

# Maéva DORMANT

## Informations générales :

- **UE :** MU4IN011, Processus d'innovation, M1 Management de l'innovation
- **Intervenant :** M. SIMON, Directeur du Fablab de Sorbonne Université
- **Étudiante :** maeva.dormant@etu.sorbonne-universite.fr

## Notes de cours :

### Cours n°1 : Introduction à .openscad

#### 1. Définition

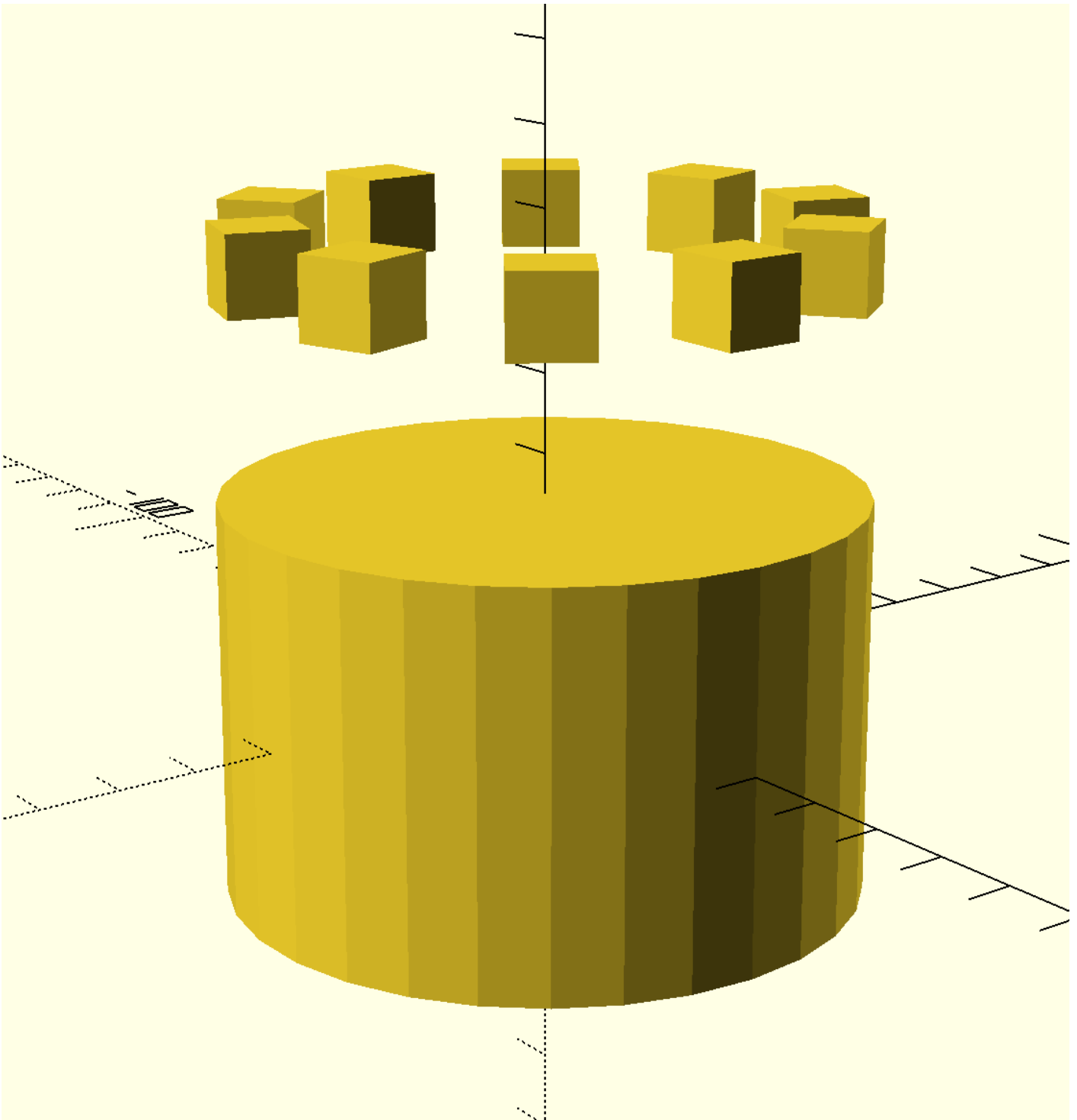
**OpenSCAD est un logiciel de modélisation 3D basé sur le texte** qui permet de créer des modèles 3D paramétriques en utilisant du code. Il est utilisé pour la personnalisation de pièces, la conception d'objets d'impression 3D, la réplique de pièces avec des variations et l'automatisation. C'est un outil utile dans l'éducation pour enseigner la modélisation 3D, la programmation et la géométrie.

