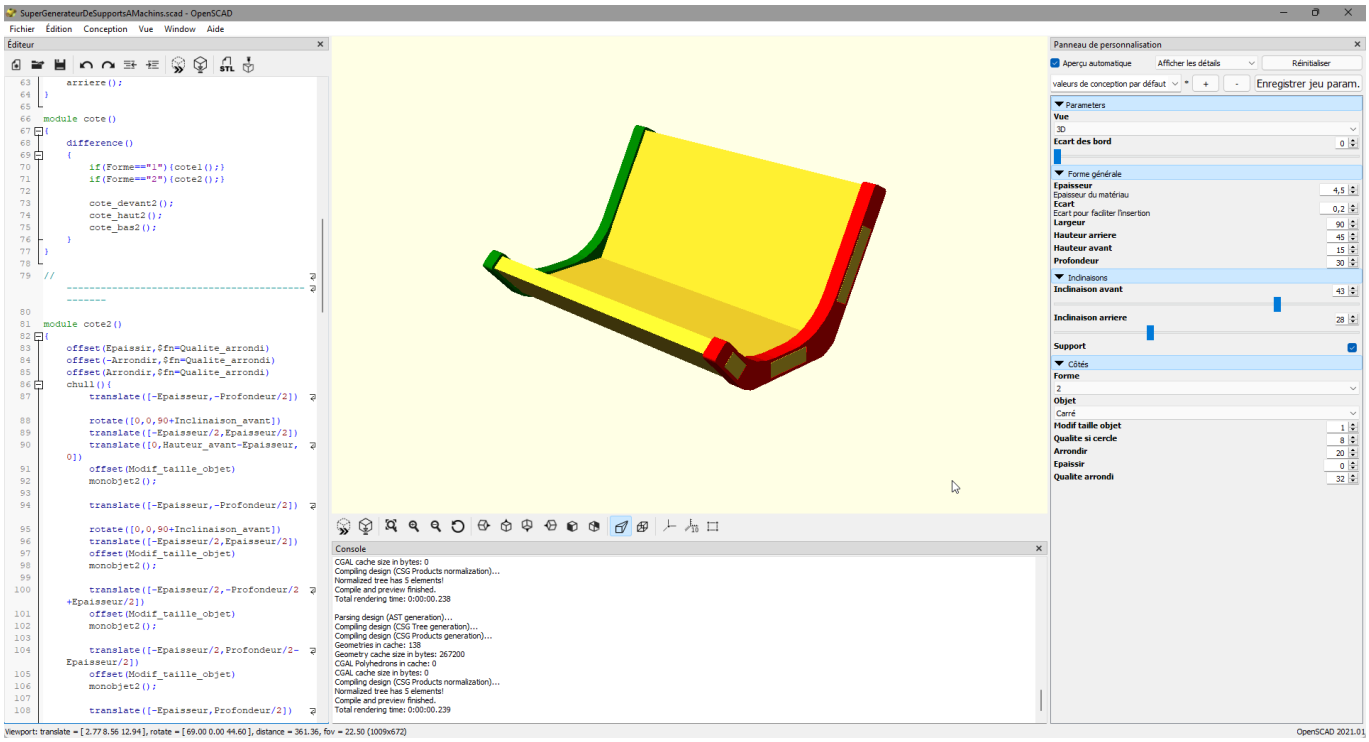
Super Générateur de Supports à Machins

Le **Super Générateur de Supports à Machins** est un générateur de supports à cartes de visite.

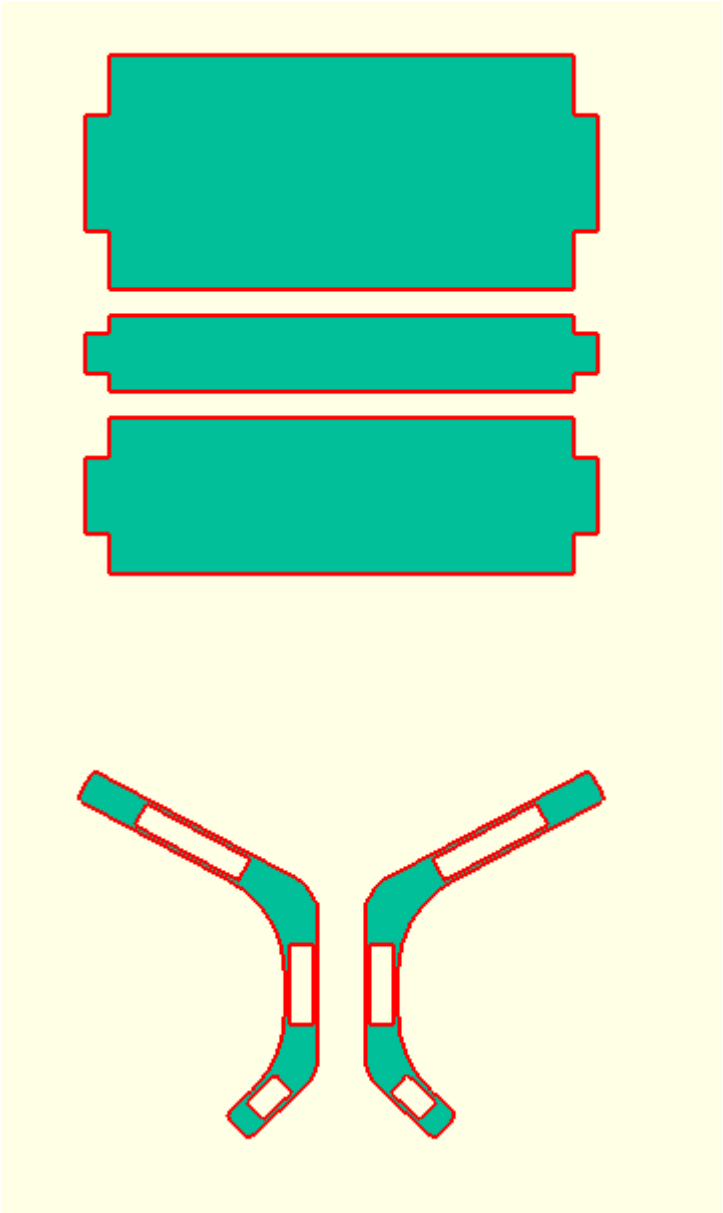
Etant paramétrable, il peut servir à d'autres choses, allant jusqu'à un support pour magazines, il suffit de prévoir une épaisseur suffisante pour le matériau de base.

