

# Projet « Optimiseur de coupe »

Xavier JANIN – 12316706

Licence 2 - année 2015

IED Paris 8

## 1 Sommaire

### Table des matières

1 Sommaire.....	1
2 Un peu d'histoire.....	1
3 La définition du cahier des charges.....	2
3.1 Mes apports.....	2
3.2 Les difficultés rencontrées.....	2
3.3 Les pistes d'amélioration.....	2
3.4 Les écarts entre le programme et le cahier des charges.....	4
3.5 Conclusion.....	4
4 Le mode d'emploi.....	4
5 Le packaging.....	5
5.1 Les solutions envisagées.....	5
5.1.1Exécutable d'installation sous Windows.....	5
5.1.2Script d'installation.....	5
5.1.3Paquet d'installation deb.....	5
5.2 Difficultés rencontrées.....	5
5.3 Le paquet d'installation.....	6
5.4 Le script add_Qt_ppa.....	6
6 Conclusion.....	6

## 2 Un peu d'histoire...

Je suis arrivé dans ce projet pour compléter l'équipe déjà constituée par Rebecca et Pierre-Emmanuel. Le partage des tâches étant déjà fait, il m'incombait de trouver une place dans ce projet.

Il a donc été décidé que j'aurais à m'occuper de réfléchir à la gestion des erreurs dues aux saisies, de rédiger le cahier des charges, de rédiger un mode d'emploi et de prendre en charge le packaging final.

## 3 La définition du cahier des charges

### 3.1 Mes apports

Le vrai défi pour un informaticien consiste à réussir à s'approprier le point de vue de l'utilisateur. C'est là dessus je pense que mon apport a été le plus significatif. J'ai ainsi suggéré quelques ajouts de fonctionnalités : perte à la coupe (due à l'épaisseur de la lame servant à la découpe), choix des unités de mesure, et possibilité d'avoir plusieurs longueurs de matière première. Mes expériences professionnelles sur des machines outil industrielles m'ont en cela grandement aidé, j'ai vraiment pu me mettre dans la peau du client pour connaître ses besoins et ses contraintes.

### 3.2 Les difficultés rencontrées

Les choses sont devenues plus compliquées lorsqu'il a fallu définir les valeurs limites des saisies. En effet, outre les erreurs évidentes (valeurs négatives, égales à zéro, interactions entre les valeurs saisies, ..., cf cahier des charges qui détaille cela précisément), il a fallu choisir des valeurs min et max plausibles... Plausibles, oui, mais pour qui ? Et pourquoi ? La réponse à cette dernière question est simple, des valeurs abracadabrantes auraient fait planter le programme, ou auraient dans le meilleur des cas demandé un temps de calcul prohibitif. Donc il a fallu décider de ce qui était plausible. Les choix ont été détaillés dans le cahier des charges, je ne reviendrai pas dessus.

### 3.3 Les pistes d'amélioration

Il a été envisagé d'utiliser un fichier de configuration (xml), paramétrable grâce à un onglet supplémentaire, cela permettrait d'adapter le programme à des utilisations plus exotiques. Cette solution a été abandonnée pour cette version de programme, mais est à envisager en tant que piste d'amélioration.

La raison de l'abandon de cette solution est qu'il aurait fallu définir un nouvel algorithme basé sur des heuristiques pour réduire les temps de calcul lors de saisies de grandes quantités (autorisées par modification du fichier de configuration). Or nous ne sommes pas prêts pour ça, et le temps nous aurait manqué.

Il a aussi été envisagé d'exporter les résultats, soit au format html, soit sous un format propriétaire que le programme serait capable de décoder pour restitution. Cette solution a été abandonnée pour cause de timing serré, mais est à envisager comme piste d'amélioration.

On a donc en résumé :

- Ajout d'un onglet de configuration
- Ajout d'un algorithme de calcul basé sur des heuristiques
- Ajout d'une possibilité d'exportation des résultats

### 3.4 Les écarts entre le programme et le cahier des charges

Il n'avait pas été envisagé que les restitutions graphiques et textuelles conjointes soient si peu lisibles, il a donc fallu les séparer sous deux onglets distincts.

De même l'onglet « A propos » a été ajouté après coup.

Enfin, le mode de packaging n'avait pas été envisagé dans le cahier des charges.

### 3.5 Conclusion

Le diable est dans les détails, disait Nietzsche... Je retiendrai que c'est particulièrement vrai lorsqu'il s'agit de définir un cahier des charges ! En effet, en discutant avec Pierre-Emmanuel et Rebecca concernant la gestion des saisies, je me suis rendu compte que ce que je prenais pour des certitudes ou des évidences ne l'étaient pas forcément, et qu'il a fallu pousser loin la réflexion pour finalement aboutir à des solutions simples et efficaces, mais restreignant les possibilités du programme. Il a fallu changer maintes fois de point de vue, réparer les oublis, ... En fait, un cahier des charges reste approximatif, il semble que l'on soit condamné à rencontrer des imprévus.