#### 2. Première approche du logiciel

```
d=5;
a=10;
H=50;
R=40;
N=10;

for (i=[1:1:N]) {
    rotate([0,0,i*360/N]) translate([(R-a/2),0,H+a/2]) cube(a,center=true);
}

cylinder(h=H,r=R,center=true);
```



## Cours n°2 : Polyvalence des filaments et techniques d'impressions

### 1. Introduction

Les **imprimantes 3D offrent une polyvalence en matière de choix de filaments et de techniques d'impression**, permettant une vaste gamme de projets. Elles sont compatibles avec divers types de filaments. Chaque filament présente des propriétés spécifiques, offrant aux utilisateurs la flexibilité nécessaire pour choisir le matériau adapté à leurs besoins.

En outre, les imprimantes 3D sont compatibles avec des techniques avancées d'impression, telles que l'impression multi-matériaux et multi-couleurs, l'impression en double extrusion, ainsi que

l'impression avec des supports solubles. Ces fonctionnalités élargissent considérablement les possibilités de création.

## 2. Exemples de filaments

### PLA :

- Caractéristiques : Biodégradable, facile à imprimer, idéal pour le prototypage rapide.
- Utilisations courantes : Projets non soumis à des contraintes de chaleur importantes.

### Fibres de carbone, de verre ou de métal :

- Caractéristiques : Filaments renforcés, offrent une résistance et une rigidité supérieures.
- Utilisations courantes : Pièces structurelles, prototypes haute performance.

### PVA :

- Caractéristiques : Soluble dans l'eau, utilisé comme support pour les impressions complexes.
- Utilisations courantes : Structures avec des surplombs, géométries complexes.

## Cours n°3 : Les micro-contrôleurs

### 1. Introduction

Les **micro-contrôleurs Arduino sont de petits ordinateurs sur une puce**, conçus pour rendre la programmation et le prototypage électronique accessibles à tous. Que l'on soit débutant ou confirmé, ils offrent un moyen puissant et abordable de donner vie à aux idées.

### 2. Utilisation simplifiée

Pour les utiliser, il suffit de **connecter l'Arduino à un ordinateur**, d'écrire un code simple dans l'environnement de développement Arduino, puis de téléverser ce code sur le micro-contrôleur. Le programme peut contrôler des lumières, des moteurs, des capteurs, etc.

### 3. Langage accessible

Le **langage de programmation Arduino est basé sur C/C++**, mais simplifié pour faciliter l'apprentissage. De nombreux exemples de code sont disponibles en ligne, ce qui facilite la création de projets amusants, éducatifs ou même pratiques.

## Cours n°4 : Développement d'applications mobiles pour les non-informaticiens

### 1. Un peu d'histoire

Auparavant (avant 2007), les applications étaient natives en C++/C#. À la fin 2008, Apple et Android deviennent les 2 grands leaders du marché. À ces époques, les applications web étaient des **pages de texte statique** (HTML). Par la suite, le **web 2.0** (pages actives) entrent en vigueur (HTML, CSS, JS). Les frameworks Javascript sont : React JS, Vue JS et ANGULAR JS.

## 2. App inventor

C'est une seule application pour designer et développer créée par le MIT. Il permet de **transformer le code de l'application en "bloc" en code JS**. En phase de test, il génère un QR code et l'application "AI Companion" permet de la visualiser. Néanmoins, si l'on souhaite l'envoyer sur l'App store (par exemple), il faut une licence Xcode.

---

## Exercices :

Exercices donnés lors du cours n°1 : Réalisation de l'objet de notre choix

### 1. Réaliser un objet 3D sur .openscad utilisant la différence ou l'addition.

Mon choix s'est porté sur la réalisation d'un [bonhomme de neige](#). J'utilise la différence pour créer des sphères creuses.

