

Licence Informatique

La Rochelle Université

*Création d'installateurs Windows et MACOs
pour la plateforme GALACTIC*



Martin Ehlinger

2022

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

Table des matières

1	Introduction	4
1.1	Entreprise	4
2	Stage	5
2.1	La plateforme GALACTIC	5
2.2	Sujet de stage	8
3	Déroulement du stage	9
3.1	Documentation	9
3.2	Installation	10
3.3	Développement	11
3.3.1	Raccourcis	11
3.3.2	Plugins	12
3.3.3	Optimisation	12
3.3.4	Installation finale	13
3.3.5	Désinstallation	14
3.4	Résultat et Option	14
4	Travail restant à réaliser	16
5	Conclusion	17
6	Annexe	18
6.1	Liens	18

List of Figures

1	Icône de la plateforme GALACTIC	5
2	Architecture de la plateforme GALACTIC.	7
3	NSIS	9
4	MSYS2	11
5	ToDo list	19

Résumé

Je suis Martin EHLINGER, étudiant en 3ème année de licence Informatique à l'Université de La Rochelle. Actuellement en stage au sein du L3i de La Rochelle.

Mon rôle lors de ce stage, est la création d'installateurs pour les systèmes d'exploitation majeur : Windows et macOS (Linux déjà réalisé). Mon objectif est atteint pour l'installateur pour Windows, quelques optimisations ainsi qu'une documentation qui explique le fonctionnement de mon travail est nécessaire et est en cours de développement. La partie macOS sera réalisée, sur demande de ma part et avec condition qu'elle soit acceptée, après mon stage durant les mois de juillet/août.

J'ai eu l'occasion de travailler sur la plateforme GALACTIC, 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.

I am Martin EHLINGER, a 3rd year Bachelor of Computer Science student at the University of La Rochelle. Currently in internship at the L3i de La Rochelle.

My role during the internship, is the creation of installers for major operating systems: Windows and macOS (Linux is already done). My goal is achieved for the installer for Windows, some optimizations as well as documentation that explains the operation of my work is necessary and is under development. The macOS part will be carried out, on my request and on condition that it is accepted, after my internship during the months of July/August.

I had the opportunity to work on the GALACTIC platform, the objective of the platform is to be able to put in place a framework to work on the Formal Analysis of Concepts as well as the Lattice Theory.

Remerciements

Je tiens à remercier Monsieur Yacine GHAMRI-DOUDANE, Responsable du L3i et Monsieur Muhammad Muzzamil LUQMAN, Responsable du recensement des stagiaires au sein du L3i, pour l'accueil chaleureux qu'ils m'ont témoigné au sein du Laboratoire Informatique, Image et Interaction durant toute la durée de mon stage.

Je souhaite remercier tous les membres travaillant sur le projet Galactic, je vous remercie de m'avoir accueilli au sein de l'équipe en tant que stagiaire.

Je voulais également remercier Madame Karell BERTET d'avoir accepté d'être ma tutrice de stage ainsi d'avoir expliqué le déroulement d'un travail de recherche dans un laboratoire, mais aussi pour la présentation et l'explication de l'algorithme NextPriorityConcept

Mais remercie également Monsieur Christophe DEMKO d'avoir pris le temps de m'expliquer la plateforme Galactic et de m'avoir aidé tout au long de mon stage sur la réussite de mon objectif de stage.

Pour finir, je remercie, Eléonore THONNEAU, Jules BRICOU, Alexandre BUFFARD et Kévin VOISIN qui ont été de bons collègues de travail et qui m'ont aidé durant le stage. Ainsi que pour la rédaction commune de la présentation de l'entreprise et de la plateforme GALACTIC.

1 Introduction

1.1 Entreprise



La présentation de l'entreprise est issue du site de l'entreprise (<https://l3i.univ-larochelle.fr/Presentation-317>) et il s'agit d'un travail fait en commun avec Jules BRICOU, Kévin VOISIN, Alexandre BUFFARD, Eléonore THONNEAU 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 Stage

2.1 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 Jules BRICOU, Kévin VOISIN, Alexandre BUFFARD, Eléonore THONNEAU et moi-même

