

# Polygones dégénérés

Au fablab @ Flux, 8-12 novembre 2025

Pad : [https://annuel.framapad.org/p/rda\\_polygone\\_2025](https://annuel.framapad.org/p/rda_polygone_2025)

## XY LDC

[XY LDC](#)

## mesh 2 svg

Meshlab : <https://www.meshlab.net/> Rien tiré de meshlab pour transformer un mesh (stl, obj) en svg

Essai concluant avec <https://www.svgai.org/convert/stl-to-svg>, le fichier s'ouvre bien avec inkscape, l'épaisseur des traits est bien trop élevée mais ça s'arrange facilement

**Conseil de Laurent : utiliser «ln» de Michael Fogleman** : <https://github.com/fogleman/ln> C'est programmé en Go, jamais utilisé

```
# *****  
sudo apt update  
sudo apt install golang  
go version  
go env GOPATH  
go install github.com/fogleman/ln/ln@latest  
mkdir -p ~/go/bin  
# installation du langage Go sur Debian 12 @ tenko  
# go version go1.19.8 linux/amd64  
# ok : /home/emoc/go
```

Aucun des tests d'installation ne fonctionne correctement ... (ci-dessous)

```
git clone https://github.com/fogleman/ln.git  
cd ln  
go mod init lnbuild  
go mod tidy  
go build -o ~/go/bin/ln ./ln  
chmod +x ~/go/bin/ln  
~/go/bin/ln --help  
# erreur
```

alors plutôt j'essaie de comprendre comment doit fonctionner go...

### Test d'un helloworld

Créer un fichier vide hello.go

```
nano hello.go
```

Y écrire

```
package main  
  
import "fmt"  
  
func main() {  
    fmt.Println("HelloWorld, Golang!")  
}
```

Puis

```
go run hello.go
```

### Transformation d'un .OBJ en .SVG

```
git clone https://github.com/fogleman/ln.git  
cd ln  
go mod init ln/ln  
go mod tidy
```

```
go run teapot.go
```

placer le fichier teapot.obj dans le dossier et créer le fichier teapot.go :

```
package main

import "github.com/fogleman/ln/ln"

func main() {
    scene := ln.Scene{}
    mesh, err := ln.LoadOBJ("teapot.obj")
    if err != nil {
        panic(err)
    }
    mesh.UnitCube()
    scene.Add(ln.NewTransformedShape(mesh, ln.Rotate(ln.Vector{0, 1, 0}, 0.5)))
    // scene.Add(mesh)
    eye := ln.Vector{-0.5, 0.5, 2}
    center := ln.Vector{}
    up := ln.Vector{0, 1, 0}
    width := 1024.0
    height := 1024.0
    paths := scene.Render(eye, center, up, width, height, 35, 0.1, 100, 0.01)
    paths.WriteToPNG("teapot.png", width, height)
    paths.WriteToSVG("teapot.svg", width, height)
}
```

## Petite parenthèse VRML

Visualiseur en ligne pour fichiers VRML : <https://imagetostl.com/view-wrl-online#convert>

Navigateur VRML à compiler : <https://freewrl.sourceforge.io/examples.html>

Peut être possible d'afficher un fichier VRML avec : <https://castle-engine.io/>

<https://www.qiew.org/>

<https://web.archive.org/web/20140412054654/http://cic.nist.gov/vrml/vbdetect.html>

Article extrait de : <http://lesporteslogiques.net/wiki/> - **WIKI Les Portes Logiques**

Adresse : [http://lesporteslogiques.net/wiki/recherche/residence\\_polygones/start?rev=1762637010](http://lesporteslogiques.net/wiki/recherche/residence_polygones/start?rev=1762637010)

Article mis à jour : **2025/11/08 22:23**