☒ Étape 1 : Code (Openscad)

```
$vpt=[4,2.5,38];
$vpr=[83,0,29];
$vpd=250;

// Boules

difference(){
translate([0,0,10])
color("#FFFFFF") sphere(20);
translate([-30,-30,-10])
  color("#FFFFFF") cube([100,100,10]);
color("#FFFFFF") sphere(13);
}

difference(){
translate([0,0,40])
color("#FFFFFF") sphere(15);
translate([-30,-30,-10])
  color("#FFFFFF") cube([100,100,10]);
color("#FFFFFF") sphere(13);
}
```

```
difference(){  
  translate([0,0,60])  
  color("#FFFFFF") sphere(10);  
  translate([-30,-30,-10])  
    color("#FFFFFF") cube([100,100,10]);  
  color("#FFFFFF") sphere(10);  
}
```

// Bras

```
translate([-30,0,45])  
rotate([0,100,0])  
color("#582900") cylinder(20,2,2);
```

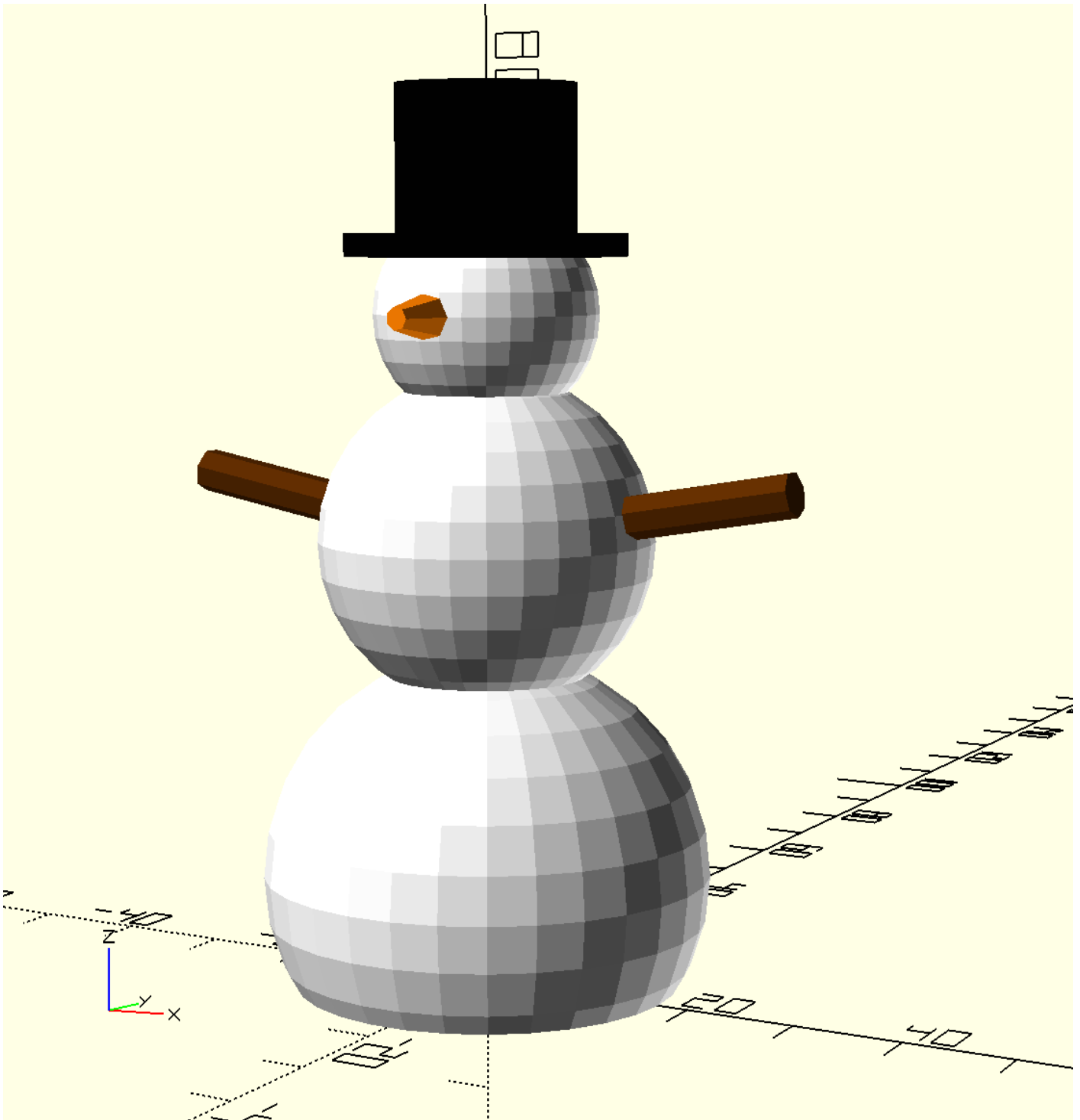
```
translate([10,0,42])  
rotate([0,80,0])  
color("#582900") cylinder(20,2,2);
```