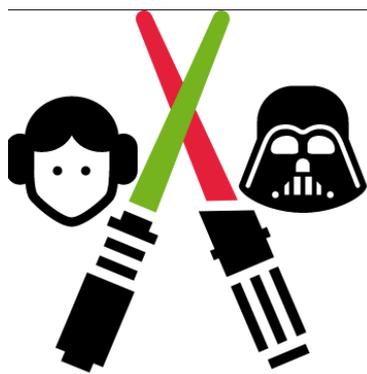


Figure 1: Icône de la plateforme GALACTIC

Le logo de la plateforme a été pensé de la façon suivante :

- Les deux sabres lasers croisés sont en fait un hommage au mathématicien Évariste Galois qui est mort au cours d'un duel galant lorsqu'il était âgé de vingt ans.
- Quant aux images représentant la Princesse Leia Organa et Dark Vader de Star Wars, c'est pour représenter le terme "GALACTIC".

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.

L'Analyse Formelle des Concepts implique la définition de concepts (paires formées par un objet et un attribut) dans un contexte donné, ils sont tous deux complètement et précisément définis. L'Analyse Formelle des Concepts est utilisée pour la classification et le regroupement car elle permet l'association de plusieurs objets à une même description.

En mathématiques, les treillis sont des structures algébriques utilisés en algèbres. Il s'agit d'ensembles partiellement ordonnés dans lesquels chaque paire d'éléments admet une borne inférieure et une borne supérieure.

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 structure.
- Des plugins, représentés par les pétales de la fleur:
 - Caractéristiques : Booléenne, Numérique, Catégorique, String, Séquentielle, Chain, Triadic. Ils permettent de définir les caractéristiques.
 - Descriptions : Booléenne, 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 : Booléenne, 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, sucesor 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 interactice les treillis contruits.
 - 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.

