LA PRÉPA DES INP

Refonte des profils d'imprimante 3D du logiciel IceSL

Stage de recherche S4

Rapport de :
Yan RABEFANIRAKA
Maître de stage :
Sylvain LEFEBVRE
Superviseurs :
Pierre-Alexandre HUGRON
Salim PERCHY

Laboratoire :
MFX - INRIA
615 RUE DU
JARDIN-BOTANIQUE, 54600
VILLERS-LÈS-NANCY

06 Mai 2024 - 14 Juin 2024

Résumé

Dans le cadre du stage de recherche de 2^{me} année à la Prépa des INP, j'ai eu l'occasion d'assister, pendant 6 semaines, l'équipe MFX de l'INRIA. Ayant eu un rôle proche de l'ingénieur informatique, j'ai pu y découvrir sa place ainsi que sa relation avec les chercheurs et leurs innovations. J'y ai également découvert le fonctionnement d'un laboratoire de recherche. Ainsi, dans ce rapport, je vais détailler le fonctionnement de MFX, tout en parlant des missions qui m'ont été confiées et des activités auxquelles j'ai pu participer. Enfin, j'aborderai mon ressenti sur ce stage.

Table des matières

Introduction **2**

1 MFX : Matter From Graphics **3**
1.1 Aperçu . . . 3
1.1.1 Membres et Organisation de MFX . . . 4
1.1.2 Les ingénieurs et IceSL . . . 5
1.2 Recherches et publications . . . 7
1.2.1 Apparences anisotropes et matériaux déformables . . . 7
1.2.2 PCBend . . . 9
1.2.3 Shrink and Morph . . . 10
1.3 IceSL . . . 11

2 Ma place et ma mission **14**
2.1 La Piscine et l'environnement de travail . . . 14
2.2 Comprendre IceSL . . . 16
2.2.1 Format des profils de machine . . . 16
2.2.2 Récupération et Étude de Données . . . 17
2.2.3 Conséquences dans la prise de décision . . . 18
2.3 *Textual* et Python . . . 18
2.3.1 Module complémentaire . . . 18
2.3.2 Angle d'approche . . . 19
2.4 Mêler matériel et virtuel . . . 22

3 Résultats et Validation **24**

Ressenti personnel et conclusion **25**

Annexes **27**

Remerciements

Avant tout, je tiens à remercier mon maître de stage Sylvain Lefebvre, qui a accepté de m'accueillir pour 6 semaines dans les locaux de MFX, et qui m'a accordé sa confiance pour mener à bien ma mission.
Je remercie également Pierre-Alexandre Hugron qui a été tout particulièrement présent pour m'assister dans mes missions, mais qui a surtout su me proposer des activités qui sortent de l'ordinaire—néanmoins toujours dans le thème de l'impression 3D. Ces remerciements vont également à Salim Perchy, qui m'a aussi largement assisté pour éclairer ma mission, mais aussi à Xavier Chermain, qui a su nous prendre—mon collègue stagiaire et moi—sous son aile.
Je n'oublie pas le reste de l'équipe qui m'a gentiment accueilli, avec qui les repas étaient toujours divertissants et intéressants, et qui a contribué à l'ambiance chaleureuse et humaine de mon séjour.
Enfin, des remerciements particuliers vont à mon collègue stagiaire Antoine Marion, avec qui nous nous sommes assurés assistance et compagnie mutuelle tout au long du stage.

Introduction

Ayant intégré pendant 6 semaines l'équipe de recherche MFX spécialisée en modélisation et en impression 3D, J'y ai accompagné une partie de l'équipe dans l'amélioration de leur logiciel de tranchage 3D fait-maison *IceSL*, pour l'impression 3D.

L'impression 3D, ou fabrication additive, est une technique de fabrication consistant en un dépôt continu de matières couche par couche, et ce afin de créer une pièce en volume. Cette technologie prend une place de plus en plus importante, notamment dans le monde industriel, avec par exemple des machines de dépôt de béton, de céramique, de chocolat, ou encore avec des vêtements de haute couture et des chaussures imprimées en 3D. Cet intérêt vient de son opposition avec la technique traditionnelle dite de fabrication soustractive (comme la sculpture par exemple), dont le principe est de retirer de la matière jusqu'à obtenir la forme désirée, et qui par conséquent consomme bien plus de ressources. Mais la pratique se démocratise surtout auprès de particuliers souhaitant en faire un hobby ou un outil professionnel / artistique. Ainsi, avec l'explosion sur ces dernières années du marché, de très nombreuses imprimantes 3D, de la plus classique à la plus étriquée, sont disponibles à l'achat de tous. Rien que dans le logiciel de tranchage (souvent appelé *slicer*) 3D le plus populaire *Ultimaker Cura* [1], on compte environ 700 machines **répertoriées**. Ces fameux logiciels de tranchage permettent justement de faire le pont entre le modèle 3D que l'utilisateur veut fabriquer et son modèle d'imprimante, en le traduisant en instructions compréhensibles uniquement par ce genre de machines (ainsi que d'autres machines industrielles) : le *G-Code*. En effet, d'une imprimante à l'autre, les fonctionnalités, l'interprétation du *G-Code*— en clair tout ce qui est nécessaire pour imprimer en 3D— peut drastiquement différer. Les logiciels de tranchage ont donc généralement un répertoire d'imprimantes 3D avec les informations essentielles leur concernant, ce qui leur permet de créer des profils d'impression facile à comprendre pour l'utilisateur de tous les jours. Chez MFX, *IceSL* possède 55 profils de machines disponibles, mais il est très difficile pour un utilisateur débutant d'y décrire sa propre machine, et donc de pouvoir utiliser le logiciel.

Ce séjour au sein de l'équipe MFX, abordé au Chapitre 1, m'a permis de découvrir de nombreuses facettes du métier d'ingénieur, notamment ses liens avec les chercheurs. Pour ce qui est de mon rôle j'ai eu principalement pour mission, détaillée aux Chapitres 2 et 3), pendant ces 6 semaines de stage, d'effectuer la refonte du système de gestion des profils de machine sur le logiciel IceSL, et ce afin de faciliter l'ajout de nouveaux modèles mais aussi la modification des entrées existantes. Enfin, je parle aussi dans le Chapitre 3 de mon ressenti personnel.

1. https://ultimaker.com/software/ultimaker-cura/

Chapitre 1

MFX : Matter From Graphics

1.1 Aperçu

FIGURE 1.1 – Locaux du LORIA / INRIA.

Juste à côté du Jardin Botanique Jean-Marie-Pelt à Villers-Lès-Nancy, se situe dans les laboratoires du LORIA l'équipe MFX (*Matter From Graphics*). Il s'agit d'une équipe de recherche collaborative de l'INRIA (*Institut National de Recherche en Informatique et en Automatique*), du CNRS (*Centre National de la Recherche Scientifique*) et de l'Université de Lorraine, créée en 2018.

Étant au laboratoire de l'INRIA, son domaine d'expertise est évidemment l'informatique : Plus particulièrement, elle s'intéresse à tout ce qui concerne la 3D, de la modélisation au rendu graphique, en passant par l'impression 3D. Elle cherche en particulier à développer des algorithmes novateurs permettant à des utilisateurs de tout horizon d'exploiter tout le potentiel de la fabrication additive.

Cependant, elle innove aussi dans des domaines encore peu explorés, comme les matériaux à mémoire de forme (1.2.3 et [4]), nécessitant des travaux de recherche pluridisciplinaires et incitant régulièrement MFX à collaborer avec d'autres équipes, françaises comme étrangères.

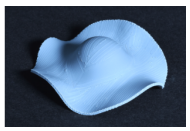


FIGURE 1.2 – [4] Papier à mémoire de forme.

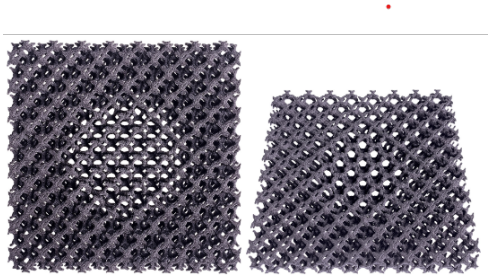


FIGURE 1.3 – [2] Nivellement concentrique imprimé en 3D.

En effet, bien que la fabrication additive se démocratise casuellement auprès des particuliers qui veulent rendre concrets des objets virtuels, elle est avant tout en ce moment même au coeur d'une révolution technologique significative pour les industries. C'est en particulier le cas du côté des ingénieurs et designers qui cherchent à créer des formes géométriques plus complexes (*i.e* 1.3), aux méthodes de fabrication strictes et demandantes. Elle offre un gain de temps considérable et

une nouvelle facilité à prototyper tout produit en Recherche et Développement. Ceci rend les travaux de cette équipe essentielles dans l'intégration de ces nouvelles technologies aux circuits industriels actuels : outre l'apport important d'innovations (c.f section 1.2 où seront détaillés quelques travaux de recherche innovants), les objets très détaillés et fonctionnels produits par leurs techniques d'impression nécessitent de pouvoir être précisément visualisés, interactifs ; cela passe donc par leurs algorithmes de visualisation.

1.1.1 Membres et Organisation de MFX

FIGURE 1.4 – Photo de l'équipe au complet.

MFX est une équipe-projet [1] à taille humaine, avec 8 employés titulaires, appuyés de temps en temps par des doctorants / post-doctorants ou encore des étudiants en stage de validation de Master. Pour autant, elle aura vu passer de nombreux collaborateurs et anciens membres, qui ont soit changé d'équipe de recherche, soit été recruté(e)s par de grandes entreprises privées, comme le géant *Adobe* par exemple.

Aussi, le travail de chercheur demande très régulièrement de partir en déplacement professionnel pour divers conférences, travaux de recherche conjoints, présentations ou conventions, dans toute la France comme à l'étranger. À titre d'exemple, mon maître de stage Sylvain Lefebvre était en conférence à New York la semaine de mon arrivée dans l'équipe. Ainsi, je n'ai pu fréquenter quotidiennement qu'une partie de l'équipe lors de mon stage. Voici donc un organigramme détaillant le rôle des membres que j'ai eu l'occasion de rencontrer.

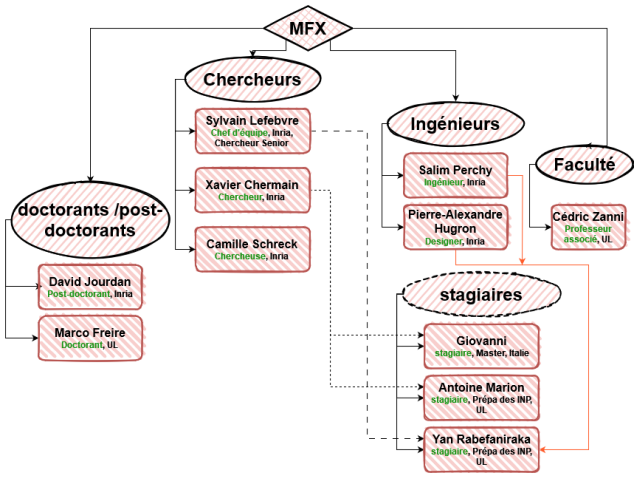


FIGURE 1.5 – Organigramme des membres que j'ai fréquenté(e)s.

