

# Projet info223

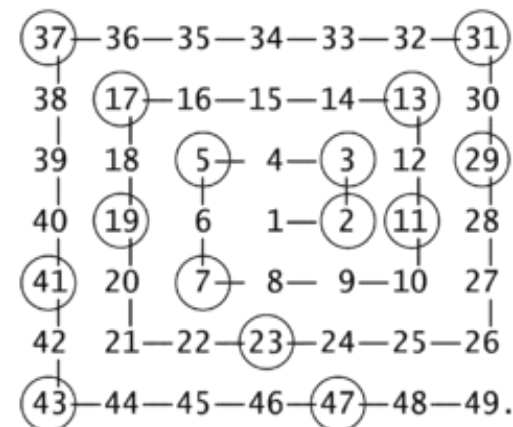
## Spirale d'Ulam en 3 dimensions

Chloe Makoundou

L2 info groupe 1 - Frederic Vernier

<https://github.com/Chloooow/ProjetUlam1>  
GSD

```
Array  
14 float[] observe;  
15  
16 void setup() {  
17   size(640, 360);  
18   observe = new float[width];  
19   for (int i = 0; i < width; i++) {  
20     float amount = map(i, 0, width, 0, PI);  
21     observe[i] = abs(cos(amount));  
22   }  
23   background(255);  
24   noLoop();  
25 }  
26  
27 void draw() {  
28   int y1 = 0;  
29   int y2 = height/2;  
30   for (int i = 0; i < width; i++) {  
31     stroke(observe[i]*255);  
32   }
```

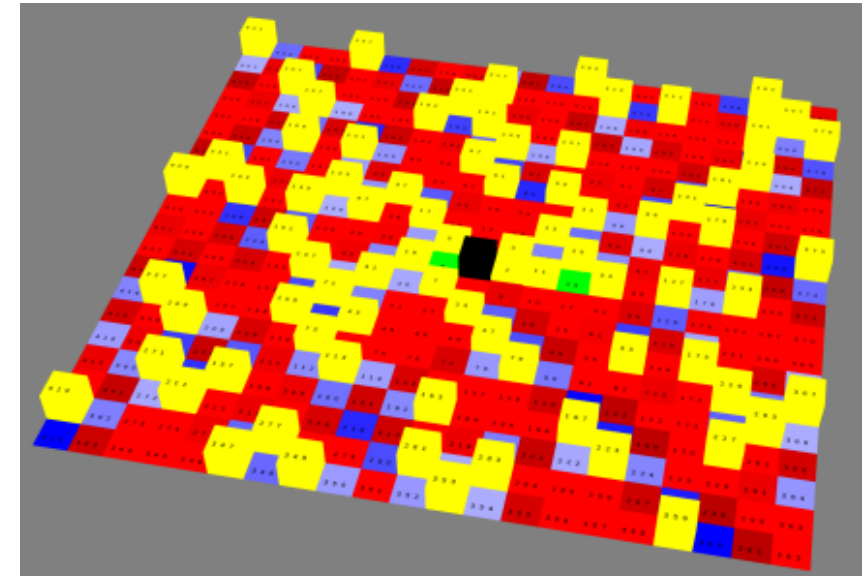
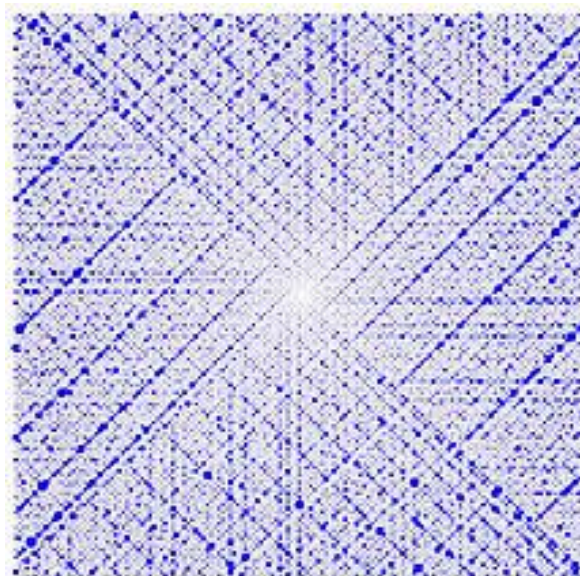
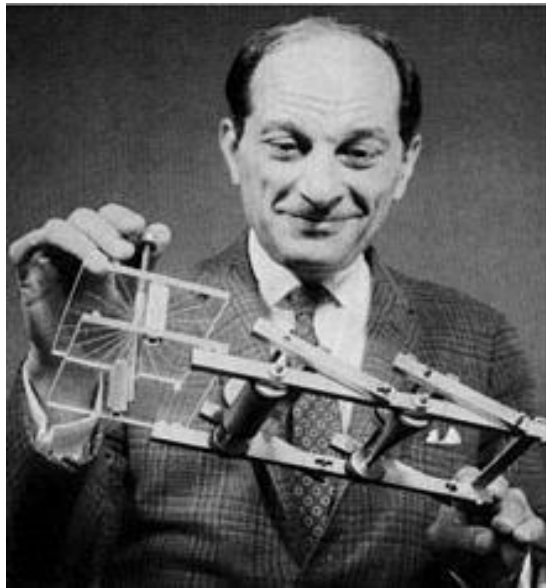