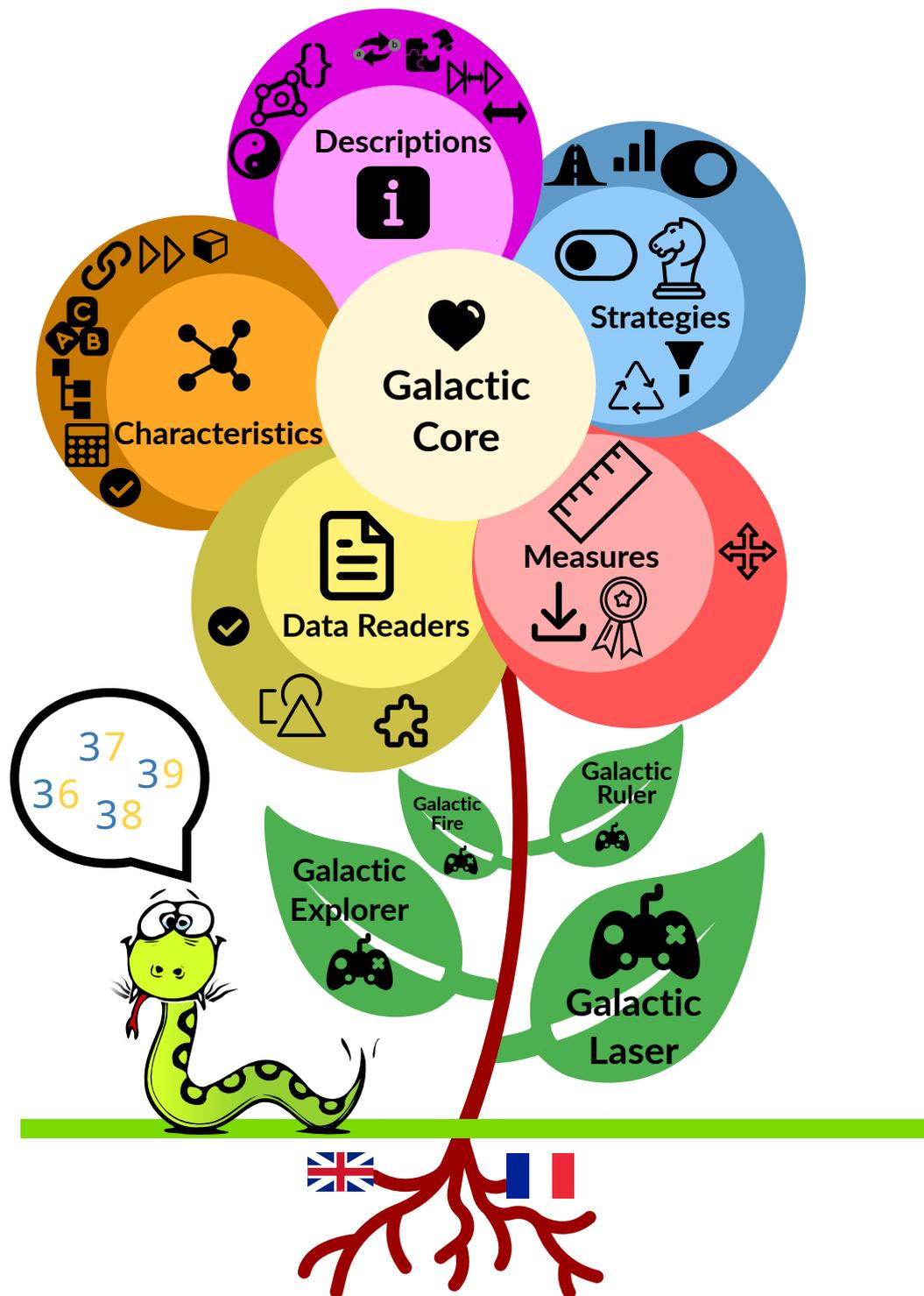


Figure 2: Architecture de la plateforme GALACTIC.

2.2 Sujet de stage

Lors de ce stage, j'avais deux objectifs à réaliser. Le premier était de réaliser un installateur pour le système d'exploitation Windows. Le deuxième objectif était de faire l'équivalent, mais pour le système d'exploitation macOS via le gestionnaire de paquet Homebrew.



Le système d'exploitation Windows étant installé sur la grande majorité des ordinateurs personnels dans le monde, la création d'un installateur de la plateforme Galactic était donc prioritaire sur ce système d'exploitation. De plus, l'installateur pour le système d'exploitation Linux étant déjà réalisé, la création de l'installateur de la plateforme Galactic pour macOS était donc plus facilement réalisable. En effet, macOS et Linux appartiennent tous deux à la famille UNIX.

3 Déroulement du stage

3.1 Documentation

Le premier jour de mon stage, mon maître de stage m'a présenté la plateforme sur laquelle j'allais travailler ainsi qu'une idée de départ de comment j'allais aborder mon stage. Ma priorité était donc de me concentrer sur la création d'un installateur pour le système d'exploitation Windows. Utilisant macOS depuis des années, le développement et la création d'un exécutable sous Windows m'était inconnus. J'ai passé les deux premières semaines de mon stage à me documenter sur le sujet. J'ai trouvé quelques logiciels dans la création d'exécutable pour Windows, comme par exemple InstallShield. Malheureusement ce logiciel, comme la plupart des autres logiciels de ce type, est payant.

Dans mes recherches, j'ai pu élaborer quelques idées possibles afin de développer l'installateur. D'abord avec WSL (Windows Subsystem for Linux), travailler avec un véritable noyau Linux aurait pu être une bonne idée, mais ça ne correspondait pas aux attentes. J'ai également eu l'idée de travailler avec l'IDE Visual Studio de Microsoft qui propose de créer un installateur à partir d'un programme. Malheureusement, le programme doit être écrit en C ou C++, or la plateforme GALACTIC est écrite en Python.

Un ancien stagiaire avait déjà essayé de faire quelque chose. J'ai remarqué qu'il avait utilisé le logiciel NSIS (Nullsoft Scriptable Install System), j'ai donc décidé de travailler avec ce dernier. NSIS, est un logiciel libre contrôlable par script, qui permet la création d'installateurs pour Windows (Wikipedia).



Figure 3: NSIS

Cependant, pour des raisons matérielles, je n'ai pas pu commencer à travailler immédiatement le code. Ayant un MacBook depuis le début de mes études et utilisant le gestionnaire de paquets Homebrew, je me suis donc renseigné pour la création du paquet Galactic sur Homebrew. Homebrew possède deux sortes de paquets, les "cask" qui sont les applications comme Firefox par exemple et les "formulae" qui sont des paquets comme nous avons l'habitude de voir sur Linux. Tout comme Linux, Homebrew sait où installer les paquets et respecte la sandbox qu'Apple impose sur macOS.