Par ailleurs, David Jourdan ayant terminé son post-doctorat lors de mon séjour, il nous a quitté pour présenter son sujet d'étude au jury du concours CRCN [2] afin de devenir chercheur à l'INRIA

1. dirigée par un chef de projet, une équipe-projet se consacre à un unique thème à travers un prisme pluridisciplinaire.
2. https://www.inria.fr/fr/chargees-et-charges-de-recherche-de-classe-normale-crcn

Grenoble. Dans ce genre de situations où un membre vient à présenter ses travaux à un jury externe, toute l'équipe profite de la réunion du Jeudi matin pour lui faire réviser sa soutenance, en donnant des conseils/suggestions et des critiques constructives sur sa présentation, afin de le/la préparer au mieux. Chacun(e) en profite évidemment pour lui souhaiter bonne chance pour la suite.

1.1.2 Les ingénieurs et IceSL

FIGURE 1.6 – logo IceSL.

Avant tout, chez MFX, les résultats de recherches sont principalement implémentés dans le trancheur *IceSL* [3]. Son but premier est de regrouper leurs innovations dans un logiciel libre d'accès (avant tout pour les autres chercheurs) : cela leur permet de prototyper de nouveaux algorithmes à l'aide des machines disponibles au local, mais il s'agit surtout de distribuer leurs travaux à toute la communauté scientifique.

Le site de l'équipe [4] arbore d'ailleurs fièrement le *Graphics Replicability Stamp* [5] ou *Tampon de Réplicabilité*. Il s'agit d'une initiative communautaire, qui consiste en l'approbation de travaux de recherche non-commerciaux jugés reproductibles et rendant service au monde de la recherche graphique, en fournissant des implémentations prêtes et facilement modifiables.


FIGURE 1.7 – Tampon de Réplicabilité Graphique.

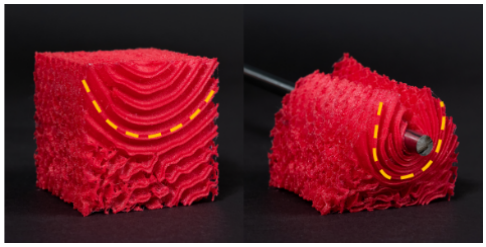


FIGURE 1.8 – Bloc déformable par remplissage anisotrope. En appuyant sur les faces aux remplissages non-orientées du bloc, il ne se déforme pas.

Ainsi, que ce soit de nouvelles techniques d'affichage 3D, de rendu de matériaux ou de simulation de reflets lumineux sur une surface, toutes les innovations graphiques développées par l'équipe finissent tôt ou tard accessibles sur *IceSL*. On y retrouve également des techniques d'impression 3D expérimentales, avec par exemple des remplissages anisotropes progressifs(1.2.1) qui font que l'objet imprimé va se déformer différemment selon la direction de la pression physique appliquée. Le logiciel est abordé dans la section 1.3 et nécessaire à mes missions dans la section 2.2.

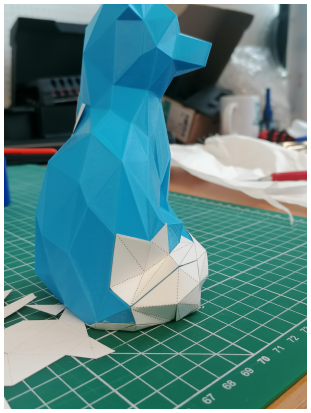
Toutes ces nouvelles techniques nécessitent donc une application pratique : en passant d'abord par le tranchage sur *IceSL*, elles peuvent être ensuite testées directement sur les machines du laboratoire. Alors, il faut aussi gérer, entretenir et modifier en adéquation le matériel disponible. Cela inclut des imprimantes 3D à filament traditionnelles sans oublier celles à résine, mais aussi des découpeuses laser (c.f 1.9a), des bras industriels, des imprimantes à argile ou même à chocolat (c.f 1.10). Ces instruments permettent des réalisations pratiques pour d'autres

3. https://icesl.loria.fr/
4. https://mfx.loria.fr/software/
5. https://www.replicabilitystamp.org/

sujets de recherche comme développé en 1.2.2 avec des circuits imprimés éclairés.

(a) Découpeuse laser Trotec Speedy 100.

(b) Découpage du patron.



(c) Renard imprimé en 3D. Le patron du papier est automatiquement généré en envoyant le modèle du renard dans IceSL pour le préparer à la découpe laser.

(d) Renard imprimé en 3D et recouvert de papier découpé au laser.

FIGURE 1.9 – Matériel et réalisation pratique. Fonctionnalité trouvable sur IceSL.

C'est dans ce contexte que Salim Perchy et Pierre-Alexandre Hugron, les ingénieurs de l'équipe, évoluent. Leur rôle est d'implémenter les travaux de recherche de leurs camarades chercheurs, ainsi que de vérifier leur applicabilité sur les machines. On peut les qualifier d'ingénieurs en recherche appliquée ou *RD Engineers* en anglais.

Salim Perchy est avant tout membre du Service d'Expérimentation et de Développement (*SED*) de l'INRIA, un service ne regroupant que des ingénieurs. L'objectif de ces derniers est de travailler avec les chercheurs des différentes équipes afin d'expérimenter avec leurs résultats de recherche et de les implémenter. Néanmoins, par une initiative de l'INRIA, certaines équipes se sont vues attribuer un ingénieur du SED pour 4 ans avec pour volonté d'améliorer la cohésion entre les chercheurs et les ingénieurs, et ainsi d'améliorer les implémentations. C'est ainsi que Salim Perchy travaille au sein de MFX en tant qu'ingénieur / développeur. C'est lui qui se charge principalement de l'implémentation des nouvelles fonctionnalités sur le logiciel *IceSL*, avec Sylvain Lefebvre et Pierre-Alexandre Hugron. Il a par exemple comme projet, tout juste terminé, l'implémentation en langage *Python* de *IceSL* (dénommé en conséquence *PyiceSL*), qui est normalement codé en *Lua* [6] et en *C++*.

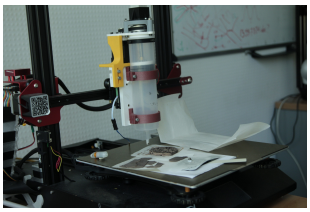


FIGURE 1.10 – Imprimante à chocolat.

Quant à Pierre-Alexandre Hugron, il est le designer de l'équipe. Il a établi le design et l'interface du logiciel, mais il gère aussi toutes les machines, de l'imprimante 3D à la découpeuse laser, sans oublier l'imprimante à argile ou encore celle au chocolat. Il en assure la maintenance régulière, les paramétrages, teste les configurations possibles, et lance des impressions pour prototyper les trouvailles de ses collègues.

6. langage de script pouvant être embarqué dans d'autres applications afin de les étendre. Particulièrement utilisé dans les systèmes embarqués, le réseau et les jeux vidéo.

À mon arrivée, il en était même à monter de toute pièce sa propre imprimante 3D, afin de mieux comprendre leur fonctionnement et afin d'incorporer toutes les fonctionnalités dont a besoin le reste de MFX. De cette façon, c'est avec lui que j'ai passé le plus de temps lors de mon stage, avec des activités pratiques notamment (c.f 2.4).

Mais son travail ne s'arrête pas là : il s'occupe aussi de prendre des photos, avec du matériel professionnel, de leurs impressions 3D ou découpages laser depuis la petite installation photo dans son bureau. Il en fait également la retouche sur *Photoshop*. Cela permet d'illustrer correctement les documents de ses collègues, mais aussi de promouvoir leur travail auprès de tous, depuis les différentes pages *Instagram* / *Twitter* de l'équipe.

FIGURE 1.11 – Installation photo.

1.2 Recherches et publications

L'équipe arrive tout particulièrement à innover dans l'anisotropie [7] au sein de l'impression 3D, tout en gardant un domaine de recherche élargi, comme la découpe-laser de lumières LED pour embellir les objets électroniques ou les impressions 3D (c.f 1.2.2). Cependant, elle travaille aussi régulièrement en étroite collaboration avec différents acteurs scientifiques de la région sur des projets pluridisciplinaires : on peut nommément parler du projet ORTHOSIS 4D, qui regroupe l'Institut Jean Lamour (IJL), le LORIA, le Laboratoire Réactions et Génie des Procédés (LRGP), l'Équipe de Recherche sur les Processus Innovatifs (ERPI), l'Institut Régional de Médecine Physique et de Réadaptation de Nancy (IRR) ainsi que le CHU de Nancy. Le but de ce projet est d'étudier la fabrication d'objets fins et déformables dans le domaine de la santé et du sport, et ainsi d'étendre l'applicabilité de l'impression 3D en octroyant aux objets imprimés la capacité de réagir à leur environnement comme attendu d'eux.

FIGURE 1.12 – ORTHOSIS 4D s'inscrit dans le cadre de l'initiative Lorraine Université d'Excellence.

1.2.1 Apparences anisotropes et matériaux déformables

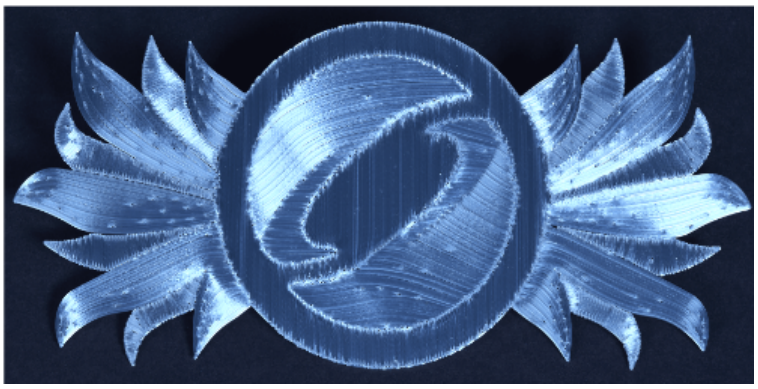


FIGURE 1.13 – Impression 3D à l'apparence de métal brossé. La direction de la trajectoire détermine la capacité de réflexion de la lumière par la surface.

7. se dit d'un objet dont les propriétés dépendent de la direction.

FIGURE 1.14 – Tapis rugueux à poils longs. Un motif s'y dessine de par l'orientation différente des poils qui reflète inégalement la lumière.

Les travaux de recherche de l'équipe sur la fabrication d'apparences anisotropes [1] offrent une nouvelle technique d'impression 3D où les résultats ressemblent à des gravures ou à du métal brossé. Cela se fait à partir de l'orientation de la buse d'impression quand elle vient déposer le filament dans une direction différente de ce qui l'entoure, similaire à un tapis à poils longs dont on aurait brossé les poils dans une direction au lieu de l'autre, changeant la manière à laquelle la lumière se reflète dessus. En effet, la direction du chemin de dépôt appose une rugosité variable au matériau, offrant donc ce visuel anisotrope. Une partie de ces recherches a d'ailleurs servi de base pour le projet d'informatique du S3 (c.f annexe 3) avec la couture de cycles servant d'étape charnière pour permettre cette rugosité variable.

Cet algorithme peut être appliqué à des photos ou des tableaux afin d'en faire des décorations murales après impression. C'est ainsi que l'image 1.15 décore fièrement la salle des machines.