Processing

# INTRO

Le but de ce projet est de réaliser deux modèles 3D composées de morceaux (cubes, sphères, cylindres, etc..) dont chacun représente un nombre à partir de 1. Chaque morceaux est placé dans l'espace 3D par une technique que vous choisirez et implémenterez. Pour cela on utilisera Processing.

En mathématiques, la spirale d'Ulam, ou spirale de nombres premiers, est une méthode simple de représentation des nombres premiers qui révèle un schéma qui n'a jamais été entièrement expliqué. On doit sa découverte au mathématicien Ulam Stanislaw en 1963.



# But et Objectifs

Mon objectif est de faire :

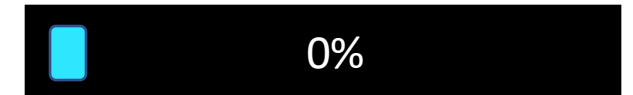
- Les formes



Ce que j'ai réalisé en terme de pourcentage



- L'animation



- Textures



- Polynome



# Répartition du travail

J'ai décidé de couper le travail à effectuer en 5 parties pour en avoir une sauvegarde stable et ne pas tout perdre

J'ai également réalisé des fichiers de test pour certain modules avant de les rajouter à une version final pour éviter les bug direct sur les fichiers à rendre



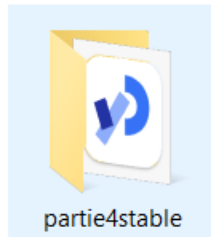
partie1stable



partie2stable



partie3stable



partie4stable



partie4stablebis



partie5stableFIN



J'ai également utilisé la plateforme Github afin de pouvoir travailler chez moi et de même lorsque nous avons TP en 336

<https://github.com/Chlooow/ProjetUlamIGSD>

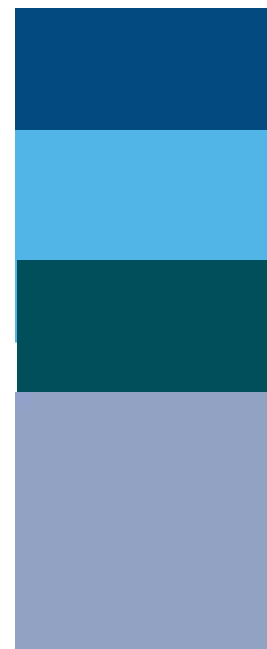
# Etape 1 : somme des diviseurs

En arithmétique, la fonction somme des diviseurs est la fonction arithmétique qui, à un entier naturel non nul, associe la somme de ses diviseurs positifs, souvent notée  $\sigma$ .

```
// ===== Somme des diviseur =====  
  
int sommeDesDiviseur(int n) {  
    int somme = 0;  
    for (int i = 1; i <= n ; i++) {  
        if (n % i == 0) {  
            somme += i;  
        }  
    }  
    return somme;  
}  
  
// =====
```