Homebrew utilise le langage de programmation Ruby, ce fut pour moi une découverte amusante et intéressante d'apprendre ce langage. L'avantage qu'il y a avec Homebrew, c'est qu'il suffit de lui indiquer le dossier/fichier compilé et il s'occupe de la création de la "formulae/cask" en lui spécifiant de quel type de paquet il s'agit. Une fois ceci fait, Homebrew crée un fichier nomDeLaFormulae.rb (rb est l'extension de fichier pour le langage Ruby), ce fichier comprend de base, le hashage afin de vérifier l'intégrité du paquet lors du téléchargement et l'installation de ce dernier une fois qu'il sera mis en ligne et à la disposition de tous.

3.2 Installation

Après la documentation, j'ai pu commencer à travailler sur l'exécutable d'installation pour Windows. Avec mon Maitre de stage, nous avons discuté sur la façon dont j'allais aborder la création de l'exécutable, en effet de nombreuses choses ont été testé avant mon arrivée, sans résultat concluant. Pour commencer, je devais installer MSYS2 afin de pouvoir installer les dépendances nécessaires à la plateforme Galactic comme par exemple GTK. Une fois ceci fait, j'ai vérifié si il était possible de déplacer le dossier d'installation de MSYS2 sans que cela impact l'installation. S'il y avait un impact, je devais trouver une autre solution, heureusement, il n'y en avait pas. J'ai donc cherché à installé les paquets Galactic avec la commande suivante.

```
>_ | pip install --upgrade --find-links https://galactic.univ-lr.fr/packages  
    ↪ --pre py-galactic[all]
```

Cependant, une erreur survint à chaque tentative. J'ai essayé d'isoler le problème en installant les paquets via l'invite de commande de Windows et avec WSL (Windows Subsystem for Linux). Les deux méthodes fonctionnaient parfaitement, le problème venait donc de MSYS2. En ajoutant "--verbose" à la commande d'installation, nous remarquons (mon Maitre de stage et moi-même) que le soucis venait d'un paquet qui essayait de se réinstaller. Après quelques recherches nous comprenons donc que la "--pre" dans l'installation cherchait à installer les dernières versions de tous les paquets, y compris les dépendances. Cependant dans ces paquets, certains servent à la compilation et dans ces paquets, certains étaient des versions instables.

Une fois le problème réglé, un autre problème survint venant d'un autre paquet. Cette fois-ci, et malgré différentes recherches, impossibles de trouver une solution au problème. Sur conseil de mon maître de stage, j'ai donc testé l'installation de chaque paquet un par un. J'ai remarqué à la fin de chaque installation que les paquets de la [docs] de la plateforme Galactic étaient la source du problème. En effet la plateforme Galactic est composée de différents groupes de dépendance : [docs], [apps], [intl], [fr_FR], [data] et enfin [all] qui réunit les 5 précédents.

J'ai installé, avec réussite, les paquets sans la partie [docs] :

```
>_ | pip install --find-links https://galactic.univ-lr.fr/packages  
    ↪ "py-galactic[apps, intl, fr_FR, data]>=0.5.0.dev0"
```

Une fois l'installation réussie, une question s'est rapidement posée : pouvons-nous toujours déplacer le fichier MSYS2 sans soucis ? Malheureusement, ce ne fut pas le cas, en effet les chemins des exécutables de la plateforme, galactic-laser-ui.exe et galactic-ruler-ui.exe, étaient écrit en dur. Par conséquent, nous devons donc, lors de l'installation, récupérer le fichier MSYS2, l'installer à l'endroit voulu puis uniquement à partir de ce moment installer les paquets de Galactic. MSYS2 est une plate-forme de distribution et de développement de logiciels qui permet de porter des logiciels du monde Unix sur Microsoft Windows. (Wikipedia)



Figure 4: MSYS2

3.3 Développement



Note

Tout au long de cette partie, vous allez voir des variables qui commencent par un \$ comme par exemple \$INSTDIR ou \$SMPROGRAMS. Ce sont des variables natives à NSIS (<https://nsis.sourceforge.io/Docs/Chapter4.html#variables>).

Une fois l'installation des paquets réussie, j'ai commencé à travailler sur le script qui va générer l'exécutable d'installation. Avant de commencer, je me suis fait une To Do list afin de mieux m'organiser dans mon travail. Tout d'abord, j'ai récupéré le script d'un ancien stagiaire en supprimant les parties inutiles selon moi. La partie graphique lors de l'installation était déjà faite ainsi que le plus important de la partie de désinstallation du programme.