## 4 Le mode d'emploi

Pas grand-chose à dire là dessus, il s'agit de deux fichiers html (un en français, l'autre en anglais) accessibles pas le bouton « aide » (ou « help ») du programme.

Ces pages consistent majoritairement en copies d'écrans décrivant l'usage pouvant être fait du programme et des erreurs pouvant être rencontrées.

Le programme n'ayant pas d'argument en ligne de commande, aucune page man n'a été créée.

## 5 Le packaging

### 5.1 Les solutions envisagées

#### 5.1.1 Exécutable d'installation sous Windows

Cela semble assez simple à faire, mais cette solution n'a pas été retenue car à l'heure actuelle le programme ne servira que dans le cadre d'une démonstration sur plate-forme logicielle libre type Linux.

#### 5.1.2 Script d'installation

Cela a été envisagé aussi, mais il aurait été compliqué de gérer les dépendances.

#### 5.1.3 Paquet d'installation deb

C'est la solution retenue, à la base pour la bonne gestion des dépendances. Il s'avère que cette solution est plus simple à l'installation, plus ergonomique à l'usage (le programme est visible dans la liste des programmes installés, l'icône du programme apparaît lorsqu'on lance le programme). Par contre elle est moins simple lorsqu'il s'agit de la mettre en pratique.

## 5.2 Difficultés rencontrées

Cela se résume à « How to ? ». En effet les informations ne manquent pas sur ce sujet, malheureusement elles ne sont pas toutes fiables, et lorsqu'elles le sont elles apportent plus de confusion que de réponses aux questions, alors que cela pourrait être simple. Le lien suivant m'a servi à définir l'arborescence utilisée lors de l'installation du programme :

[http://docs.abuledu.org/abuledu/mainteneur/contenu\\_minimal\\_deb](http://docs.abuledu.org/abuledu/mainteneur/contenu_minimal_deb)

L'auteur va droit au but...

Pour le reste, (les fichiers « control », « COPYRIGHT », « cutting\_optimizer.desktop », ...), il m'a fallu séparer le bon grain de l'ivraie, et cela n'a pas été trivial.

Le plus compliqué a été de créer la liste des dépendances, la fonction ldd retournant une liste de librairies, et non pas une liste de paquets à installer... Le programme dpkg-shlibdeps a permis de contourner cet écueil de taille.

## 5.3 Le paquet d'installation

Fichiers installés :

- /usr/bin/cutting-optimizer : le binaire du programme
- /usr/share/pixmaps/cutting\_optimizer.xpm : l'icône utilisée par le programme
- /usr/share/applications/cutting\_optimizer.desktop : entrée de menu de l'application
- /usr/share/cutting\_optimizer/help/ : dossier où se trouvent les fichiers du mode d'emploi
- /usr/share/cutting\_optimizer/src/ : fichiers source du programme



## 5.4 Le script add Qt ppa

Le repository propre a Qt5 ne fait pas partie des repositories présents par défaut dans les distributions Ubuntu. Ce script a donc pour fonction d'installer le programme apt-add-repository, puis d'ajouter le repository de Qt5.

## 6 Conclusion

Ce projet a été captivant sur cet aspect : « Yes ! We can ! ».

Et oui, on l'a fait.

Mais il met aussi en avant le chemin qu'il nous reste à accomplir : la mise en place d'un nouvel algorithme basé sur des heuristiques n'est pas encore à notre portée.

Mais surtout ce que je retiendrai, c'est d'avoir eu la chance de pouvoir travailler sur des aspects un peu « satellites » de la programmation, tels que l'élaboration d'un cahier des charges et le packaging.

En effet, lorsqu'il a des besoins concernant la création de scripts, mon donneur d'ordres actuel me donne de vagues consignes, à charge pour moi de définir dans le détail quelles seront les missions que rempliront les scripts, puis de les écrire. Je suis donc à la fois « juge et partie ». Jusqu'à présent, il ne m'avait pas encore été donné de pouvoir à ce point rentrer dans la peau du client exprimant ses besoins, puis de formaliser ces besoins.

Surtout, nous avons pu mener à bien ce projet en mode collaboratif, toutes les décisions ont été prises en commun, l'accent ayant été mis sur la communication grâce notamment à l'outil formidable que constitue github.

Le bilan que je fais de ce projet est donc extrêmement positif.