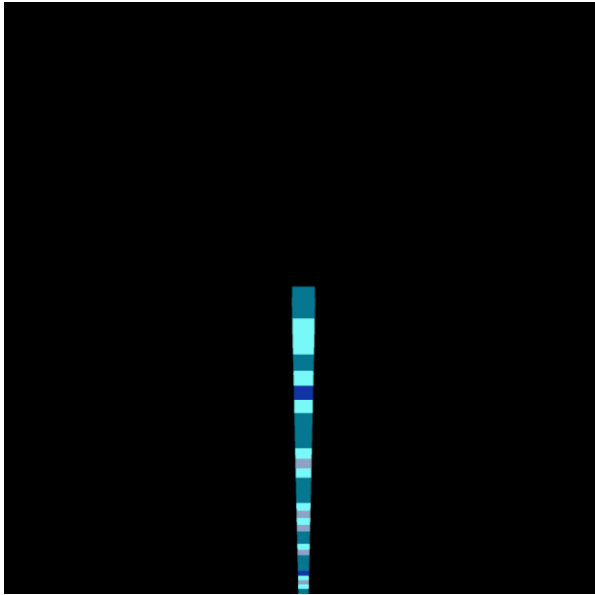
Code couleur :

- Nombre parfait
- Nombre premier
- Nombre Deficient
- Nombre abondant

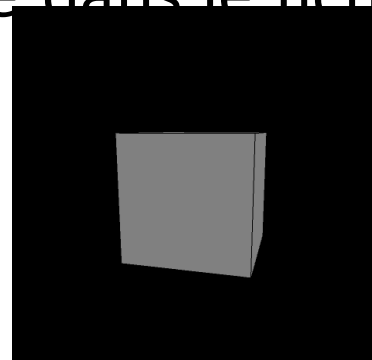
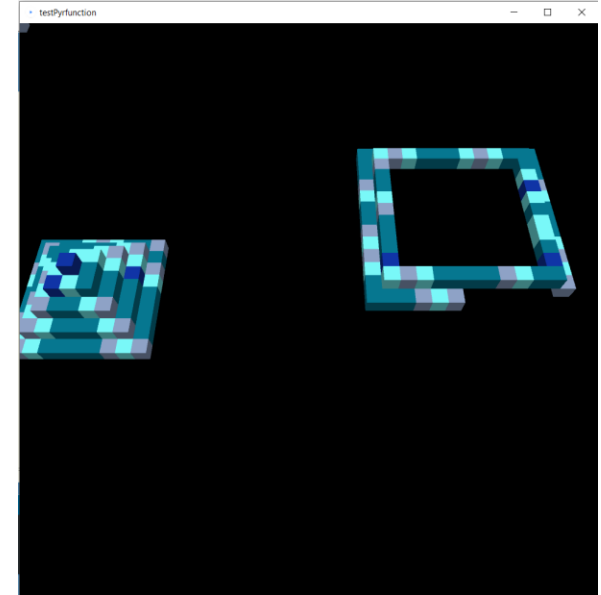


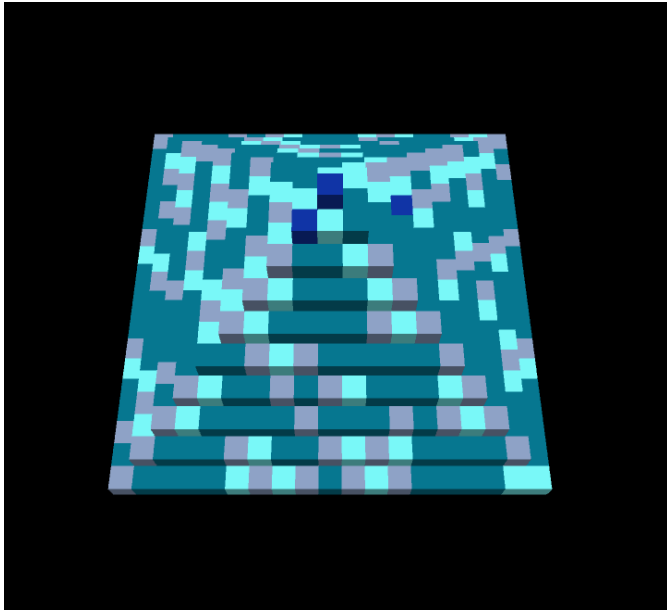


# Etape 3 : Camera + Pyramide



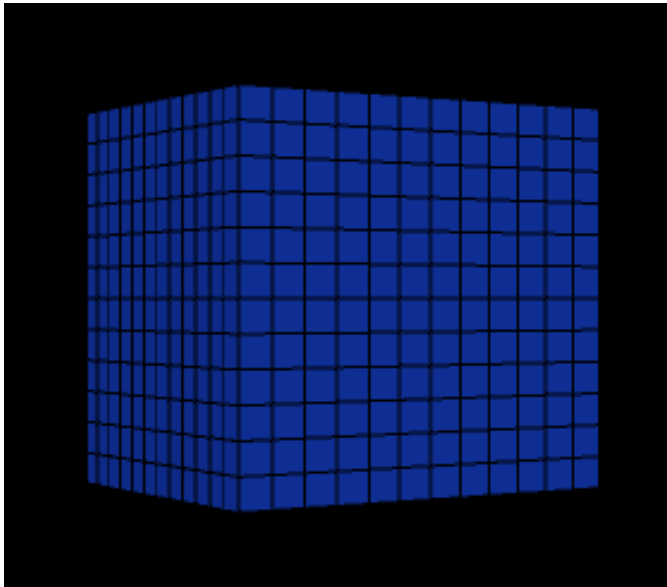
Lorsque j'ai essayé de faire la caméra, elle tournait simplement autour du sommet de la pyramide et l'a disloquait en tournant je n'ai malheureusement pas réussi à régler le problème, mais la compétence a été acquise dans le fichier test





## Etape 4 : Pyramide

J'ai décidé de faire une pyramide car la forme est plutôt évidente. J'ai également essayé de faire un cube en empilant des box mais le modèle de spiral pour un cube semble difficile alors j'ai continué avec la pyramide pour ne pas perdre de temps.

