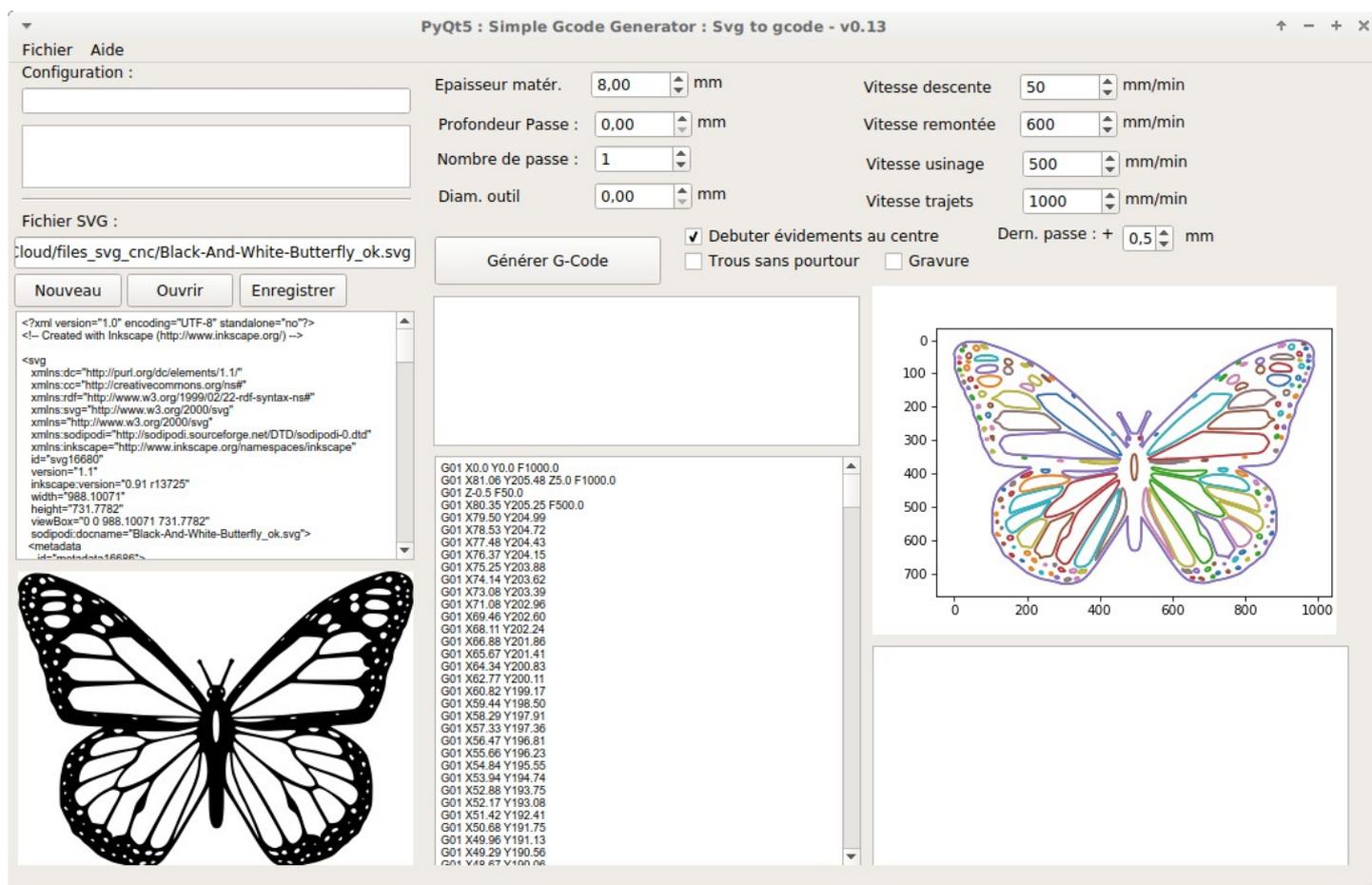


PyQt5.x / Python3.x : Installation de Simple G-Code Generator

Par X. HINAULT – www.mon-club-elec.fr | www.mon-fablab.fr – Juin 2017 - Tous droits réservés - Licence Creative Commons BY NC SA



De quoi parle-t-on ?