// Nez

```
translate([0,-10,60])  
rotate([90,0,0])  
color("#ff8000") cylinder(5,2,1);
```

// Chapeau

```
translate([0,0,65])  
color("#000000") cylinder(2,12.5,12.5);  
translate([0,0,65])  
color("#000000") cylinder(15,8,8);
```



☒ Étape 2 : Définition des paramètres d'impression (IdeaMaker)

Pour imprimer en 1 heure(s) 57 min 18 sec, un objet de 50.000 x 32.936 x 66.235 :

- Type de filament : PLA 1,75 mm
- Couche : Hauteur de couche - 0,2500 mm / Coques - 1,0
- Remplissage : Densité - 25% / Motif : Cubique
- Aide à l'adhérence : Radeau uniquement

☒ Étape 3 : Impression (Raise3D)

Il suffit de lancer l'impression en respectant les paramètres d'impression.

## 2. Réaliser un objet 3D sur .openscad. Il doit être uniquement réalisable via l'impression 3D.

Mon choix s'est porté sur la réalisation d'un [chat assis](#).

☒ Étape 1 : Code (Openscad)

```
$fn=30;

// TÊTE

difference(){
color("#E5E4E2") hull(){
translate([0,0,15])
scale([1,0.75,1])
sphere(7);
}
}

// OREILLES

module oreilles(){
hull(){
translate([2,0,0])
    sphere(1);
translate([-2,0,0])
    sphere(1);
translate([0,2,0])
    sphere(1);
}}

translate([5,0,20])
rotate([-90,100,0])
color("#808080") oreilles();

translate([-5,0,20])
rotate([-90,-100,0])
color("#808080") oreilles();
```

```
// CORPS
```

```
translate([0,1,1])  
scale([1.8,1.5,1])  
color("#E5E4E2") sphere(3);
```

```
translate([0,1,2])  
rotate([4,0,0])  
color("#808080") cylinder(7.5,4,1.5);
```

```
// QUEUE
```

```
module queue(){  
  hull(){  
    translate([0,.5,13])  
    sphere(0.5);  
    translate([0,4,18])  
    sphere(1);  
  }  
}
```

```
translate([0,4,-12])  
color("#E5E4E2") queue();
```

```
// BRAS
```

```
translate([-6,0,4.5])  
rotate([0,75,0])  
color("#808080") cylinder(5,1,1.5);
```

```
translate([6,0,4.5])  
rotate([0,-75,0])  
color("#808080") cylinder(5,1,1.5);
```

```
// MAINS
```

```
module mains(){  
  scale([2,2,1])  
  sphere(.8);  
  translate([0,-2,0])
```

```
scale([1,3,1])
sphere(.6);
translate([-1,-2,0])
rotate([0,0,-20])
scale([1,3,1])
sphere(.6);
translate([1.2,-1.5,0])
rotate([0,0,20])
scale([1,3,1])
sphere(.6);
    translate([-2,-.5,0])
rotate([0,0,-60])
scale([1,3,1])
sphere(.6);
}
```

```
translate([5.6,0,5 ])
rotate([-90,0,80])
color("#E5E4E2") mains();
```

```
translate([-5.6,0,5 ])
rotate([-90,0,80])
color("#E5E4E2") mains();
```

```
// JAMBES
```

```
translate([-3,1,1])
rotate([-90,0,135])
color("#808080") cylinder(12,2,1.5);
```

```
translate([3,1,1])
rotate([-90,0,-135])
color("#808080") cylinder(12,2,1.5);
```