Ce générateur a été créé en utilisant le logiciel OpenSCAD, logiciel de CAO 2D et 3D, dont le but est de s'affranchir de toute interface graphique pour ne permettre la modélisation que par un code écrit dans un langage spécifique à ce logiciel.

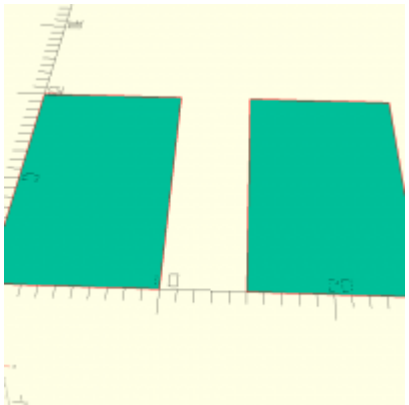
Les scripts OpenSCAD servent de base aux générateurs proposés sur le site Thingiverse [1] ainsi qu'à la nouvelle plateforme SOL75 [2].

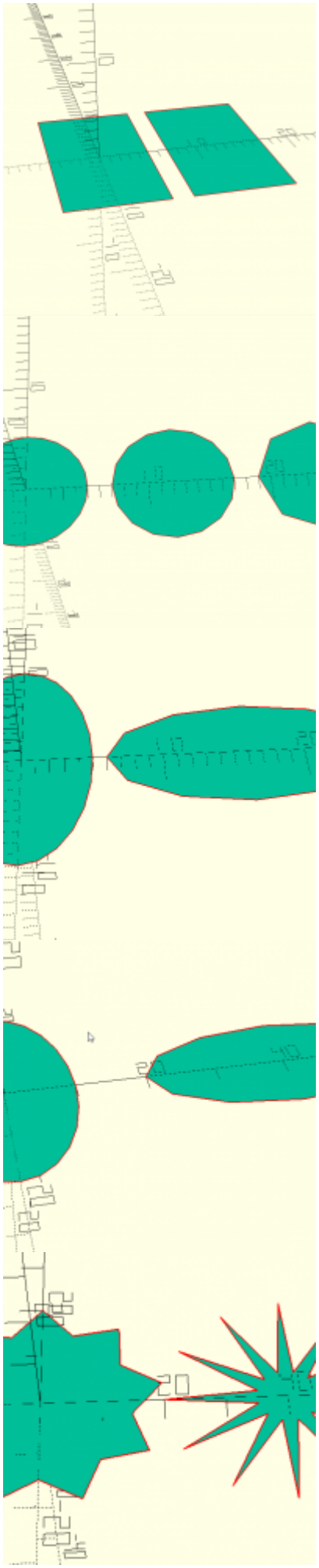


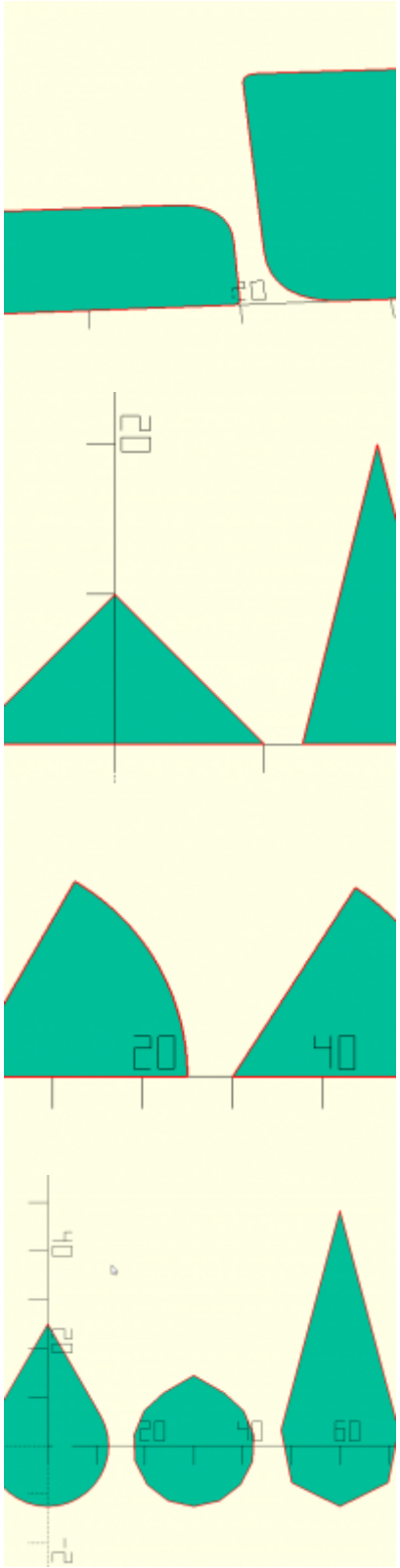
Une fois le modèle voulu acceptable, il est possible d'en créer un modèle 2D exportable en DXF ou en SVG :