3.3.1 Raccourcis

À partir de ce script, j'ai fait les raccourcis de Windows pour les applications afin d'avoir une meilleure expérience utilisateur.

```
# create shortcut product_name.exe with icon
CreateShortcut "$SMPROGRAMS\product_name.lnk" "$INSTDIR\product_name.exe"
↳ "" "img/icon"
```

“CreateShortcut” permet de créer un raccourci au chemin indiqué qui suit. \$INSTDIR est une variable initialisée au début du script et/ou au début de chaque “section”, elle indique où va se trouver le dossier d'installation et autres fichiers/dossiers si besoin.

Voici un exemple :

```
# this is the installation directory
InstallDir "C:\${PRODUCT_NAME}"

# this is a section
Section

    # set the installation directory as the destination for the following
    ↳ actions
    SetOutPath $INSTDIR
```

Comme beaucoup de logiciels, on peut les désinstaller depuis le Panneau de configuration de Window. Pour ce faire, nous devons utiliser “WriteRegStr HKLM” de NSIS. Afin d'afficher les informations dans

le panneau de configuration nous devons préciser le type de l'information avec "DisplayName" ou encore "Publisher" qui servent respectivement à afficher le nom et l'éditeur du logiciel. Nous pouvons également utiliser des informations imprécises comme la taille du logiciel avec "EstimateSize"

```
# Add uninstall and product informations to "Add/Remove Programs"
WriteRegStr HKLM
  ↳ "Software\Microsoft\Windows\CurrentVersion\Uninstall\name"
  ↳ "DisplayName" "name"
```

Il est également possible d'ajouter des icônes aux raccourcis ainsi que dans le panneau de configuration. Il suffit d'indiquer au script une image et d'indiquer le chemin de cette image (une fois l'installation faite) aux raccourcis. Pour le projet, j'ai décidé de copier-coller les images avec **File /r "img"** cependant je pense trouver une solution plus optimale comme par exemple ajouter les images directement depuis le dossier en ligne.

3.3.2 Plugins

Par la suite, j'ai cherché à installer le dossier MSYS2 sans les paquets galactic, le plus simple pour les tests a été de copier-coller un dossier préparé. Cependant de façon optimale il faudrait pouvoir télécharger un fichier compressé puis de le décompresser à l'endroit voulu. Nativement, NSIS ne supporte pas ces deux options, il faut donc ajouter des plugins. Les plugins doivent être ajoutés manuellement dans le dossier d'installation NSIS, une page sur le site officiel explique comment faire (https://nsis.sourceforge.io/How_can_I_install_a_plugin). Dans ce cas précis, j'avais besoin de deux plugins **Inetc** pour télécharger le dossier 7z (ce plugin supporte HTTPS) et **Nsis7z** pour l'extraction des fichiers téléchargés.

Avant d'installer ces deux plugins, j'avais testé d'autres plugins. ZipDLL pour les dossiers en .zip et le plugin natif de NSIS, NSISdl. Pour une raison inconnue ZipDLL ne fonctionnait pas, le plugin était bien chargé et appelé, mais ce dernier ne décompressait pas le dossier. Pour NSISdl, il ne prenait pas en charge HTTPS, mais uniquement HTTP. Pour des raisons de sécurité, nous avons donc opté pour Inetc.

3.3.3 Optimisation

Afin de faciliter les mises à jour et d'éviter aux futurs développeurs de chercher parmi les dizaines de lignes de codes du script la bonne valeur, des variables sont initialisées au début du script. On a par exemple la version ou encore le nom du produit. Il suffit de les appeler avec \${VARIABLE}.

Exemple :

```
!define PRODUCT_NAME "Galactic" # Name of the product
```

```
# create the uninstaller
WriteUninstaller "$INSTDIR\${PRODUCT_NAME}-uninstall.exe"

# create galactic directory
CreateDirectory "$SMPROGRAMS\${PRODUCT_NAME}"
```

3.3.4 Installation finale

Comme vu plus haut, le chemin de l'exécutable est écrit en dur, nous devons donc effectuer le pip install après avoir installé le dossier. Nous devons faire appel au terminal de MSYS2, nommé "mingw64". Au début, j'ai cherché un moyen de l'exécuter directement sans passer par un logiciel tiers. J'ai très vite opté pour une autre option qui était de faire appel à l'invité de commande de Windows qui se chargera d'exécuter le pip install via mingw64. Je remercie Monsieur DEMKO pour avoir trouvé la solution au problème, en effet, mes diverses tentatives avaient échoué. Lors de l'installation une fenêtre pop up affichait l'installation du pip install. Bien que cela soit pratique pour vérifier le bon fonctionnement, cela peut réduire l'expérience utilisateur. Il fallait donc afficher les étapes d'installation dans la fenêtre d'installation en cliquant sur "plus d'infos."