Simple G-Code Generator est une interface graphique Python (PyQt) libre que nous avons développée (X. HINAULT) et qui permet de générer simplement du G-Code à partir d'un simple dessin SVG segmentisé.

En clair, Simple-Gcode Generator permet de générer du G-Code utilisable avec l'[Open Maker Machine](#) ou l'[Open Maker Machine PRO](#) ou l'[Open Maker Machine PLUS](#) à partir d'un simple dessin réalisé dans Inkscape.

Potentiellement le G-Code produit avec Simple G-Code Generator est utilisable avec toute machine de découpe 2D propulsée par [GRBL](#), le micrologiciel décodeur de G-Code opensource programmé dans l'Arduino.

Pour info, le Simple G-Code Generator fait partie d'une petite suite logicielle comprenant :

- **Simple G-Code Generator** : un générateur de G-Code à partir de SVG segmentisé.
- **Simple G-Code GUI** : une interface de contrôle d'une machine de fabrication numérique 2D, notamment utilisable avec nos machines l'[Open Maker Machine](#) ou l'[Open Maker Machine PRO](#) ou l'[Open Maker Machine PLUS](#)
- **Simple G-Code Decoder** : un micrologiciel Arduino décodeur de G-Code minimaliste (10Ko) mais permettant de créer facilement de petites machines numériques. Je conseille cependant plutôt l'utilisation de GRBL, notamment pour l'[Open Maker Machine PRO](#) ou l'[Open Maker Machine PLUS](#)

Ces logiciels constituent un ensemble cohérent opérationnel et opensource permettant de contrôler une machine de fabrication numérique 2D : ils sont cependant utilisables indépendamment les uns des autres, entre-eux ou non.

Note aux utilisateurs :

Il existe de nombreux générateurs de G-Code 2D, souvent très complets mais un peu déroutants pour des besoins simples... Plus rares sont les générateurs de G-Code 2D libres et open-source. Notre but ici est clairement de proposer un outil pragmatique, simple et rapide à utiliser par des « néophytes » à l'aide des outils habituels des « makers », tout en étant libre et open-source.

Nous utilisons Simple G-Code Generator tous les jours en production et nous l'enrichissons progressivement lorsque des besoins se font sentir.

Pré-requis PyQt5.x / Python 3.x

On présume que [l'installation des pré-requis présentés](#) par ailleurs a été faite.

Note : Ce logiciel étant développé avec PyQt5.x, il est utilisable potentiellement sur Windows et Mac mais je n'ai pas à ce jour écrit les versions pour ces plateformes.

Dépendances utiles

En complément des pré-requis, il est nécessaire d'installer les dépendances suivantes :

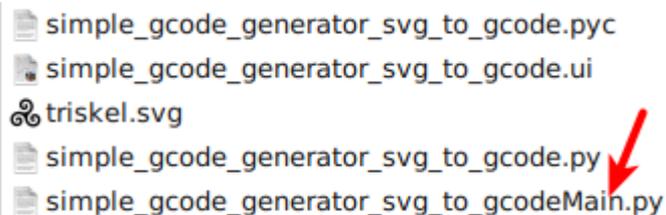
```
sudo apt-get update
sudo apt-get install python3-shapely
sudo apt-get install python3-matplotlib
sudo apt-get install python3-pyqt5.qtsvg
```

Téléchargement

Le Simple G-Code Generator est téléchargeable ici :

http://cloud-mon-club-elec.fr:8080/pyqt5_apps/pyqt5_simple_gcode_generator/

Choisir de préférence la version la plus récente. Dézipper l'archive obtenue. Le répertoire contient :



- simple_gcode_generator_svg_to_gcode.pyc
- simple_gcode_generator_svg_to_gcode.ui
- triskel.svg
- simple_gcode_generator_svg_to_gcode.py
- simple_gcode_generator_svg_to_gcodeMain.py

Lancement

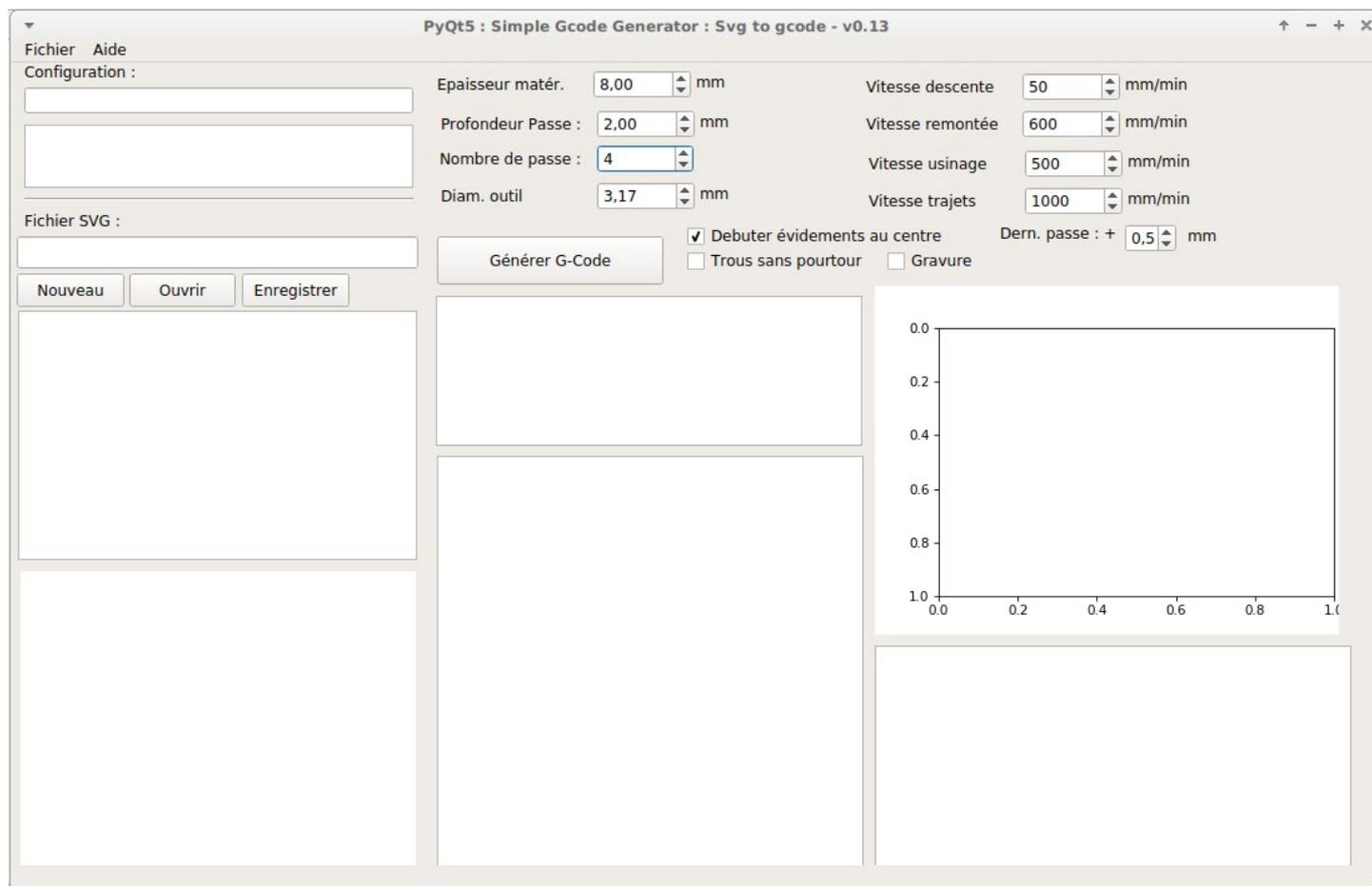
Dans le répertoire obtenu, le fichier à exécuter est le *.main :

