



OpenPythonSCAD library for moving a tool in lines and arcs so as to model how a part would be cut using G-Code or described as a DXF.

LGPL-2.1 license

32 stars

4 forks

3 watching

Branches

Activity

Tags

Public repository

main

1 Branch

0 Tags

Go to file

Go to file

+

Add file

Code

WillAdams

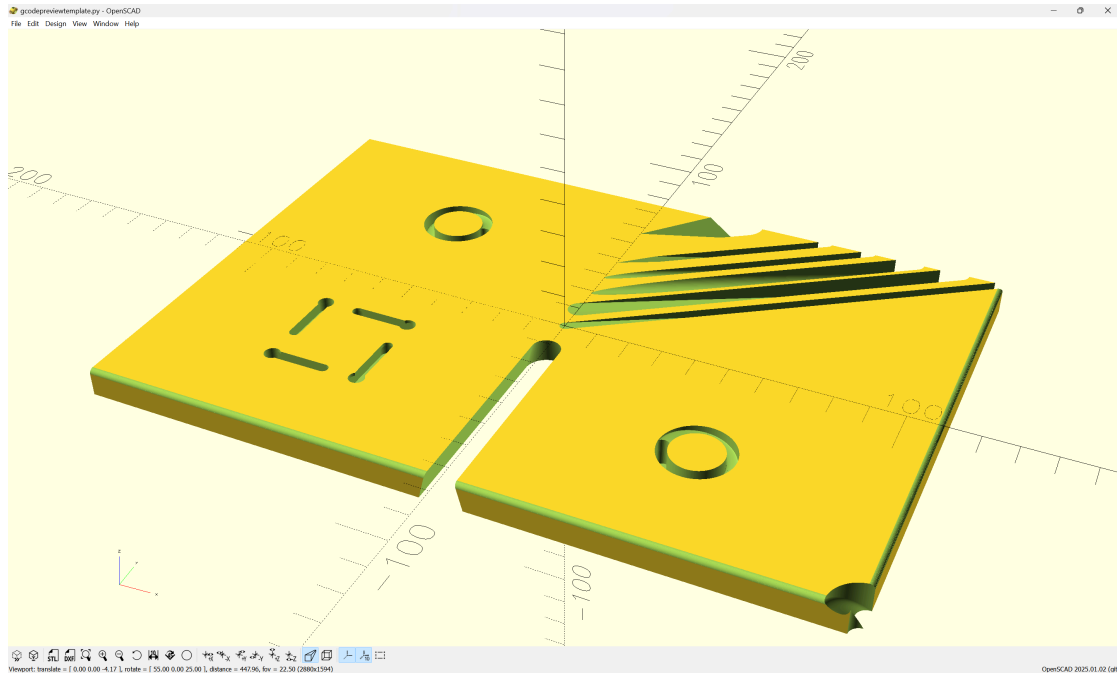
Update README.md

6c59192 · 2 minutes ago

images	update gcpdxf for nimport	last week
.gitattributes	Create .gitattributes	4 months ago
.gitignore	Update .gitignore	5 months ago
LICENSE	Initial commit	4 years ago
OSGE_cutjoinery.png	Small business Saturday	2 years ago
README.md	Update README.md	2 minutes ago
cut2Dshapes.scad	Folklore Day	5 months ago
cut2Dshapes.tres	Small business Saturday	2 years ago
dxlexport.dxf	DXF	last week
export.102.dxf	Literate corrections	9 months ago
export.dxf	Update export.dxf	2 months ago
export.nc	Literate corrections	9 months ago
flatten.graph.tres	Add files via upload	2 years ago
gcodepreview-openscad_0_6.pdf	Upload v0.6 under identifying names	2 months ago
gcodepreview-openscad_0_6.tex	Upload v0.6 under identifying names	2 months ago
gcodepreview.pdf	shapes	2 months ago
gcodepreview.py	shapes	2 months ago
gcodepreview.scad	Black Friday	2 months ago
gcodepreview.tex	shapes	2 months ago
gcodepreview_PYunittests.scad	National Virginia Day after	4 months ago
gcodepreview_template.png	Create gcodepreview_template.png	9 months ago
gcodepreview_unittests.png	Update gcodepreview_unittests.png	3 minutes ago
gcodepreviewtemplate.py	Small Business Saturday 2024	2 months ago
gcodepreviewtemplate.scad	Black Friday	2 months ago
gcp_template.graph.tres	Add files via upload	last year
gcpdxf.py	DXF	last week
literati.sty	Small Business Saturday 2024	2 months ago
openscad_cutjoinery.png	Small business Saturday	2 years ago
openscad_gcodepreview_cutjoinery.tres	snow day fix	2 years ago
osge_cutjoinery.png	Small business Saturday	2 years ago
pygcodepreview.scad	Black Friday	2 months ago
readme.pdf	National Virginia Day ongoing	4 months ago

## gcodepreview

PythonSCAD library for moving a tool in lines and arcs so as to model how a part would be cut using G-Code, so as to allow PythonSCAD to function as a complete CAD/CAM solution for subtractive 3-axis CNC (mills and routers) by writing out G-code in addition to 3D modeling (in some cases toolpaths which would not normally be feasible), and to write out DXF files which may be imported into a traditional CAM program to create toolpaths.