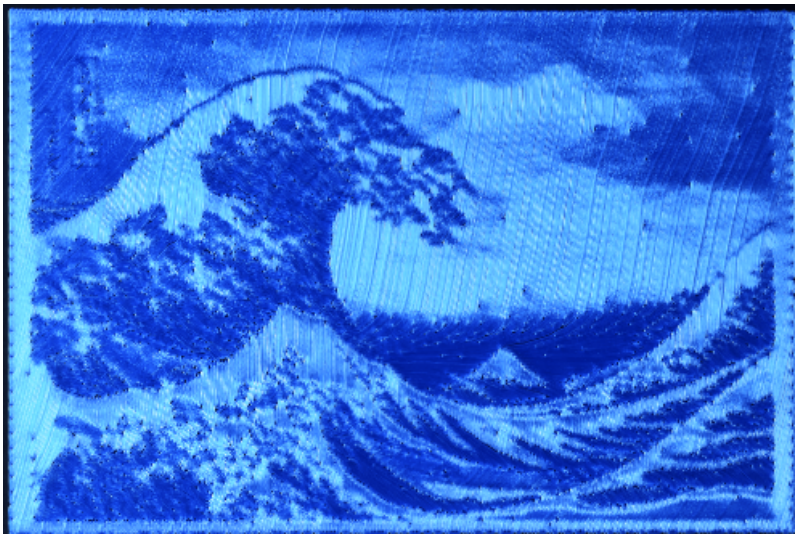


FIGURE 1.15 – Illustration supplémentaire : Impression 3D à apparence de métal brossé de La Grande Vague de Kanagawa, Hokusai.

On peut aussi mentionner le remplissage anisotrope flexible mentionné en 1. Ce dernier consiste en un remplissage progressif de l'intérieur de la forme à l'aide de bruit généré procéduralement, en changeant les angles de la trajectoire afin de provoquer l'anisotropie et donc de permettre la déformation.

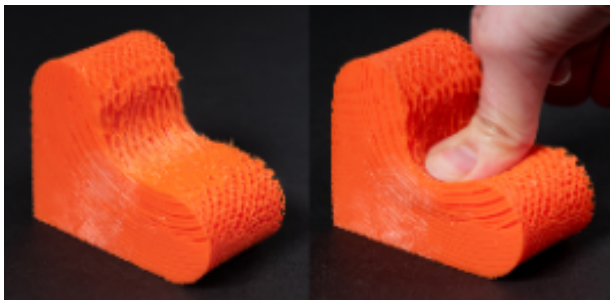


FIGURE 1.16 – Prototype de siège miniature déformable et à mémoire de forme. Le remplissage anisotrope orienté autour de l'assise permet de le rendre déformable uniquement à cet endroit.

1.2.2 PCBend

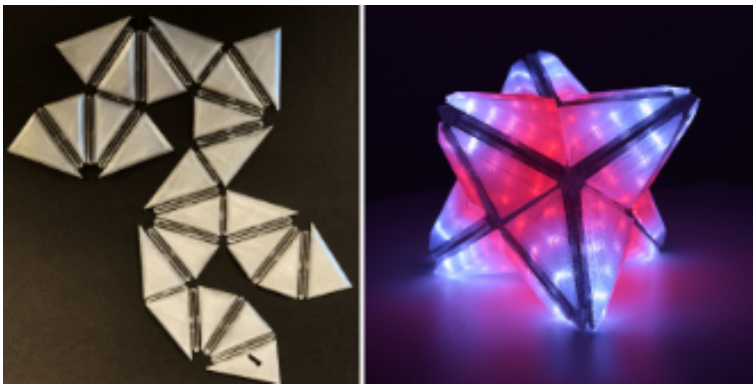
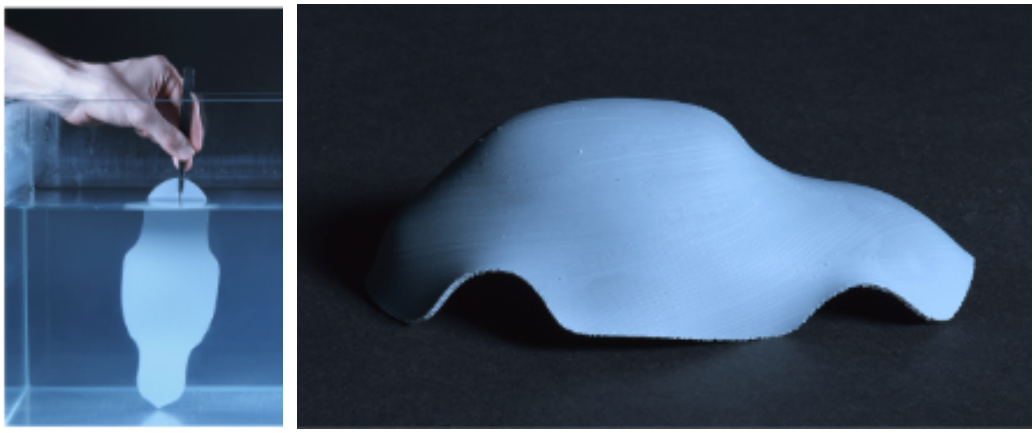


FIGURE 1.17 – Étoile diffusant de la lumière avec un motif intérieur-vers-extérieur.

PCBend[3] est un projet conjoint entre MFX et l'équipe *Computer Graphics and Digital Fabrication*, de l'*Institute of Science and Technology Austria* (ISTA). Menés par Marco Freire (doctorant à l'INRIA)et Manas Bhargava(doctorant à l'ISTA), le but de se projet est de créer un algorithme permettant de découper efficacement au laser des circuits imprimés équipés de LED afin de créer un patron pliable de la forme géométrique recherchée. Cela inclut aussi la résolution des problèmes liés à la flexibilité limitée des circuits imprimés, en découpant automatiquement des motifs qui serviront de jointures. À ce stade, les prototypes ont été réalisés avec des circuits imprimés dont la densité de LEDs insallés est basse. Il est en théorie possible cependant d'y mettre des écrans plus détaillés et puissants.

En se projetant dans l'avenir, cette technologie pourra alors être utilisée pour continuer à développer les écrans pliables, mais sera aussi utile pour les constructeurs automobiles et les designers urbains afin qu'ils se séparent du format d'écran rectangulaire et adoptent des formes plus complexes qu'ils pourront appliquer à des surfaces.

1.2.3 Shrink and Morph



(a) Immersion de la plaque à température de transition vitreuse [8].

(b) Résultat après déformation de la plaque à haute température.

FIGURE 1.18 – Déformation d'une plaque en PLA (Acide polyactique, très fréquemment utilisé pour les impressions 3D car biodégradable et peu cher à produire) par rétrécissement afin d'obtenir une forme désirée, "programmée" par les trajectoires de dépôt de filament.

Le projet *Srhink and Morph* est le sujet de post-doctorat de David Jourdan qu'il a présenté au jury du CRCN. Il s'agit d'un projet hautement pluridisciplinaire, réunissant l'impression 3D, la **physique des matériaux déformables** et le génie des matériaux, avec un grand champ d'applicabilité.

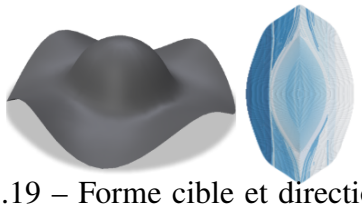


FIGURE 1.19 – Forme cible et directions d'impressions pour un chapeau.

Le but de se projet est de créer un algorithme qui peut exploiter la rétraction des matériaux thermoplastiques comme le PLA afin d'en faire des plaques imprimées en 3D, se changeant en une fine forme désirée, après réchaud. En fait, ces matériaux conservent les tensions internes appliquées dans les directions du dépôt de matière, et les relâchent quand chauffés à température de transition vitreuse. Cela cause une déformation anisotrope et courbe la plaque ; il faut donc placer adéquatement les dépôts de matière pour que la plaque se déforme jusqu'à plus ou moins prendre la forme cible. Tout ceci a été expérimentalement implémenté sur des imprimantes FDM standards, après que l'algorithme ait fourni l'ensemble de chemins.

Ces expérimentations avec la mémoire de forme sont particulièrement intéressantes dans l'optimisation de l'utilisation de matière en impression 3D : en effet, fabriquer des formes fines cause régulièrement des problèmes d'alignement de couche, causant des fragilités structurelles. C'est pour cela que l'impression de formes fines nécessite actuellement plus de ressources pour générer les supports d'impression [9] que pour générer la forme finale.

9. analogues à des échafaudages, un support d'impression vient soutenir la masse de matière afin d'éviter qu'elle ne tombe ou ne se déforme grandement.

1.3 IceSL

L'apogée de leurs travaux de recherche se retrouve dans le logiciel libre *IceSL*, où toutes leurs approches novatrices sont intégrées et utilisables. On peut entres autres citer les différentes innovations anisotropiques comme le remplissage flexible (c.f 1.2.1) dénommé "*Phasor*" dans les paramètres du slicer, et les apparences anisotropes de métal brossé (c.f 1.2.1) appelées "*Optizor*". Dans sa version la plus complète *IceSL-Forge*, il fait à la fois logiciel de modélisation 3D et trancheur pour imprimantes.

(a) IceSL Forge intègre une fenêtre de scripts en Lua dans laquelle on donne les instructions de modélisation au logiciel. Les modifications sont visualisables en temps réel.

(b) IceSL Slicer est une version allégée ne faisant que le tranchage.

FIGURE 1.20 – Les deux versions du logiciel 1.20a et 1.20b sont à installer sur la machine de l'utilisateur. Elles sont aussi accessibles en version Web sur le site du logiciel avec *Slicecrafter*, une itération moins puissante mais accessible n'importe où.

La modélisation se fait entièrement en scriptant en Lua, à la différence des logiciels habituels comme *Blender*, *Maya* ou *Rhino* qui offrent avant tout des outils interactifs de fabrication, de transformation ou de déplacement de forme (c.f annexe 3). Dans *IceSL*, cela consiste à "émettre" *ex-nihilo* des formes géométriques basiques (les plus simples étant les cubes et sphères), sur lesquelles on effectuera des opérations dites *booléennes*, en référence à l'l'Algèbre de Boole. Concrètement, on pourra assembler, soustraire et intersecter ces formes de base afin d'en faire des plus complexes. *Forge* est aussi équipé de fonctions spéciales applicables aux objets ; par exemple, en 1.21, la fonction *to_voxel_solid* indique au logiciel qu'il faut recréer le maillage de la forme en en faisant un assemblage de petits blocs, comme des *Lego*®.
Ensuite, le logiciel peut exporter le maillage au format STL (*Standard Tesselation Language*) [10], permettant notamment de l'utiliser dans d'autres trancheurs ou logiciel de modélisation 3D.

```
1  obj_1 = rotate(0, 45, 0)*ccube(40)
2  obj_2 = to_voxel_solid(sphere(25), 1)
3  emit(difference(obj_1, obj_2), 3)
```

FIGURE 1.21 – Scripting avec Lua. Les changements sont instantanément visibles comme illustré en 1.22.

10. format de fichier décrivant la géométrie de surface d'un objet en 3 dimensions. On n'obtient d'information ni sur sa couleur, ni le matériau, ni la texture. C'est le format par défaut pour la fabrication additive.