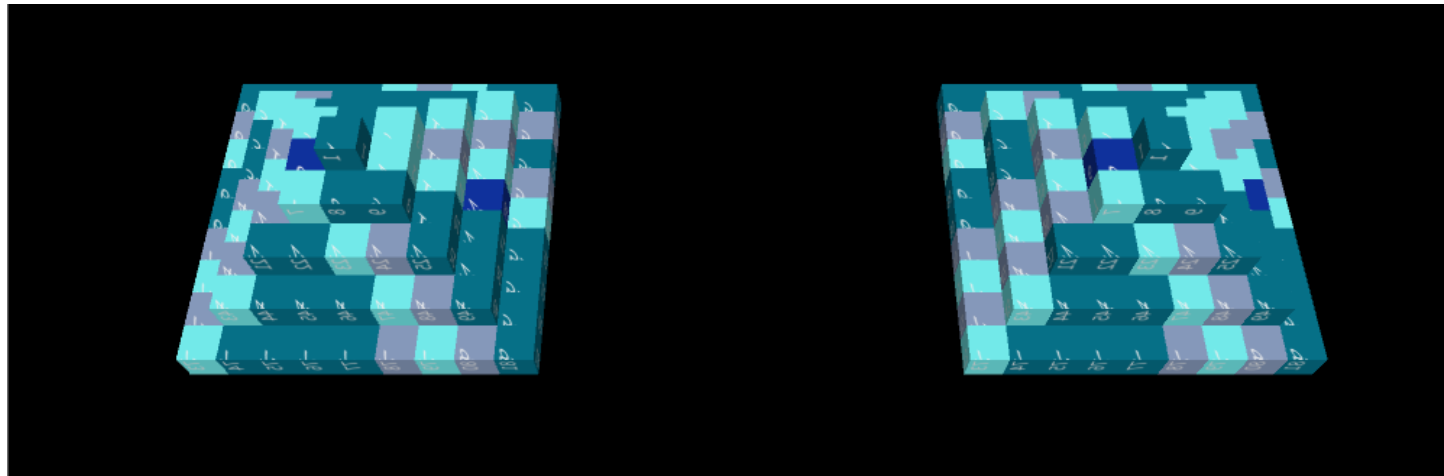




# Etape 5 : Textures

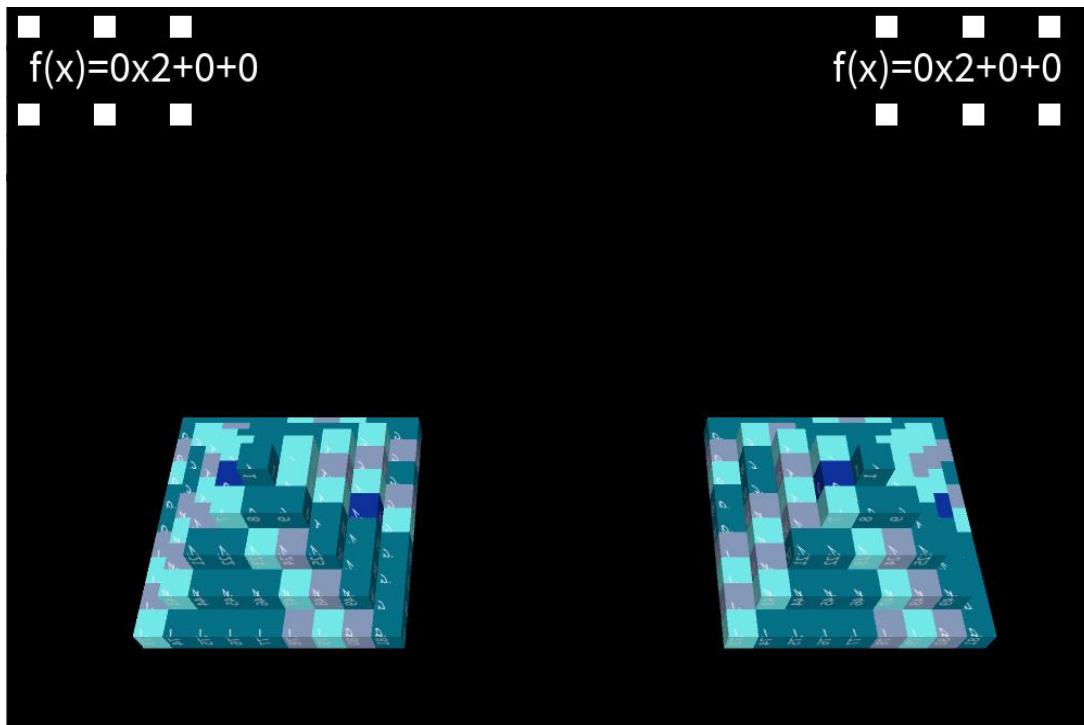
Les textures furent également ce qui m'a pris le plus de temps, et finalement j'ai partiellement réussi à les afficher.