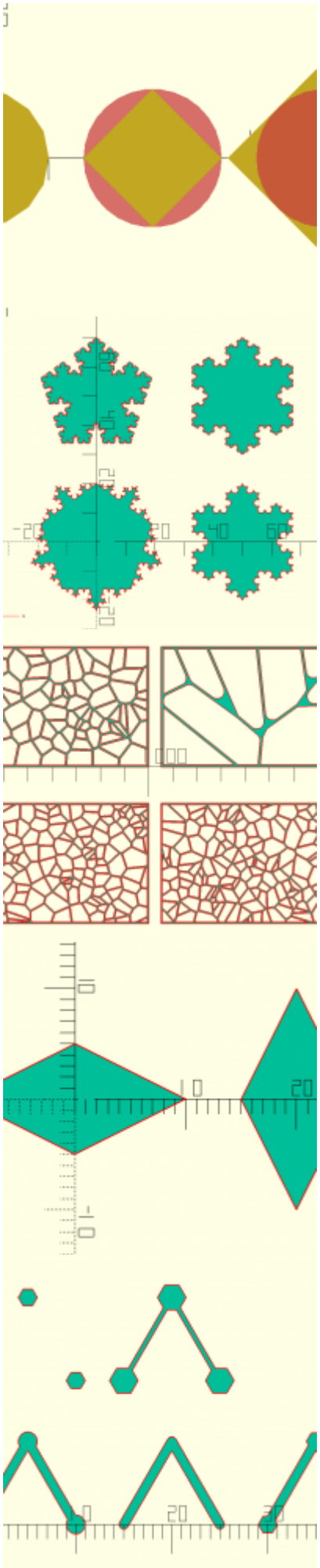


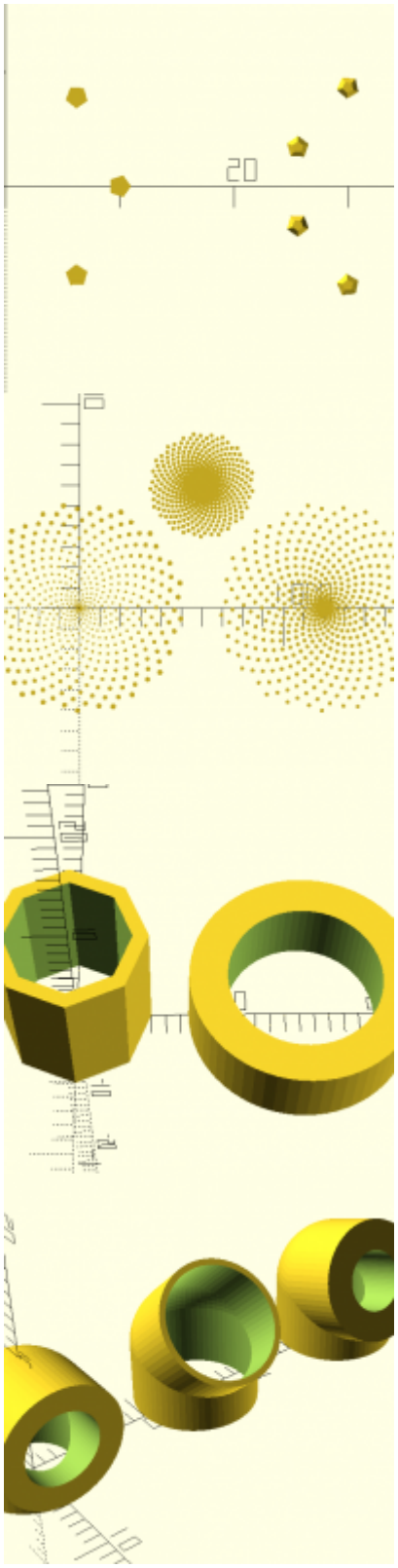
Exemples Divers

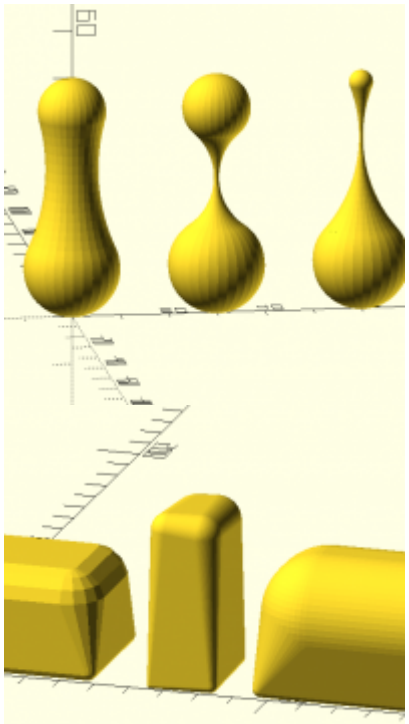


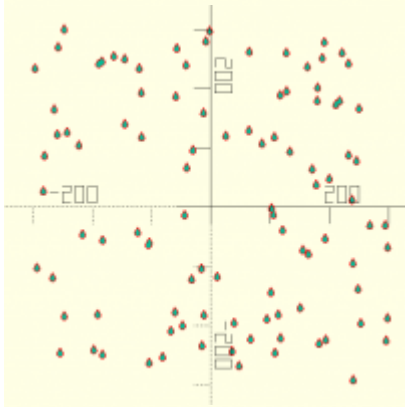


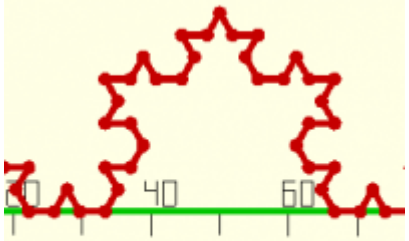

20
20
40
20
40
60

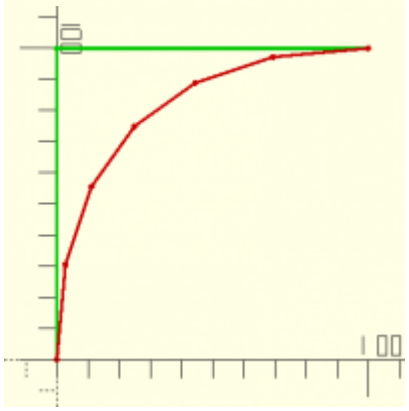


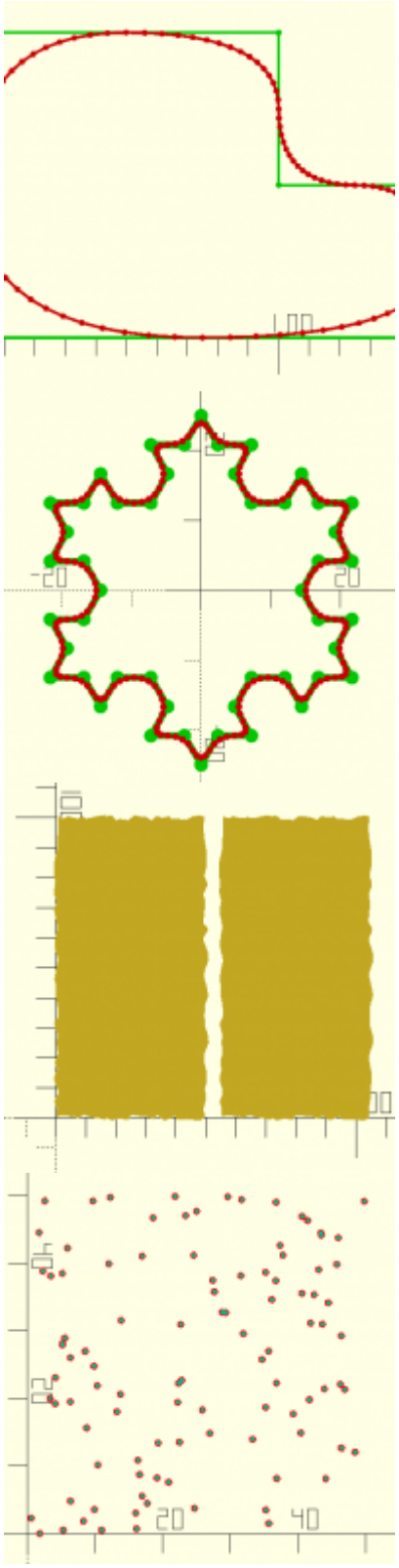


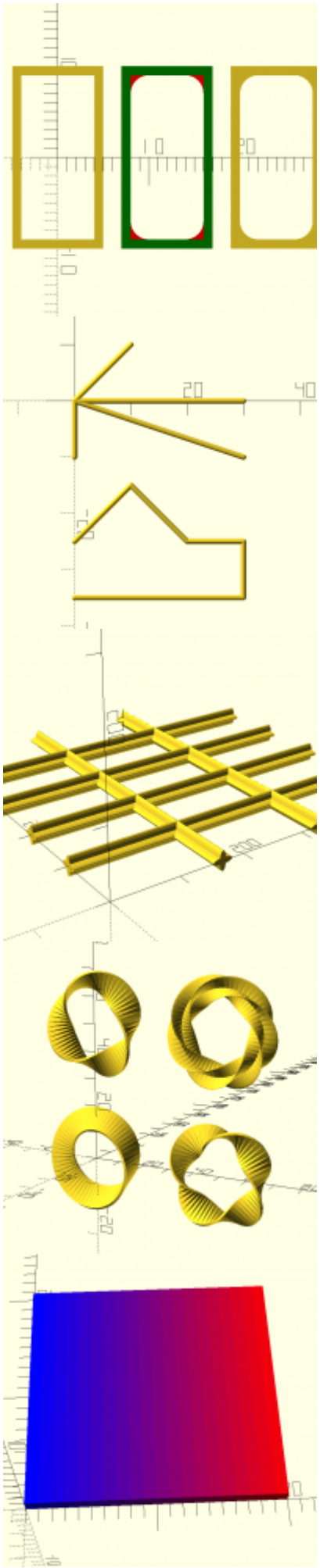


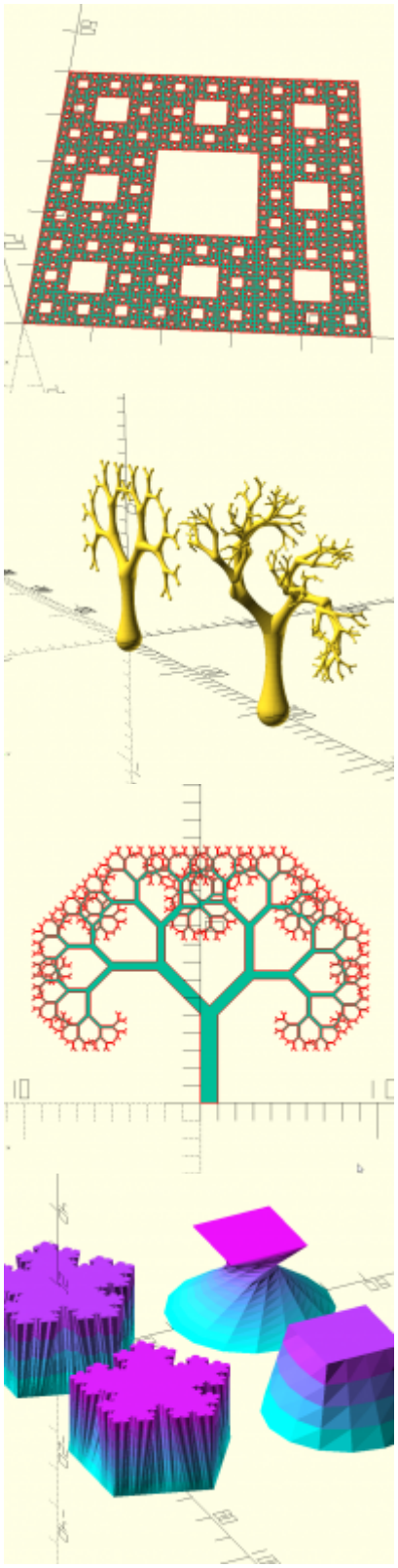

200
200
200
-200

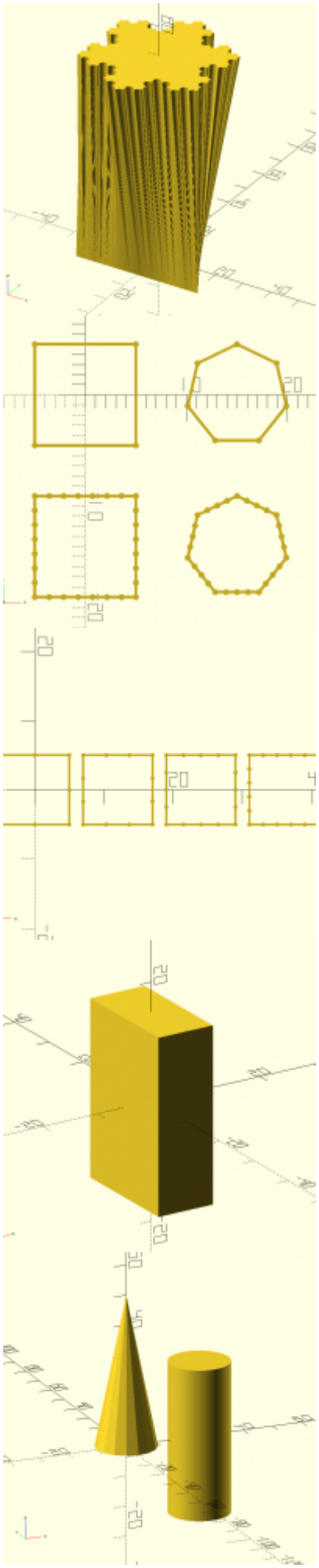

40
60

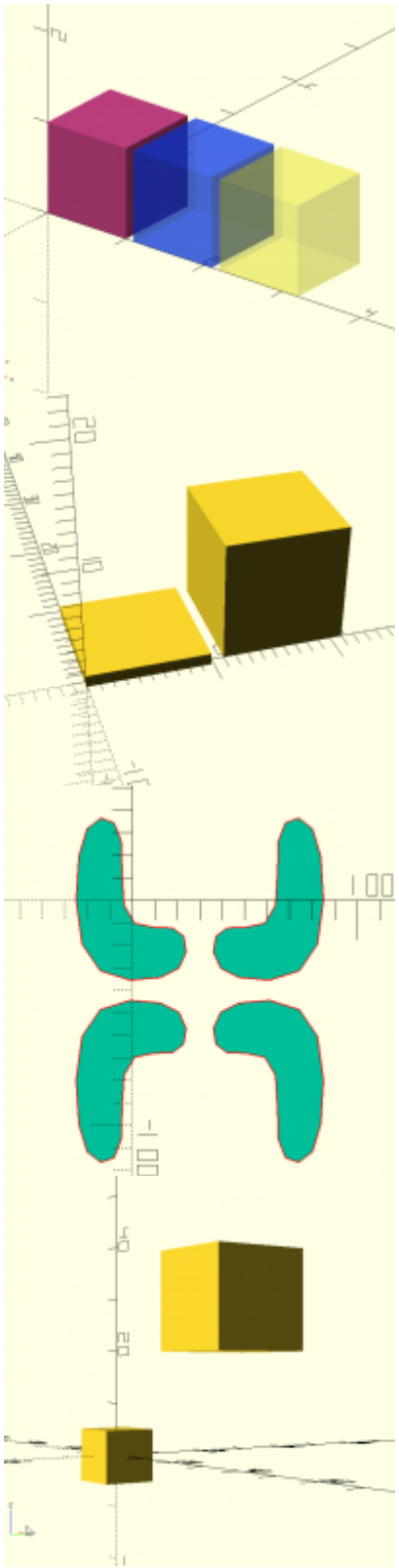

100

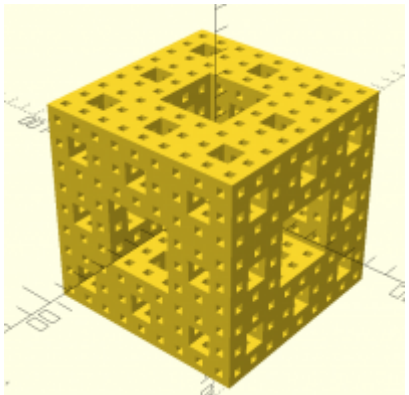


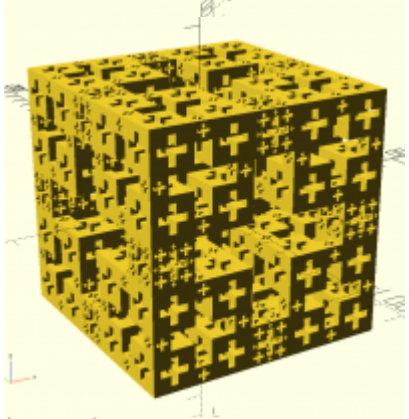


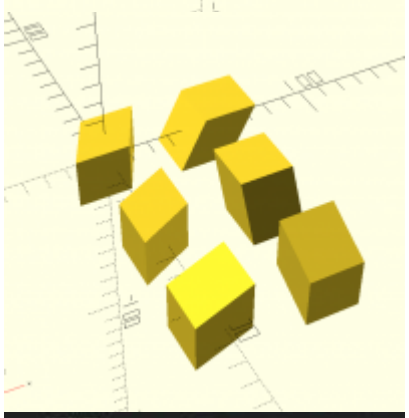


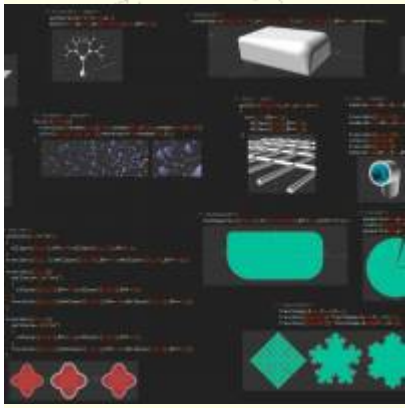


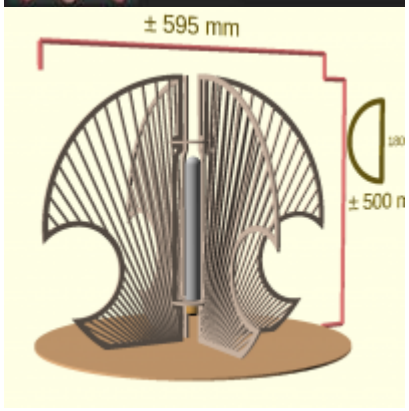


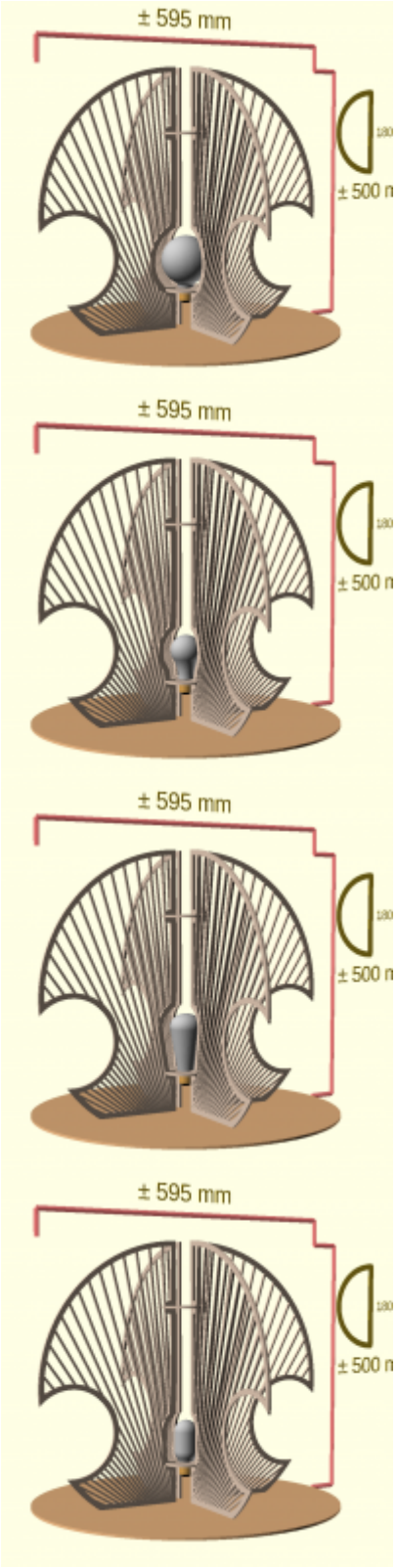


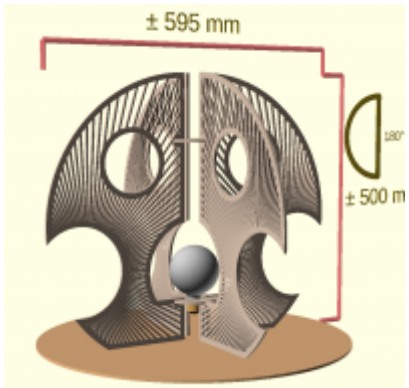


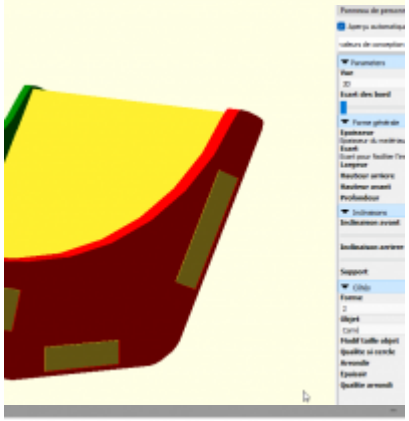


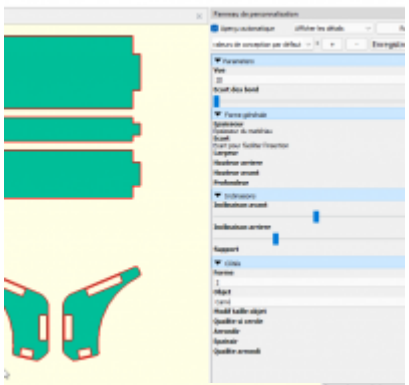


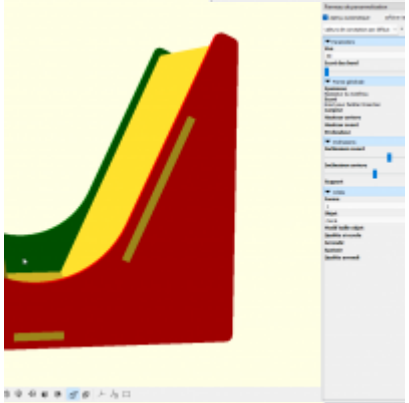

± 595 mm
± 500 m

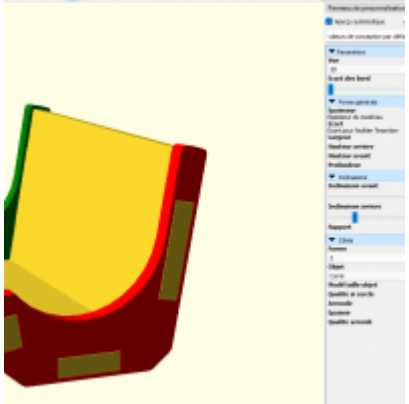

± 595 mm
± 595 mm
± 595 mm
± 595 mm

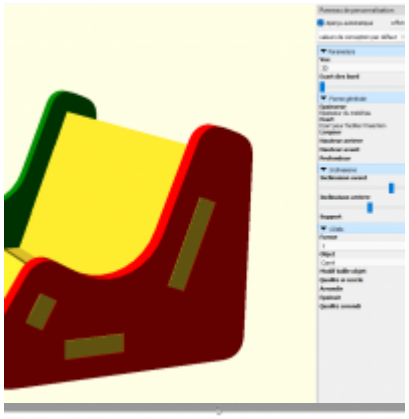

± 595 mm
± 500 m

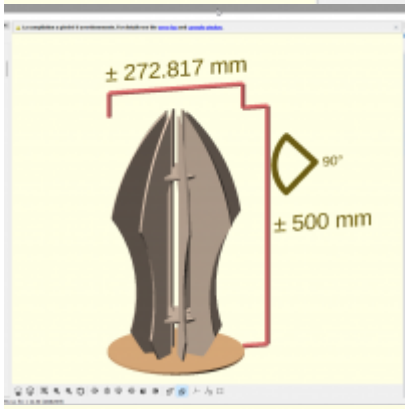


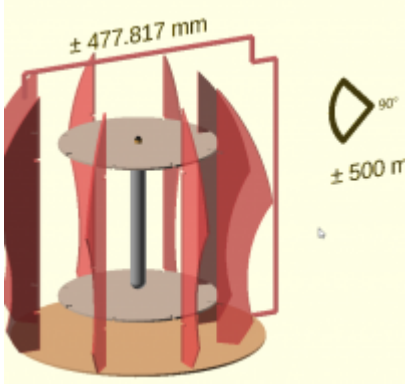


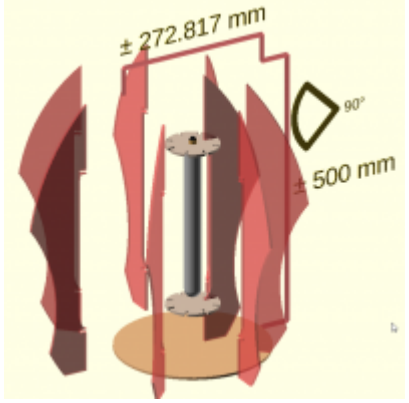


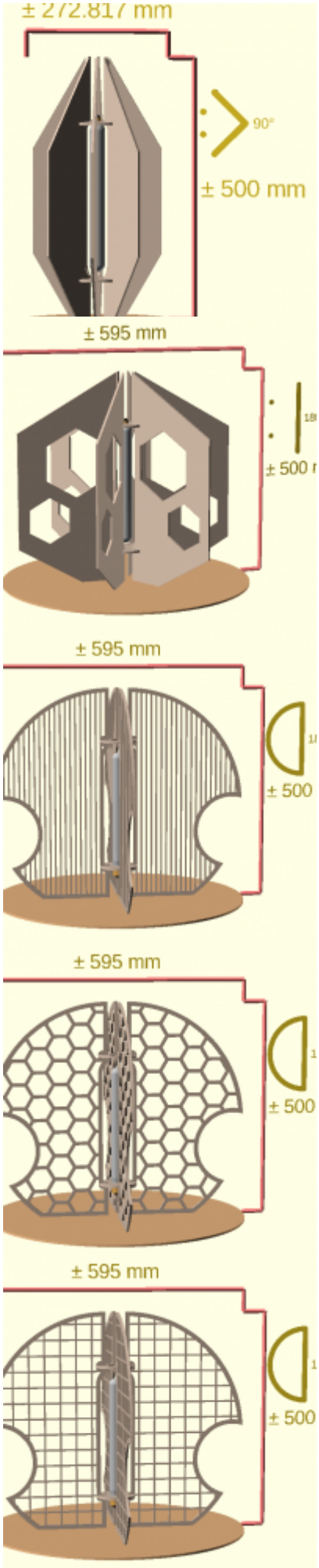


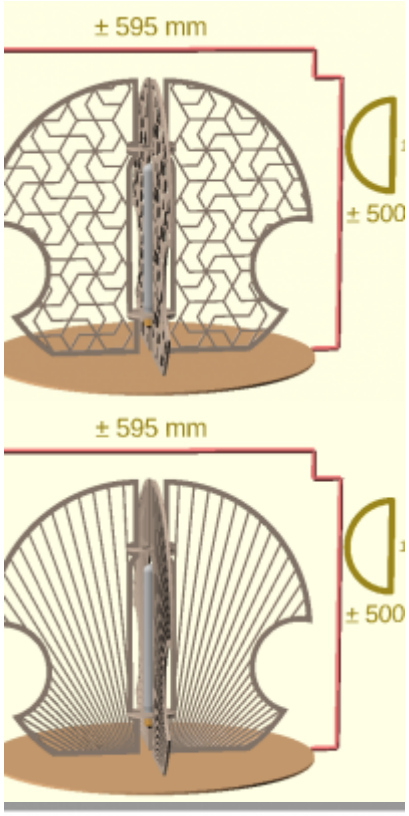


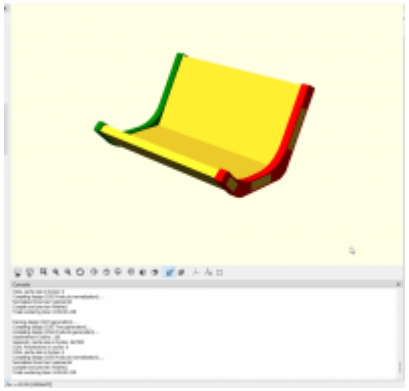

± 272.817 mm
90°
± 500 mm


± 477.817 mm
90°


± 272.817 mm
90°
± 500 mm


± 272.817 mm
90°
± 500 mm
± 595 mm
± 595 mm
± 595 mm
± 595 mm





SupportAMachins.scad

```
/*
Super générateur de support à machins
CC-BY-SA 2.0 Belgique - Marc Vanlindt (marc@vanlindt.be)

-----------------------------------------------

Carte de visite : 85*55
Magazine : 210*297

*/
// -----------------------------------------------

Vue                 ="3D";//[2D,3D]
Ecart_des_bord      =0;//[0:20]

/*[ Forme générale ]*/
```

```
//Epaisseur du matériau
Epaisseur           =4.5;
//Ecart pour faciliter l'insertion
Ecart               =0.2;
Largeur             = 90 ;
Hauteur_arriere     = 45 ;
Hauteur_avant       = 15 ;
Profondeur          = 30 ;
/*[ Inclinaisons ]*/
Inclinaison_avant   = 10;// [-90:90]
Inclinaison_arriere = 10;// [0:90]
Support=true;
/*[ Côtés ]*/
Forme               ="2";//[1,2]
Objet               ="Carré";//[Cercle, Carré]
Modif_taille_objet  =1;
Qualite_si_cercle   =8;
Arrondir            =20;
Epaissir            =0;
Qualite_arrondi     =32;

// -----------------------------------------------

if(Vue=="3D")
{
    vue3D();
}
if(Vue=="2D")
{
    vue2D();
}

module vue2D()
{
    translate([-Epaisseur-Epaissir-1,0])
    cote();

    translate([Epaisseur+Epaissir+1,0])
    mirror() cote();

    translate([0,Profondeur+Epaisseur+Epaissir+Profondeur+Profondeur])
    bas();

translate([0,Profondeur+Epaisseur+Epaissir+Profondeur+Profondeur/2+Haut
eur_avant/2+5+Profondeur])
    avant();
translate([0,Profondeur+Epaisseur+Epaissir+Profondeur+Profondeur/2+Haut
eur_avant+5+Hauteur_arriere/2+5+Profondeur])
    arriere();
}
```

```
module cote()
{
    difference()
    {