Updated to make use of Python in OpenSCAD:[\[1\]](#)

<https://pythonscad.org/> (previously this was <http://www.guenther-sohler.net/openscad/> )

A BlockSCAD file for the initial version of the main modules is available at:

<https://www.blockscad3d.com/community/projects/1244473>

The project is discussed at:

<https://willadams.gitbook.io/design-into-3d/programming>

Since it is now programmed using Literate Programming (initially a .dtx, now a .tex file) there is a PDF:

<https://github.com/WillAdams/gcodepreview/blob/main/gcodepreview.pdf> which includes all of the source code with formatted commentary.

The files for this library are:

- gcodepreview.py (gcpy) --- the Python functions and variables
- gcodepreview.scad (gcpsc) --- OpenSCAD modules and variables
- gcodepreviewtemplate.scad (gcptmpl) --- .scad example file
- gcodepreviewtemplate.py (gcptmplpy) --- .py example file (which requires PythonSCAD)
- gcpdxf.py (gcpdxfpy) --- .py example file which only makes dxf file(s) and which will run in "normal" Python

If using from PythonSCAD, place the files in C:\Users\~\Documents\OpenSCAD\libraries [\[2\]](#)

and call as:

```
use <gcodepreview.py>  
include <gcodepreview.scad>
```



Note that it is necessary to use the first file (this allows loading the Python commands (it used to be necessary to use an intermediary .scad file so as to wrap them in OpenSCAD commands) and then include the last file (which allows using OpenSCAD variables to selectively implement the Python commands via their being wrapped in OpenSCAD modules) and define variables which match the project and then use commands such as:

```
opengcodefile(Gcode_filename);  
opendxfile(DXF_filename);  
  
gcp = gcodepreview(true, true, true);
```



```

setupstock(219, 150, 8.35, "Top", "Center");

movetosafeZ();

toolchange(102,17000);

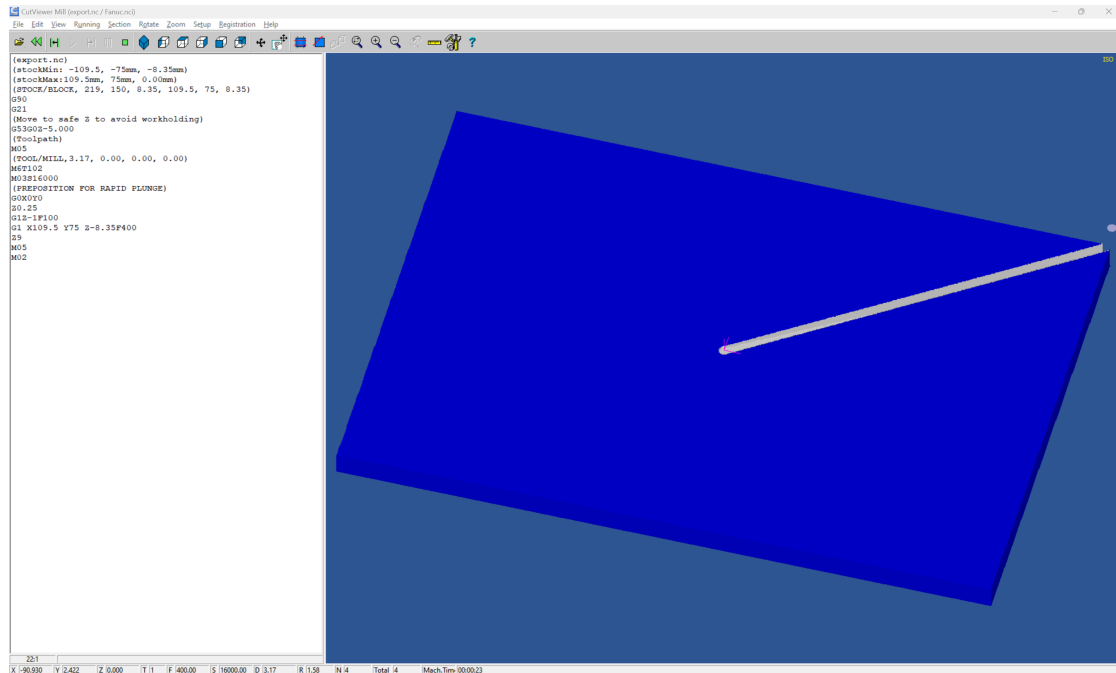
cutline(219/2,150/2,-8.35);

stockandtoolpaths();

closegcodefile();
closedxf file();

```

which makes a G-code file:



but one which could only be sent to a machine so as to cut only the softest and most yielding of materials since it makes a single full-depth pass, and of which has a matching DXF which may be imported into a CAM tool --- but which it is not directly possible to assign a toolpath in readily available CAM tools (since it varies in depth from beginning-to-end).

Importing this DXF and actually cutting it is discussed at:

<https://forum.makerforums.info/t/rewriting-gcodepreview-with-python/88617/14>

Alternately, gcodepreview.py may be placed in a Python library location and used directly from Python --- note that it is possible to use it from a "normal" Python when generating only DXFs.

Tool numbers match those of tooling sold by Carbide 3D (ob. discl., I work for them).

Comments are included in the G-code to match those expected by CutViewer, allowing a direct preview without the need to maintain a tool library.

Supporting OpenSCAD usage makes possible such examples as: openscad\_gcodepreview\_cutjoinery.tres.scad which is made from an OpenSCAD Graph Editor file:

