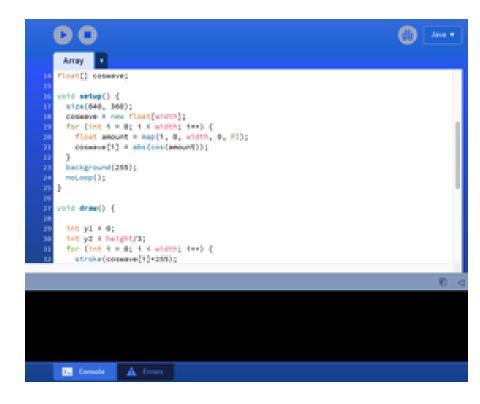
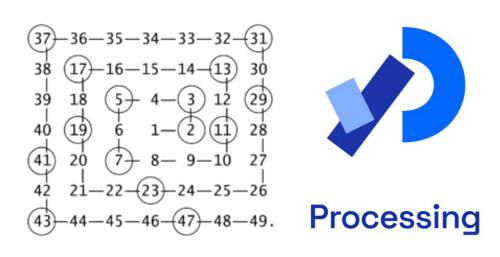
Projet info223 Spirale d'Ulam en 3 dimensions



Chloe Makoundou

L2 info groupe 1 - Frederic Vernier

https://github.com/Chlooow/ProjetUlamI GSD

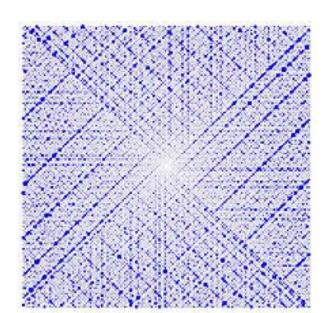


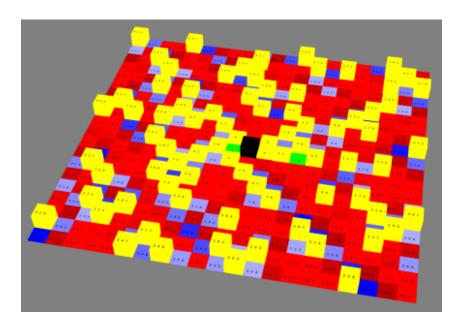
INTRO

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
L'animation
Textures
Polynome
20%

Ce que j'ai réalisé en terme de pourcentage

Répartition du travail

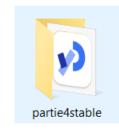
J'ai décidé de couper le travail à effectuer en 5 parties pour en avoir une sauvegarde stable et ne pas tout perdre

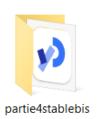
J'ai également réalisé des fichiers de test pour certain modules avant de les rajouter à une version final pour éviter les bug direct sur les fichiers à rendre













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

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 σ .

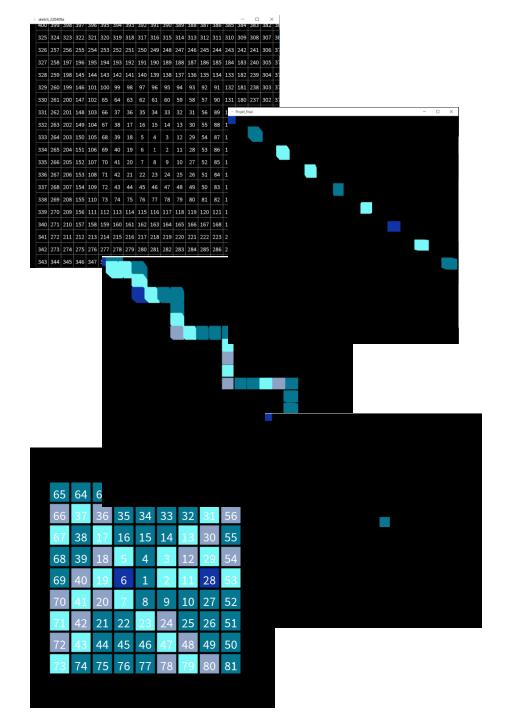
Code couleur:

- Nombre parfait

- Nombre premier

- Nombre Deficient

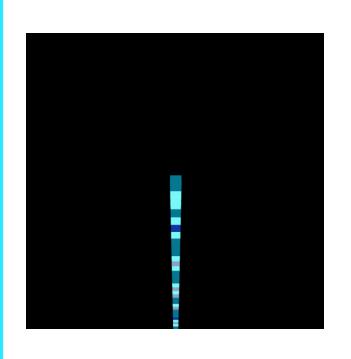
- Nombre abondant



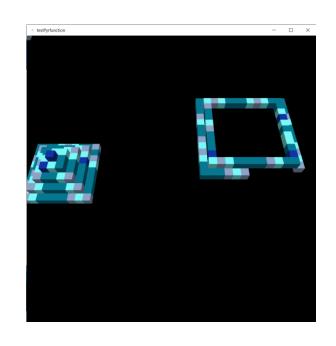
Etape 2 : Spiral d'Ulam

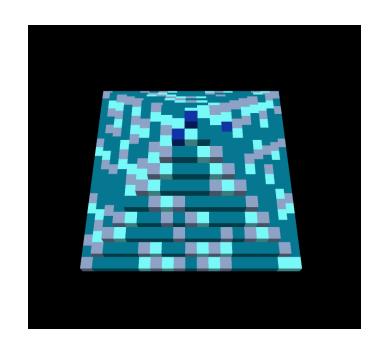
Cette partis à été l'une des plus longue car il fallait faire une spiral. J'ai perdu beaucoup de temps à comprendre son fonctionnement mais j'ai finis par comprendre grace à des videos sur youtube et des essais. Finalement j'en suis venu à ce resultat

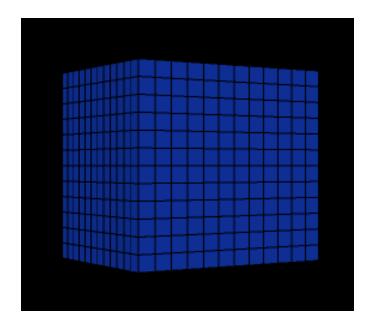
Etape 3: Camera + Pyramide



Lorsque j'ai essayé de faire la caméra, elle tournait simplement autour 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







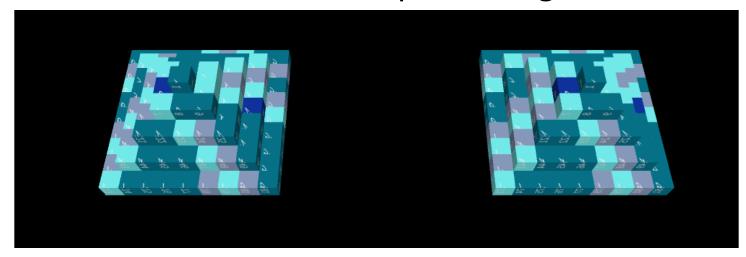
Etape 4 : Pyramide

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

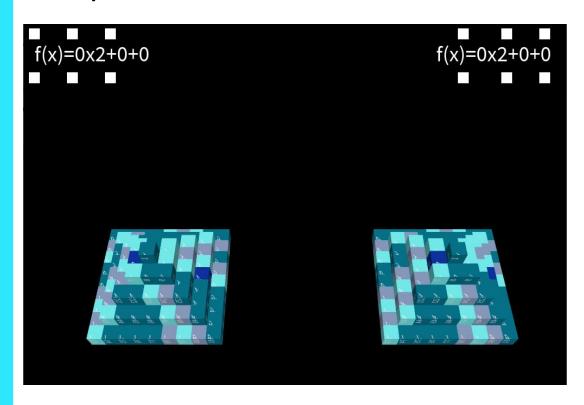
Les textures furent également ce qui m'a pris le plus de temps, et finalement j'ai partiellement réussis a les afficher.

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

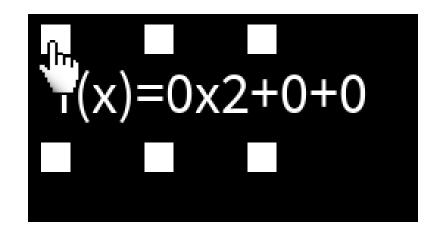


Etape 6 : Polynomes

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







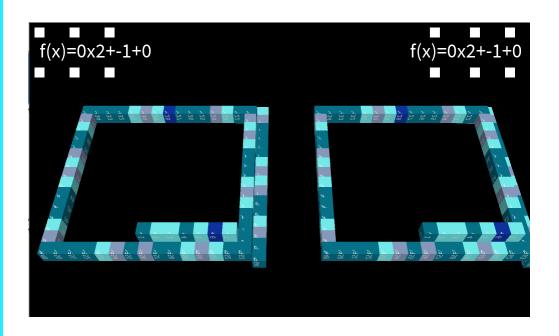
Les Etapes manquantes et problèmes rencontrés

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 antérieur ou tester ce dernier sur les ordinateurs de la fac. Mais l'objectif aurait été de s'inspirer des TP utilisant les shaders



A nasty error occurred while trying to use Shader. It may not be compatible with this version of Processing. Try updating the Mode or contact its author for a new version. java.lang.NullPointerException: Cannot invoke "javax.swing.JMenu.add(javax.swing.JMenultem)" because "this.debugMenu" is null at processing.mode.java.debug.Debugger.populateMenu(Debugger.java:131) at processing.mode.java.JavaEditor.(JavaEditor.java:107) at processing.mode.shader.ShaderEditor.(ShaderEditor.java:48) at processing.mode.shader.ShaderMode.createEditor(ShaderMode.java:26) at processing.app.Base.handleOpenInternal(Base.java:1648) at processing.app.Base.handleOpenUntitled(Base.iava:1636) at processing.app.Base.handleNew(Base.java:1210) at processing.app.Base.changeMode(Base.java:989) at processing.app.ui.Editor.lambda\$rebuildModePopup\$1(Editor.java:468) at java.desktop/javax.swing.AbstractButton.fireActionPerformed(AbstractButton.java:1972) at java.desktop/javax.swing.AbstractButton\$Handler.actionPerformed(AbstractButton.java:2313) at java.desktop/javax.swing.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:405) at java.desktop/javax.swing.JToggleButton\$ToggleButtonModel.setPressed(JToggleButton.java:411) at java.desktop/javax.swing.AbstractButton.doClick(AbstractButton.java:374) at iava.desktop/iavax.swing.plaf.basic.BasicMenultemUl.doClick(BasicMenultemUl.iava:1028) at java.desktop/javax.swing.plaf.basic.BasicMenultemUI\$Handler.mouseReleased(BasicMenultemUI.java:1072) at java.desktop/java.awt.Component.processMouseEvent(Component.java:6626) at java.desktop/javax.swing.JComponent.processMouseEvent(JComponent.java:3389) at java.desktop/java.awt.Component.processEvent(Component.java:6391) at java.desktop/java.awt.Container.processEvent(Container.java:2266) at java.desktop/java.awt.Component.dispatchEventImpl(Component.java:5001) at java.desktop/java.awt.Container.dispatchEventImpl(Container.java:2324) at java.desktop/java.awt.Component.dispatchEvent(Component.java:4833) at java.desktop/java.awt.LightweightDispatcher.retargetMouseEvent(Container.java:4948) at java.desktop/java.awt.LightweightDispatcher.processMouseEvent(Container.java:4575) at java.desktop/java.awt.LightweightDispatcher.dispatchEvent(Container.java:4516) at java.desktop/java.awt.Container.dispatchEventImpl(Container.java:2310) at java.desktop/java.awt.Window.dispatchEventImpl(Window.java:2780) at java.desktop/java.awt.Component.dispatchEvent(Component.java:4833) at java.desktop/java.awt.EventQueue.dispatchEventImpl(EventQueue.java:773) at java.desktop/java.awt.EventQueue\$4.run(EventQueue.java:722) at iava.desktop/iava.awt.EventQueue\$4.run(EventQueue.iava:716) at java.base/java.security.AccessController.doPrivileged(AccessController.java:399) $at\ java.base/java.security.ProtectionDomainSJavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:86)$ at java.base/java.security.ProtectionDomain\$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:97) at java.desktop/java.awt.EventQueue\$5.run(EventQueue.java:746) at java.desktop/java.awt.EventQueue\$5.run(EventQueue.java:744) at java.base/java.security.AccessController.doPrivileged(AccessController.java:399) at java.base/java.security.ProtectionDomain\$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:86) at java.desktop/java.awt.EventQueue.dispatchEvent(EventQueue.java:743) at java.desktop/java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:203) at java.desktop/java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:124) at iava.desktop/iava.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.iava:113) at java.desktop/java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:109) at java.desktop/java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:101) at java.desktop/java.awt.EventDispatchThread.run(EventDispatchThread.java:90)

J'ai trouvé mon erreur!



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 a 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 la 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.

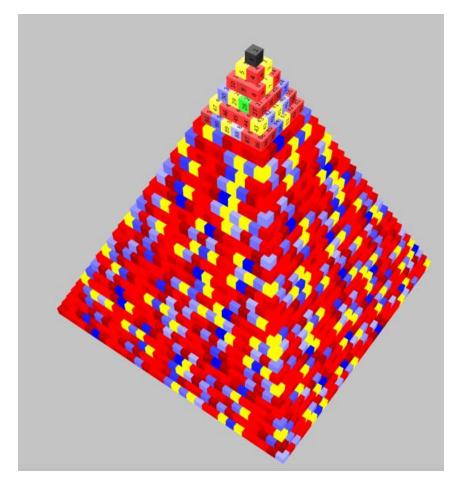
When noLoop() is used, it's not possible to manipulate or access the screen inside event handling functions such as mousePressed() or keyPressed(). Instead, use those functions to call redraw() or loop(), which will run draw(), which can update the screen properly. This means that when noLoop() has been 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 réussis 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 serais interessant de le continuer ou de l'améliorer.



Sources

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