- soit en le rendant exécutable (clic droit > propriétés > Permissions > cocher rendre exécutable)
- soit en ouvrant un terminal dans le répertoire où il se trouve et en saisissant

```
./simple_gcode_generator_svg_to_gcodeMain.py
```

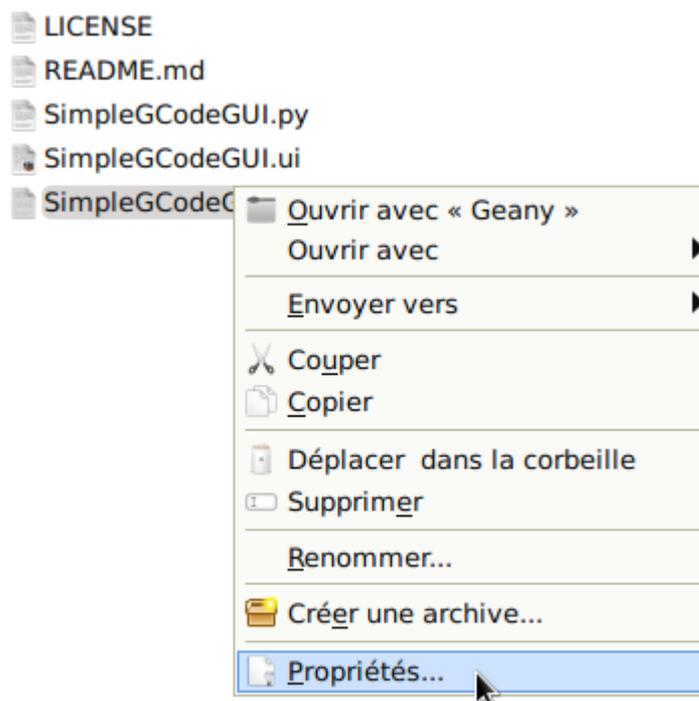
- soit en l'ouvrant avec l'éditeur Geany et en l'exécutant.

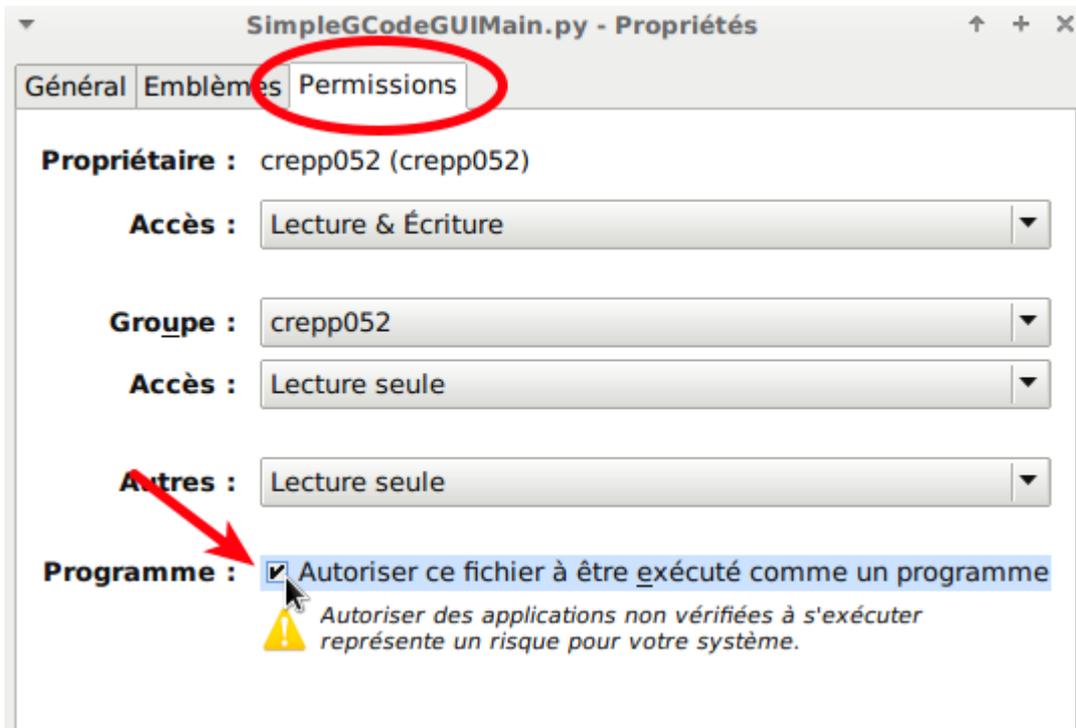
Au lancement, si aucune dépendance n'est manquante, on obtient :