Pour ce faire, il suffisait d'ajouter ToLog :

```
nsExec::ExecToLog '$INSTDIR\galactic-bin\usr\bin\bash.exe -l -c
↳ "/mingw64/bin/pip.exe install --find-links
↳ https://galactic.univ-lr.fr/packages
↳ py-galactic[apps,intl,data]==${VERSION}"'
```

3.3.5 Désinstallation

La dernière section du script sert à la désinstallation complète du programme lorsque l'utilisateur exécute l'exécutable "uninstall.exe". Pour ce faire, on doit dans un premier temps supprimer tous les raccourcis et toutes les informations liés au panneau de configuration puis finir par supprimer le dossier d'installation de façon récursive.

```
# uninstaller section start
Section "uninstall"
  # Delete informations from Add/Remove programs
  DeleteRegKey HKLM
  → "Software\Microsoft\Windows\CurrentVersion\Uninstall\product_name"

  # Delete link
  Delete "$SMPROGRAMS\product_name.lnk"
  Delete "$DESKTOP\product_name.lnk"

  # Delete product folder
  RMDir /r "$INSTDIR" # Delete all files in $INSTDIR
  RMDir $INSTDIR # Delete $INSTDIR
# uninstaller section end
SectionEnd
```

3.4 Résultat et Option

Nous avons donc enfin un installateur parfaitement fonctionnel, il installait bien à l'endroit voulu et désinstaller bien correctement tout ce qu'il fallait sans laisser aucune trace du logiciel par la suite. Cependant, tous les tests ont été effectués sur une machine en ayant les droits d'administrateur, ce qui pose problème, car sous Windows, certains dossiers comme "Program Files" sont des dossiers où il faut être administrateur pour y avoir accès. Il faut donc ajouter une condition en fonction de si l'utilisateur est administrateur ou non.

Tout d'abord, il faut modifier le dossier d'installation par défaut qui doit être dans un dossier où un utilisateur non administrateur ou administrateur puisse installer sans erreur, par exemple la racine du disque C:. Ensuite, il faut spécifier aucune restriction d'installation.

```
RequestExecutionLevel highest
```

Ici "highest" permet de récupérer le plus haut statut de l'utilisateur courant pour l'exécution. Si on met "admin" à la place, seul les administrateurs peuvent installer le logiciel.

Pour finir, il faut récupérer le type d'utilisateur puis en fonction du résultat exécuter une tâche.

```
# Function who blocked install when non-admin user try to install on
↳ "Program files" folder.
Function .onVerifyInstDir
    UserInfo::GetAccountType
        pop $1
        ${If} $1 != "admin" ;Require admin rights on NT4+
            StrLen $0 $PROGRAMFILES64
            StrCpy $0 $INSDIR $0
            StrCmp $0 $PROGRAMFILES64 0 PathGood
            Abort
        PathGood:
    ${EndIf}
```

Cette fonction empêche un utilisateur non administrateur, d’installer dans les dossiers “Program Filesx86” et “Program Files”, afin d’éviter toute erreur lors de l’installation.

En tant qu’utilisateur, il est toujours agréable de pouvoir avoir le choix d’installer ou non certaines choses comme les raccourcis sur le bureau.

```
!insertmacro MUI_PAGE_COMPONENTS

...

# Section for installing desktop shortcut
Section "desktop shortcut"

...
...

SectionEnd

# Section for add fr_FR packages, unchecked by default
Section /o "fr_FR"

...

SectionEnd
```

Pour ce faire, il faut ajouter l’affichage avec “!insertmacro MUI_PAGE_COMPONENTS”, puis nommer les sections afin qu’elles s’affichent en option.

Le “/o” permet de décocher la valeur par défaut.



Warning

Si vous nommez d’autres sections, celles-ci seront également affichées en option. Cependant la section uninstall doit être nommée, cette dernière ne sera pas en option, mais si vous ne le faites pas, le logiciel ne pourra pas se désinstaller.