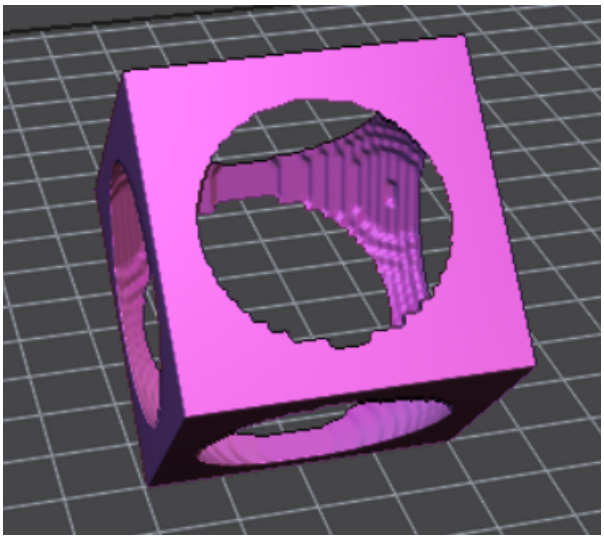


FIGURE 1.22 – Maillage obtenu à partir de la commande en 1.21. On a soustrait ce qui était une sphère recomposée en petits blocs (les fameux "*voxels*) à un cube.

Quant au slicer, il n'intègre pas qu'une fonction de tranchage (de fichier STL à G-code donc) ; il propose en prime plusieurs services comme la conversion en un patron découpable au laser, et la conversion couche par couche du modèle en fichier SVG (*Scalable Vector Graphics*) [11].

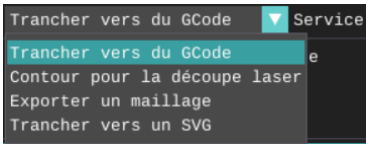
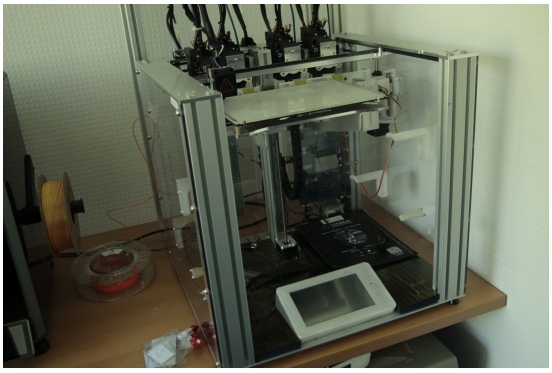


FIGURE 1.23 – Fonctionnalités intégrées au Slicer.

Pour ce qui est du support de machines, le logiciel possède pour l'instant 55 machines prises en charge, allant de la plus généraliste comme la *Creality Ender3* (en 1.24a) à la plus raffinée comme la *E3D Toolchanger* (en 1.24b), qui peut changer d'extrudeur en pleine impression en attrapant le suivant avec un aimant. Les profils les plus détaillés et compatibles sont ceux des machines disponibles au laboratoire, sur lesquelles l'équipe a pu longuement expérimenter et a en conséquence pu établir les meilleurs paramètres d'impression. Cela représente entre une dizaine et une quinzaine de machines. Pour les autres, la grande majorité est le résultat de contributions d'utilisateurs qui, désireux de voir leur machine supportée, ont créé leur propre profil.

11. format de données décrivant un ensemble de graphiques vectoriels 2D. Très utilisé en graphisme ou en cartographie pour créer des images infiniment agrandissables sans perdre de netteté.

(a) Creality Ender3. Une imprimante accessible, ciblant les particuliers, surtout les débutants.



(b) E3D ToolChanger de la salle des machines. Possède 4 extrudeurs différents, qu'elle échange et attrape à l'aide d'un aimant.

FIGURE 1.24 – Deux des machines supportées par *IceSL*. La 1.24b est trouvable au laboratoire.

Cependant, cet aspect communautaire ainsi que l'ajout progressif de machines très distinctes sont paradoxalement une source de problèmes au sein du code source de *IceSL*. Effectivement, comme il n'y a pas de cadre strict et arbitraire sur l'écriture de ces profils, chacun d'entre eux est écrit d'une manière plus ou moins différente des autres. Combiné aux idiosyncrasies de chaque imprimante, cela crée des écarts notables dans l'intégration de certaines machines, en particulier les plus complexes. On compte alors, dans la plupart des profils de machine, quelques redondances de code, le rendant ballonné et plus difficile à mettre à jour pour chaque machine. C'est ainsi que certaines se retrouvent privées de nouvelles fonctionnalités puisque cela demanderait d'intégrer individuellement à chacune dans leur profil ladite fonctionnalité. Il était donc nécessaire d'uniformiser le format global des profils de machine afin de faciliter l'évolution du logiciel.

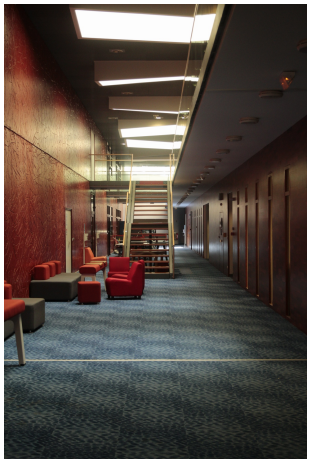
Chapitre 2

Ma place et ma mission

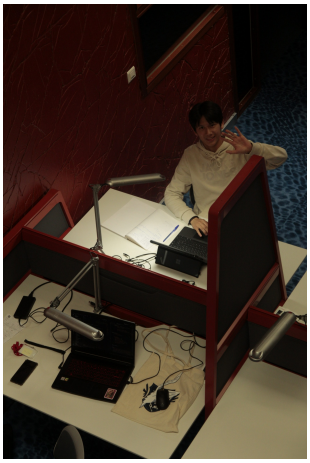
C'est dans ce contexte que ma mission s'inscrit : mon maître de stage Sylvain Lefevbre m'a donc proposé de m'intéresser à la construction actuelle des profils de machine, afin de refaçonner ceux existants en les uniformisant. La finalité serait de créer une structure générale et universelle, permettant de rajouter de nouvelles imprimantes ou d'en modifier. Ce que je fais pourrait donc s'apparenter à du *code clean-up* (nettoyage de code) ou du *code review* (ou révision de code), c'est-à-dire que mon but est d'optimiser un code déjà existant en l'épurant. On verra tout au long de ce chapitre que j'ai tout de même dû, pour m'aider, créer un outil numérique. Pour cela, j'ai été supervisé par Salim Perchy et Pierre-Alexandre Hugron. Le premier m'éclaire principalement sur le code et la manière de s'y prendre, tandis que le second m'aide sur l'aspect matériel et pratique, sans oublier les spécificités des machines.

2.1 La Piscine et l'environnement de travail

Tout d'abord, il me fallait trouver un endroit où travailler car, dans le cas d'un stage si court, il est compliqué pour l'équipe de trouver une place aux stagiaires dans les bureaux. C'est alors que je me suis installé dans l'*Open Space* du LORIA, la *Piscine*(c.f 2.1a). Il s'agit d'un très long et large couloir meublé de plusieurs espaces de repos confortables et de travail mobile. Elle est aussi ornée d'une moquette bleue, à motif océanique très distinctif, ce qui lui a valu son nom si particulier de *Piscine*.
Beaucoup d'employés s'en servent lors de leur pause de midi pour y parler et se détendre, mais aussi pour des réunions virtuelles, des tests de machine , ou pour changer d'environnement de travail. Elle mène aussi à différentes salles de réunion réservables en un instant, analogues aux salles de travail de la Bibliothèque Universitaire que les étudiants peuvent réserver en ligne.
L'endroit est habituellement plutôt propice pour travailler efficacement (muni d'un bon casque anti-bruit, pour supprimer les discussions de couloir qui résonnent au loin). Pour autant, il y a des moments légèrement plus difficiles sur le plan sonore : entre les réunions improvisées en dehors des salles prévues à cet effet et les discussions autour d'un café, il peut être compliqué de se concentrer si on n'est pas équipés pour.



(a) Open Space du Loria, appelée la *Piscine*.



(b) Notre bureau avec Antoine Marion. On amène nos propres ordinateurs. On peut les brancher sur les prises de la table et les éclairer avec une lampe amovible.

(c) Borne d'Arcade de la *Piscine*.

FIGURE 2.1 – Notre installation à la *Piscine*.

De plus, trône au milieu de la salle une borne d'arcade (c.f 2.1c) ; celle-ci contient aux alentours de 300 jeux-vidéo rétros [1], souvent tirés de vieilles bornes d'arcade. Pour entretenir l'esprit de compétitivité entre les joueurs, elle a à son côté gauche un tableau blanc où sont annotés les meilleurs scores du jeu *Tetris*.

Mais il ne s'agit pas du seul dispositif cherchant à améliorer le confort de travail du laboratoire ; Il y a aussi la *Napbox*, qui est en fait une boîte fabriquée en carton, où sont installés deux poufs et un plaid afin de permettre aux employés de prendre une sieste si nécessaire. C'est ainsi que la plupart de mes pauses du midi se sont passées à l'intérieur de la *Napbox*, afin de me préparer pour l'après-midi.

FIGURE 2.2 – La *Napbox*.

C'est donc dans la *Piscine* que je suis installé tous les jours (c.f 2.1b), de 8h30 à 17h. Ces horaires sont flexibles et n'ont pas été établis par le laboratoire mais selon mes préférences. En n'effectuant qu'1h30 de pause, de 11h30 à 13h, je peux me libérer aux alentours de 17h, moyennant l'arrivée du prochain bus. En effet, la flexibilité de ces horaires tend plus à allonger la durée de travail pour ma part, puisque l'unique bus pour rentrer chez moi, bien que passant sur la rue du laboratoire, ne passe qu'à des créneaux bien écartés les uns des autres (faut-il encore qu'ils passent bel et bien ou qu'ils s'arrêtent comme prévu...). Ainsi, il m'arrive régulièrement de partir à 17h30, voire 18h30. Cela peut se révéler tantôt avantageux pour avancer sur ma mission, tantôt éreintant mentalement après une journée tout particulièrement remplie, mais la cause est avant tout le mauvais desservice de ma commune de résidence.
Pour autant, le travail n'est pas spécialement physique : Je passe la plupart du temps assis sur mon emplacement dans la *Piscine*, et me permets régulièrement quelques minutes de pause pour me dégourdir les jambes en marchant dans l'*Open Space*. En effet, il s'agit d'un poste de

1. Se dit des jeux-vidéo d'anciennes générations, aux technologies bien moins poussées.

bureau—un travail particulièrement sédentaire— dans la même veine qu'un ingénieur logiciel.

2.2 Comprendre IceSL

IceSL est le produit direct du travail des ingénieurs Salim Perchy et Pierre-Alexandre Hugron, avec la participation de Sylvain Lefevbre au développement. Comme dit précédemment, il permet d'implémenter le travail des chercheurs de l'équipe, et de le rendre accessible à d'autres acteurs du monde de la recherche gratuitement et facilement. Mais pour cela, il est nécessaire d'intégrer des profils machine. Il s'agit, dans le cas d'IceSL, de dossiers uniques à chaque imprimante prise en charge par le logiciel (disponibles sur le répertoire Github), contenant des informations sur les caractéristiques techniques des imprimantes et permettant à IceSL de les exploiter pour générer correctement du G-Code. La nécessité de les harmoniser s'est vite présentée lorsque les utilisateurs demandent régulièrement un profil spécialement conçu pour leur machine, là où ceux qui sont déjà intégrés possèdent du code redondant, qui n'est pas forcément mis à jour ou qui est ballonné.