        if(Forme=="1"){cote1();}
        if(Forme=="2"){cote2();}

        cote_devant2();
        cote_haut2();
        cote_bas2();
    }
}

// ------------------------------------------------

module cote2()
{
    offset(Epaissir,$fn=Qualite_arrondi)
    offset(-Arrondir,$fn=Qualite_arrondi)
    offset(Arrondir,$fn=Qualite_arrondi)
    chull(){
        translate([-Epaisseur,-Profondeur/2])
        rotate([0,0,90+Inclinaison_avant])
        translate([-Epaisseur/2,Epaisseur/2])
        translate([0,Hauteur_avant-Epaisseur,0])
        offset(Modif_taille_objet)
        monobjet2();

        translate([-Epaisseur,-Profondeur/2])
        rotate([0,0,90+Inclinaison_avant])
        translate([-Epaisseur/2,Epaisseur/2])
        offset(Modif_taille_objet)
        monobjet2();

        translate([-Epaisseur/2,-Profondeur/2+Epaisseur/2])
        offset(Modif_taille_objet)
        monobjet2();

        translate([-Epaisseur/2,Profondeur/2-Epaisseur/2])
        offset(Modif_taille_objet)
        monobjet2();

        translate([-Epaisseur,Profondeur/2])
        rotate([0,0,180-Inclinaison_arriere])
        translate([Epaisseur/2,-Epaisseur/2])
        offset(Modif_taille_objet)
        monobjet2();

        translate([-Epaisseur,Profondeur/2])
        rotate([0,0,180-Inclinaison_arriere])
```

```
        translate([Epaisseur/2,-Epaisseur/2])
        translate([Hauteur_arriere-Epaisseur,0])
        offset(Modif_taille_objet)
        monobjet2();

        translate([-Epaisseur,Profondeur/2])