Je me suis inspiré du TP sur le labyrinthe et de ressources retrouvés sur internet notamment dans la documentation de processing



# Etape 6 : Polynomes

J'ai a peu près compris en quoi consistait le polynôme mais je me suis arrêté à l'étape de cliquage de bouton. Malheureusement lorsque je clic j'ai une erreur que je ne saurais régler. Lorsque je clique sur un bouton, la fenêtre se ferme puis affiche le message

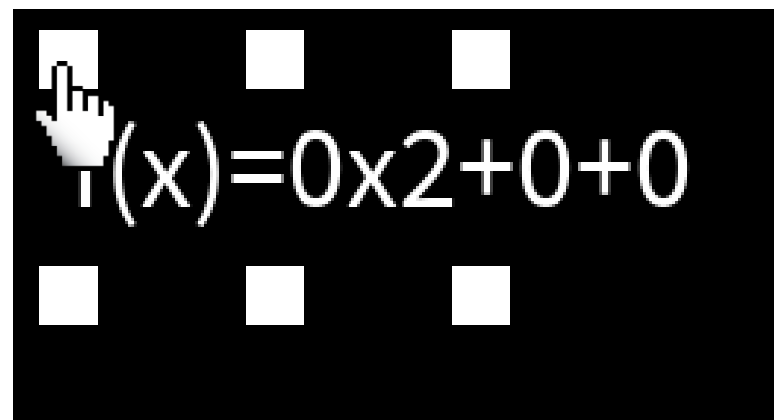


L'enregistrement est terminé.

# See problematic frame for where to report the bug.

#

Could not run the sketch (Target VM failed to initialize).  
For more information, read Help ? Troubleshooting.