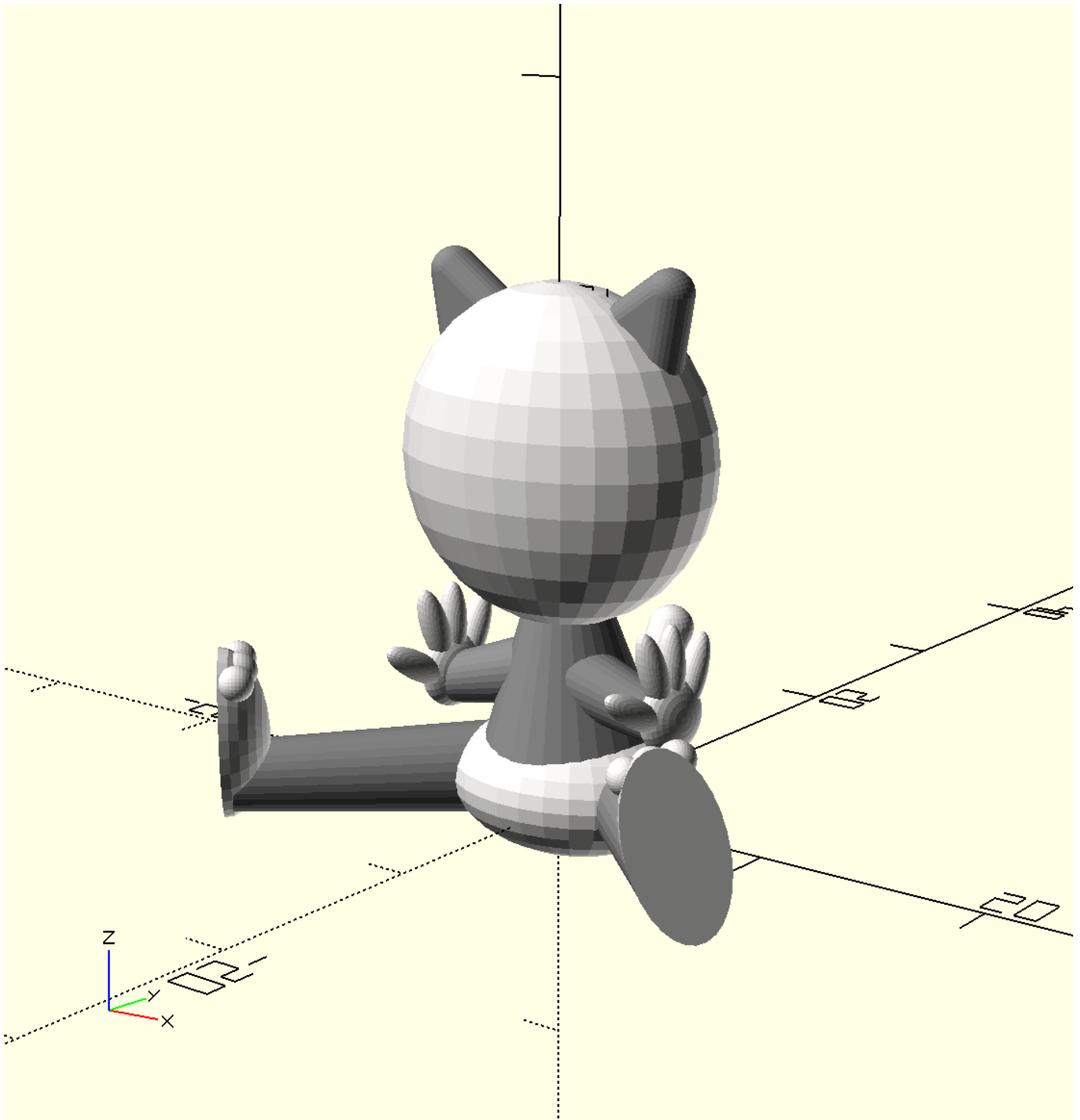
```
// PIEDS
```

```
module pieds() {
    difference(){
        scale([1,2,1])
        sphere(2);
```

```
translate([-2,-10,-2])
  cube([5,20,2]);
}
translate([0,-3,.80])
sphere(.8);
translate([-1,-3,.75])
sphere(.75);
  translate([1,-3,.75])
sphere(.75);
}
```

```
translate([-11.5,-8,3])
rotate([-90,15,-50])
color("#E5E4E2") pieds();
```

```
translate([11.5,-8,3])
rotate([-90,-15,50])
color("#E5E4E2") pieds();
```