        rotate([0,0,180-Inclinaison_arriere])
        translate([Epaisseur/2,-Epaisseur/2])
        offset(Modif_taille_objet)
        monobjet2();

        translate([-Epaisseur,Profondeur/2])
        rotate([0,0,180-Inclinaison_arriere])
        translate([Epaisseur/2,-Epaisseur/2])
        translate([Hauteur_arriere-Epaisseur,0])
        offset(Modif_taille_objet)
        monobjet2();
    }
}

// ------------------------------------------------

module cote1()
{
    offset(Epaissir,$fn=Qualite_arrondi)
    offset(-Arrondir,$fn=Qualite_arrondi)
    offset(Arrondir,$fn=Qualite_arrondi)
    {
        hull()
        {
            translate([-Epaisseur/2,Profondeur/2+Epaisseur/2-
sin(Inclinaison_arriere)*Epaisseur/2])
            monobjet();
            if(Support==true)
            {
                translate([-
Epaisseur/2,Profondeur/2+Epaisseur/2+sin(Inclinaison_arriere)*Hauteur_a
rriere-sin(Inclinaison_arriere)*Epaisseur/2])
                monobjet();
            }
            translate([-Epaisseur/2,Profondeur/2+Epaisseur/2-
sin(Inclinaison_arriere)*Epaisseur/2])
            rotate([0,0,-180-Inclinaison_arriere])
            translate([Hauteur_arriere,0,0])
            monobjet();
        }
        hull(){
            translate([-Epaisseur/2,-Profondeur/2-
Epaisseur/2+sin(Inclinaison_avant)*Epaisseur/2])
            monobjet();
            translate([-Epaisseur/2,Profondeur/2+Epaisseur/2-
```

```
sin(Inclinaison_arriere)*Epaisseur/2])
            monobjet();
        }
        hull(){
            translate([-Epaisseur/2,-Profondeur/2-