# Les Etapes manquantes et problèmes rencontrés

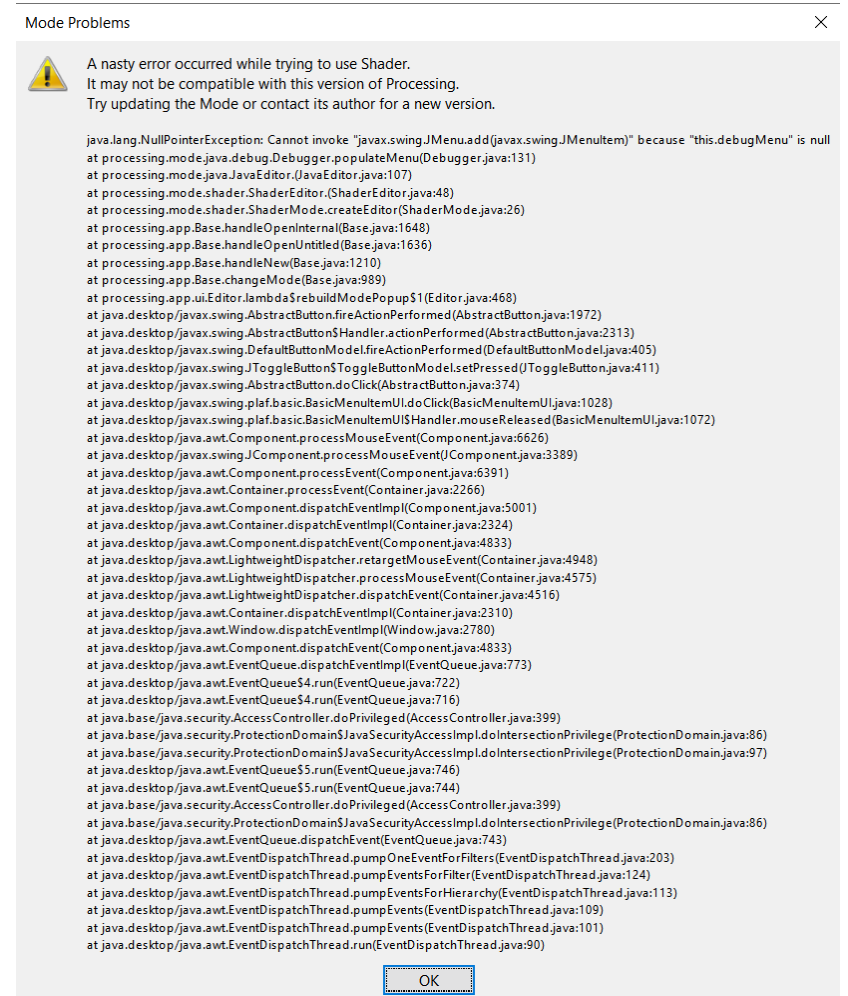
J'ai essayé de faire les shaders mais j'ai rencontré un problème lors du passage à ce mode. Par manque de temps et de ressources je n'ai pas pu installé une version antérieure ou tester ce dernier sur les ordinateurs de la fac. Mais l'objectif aurait été de s'inspirer des TP utilisant les shaders

```
tp5 firstFragmentShader_glsl myVertexShader_glsl ▼
PShader firstProgrammeShader;
PShader myProgrammeShader;

void setup(){
    firstProgrammeShader = loadShader("firstFragmentShader.glsl", "firstVertexShader.glsl");
    myProgrammeShader = loadShader("myFragmentShader.glsl", "myVertexShader.glsl");
    size(1000, 1000, P3D);
}

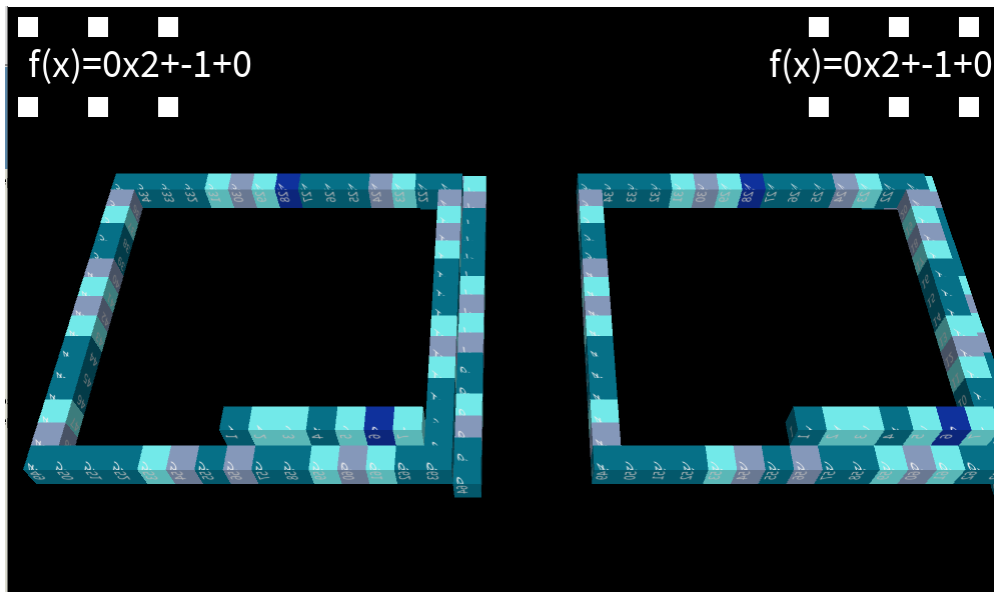
void draw(){
    pointLight(255, 255, 255, mouseX, mouseY, 400);
    noStroke();
    fill(192, 128, 64);

    translate(width/2, height/2);
    shader(myProgrammeShader);
    sphereDetail(30);
    sphere(300);
}
```



# J'ai trouvé mon erreur !

```
// ===== Void Setup =====  
  
void setup() {  
  size(1000, 1000, P3D);  
  strokeWeight(0);  
  noLoop();  
}
```



La caméra, et l'erreur des boutons est dû au `noLoop()`; Mon code est fait que pour s'exécuter 1 seule fois, du coup lorsque je l'enlève, il disloque ma pyramide car il n'est pas habitué à être appelé plusieurs fois.

Ce qu'il aurait fallu faire, c'est que j'enlève le `no loop` dès l'étape 1 ( car il était là pour tester la somme des diviseurs et la spirale).

## `noLoop()`

Stops Processing from continuously executing the code within `draw()`. If `loop()` is called, the code in `draw()` begin to run continuously again. If using `noLoop()` in `setup()`, it should be the last line inside the block.