☒ Étape 2 : Définition des paramètres d'impression (IdeaMaker)

Pour imprimer en 1 heure(s) 16 min 32 sec, un objet de 50.000 x 36.166 x 48.339 :

- Type de filament : PLA 1,75 mm
- Couche : Hauteur de couche - 0,2500 mm / Coques - 1,0
- Remplissage : Densité - 25% / Motif : Cubique
- Aide à l'adhérence : Radeau uniquement

☒ Étape 3 : Impression (Raise3D)

Il suffit de lancer l'impression en respectant les paramètres d'impression.

## Exercice donné lors du cours n°2 : Ajout de complexité

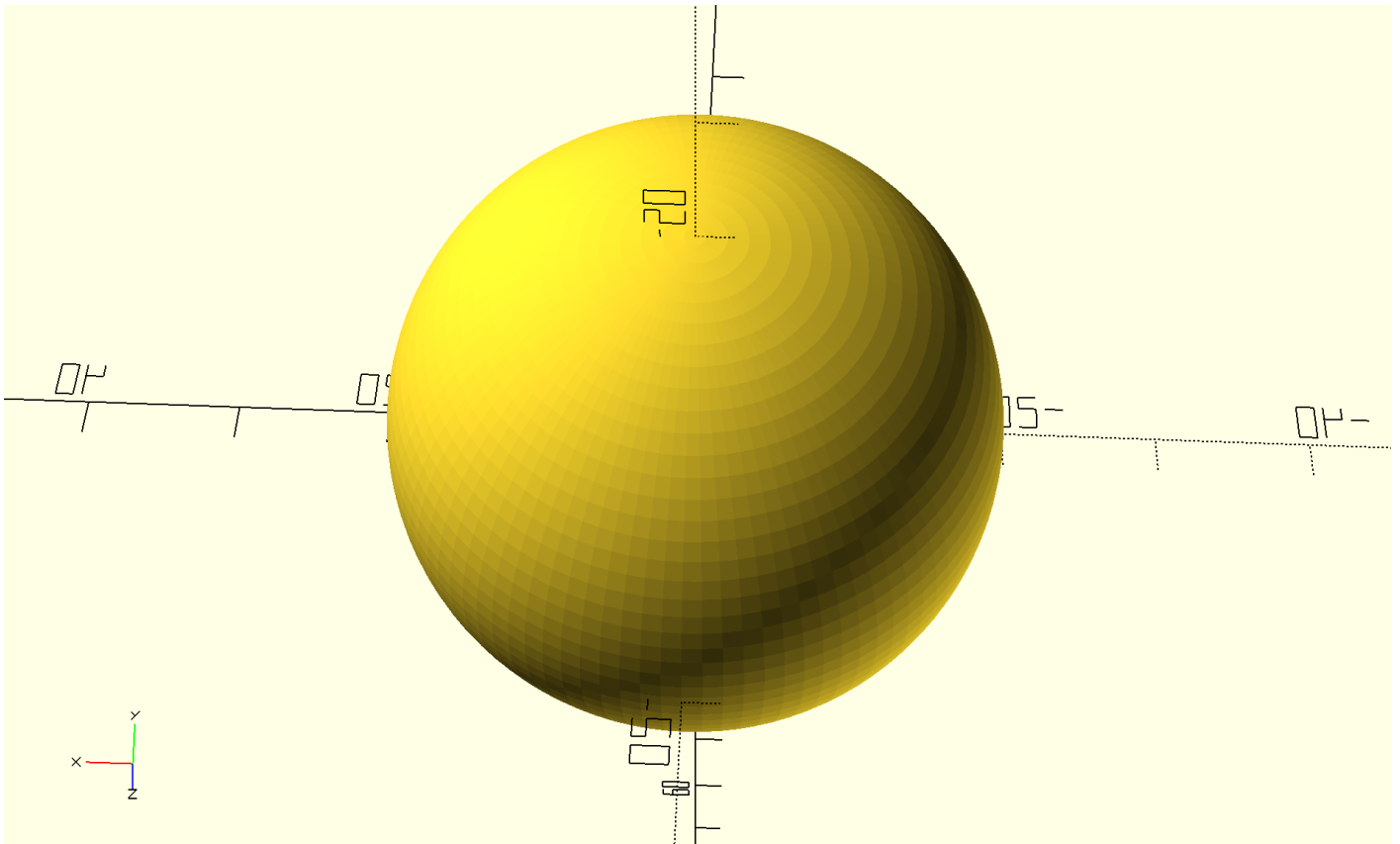
**1. Réaliser un objet plus complexe. Par exemple, mettre un objet solide dans une impression 3D ou réaliser une impression bi-couleur.**