2.2.1 Format des profils de machine

Avant même le début du stage, il a été discuté avec Sylvain Lefebvre que le meilleur moyen d'harmoniser les profils de machine était d'observer le profil le plus complexe et ballonné afin d'en faire la structure de base. Cela permettrait d'un seul coup de n'oublier aucune fonctionnalité, aucune variable, et de simplement désactiver ce qui n'est pas pris en compte par chaque machine individuelle.

Le profil des machines se définit comme suit :

— *features.lua* regroupe des constantes qui caractérisent máteriellement l'imprimante. On y retrouve principalement la taille de la zone d'impression, le nombre d'extrudeurs, le diamètre de la buse, celui du filament utilisé, la hauteur d'un dépôt de couche, ou encore la température de la buse et du lit. Par exemple, pour l'Ender3 (image en 1.24a), on définit en 2.3a la taille en millimètre du lit d'impression selon les axes x, y et z. On y définit aussi le logiciel de bord utilisé par le micro-contrôleur (*firmware*), puisqu'il change la manière d'écrire le G-code. Toutes ces valeurs seront utilisées par le logiciel à travers le fichier *printer.lua*.

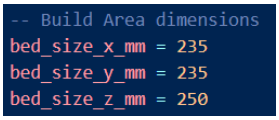
— Le fichier *printer.lua* en 2.3b contient toutes les fonctions utilisées par le logiciel pour convertir le modèle en G-code en adéquation avec les propriétés matérielles de l'imprimante. Elle utilise pour ce faire les constantes définies dans *features.lua*. Cela se constitue nommément d'une fonction *Header()*, où on lance les instructions de départ telles que le chauffage du lit et de la buse ou la mise en position 0 de l'extrudeur, et d'une fonction *Footer()* qui lance les instructions de finition telles que l'extinction du chauffage, et la remise en position 0 de l'extrudeur.

— Un dossier *materials* contient des fichiers pour apporter des détails pour chaque matériau compatible avec la machine, en révisant les constantes définies dans *features.lua*. On parle typiquement du PLA, de l'ABS [2], ou encore du PETG [3]. Les révisions sont souvent liées aux températures de chauffe de la buse et du lit, mais aussi aux vitesses de refroidissement du ventilateur.

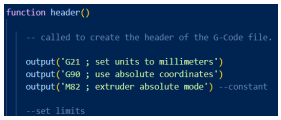
2. Acrylonitrile Butadiène Styrène. Polymère thermoplastique apprécié en fabrication aditive pour sa résistance élevée.

3. Combinaison de polyethylene terephthalate et de glycol. Apprécié en fabrication additive pour sa rigidité et sa transparence.

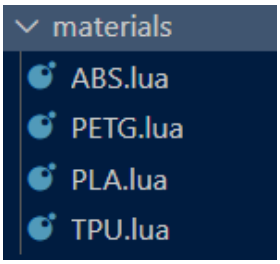
— Un dossier *profiles* regroupe des paramètres prédéfinis, qui révisent les valeurs définies dans *features.lua*. Ils sont souvent liés à la qualité ou à la vitesse attendues de l'impression. On a typiquement un fichier *low.lua* définissant des vitesses d'impression plus élevées et des hauteurs de couches plus hautes pour des prototypes rapides mais peu qualitatifs, ou encore un fichier *high.lua* avec des hauteurs de couches très basses et une vitesse d'impression plus lente pour assurer un meilleur résultat avec de meilleures finitions.



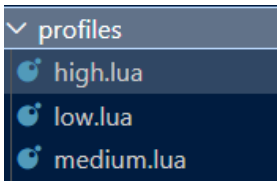
(a) Exemple de *features.lua*.



(b) Exemple de la fonction *Header()* dans *printer.lua*



(c) Exemple de dossier *materials*



(d) Exemple de dossier *profiles*

FIGURE 2.3 – Un exemple de profil de machines.

2.2.2 Récupération et Étude de Données

Ainsi, en premier lieu, afin de comprendre le fonctionnement des différents fichiers d'un profil de machine, mais aussi pour établir des chiffres et me faire une idée sur les fonctionnalités les plus présentes, j'ai commencé par créer un petit *parser* [4] en *Python* qui va lire tous les profils de machine existants et générer un grand dictionnaire dans un fichier *Json* [5](c.f 2.4). Comme attribut, on place la fonctionnalité étudiée (par exemple *support_print_speed_mm_per_sec* définit la vitesse d'impression du support en mm/s), et en contenu la valeur pour chaque imprimante sur laquelle elle est définie.

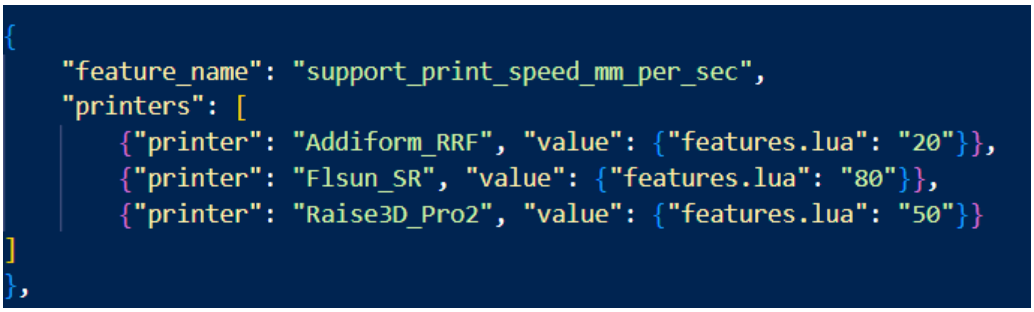


FIGURE 2.4 – Exemple d'entrée dans le dictionnaire *Json* pour la vitesse d'impression des supports.

4. Script qui va prendre des données en entrée et les mettre en page sous forme de structure de données utile à l'analyse de ces dernières.

5. Format de données textuel où les données sont définies en paires *attribut-valeur*. Souvent utilisé en science des données pour réunir beaucoup de valeurs et rendre ça lisible pour un humain.

De cette petite étude est ressorti que le profil le plus complexe était la *Addiform RRF*, qui a été ajouté par un utilisateur. Machine à échelle industrielle (aucune photo disponible malheureusement), elle est équipée de 14 extrudeurs (le second étant le ToolChanger en 1.24b avec seulement 4 extrudeurs). Quant à son profil, il contient une quantité excessive de constantes définies, qui ne seront même pas utilisées par le logiciel et qui ne sont présentes dans aucune autre machine... De cette observation, nous avons conclu que commencer par les machines les plus complexes allait compliquer la tâche, notamment parce qu'elles possèdent des idiosyncrasies trop prononcées. Cela allait constituer une quantité de données superflues conséquente pour les machines les plus générales.

2.2.3 Conséquences dans la prise de décision

C'est alors qu'on a décidé de réorienter le cheminement du projet : Il vaut mieux commencer par les profils les plus généraux, en n'incluant que les fonctionnalités et caractéristiques prises en compte par toutes les imprimantes. Plus encore, Salim Perchy me propose d'y aller entièrement à contresens en débutant la mission par l'outil de création de profils qui n'était d'origine qu'un bonus s'il nous restait du temps. Avec cet outil, il serait plus simple de générer des profils reformatés et nettoyés pour les machines les plus générales, qui constituent la majorité de celles prises en charge par *IceSL*. Les plus complexes seraient alors réservées à plus tard.

C'est ainsi que j'ai réuni les caractéristiques définissant n'importe quelle imprimante 3D avant de débuter l'outil de création :

- Le nom de l'imprimante (définira aussi le nom du dossier à la création du profil).
- Le logiciel de bord du micro-contrôleur (essentiel pour l'écriture du G-code).
- Le format du lit d'impression (circulaire ou rectangulaire) et sa taille.
- Le nombre d'extrudeurs, leur diamètre et celui du filament.
- La hauteur d'une couche.
- La température cible de la buse d'extrusion et celle du lit d'impression.
- La vitesse d'impression globale.
- Quelques paramètres divers tels que le soulèvement de la buse après la fin d'une couche ou encore des paramètres d'accélération.

2.3 *Textual* et Python

Avec ces constantes par défaut, il me faut ensuite créer un outil pour générer un fichier à partir des données entrées par mon utilisateur. Salim Perchy m'a recommandé de le faire en Python puisqu'il a pertinemment observé que je sais relativement bien m'en servir. Cela peut sembler étrange sachant que le reste du code à gérer sera en *Lua*, mais il était essentiel pour lui que je me serve d'un langage que je maîtrise. En effet, ils ont déjà expérimenté avec un stagiaire qui a dû apprendre un nouveau langage difficile en son entièreté (le *C++)* et qui s'y est particulièrement empêtré.

2.3.1 Module complémentaire

Pour implémenter une telle application, principalement composée d'une interface graphique réunissant des champs d'entrée de données, j'ai décidé de passer par la librairie *Textual*. Je l'ai déjà utilisée pour le TP2 d'informatique en 1A, et suis donc un peu familier avec. Bien qu'elle n'est encore qu'en *béta*, la librairie permet d'implémenter rapidement une interface

graphique pour son application, directement au sein du *Terminal* (c.f 2.5) de son ordinateur. Qui plus est, elle est très réactive et permet de déclencher des événements à partir d'autres actions. Par exemple, on peut faire apparaître de nouveaux champs après avoir augmenté le nombre d'extrudeurs, pour que chacun soit caractérisé individuellement selon son diamètre. De même, cela permet d'inclure un bouton "mode avancé" grâce auquel certains paramètres trop techniques pour un utilisateur casuel, cachés par défaut, seront révélés et modifiables. On pense aux valeurs minimales et maximales de la vitesse d'impression par exemple. Enfin, les paramètres désactivés (grisé dans l'interface) seront renvoyés sous forme de commentaires dans le fichier final, et n'auront donc pas d'impact.

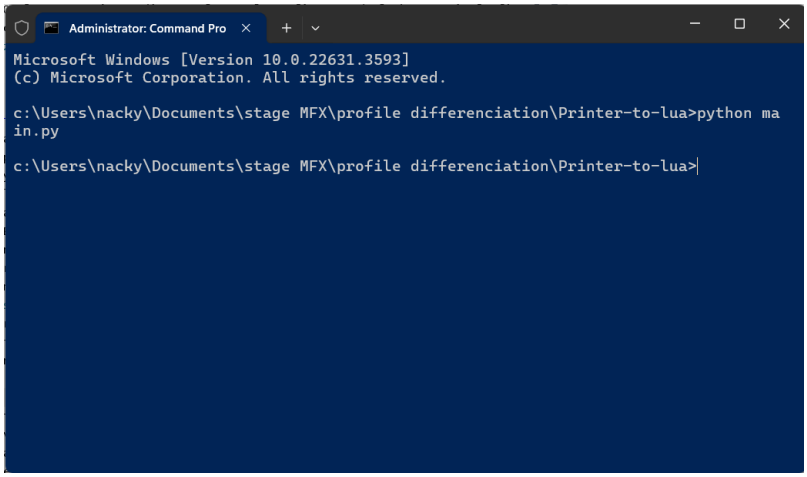


FIGURE 2.5 – Terminal Windows. Aussi appelé console de commandes, il permet d'exécuter des instructions au plus bas niveau du système.