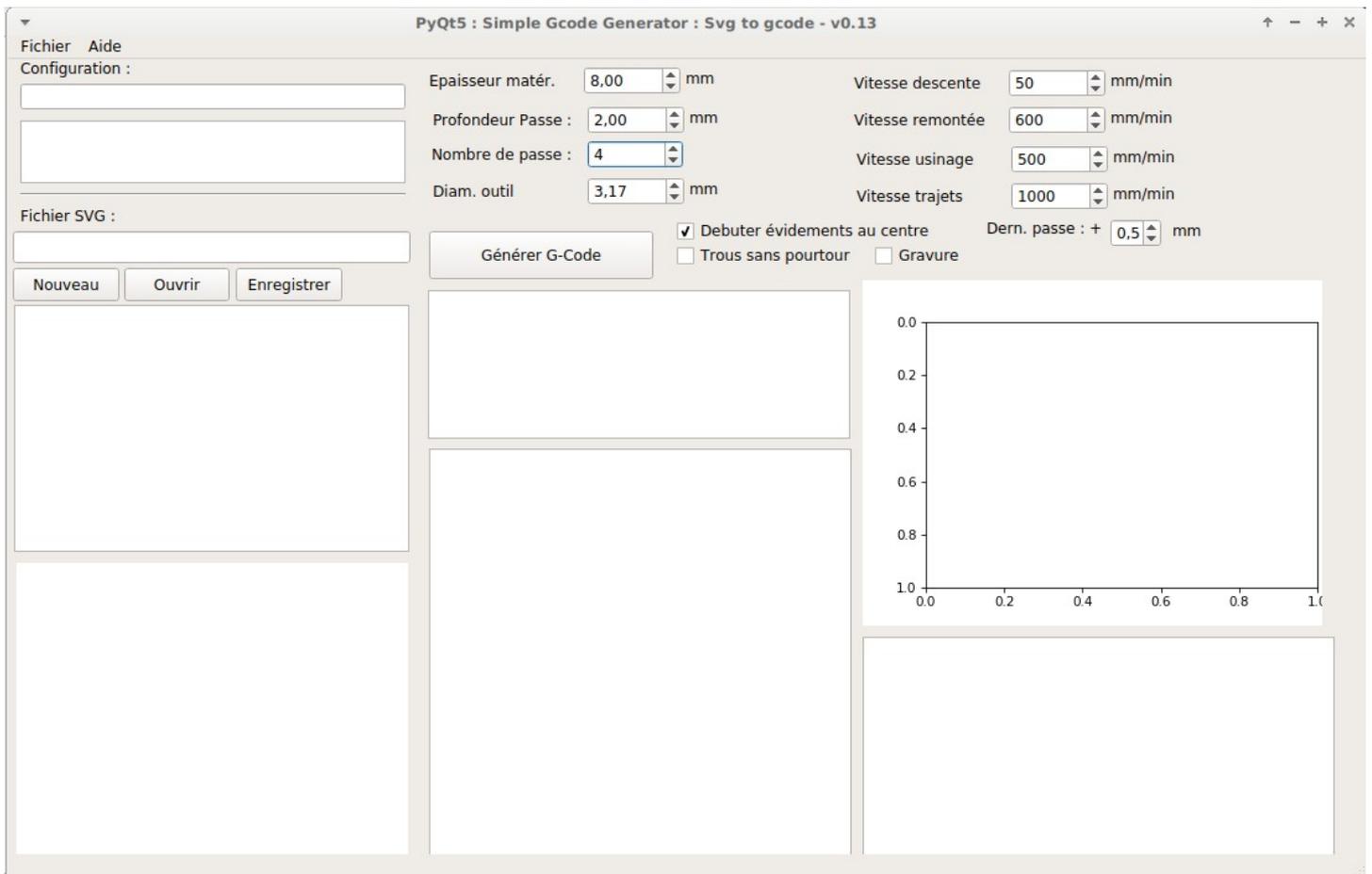
Rendre exécutable

Rendre exécutable le fichier *Main.py par clic droit > propriétés > onglet permissions >