4 Travail restant à réaliser

J'aurais réalisé le plus important de mes objectifs durant ce stage, mais malheureusement, je ne pense pas avoir le temps de finir ce que je souhaitais faire. En effet, je dois encore optimiser l'installateur pour Windows ainsi que réaliser celui de macOS. C'est pour cette raison que j'ai demandé à mon maître de stage de continuer le développement après la fin de mon stage. J'ai grandement apprécié travailler sur ce projet et je souhaite finir ce que j'ai commencé.

Pour Windows, il reste à régler les petits soucis et autres bugs pouvant être présents ainsi qu'optimiser davantage le code pour les futurs développeurs. Je dois également réaliser une documentation afin d'expliquer le script pour toute modification future. Je vais essayer de changer le thème GTK pour Windows.

Pour macOS, je dois commencer de zéro. Ayant de bonnes connaissances et l'habitude de travailler sur macOS, je pense pouvoir réaliser ce travail sans grande difficulté.

5 Conclusion

Lors de ce stage, j'ai été très bien accueilli par l'équipe du projet Galactic ainsi qu'au sein du L3i. Mon objectif durant ce stage n'était pas le plus simple et je ne pensais pas réussir dans les temps. En effet, j'ai d'abord eu des soucis matériels, m'empêchant de travailler sur l'installateur pour Windows. Ce n'était qu'un simple contretemps qui fut de courte durée et j'ai eu suffisamment de temps pour me renseigner afin de pouvoir établir une ToDo list (voir Annexe).

J'ai donc réalisé les choses dans l'ordre suivant :

- Documentation
- installation des dépendances
- création de raccourcis
- exécutable pour désinstaller
- optimisation
- télécharger le dossier et l'installer en le décompressant
- gestion des utilisateurs non administrateurs
- option d'installation
- documentation et autre optimisation (en cours de développement)

Cette ToDo list a évolué au fil du temps en fonction des attentes et des demandes de mon maître de stage. Ce stage a été une excellente expérience pour moi. En effet j'ai des soucis de concentration et d'organisation depuis toujours, par exemple lorsque je cherche quelque chose, si je trouve une réponse à un autre problème, plutôt que de garder cette solution de côté, j'applique immédiatement la solution au point d'en oublier ma recherche initiale. Je me suis donc obligé plus d'une fois à rester concentré sur ce que je faisais. La ToDo list m'a également aidé à mieux m'organiser et a plus facilement resté concentré sur mon travail.

Pour conclure ce rapport de stage, je suis très honoré et très heureux d'avoir pu travailler sur le projet Galactic. J'ai non seulement découvert de nombreuses choses intéressantes, mais j'ai également pu m'améliorer dans mon travail de manière générale et ce, malgré quelques petites difficultés. Bien que je n'ai pas pu terminer tout ce que je souhaitais faire initialement, j'ai demandé l'accord de mon maître de stage de continuer sur la plateforme Galactic après mon stage.

6 Annexe

6.1 Liens

G.

- Galactic : <https://galactic.univ-lr.fr/>

H.

- Homebrew : <https://brew.sh/>

I.

- Inetc : https://nsis.sourceforge.io/Inetc_plug-in
- InstallShield : <https://www.revenera.com/install/products/installshield>

L.

- L3i : <https://l3i.univ-larochelle.fr>

M.

- MSYS2 : <https://www.msys2.org/>
- MSYS2 Wikipedia : <https://fr.wikipedia.org/wiki/MSYS2>

N.

- NSIS : <https://nsis.sourceforge.net>
- NSIS Wikipedia : https://fr.wikipedia.org/wiki/Nullsoft_Scriptable_Install_System
- Nsis7z : https://nsis.sourceforge.io/Nsis7z_plug-in
- NSISdl : https://nsis.sourceforge.io/NSISdl_plug-in

W.

- WriteRegStr HKLM : <https://nsis.sourceforge.io/Reference/WriteRegStr>
- WSL : <https://ubuntu.com/wsl>

Z.

- ZipDLL : https://nsis.sourceforge.io/ZipDLL_plug-in

```
--TO DO--

add docs --> dont forget explain install nsis and plugins
add apps --> dont forget someapp unfoctionnal at the time

URGENT ! add  pacman -S mingw-w64-x86_64-python3-wrapt

--DONE--
icon
dl
extract
cmd popup in install
licence
non admin --> check if install on prog files (admin)
question install

--WIP--
script
installer windows gtk--> nothing change, try again
```

Figure 5: ToDo list