Cela offre à l'utilisateur la capacité d'interagir avec les différentes constantes facilement et intuitivement, mais aussi d'observer en temps réel les fichiers générés. Pour l'équipe, avoir une telle application permettrait de faciliter la maintenance numérique et la mise à jour des machines prises en charge, ainsi que d'en ajouter plus facilement dans la collection. Néanmoins, cette tâche s'est révélée plus compliquée que prévu, puisque la librairie est très riche : elle contient beaucoup de fonctions potentielles, la rendant complexe à pleinement maîtriser et appliquer malgré ma précédente expérience. Devoir apprendre un nouveau module a donc certainement contribué à ralentir mon rythme global.

2.3.2 Angle d'approche

Pour autant, l'implémentation s'est faite sans encombre étape par étape. D'abord, il a fallu penser aux **interactions entre l'utilisateur et l'interface graphique**, afin d'assurer un confort d'utilisation et une facilité optimale. C'est pourquoi cette dernière a été divisée en différents onglets, un pour chaque fichier détaillé en 2.2.1. Ils peuvent interagir entre eux, de sorte qu'une modification faite dans un onglet modifie aussi les autres qui en dépendent. Il a par la suite fallu implémenter les **champs d'entrée** pour les données de l'onglet qui correspond à *features.lua* (les onglets Profiles et Materials sont exactement les mêmes dans leur fonctionnement, à quelques champs près). *Textual* offre plusieurs catégories de champs (c.f 2.6) :

— Les *Inputs* sont des champs textuels dans lesquels on peut entrer une valeur. Le type (soit du texte soit un nombre) de celle-ci est prédéfini et on ne peut s'y déroger. Ils permettent aussi d'établir des conditions de validation sur les entrées (par exemple 'Que des nombres pairs'), utile pour éviter les erreurs et bugs.

— Les sélections sont des menus déroulants.
— Les interrupteurs retournent les valeurs "Vrai" ou "Faux".
— Les boutons lancent une action après pression.

Chaque type de champ me permet de retourner le bon type de variable pour le fichier *features.lua*. Effectivement, les interrupteurs retournent des *booléennes* (Vrai ou Faux), utiles pour les informations comme "Activer le chauffage dans la chambre d'impression" qui n'acceptent que Oui ou Non, par exemple. Les sélections permettent de restreindre les possibilités tels les diamètres de filament qui viennent soit en 1.75mm soit en 2.85mm. Les Inputs renvoient directement la valeur entrée dedans sous forme de chaînes de caractères.

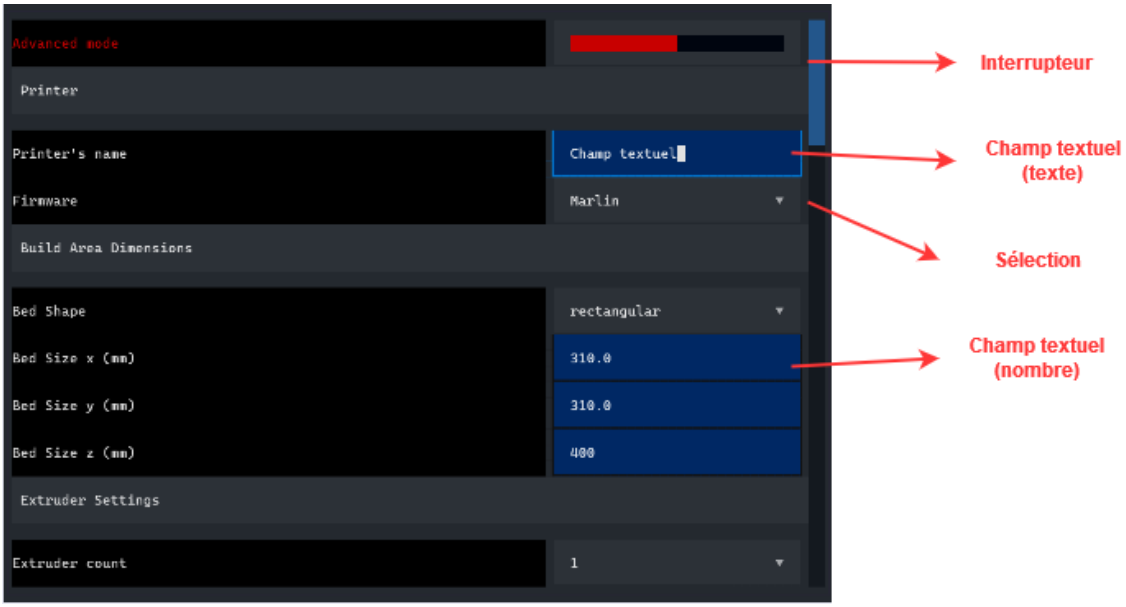


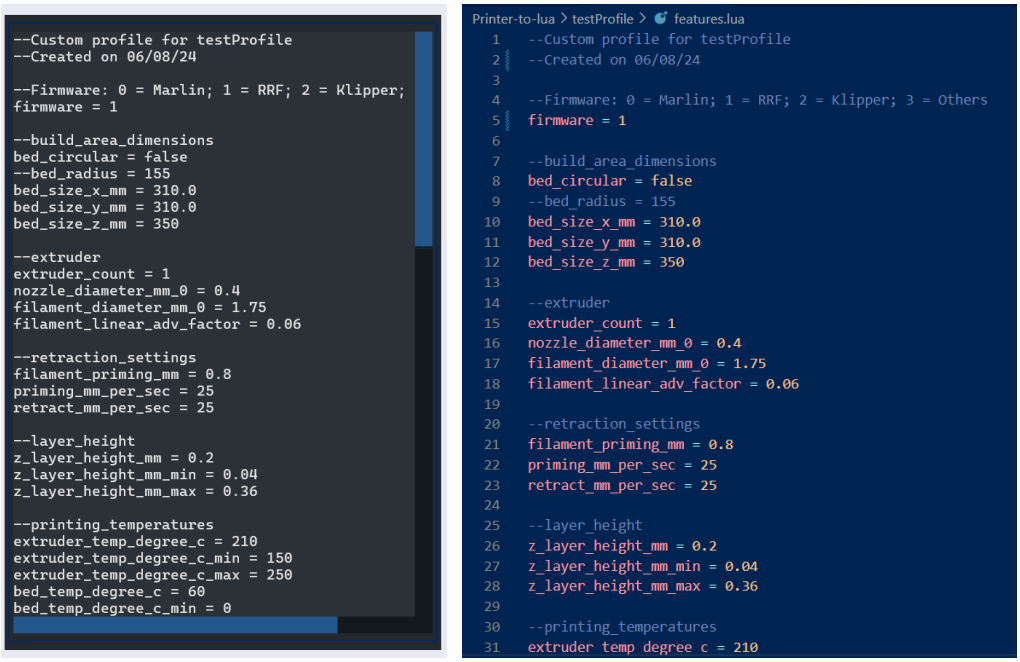
FIGURE 2.6 – Exemple de champs utilisés.

Ensuite, il a fallu créer une **fenêtre de visualisation du résultat**. Avoir un aperçu du fichier offre un gain de temps appréciable en évitant à l'utilisateur de chercher le résultat dans son dossier. Il sera affiché une fois qu'un bouton "**Créer**" aura été appuyé. Bien sûr, cela implique d'autres conditions nécessaires, dont l'obligation d'entrer un nom d'imprimante (sans quoi l'application ne pourra pas créer le dossier).

FIGURE 2.7 – Bouton "créer". Il exécute la création de fichier et affiche le résultat dans la section de droite (c.f 2.8.)

Ce genre de conditions est vérifiée grâce aux fonctions réactives de *Textual* qui surveillent activement les champs et qui ici, peuvent l'écrire dans la fenêtre de visualisation, mais aussi bloquer le bouton de création. Ainsi, des messages comme "Veuillez insérer un nom d'imprimante" peuvent apparaître si rien n'y est mis.
D'autres conditions et dépendances entre constantes ont également été ajoutées. Cela permet, dans le cas où une donnée numérique est indexée / calibrée (à un facteur près) à partir d'une autre, de les modifier simultanément et donc d'éviter à l'utilisateur des problèmes de valeurs incompatibles. On peut nommer la hauteur minimale et maximale des couches qui dépend entièrement du diamètre de la buse. Cette fonctionnalité a été tout particulièrement ardue à implémenter, nécessitant tout d'abord de saisir le sens matériel / physique de ces liaisons (détaillé en 2.4), mais

surtout de comprendre en profondeur le fonctionnement de *Textual*. En fait, ce dernier fonctionne par envoi de "messages", c'est-à-dire qu'à chaque interaction de l'utilisateur avec un élément de l'interface, ce dernier envoie au coeur de la librairie des informations sur ladite interaction. On y récupère son identifiant, sa valeur après modification, son état (caché, désactivé), ou encore si la valeur entrée respecte bien les conditions de validation susmentionnées.
En clair, cet ajout se révélera primordial en 2.4 afin d'assurer une cohérence entre les valeurs.



(a) Exemple d'aperçu de résultat dans la fenêtre de visualisation.

(b) Extrait du résultat réel, généré au sein du dossier où a été lancé l'application.

FIGURE 2.8 – Résultat de l'exécution pour l'onglet "*Features*".

Dans le cas de **l'onglet Printer** en 2.9 (ici appelé *G-code translation*), la tâche est relativement distincte : comme il s'agit non plus de constantes définies simplement comme étant des variables *lua*, mais des fonctions, il est nécessaire de permettre à l'utilisateur de modifier à sa guise le code. Heureusement, Textual possède un *widget* permettant de simuler une zone de code. Bien sûr, il est activement modifié selon les actions faites dans les autres onglets, comme précisé précédemment ; ainsi, activer l'accélération par exemple, va générer 4 lignes en plus dans le *Header()* et le *Footer()* pour que la machine prenne cela en compte.

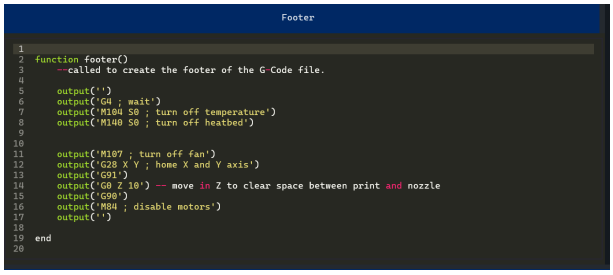


FIGURE 2.9 – Zone de code pour *footer()*. Simule aussi les indentations et les mises en surbrillance d'un éditeur de code typique.