Epaisseur/2+sin(Inclinaison_avant)*Epaisseur/2])
            monobjet();
            translate([-Epaisseur/2,-Profondeur/2-
Epaisseur/2+sin(Inclinaison_avant)*Epaisseur/2])
            rotate([0,0,-180+Inclinaison_avant])
            translate([Hauteur_avant,0,0])
            monobjet();
        }
    }
}

module monobjet()
{
    if(Objet=="Cercle")
    {
        circle(Epaisseur/2+Modif_taille_objet/2,$fn=Qualite_si_cercle);
    }
    if(Objet=="Carré")
    {
        square(Epaisseur+Modif_taille_objet,center=true);
    }
}
module monobjet2()
{
    if(Objet=="Cercle")
    {
        circle(Epaisseur/2,$fn=Qualite_si_cercle);
    }
    if(Objet=="Carré")
    {
        square(Epaisseur,center=true);
    }
}

module cote_haut2()
{
    translate([-Epaisseur,Profondeur/2,0])
    rotate([0,0,90-Inclinaison_arriere])
    translate([Epaisseur/2,Hauteur_arriere/2])
    square([Epaisseur+Ecart,Hauteur_arriere/2+Ecart],center=true);
}

module cote_devant2()
{
    translate([-Epaisseur,-Profondeur/2,0])
```

```
    rotate([0,0,90+Inclinaison_avant])
    translate([-Epaisseur/2,Hauteur_avant/2])
    square([Epaisseur+Ecart,Hauteur_avant/2+Ecart],center=true);
}

module cote_bas2()
{
    translate([-Epaisseur/2,0])
    square([Epaisseur+Ecart,Profondeur/2+Ecart],center=true);
}

module cote_haut()
{
    translate([-Epaisseur,Profondeur/2,0])
    rotate([0,0,90-Inclinaison_arriere])
    translate([Epaisseur/2,Hauteur_arriere/2])
    square([Epaisseur,Hauteur_arriere],center=true);
}

module cote_devant()