Mon choix s'est porté sur la réalisation d'une **sphère creuse**.

☒ Étape 1 : Code (Openscad)

```
$fn = 100; // Nombre de faces pour l'approximation de la sphère
$r_ext = 20; // Rayon externe de la sphère
$r_int = 15; // Rayon interne de la sphère creuse

difference() {
  sphere($r_ext);
  translate([0, 0, 0])
    sphere($r_int);
}
```



☒ Étape 2 : Définition des paramètres d'impression (IdeaMaker)

Pour imprimer en 2 heure(s) 38 min 33 sec, un objet de 60.000 x 60.000 x 60.000 :

- Type de filament : PLA 1,75 mm (transparent)
- Couche : Hauteur de couche - 0,500 mm / Coques - 1,0
- Remplissage : Densité - 10% / Motif : Cubique
- Aide à l'adhérence : Radeau uniquement

⚠ Lors de la définition de l'échelle, il faut veiller à ce que la sphère soit plus grande que l'objet à insérer.

☒ Étape 3 : Impression (Raise3D)

Lors de l'impression, stopper l'impression à mi-hauteur (approximativement la moitié du temps d'impression) et insérer l'objet à l'intérieur de la sphère.

## 2. Réaliser un post-traitement sur l'objet précédent :

<https://www.youtube.com/watch?v=Jv-CbnB8Uz4&t=83s>

- ☐ Réunir le matériel : Pièce imprimée avec du filament ABS - Boîte de conserve - Acétone
- ☐ Verser de l'acétone dans le récipient et le faire chauffer.
- ☐ Plonger la pièce dans la boîte et refermer.

## Exercice donné lors du cours n°3 : Montage simple

À l'aide du cours, réaliser un projet simple à l'aide des micro-contrôleurs Arduino.

- ☐ Récupérer le cours.
- ☒ Réaliser le code.

```
const int RED_LED_PIN = 0; // Broche pour la LED rouge
const int GREEN_LED_PIN = 1; // Broche pour la LED verte

void configurationInitiale() {
  pinMode(RED_LED_PIN, OUTPUT); // Configuration de la broche pour la LED rouge en sortie
  pinMode(GREEN_LED_PIN, OUTPUT); // Configuration de la broche pour la LED verte en sortie
}

void bouclePrincipale() {
  digitalWrite(RED_LED_PIN, HIGH); // Allumer la LED rouge
  digitalWrite(GREEN_LED_PIN, LOW); // Éteindre la LED verte
  delay(2000); // 2 secondes d'attente
```

```
digitalWrite(RED_LED_PIN, LOW); // Éteindre la LED rouge
digitalWrite(GREEN_LED_PIN, HIGH); // Allumer la LED verte
delay(2000); // 2 secondes d'attente
}

void setup() {
  configurationInitiale();
}

void loop() {
  bouclePrincipale();
}
```

☐ Simuler le montage sur Thinkercad.

☐ Réaliser le vrai montage Arduino.

## Exercice donné lors du cours n°4 : App inventor

**Mettre en place un premier programme, application et la tester sur notre téléphone.**

---

Revision #32

Created 27 October 2023 12:27:28 by Maéva DORMANT

Updated 24 January 2024 15:51:22 by Maéva DORMANT