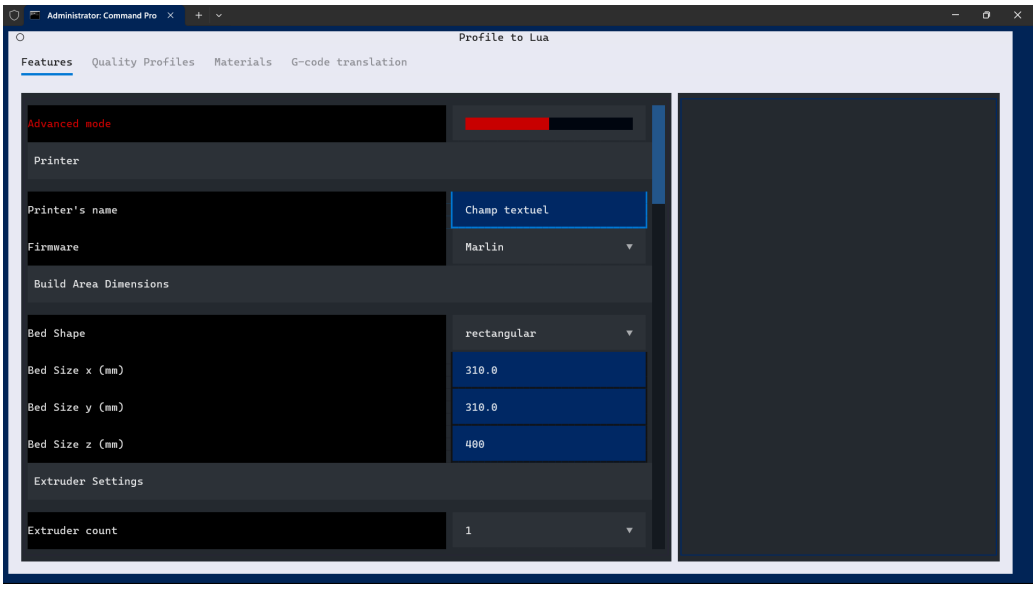


FIGURE 2.10 – Interface graphique à l'intérieur de la console. En haut, un onglet par fichier *lua*. À gauche, les champs d'entrée de données. À droite, l'aperçu du résultat.

Enfin, une fois tous les onglets implémentés, on obtient l'interface en 2.10.

2.4 Mêler matériel et virtuel

L'une des parties constitutive de ce projet est tout de même son lien avec le réel : on retranscrit à l'application des caractéristiques décrivant une vraie machine.
Ainsi, à l'aide de Pierre-Alexandre Hugron et de ses connaissances pointues en imprimante 3D, j'ai défini des valeurs par défaut à chaque champ, les pré-remplissant au cas où l'utilisateur ne sait pas quoi y mettre. Ces valeurs se doivent être générales et compatibles avec n'importe quelle machine. C'est aussi avec lui que j'ai établi la plupart des facteurs permettant d'indexer des valeurs à partir d'autres. Pour instance, la hauteur de couche minimale est définie comme étant $diametre\_de\_buse * 0.1$. La hauteur minimale est elle définie comme étant $diametre\_de\_buse * 0.9$. Cette indexation automatique des valeurs est toute particulièrement significative, puisque de mauvaises valeurs mènent au mieux à des performances médiocres, au pire à des dégâts matériels sur l'imprimante (par exemple une accélération maximale trop élevée qui cause une dislocation de l'extrudeur, ou des températures trop chaudes pouvant endommager l'intérieur de la buse).
Par ailleurs, certaines valeurs particulières (liées à l'accélération) peuvent se retrouver par des formules de conversion. On peut notamment mentionner l'accélération de la buse arrivée sur les angles de 90° ; pour les machines sous logiciel de bord *Marlin*, il s'agit de la valeur *Jerk*. Pour les machines sous *Klipper*, cela s'appelle le *Square Corner Velocity* et vaut en fait $jerk * \sqrt{2}$. Permettre la conversion instantanée de ces valeurs assure la compatibilité pour toute imprimante, et ouvre la porte au changement de logiciel de bord.
D'autre part, il m'a aussi suggéré des améliorations pour l'application ; cela inclut l'ajout d'infobulles qui apparaissent au passage de la souris sur les champs. Ils servent à détailler l'action réelle de certaines constantes sur l'impression. Leur implémentation a été très simple et m'a permis d'en apprendre plus sur beaucoup de caractéristiques des imprimantes 3D.

Enfin, il m'a proposé plusieurs activités pratiques pour l'aider dans ses missions. Je l'ai par

exemple accompagné pour découper des patrons au laser, en préparant les feuilles canson. Cela nous a servi à recouvrir le renard (c.f 1.9c) de son propre patron, qu'on a ensuite pris en photo avec l'installation photo (c.f 1.11).

Chapitre 3

Résultats et Validation

A l'heure de l'écriture de ce rapport, l'application de création de profiles est opérationnelle pour les fonctionnalités les plus générales des machines. Elle est disponible sur un Github. Elle peut générer le fichier de fonctionnalités (*features.lua)* dans lequel sont stockées les variables définissant les caractéristiques de la machine. Elle peut ensuite générer ceux pour les profils de qualité (c.f 3.1 avec *low.lua* pour la qualité basse) et de matériaux (c.f 3.1 pour le matériel *PLA*), bien que seuls 3 matériaux sont pour l'instant inclus avec des valeurs par défaut. Enfin, avec les valeurs entrées, elle génère en adéquation le fichier *printer.lua*. Ce dernier, contenant les fonctions appelées par le logiciel, se voit détaillé / réduit selon les boutons actionnés dans l'onglet "Features".

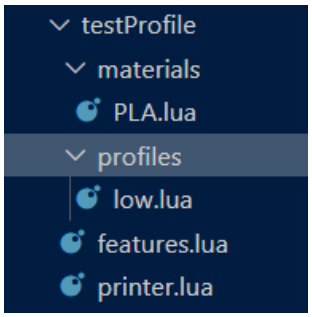


FIGURE 3.1 – Vue dans l'explorateur de fichier d'un profil test généré par l'application, pour une machine qui s'appellerait "testProfile".

Les tests sur machine seront effectués la semaine du 10 Juin, afin de valider le format des profils et de voir s'il peut effectivement correspondre à la plupart des machines. Cela se fera en tranchant, avec le slicer, un modèle 3D très simple, souvent ce qu'on appelle un Benchy Boat [1] (c.f 3.2). Il s'agira de vérifier si non seulement la machine arrive à imprimer mais aussi si elle le fait de manière optimale vis-à-vis de ces capacités individuelles.

Un *Benchy Boat* permet notamment d'évaluer la qualité des parties suspendues : celles-ci peuvent d'habitude nécessiter des supports d'impression si l'angle de la pente est trop élevé, mais ici le but est de lancer l'impression sans support afin de voir à quel point l'imprimante réussit à générer des angles aigus sans aide.

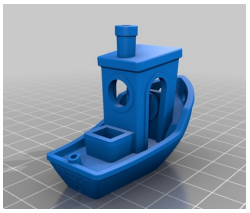


FIGURE 3.2 – Benchy Boat.

1. *Benchy* dérive de l'anglais *Benchmark* pour "évaluation" ou "repérage"

Ressenti personnel et conclusion

Ces 6 semaines de stage de recherche auprès de l'équipe MFX ont été très enrichissantes pour moi. Elles m'ont non seulement fait découvrir, en détail, le monde de la recherche scientifique et sa relation avec l'ingénieur, à travers des projets innovants et fascinants, mais m'ont aussi offert un temps et un exercice inestimables pour progresser personnellement en programmation.

L'essentiel de ce que j'ai retenu est la place de l'ingénieur dans une équipe-projet telle MFX : Il aide à implémenter, réaliser les travaux de recherche souvent plus théoriques, des chercheurs. Par ailleurs, il n'y a pas de réelle hiérarchie entre les deux métiers ; ils se complètent dans la réalisation pratique d'innovations.
C'est ainsi, en m'assimilant à un ingénieur informatique / logiciel, que j'ai pu aider l'équipe en créant mon outil pour *IceSL*, devanture par excellence de MFX. Ma mission s'inscrit typiquement dans le cadre du rôle d'un ingénieur, en alliant les nécessités matérielles et l'aspect numérique, faisant le pont entre la théorie et la pratique.
Il est important de mentionner que cette expérience a été tout particulièrement plaisante humainement. Il fait bon vivre dans cette équipe où règne une ambiance conviviale, où les discussions à table sont toujours intéressantes et originales (on y mange d'ailleurs très bien... Comment revenir au restaurant universitaire après ça ?), où l'entraide va de soi. Et de surcroît, il semble que c'est une constante dans plusieurs des équipes de l'INRIA.

Pour finir, ce séjour a fortifié mon envie de poursuivre l'informatique en école d'ingénieur, mais surtout me conforte dans mon projet de continuer dans la recherche après mon diplôme. En effet, le domaine de l'informatique est en constante évolution, mais nécessite un grand renouveau dans ses considérations éthiques et utilitaires. On peut légitimement se poser la question sur notre propre place dans la civilisation quand des Intelligences Artificielles viennent à nous remplacer dans ce que l'on fait de mieux : L'art. Ceci se vaut plus que tout dans les entreprises privées où sont relevées toutes nos données personnelles, et où est décidé par un petit nombre, sans grande conviction humaniste, l'avenir numérique de tous. La recherche publique me semble après cette expérience y faire exception et sert le bien commun grâce à des équipes de passionnés voulant innover pour le monde de demain. Cela consolide mon désir d'y poursuivre mon projet professionnel.

Annexes

Annexe 1

Le projet d'informatique se constituait de l'algorithme de couture de cycles utilisé dans [1]. Il avait pour but de réunir en un unique trait plusieurs ensembles distincts de points—les cycles—en unique cycle. Voici un diagramme représentatif de ce qu'il effectuait. Il a été reproduit et amélioré par Antoine Marion lors de son stage, et est l'une des fonctionnalités implémentées dans *IceSL*, du moins incluse dans l'*Optizor* mentionné en 1.3.

