Embedded », cherchant à être simple et flexible, tout en maintenant certaines sémantiques de

Python. L'API est extrêmement simple. Mako en général est très rapide, autant que Cheetah ou Jinja. Ce langage Python possède les fonctionnalités classiques de Python tel que la capacité d'utiliser des blocs de Python pur, ou juste des includes, ainsi que de l'héritage dynamique et navigable dans les deux sens.

- Version de Python requise : 2.7, 3.5+
- Compatibilité : Windows, Linux, MacOS.
- Dernière Mise à Jour (09/05/22): 10/03/22
- Nombre de commits sur GitHub : 907
- Nombre de téléchargements : 399 millions

5.5 Comparaison finale:

	Cheetah3	Jinja2	Template	Mako
Version de Python requise	3.6+	3.7+	3.7+	3.5+
Compatibilité	Windows, Linux, MacOS	Windows, Linux, MacOS	Linux	Windows, Linux, MacOS
Dernière mise à jour (09/05/22)	20/04/21	28/04/22	15/08/21	13/03/22
Nombre de commits sur Github	2397	2628	147	907
Nombre de téléchargements	11 millions	1 milliard	N/A	399 millions



Note

- Certaines parties de la documentation de Cheetah n'ont pas été mises à jour depuis 2003.
- Template est un add-on de Jinja.

De part le fait que certaines parties de la documentation de Cheetah n'ont pas été mises à jour depuis un certain temps, et que Template est « juste » un add-on de Jinja, j'ai décidé de choisir entre Jinja et Mako, et je me suis posé sur une utilisation de Jinja2, de par sa popularité.

6 Comparaison des différent langages pour la création de la commande \$galactic.

Nous comparerons CLICK, Cleo, et Typer, trois Command Line Interfaces (abrégé CLI).

6.1 CLICK

CLICK (Command Line Interface Creation Kit) est un package de Python permettant de créer des commandes élégantes avec le moins de code possible. Son but est de rendre le processus d'écrire des outils pour créer et modifier des lignes de commandes rapide et facile, en essayant aussi d'empêcher des frustrations causé par l'incapacité d'implémenter des API de Command Line Interface.

- Nombre de téléchargements : 1 milliard
- Nombre de commits sur GitHub : 2184



Note

Il existe plusieurs problèmes apparent sur Windows et Venv. Cela me semble être un gros malus de quand à l'utilité de ce package.

6.2 Typer

Typer est une library pour construire de interfaces par lignes de commandes basées sur les type hints de Python, des outils permettant de préciser un type de variable. L'écriture sur Typer est intuitive, avec de l'autocomplétion, un design fait pour une aise d'utilisation, et ayant pour but d'obtenir du code succin.

- Nombre de téléchargements : 51 millions
- Nombres de commits sur GitHub : 291

6.3 Cleo

Cleo est un outil de création de CLI, qui est un wrapper de haut niveau de CliKit, partageant donc des composants et utilitaires de ce CLI. Il est aussi inspiré par le Symfony Console Component.

- Nombre de téléchargements 68 millions
- Nombre de commits sur GitHub : 295



Note

Cleo est intégré dès le départ dans Poetry, qui est un outil que l'on utilisera pour permettre d'installer plus aisément les plugins externes.

6.4 Comparaison finale:

	CLICK	Typer	Cleo
Downloads	1G	51M	68M
Commit	2184	291	295



Note

- Cleo est déjà inclus dans Poetry.
- CLICK a certains problèmes avec Virtual Environment Venv et Windows.

De part le fait que CLICK contient quelques problèmes et bogues sur Windows et Venv, et de part le fait que Cleo est déjà intégré dans le package Poetry que l'on utilise déjà dans le projet, on utilisera Cleo pour coder l'interface de ligne de commande.

7 Installation de Poetry, Jinja, et Cleo.

Après s'être décidé sur l'utilisation de Jinja2 et de Cleo, il faut maintenant installer Poetry, puis Jinja et Cleo avec Poetry, car ils seront inclus par défaut dans la création de chaque squelette d'extension.

Poetry est un outil d'installation et de dépendance d'extensions de Python. Il permet de déclarer les bibliothèques dont le projet dépend et Poetry permet d'automatiser l'installation et la mise à jour des extensions.

Dû au fait que je dois rendre ce rapport bientôt, je m'étais attelé à la tâche mais je l'ai interrompu pour finir ce rapport de stage. Je dois malheureusement interrompre ce rapport ici.

8 Conclusion

Pour conclure, j'ai effectué ce stage de fin de Licence Informatique en tant que stagiaire en programmation Python pour le projet GALACTIC au sein du Laboratoire Informatique, Image et Interaction dans l'Université de La Rochelle. Durant ce stage de 6 semaines, j'ai été appelé à savoir utiliser mes capacités de recherches pour pouvoir déduire des outils les plus pertinents pour créer la commande, et, après ce rapport, je serais appelé à utiliser mes capacités de programmeur en Python pour pouvoir créer une commande, ajouter des options à cette commande, créer le squelette de base pour les extensions, et enfin créer une interface par terminal pour la création de nouvelles extensions.

Ce stage jusqu'à présent a été très enrichissant de part le fait que qu'il m'a permis de découvrir le domaine de la recherche informatique, ainsi que le travail qui l'accompagne. Il m'a permis de participer à ses enjeux au travers de réunions et de recherche sur les différents outils nécessaires à la création de programmes. Ce stage m'a permis de comprendre que la création d'une commande ne nécessite pas

que de coder, mais demande aussi de la recherche et - dans le cas de ce projet - de comprendre et apprendre certaines notions mathématiques pour pouvoir réaliser un produit informatique.

Cette expérience de stage dans un département de recherches universitaire fut très constructive, et m'a permis d'obtenir une nouvelle vision de la création de codes, ainsi que du travail et de la recherche lié à la recherche informatique. Le projet GALACTIC étant durant une période d'épanouissement et ayant la possibilité de pouvoir être utilisé par tout le monde, je suis fier de pouvoir y contribuer.

For de cette expérience, dans la suite de mon stage, j'espère pouvoir apprécier plus et j'ai hâte de réaliser le travail de codage de cette commande en ayant connaissance des outils que j'ai sélectionné.

9 Annexes

9.1 Sources

9.1.1 Présentation de l'entreprise

- <https://l3i.univ-larochelle.fr/Presentation-317>
- <https://l3i.univ-larochelle.fr/Equipes>
- <https://www.univ-larochelle.fr/wp-content/uploads/pdf/ULR-FICHES-LABO-L3i-2020-vf-web.pdf>

9.1.2 Présentation de la Plateforme GALACTIC

<https://galactic.univ-lr.fr/slides/architecture/Galactic-Architecture-slides-20min-complete.pdf>

9.1.3 Les moteurs de squelettes

- Cheetah3 - <https://cheetahtemplate.org/>
- Jinja2 - <https://jinja.palletsprojects.com/en/3.1.x/>
- Template - <https://git.shore.co.il/nimrod/template>
- Mako - <https://www.makotemplates.org/>

9.1.4 Les interfaces de lignes de commande

- CLICK - <https://click.palletsprojects.com/en/8.1.x/>
- Typer - <https://typer.tiangolo.com/>
- Cleo - <https://cleo.readthedocs.io/en/latest/>

9.1.5 Poetry

Poetry - <https://python-poetry.org/>