{
    translate([-Epaisseur,-Profondeur/2,0])
    rotate([0,0,90+Inclinaison_avant])
    translate([-Epaisseur/2,Hauteur_avant/2])
    square([Epaisseur,Hauteur_avant],center=true);
}

module cote_bas()
{
    translate([-Epaisseur/2,0])
    square([Epaisseur,Profondeur],center=true);
}

module vue3D(){
    translate([0,Profondeur/2,Epaisseur])
    rotate([-90-Inclinaison_arriere,0,0])
    translate([0,-Hauteur_arriere/2,0])
    linear_extrude(Epaisseur)
    arriere();
    translate([0,-Profondeur/2,Epaisseur])
    rotate([90+Inclinaison_avant,0,0])
    translate([0,Hauteur_avant/2,0])
    linear_extrude(Epaisseur)
    avant();
    linear_extrude(Epaisseur)
    bas();

    color("red")
    translate([Largeur/2+Ecart_des_bord,0,0])
    rotate([90,90,90])
    linear_extrude(Epaisseur)
```

```
    cote();

    color("green")
    translate([-Largeur/2-Epaisseur-Ecart_des_bord,0,0])
    rotate([90,90,90])
    linear_extrude(Epaisseur)
    cote();
}

module bas()
{
    square([Largeur, Profondeur],center=true);
    square([Largeur+Epaisseur*2, Profondeur/2],center=true);
}

module avant()
{
    square([Largeur, Hauteur_avant],center=true);
    square([Largeur+Epaisseur*2, Hauteur_avant/2],center=true);
}
module arriere()
{
    square([Largeur, Hauteur_arriere],center=true);
    square([Largeur+Epaisseur*2, Hauteur_arriere/2],center=true);
}

module chull(m){
  for(i=[0:$children-2]){
    hull(){
      children(m==true?0:i);
      children(i+1);
    }
  }
}
```

1)
Customizer Thingiverse
2)
Site béta - SOL75