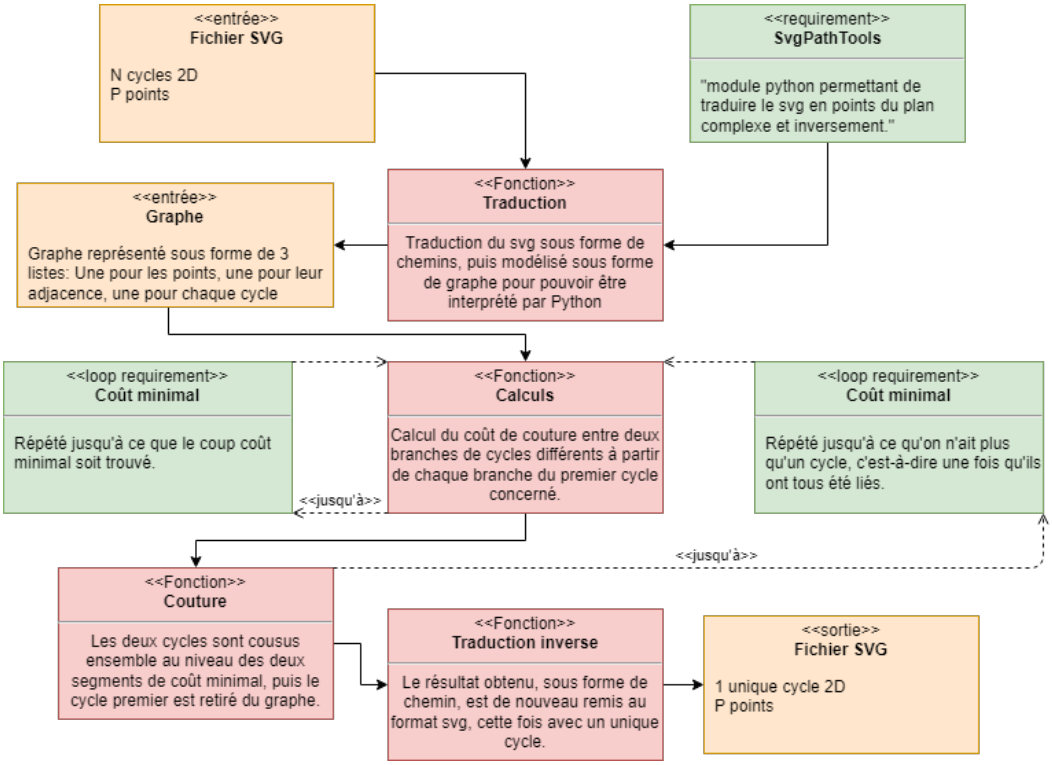
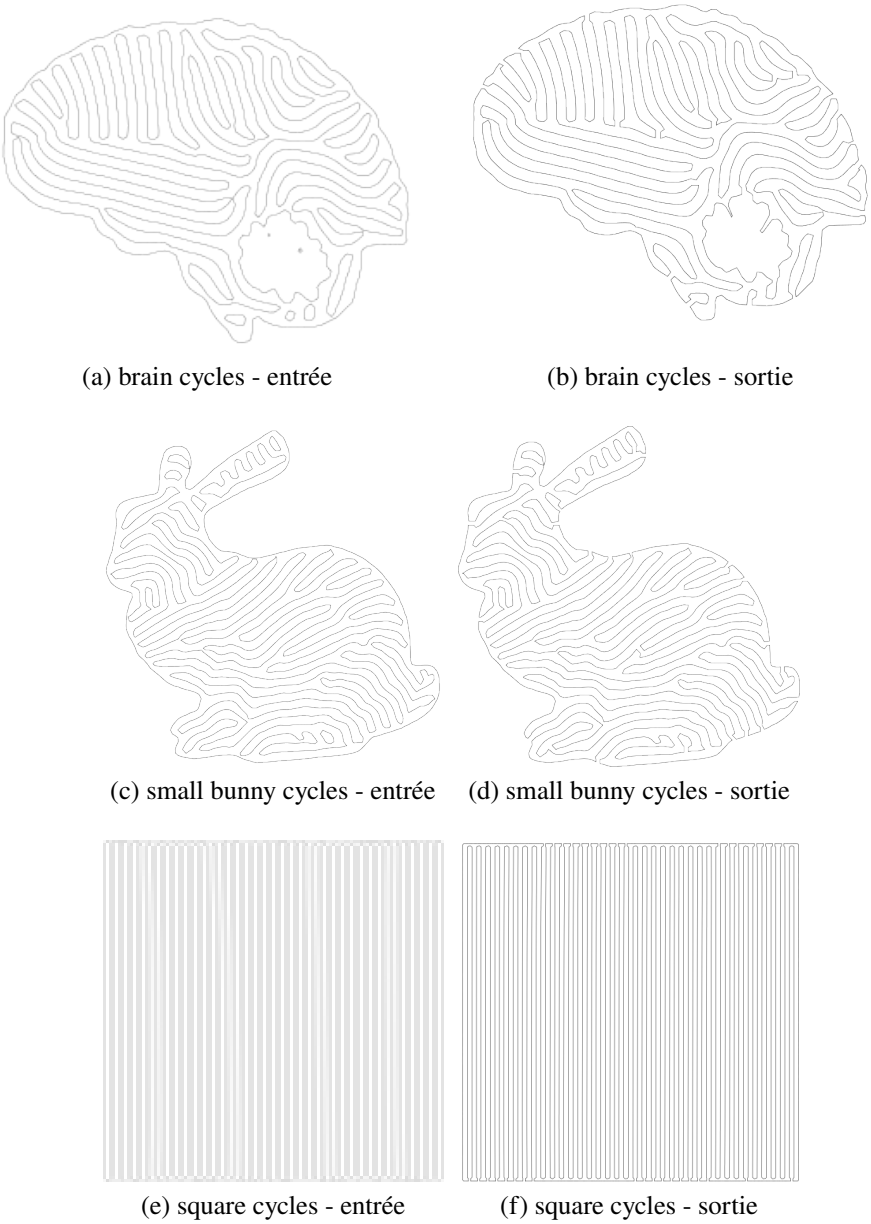


FIGURE 3.3 – Diagramme des entrées et sorties

Avec ce projet, on obtient les résultats suivants :



(a) brain cycles - entrée (b) brain cycles - sortie

(c) small bunny cycles - entrée (d) small bunny cycles - sortie

(e) square cycles - entrée (f) square cycles - sortie

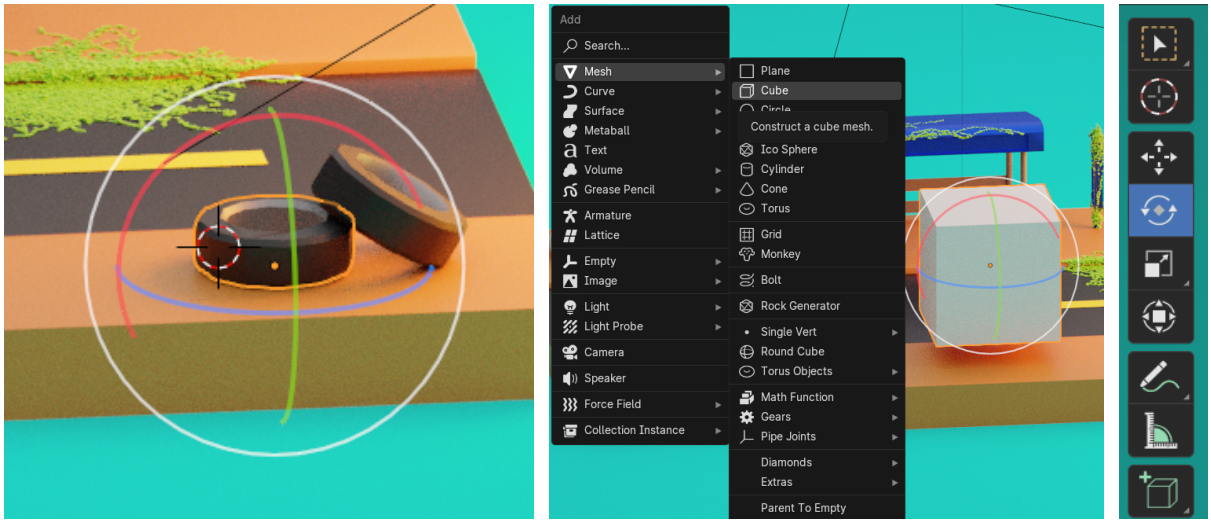
FIGURE 3.4 – Exemples de différents résultats

Annexe 2

Pour ma part, j'utilise énormément *Blender*. Il s'agit d'un logiciel de modélisation 3D, très utilisé dans l'industrie de l'animation 3D pour la facilité avec laquelle il permet de créer, modifier, sculpter, colorier un modèle. Il est félicité pour sa prise en main facile et son accessibilité : le logiciel est *open source*, signifiant qu'il est non seulement gratuit mais aussi, tout comme IceSL, modifiable à souhait car le code source est ouvert à tous.

FIGURE 3.5 – Logo Blender.

Blender est un logiciel type dans les outils de modélisation qu'il propose (outre quelques fonctionnalités uniques et innovantes), ainsi nous pourrons montrer les outils qui diffèrent grandement d'IceSL-Forge et sa modélisation par *scripting*. Pour ce faire, on peut observer l'illustration en 3.6 :



(a) L'outil "rotation" est utilisé sur la roue de voiture en surbrillance orange. Les arcs de cercle colorés correspondent chacun à un axe cartésien. En déplaçant un arc avec la souris, on fait tourner l'objet sur l'axe.

(b) Menus d'ajout de forme. Avec *IceSL*, on utiliserait la fonction *emit()* avec pour argument la forme à générer. Sa position et sa rotation seraient gérées par les fonctions *translate(x, y, z)* et *rotate(x, y, x)*

(c) Menus d'outils Blender.

FIGURE 3.6 – Vue rapide des outils Blender. À la différence d'IceSL, tout peut se faire à la souris avec des outils interactifs. Mais, cela implique aussi une faiblesse dans la précision et l'importance des décisions de l'utilisateur. De même, les opérations booléennes de Blender tendent à amener des problèmes dans le maillage, comme des trous, des points superposés, ou une mauvaise différenciation de l'intérieur / extérieur d'un objet.

Les autres logiciels de modélisation possèdent typiquement des outils similaires à Blender. La différence se fait souvent dans leur utilisation la plus courante, et dans leur prix.

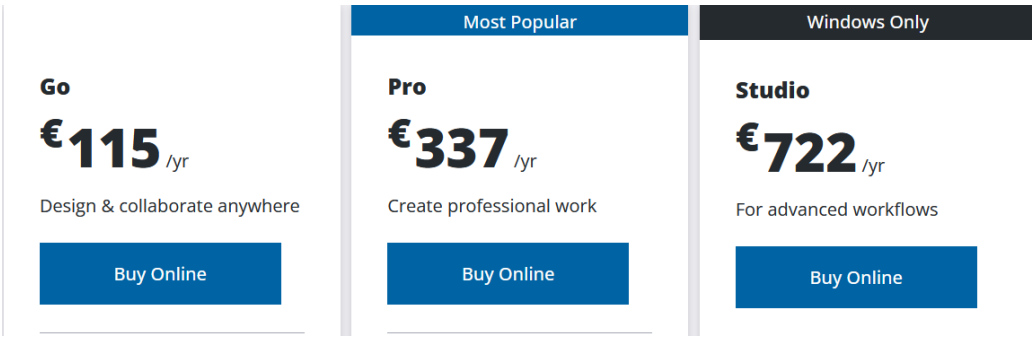


FIGURE 3.7 – Les coûts **annuels** du logiciel Sketchup, primé en architecture et design d'intérieur.

Bibliographie

[1] Xavier Chermain, Cédric Zanni, Jonàs Martínez, Pierre-Alexandre Hugron, and Sylvain Lefebvre. "orientable dense cyclic infill for anisotropic appearance fabrication". *ACM Trans. Graph. (Proc. SIGGRAPH)*, 42(4), 2023.

[2] Semyon Efremov, Jonàs Martínez, and Sylvain Lefebvre. 3d periodic cellular materials with tailored symmetry and implicit grading. *Computer-Aided Design*, 140 :103086, 2021.

[3] Marco Freire, Manas Bhargava, Camille Schreck, Pierre-Alexandre Hugron, B. Bickel, and Sylvain Lefebvre. Pcbend : Light up your 3d shapes with foldable circuit boards. *ACM Transactions on Graphics (TOG)*, 42 :1 – 16, 2023.

[4] David Jourdan, Pierre-Alexandre Hugron, Camille Schreck, Jonas Martinez, and Sylvain Lefebvre. Shrink & Morph : 3D-printed self-shaping shells actuated by a shape memory effect. *ACM Transactions on Graphics*, 42(6), December 2023.