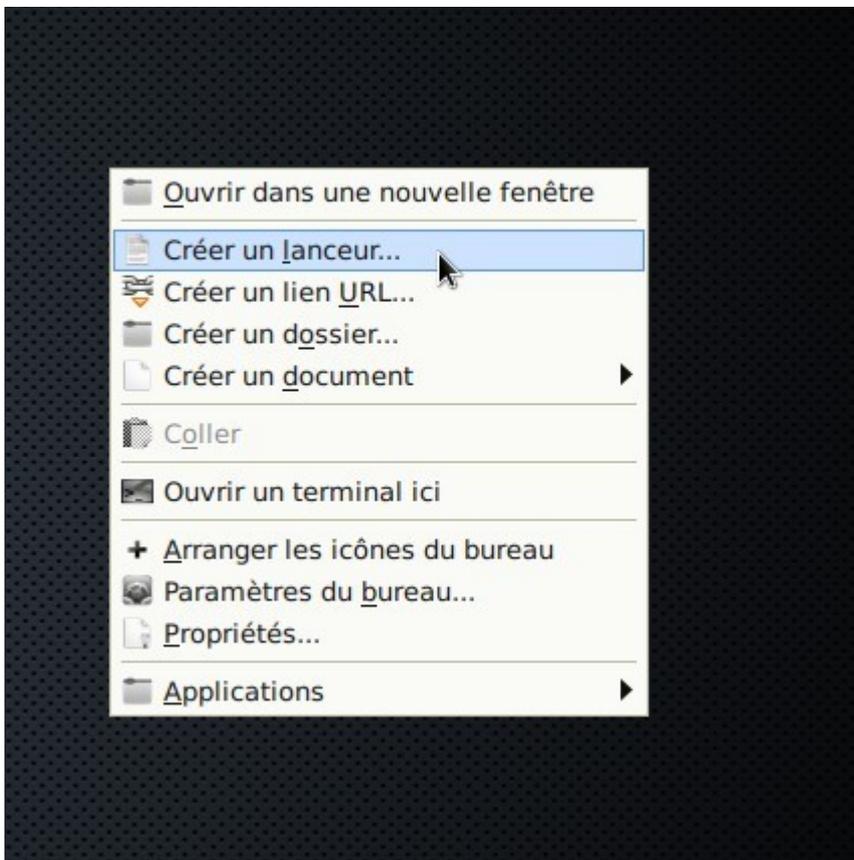


Une fois fait, un double clic sur le fichier *Main.py lance l'application.