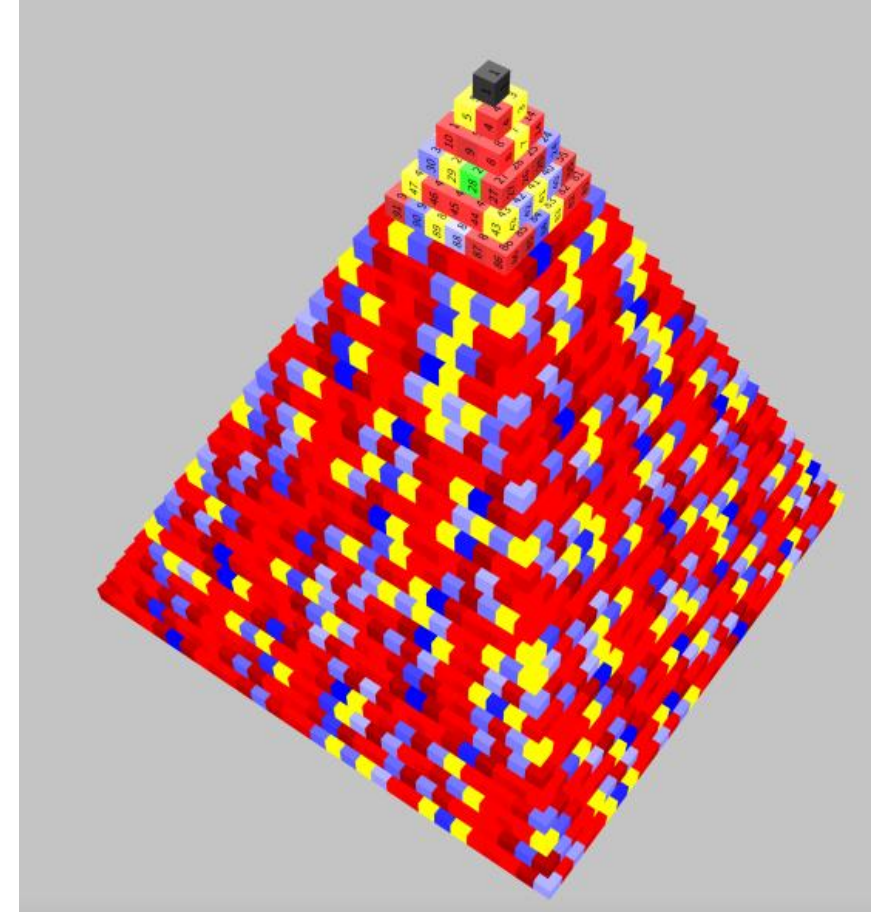
When `noLoop()` is used, it's not possible to manipulate or access the screen inside event handling functions such as `mousePressed()` or `keyPressed()`. Instead, use those functions to call `redraw()` or `loop()`, which will run `draw()`, which can update the screen properly. This means that when `noLoop()` has been called, no drawing can happen, and functions like `saveFrame()` or `loadPixels()` may not be used.

Note that if the sketch is resized, `redraw()` will be called to update the sketch, even after `noLoop()` has been specified. Otherwise, the sketch would enter an odd state until `loop()` was called.

# Conclusion

Malgré plusieurs parties manquantes, j'ai tout de même réussi à satisfaire mes objectifs du début. Je pense qu'avec plus de temps et de ressources, j'aurais pu terminer le projet dans son entièreté. Il manquait peut-être un peu plus de guide dans le sujet qui était un peu flou.

J'ai eu du mal à poser les bonnes questions, mais j'ai eu de l'aide de certains de mes camarades afin de comprendre le but du projet qui était de retranscrire ce que l'on a appris au cours des TP/Cours et l'utilisation de Processing. Ce projet fut amusant et ce serait intéressant de le continuer ou de l'améliorer.



# Sources

- CodinTrain pour la spiral  
<https://youtu.be/a35KWEjRvc0>
- Documentation Processing
- Forum Processing
- Google...etc.
- TPs