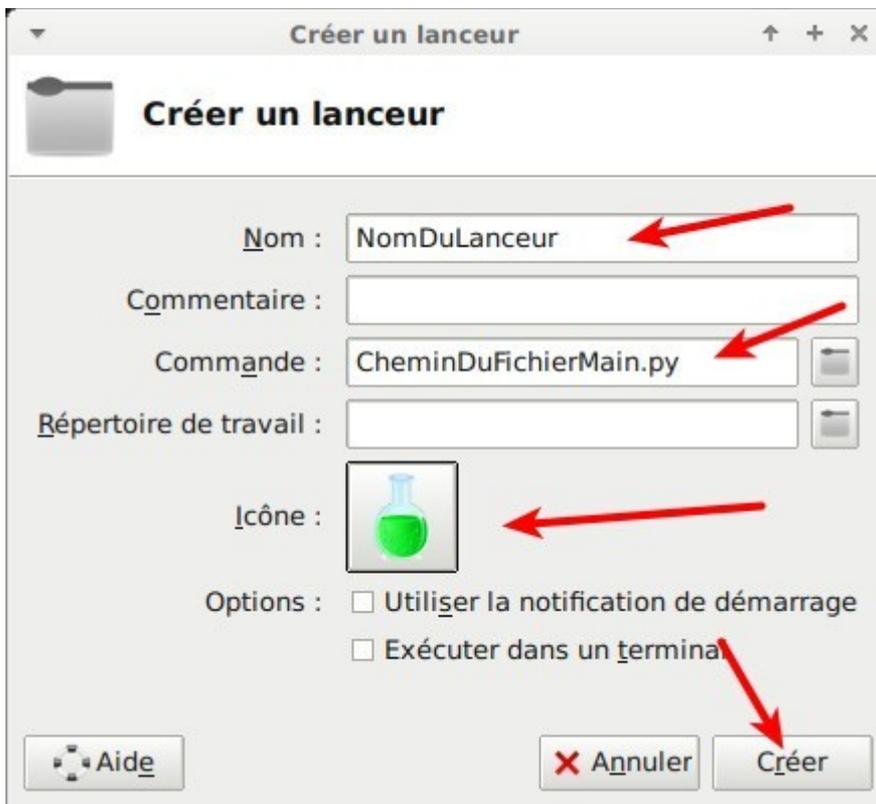


Créer un lanceur

Si l'on souhaite créer un lanceur classique sur son bureau pour cette application, créer un lanceur sur le bureau du système, ce qui se fait par exemple sous XFCE par clic-droit > Créer un lanceur.



Puis ensuite, dans la fenêtre qui s'ouvre, donner le nom voulu et définir le chemin de l'application, plus précisément du fichier *Main.py précédemment téléchargé :



Une fois validé, on obtient le lanceur sur le bureau :



Lancer en ligne de commande

Il est également possible de le lancer en ligne de commande avec (ceci présente l'avantage de montrer les messages dans le terminal système pendant l'exécution) :

```
/chemin/ou/se/trouve/le/fichier/SimpleGCodeGUIMain.py
```

Présentation de l'interface

Simple G-Code Generator se veut simple et logique... On a donc :

- une zone de **chargement du fichier de configuration à utiliser et un menu** (NOUVEAU)
- une zone de **chargement du fichier SVG** source qui comprend :
 - un champ + boutons d'ouverture d'un fichier SVG
 - une zone texte éditable où sera affiché le source du SVG ouvert
 - une fenêtre de visualisation du SVG ouvert. Le Triskell affiché par défaut l'est dans un simple but de test.
- D'une série de **champs de paramétrage** comprenant :
 - des paramètres de base limités à l'essentiel utile en pratique
 - des paramètres « avancés » correspondant aux vitesses de progression et à la sélection du mode de génération du G-Code. Ces paramètres peuvent être laissés en l'état par défaut.
- D'une zone de **gestion du G-Code** généré comportant :
 - un bouton à cliquer pour lancer la génération du G-Code
 - d'une zone texte d'affichage du G-Code obtenu
 - d'une visualisation de la trajectoire du G-Code généré superposée aux chemins extraits du SVG source.

PyQt5 : Simple Gcode Generator : Svg to gcode - v0.13

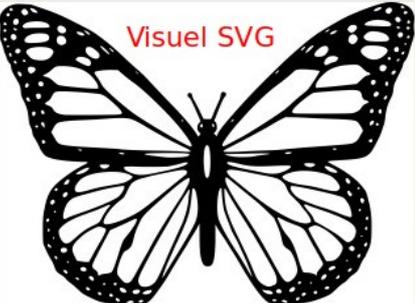
Fichier Aide

Configuration :
 /home/xavier072/config_dessin.txt
 epaisseur=0.0
 profondeur=0.0
 nombre_passes=1

Paramètres G-Code
 Epaisseur matér. : 0,00 mm
 Profondeur Passe : 0,00 mm
 Nombre de passe : 1
 Diam. outil : 0,00 mm
 Vitesse descente : 200 mm/min
 Vitesse remontée : 600 mm/min
 Vitesse usinage : 1000 mm/min
 Vitesse trajets : 1000 mm/min

Fichier SVG :
 cloud/files_svg_cnc/Black-And-White-Butterfly_ok.svg
 Nouveau Ouvrir Enregistrer

Source du SVG
 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
 <!-- Created with Inkscape (http://www.inkscape.org/) -->
 <svg
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:cc="http://creativecommons.org/ns#"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:svg="http://www.w3.org/2000/svg"
 xmlns="http://www.w3.org/2000/svg"
 xmlns:sodipodi="http://sodipodi.sourceforge.net/DTD/sodipodi-0.dtd"
 xmlns:inkscape="http://www.inkscape.org/namespaces/inkscape"
 id="svg16680"
 version="1.1"
 inkscape:version="0.91 r13725"
 width="988,10071"
 height="731,7782"
 viewBox="0 0 988,10071 731,7782"
 sodipodi:docname="Black-And-White-Butterfly_ok.svg">
 <metadata
 id="metadatas16680">

Visuel SVG


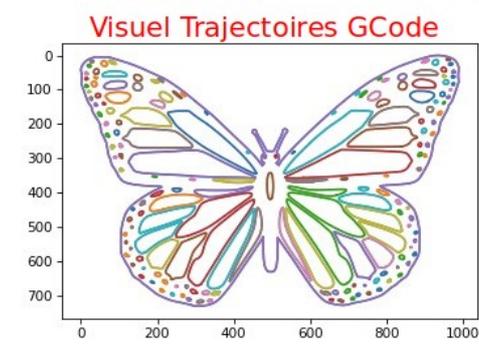
Générer G-Code
 Debuter évidements au centre
 Trous sans pourtour
 Gravure
 Dern. passe : + 0,0 mm

Options

Génération GCode


GCODE généré (éditable)

```
G01 X0.0 Y0.0 F1000.0
G01 X81.06 Y205.48 Z5.0 F1000.0
G01 Z-0.0 F200.0
G01 X80.35 Y205.25 F1000.0
G01 X79.50 Y204.99
G01 X78.53 Y204.72
G01 X77.48 Y204.43
G01 X76.37 Y204.15
G01 X75.25 Y203.88
G01 X74.14 Y203.62
G01 X73.08 Y203.39
G01 X71.08 Y202.96
G01 X69.46 Y202.60
G01 X68.11 Y202.24
G01 X66.88 Y201.86
G01 X65.67 Y201.41
G01 X64.34 Y200.83
G01 X62.77 Y200.11
G01 X60.82 Y199.17
G01 X59.44 Y198.50
G01 X58.29 Y197.91
G01 X57.33 Y197.36
G01 X56.47 Y196.81
G01 X55.66 Y196.23
G01 X54.84 Y195.55
G01 X53.94 Y194.74
G01 X52.88 Y193.75
G01 X52.17 Y193.08
G01 X51.42 Y192.41
G01 X50.66 Y191.75
G01 X49.96 Y191.13
G01 X49.29 Y190.56
G01 X48.67 Y190.08
```

Visuel Trajectoires GCode


Conclusion

Tout est OK ? Alors vous allez pouvoir apprendre à l'utiliser.