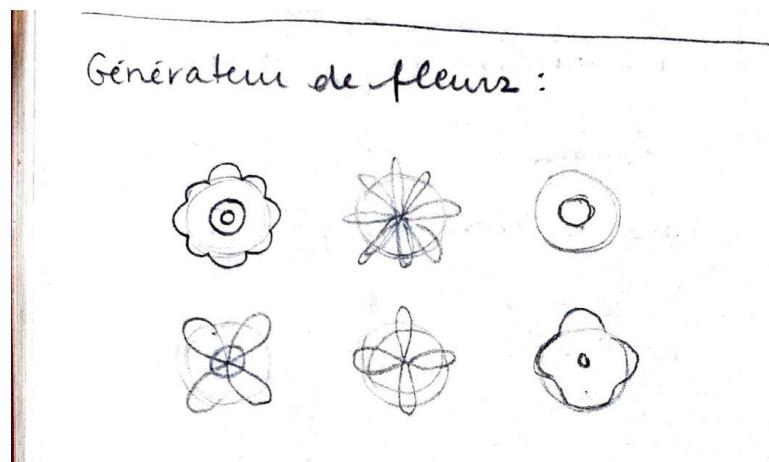


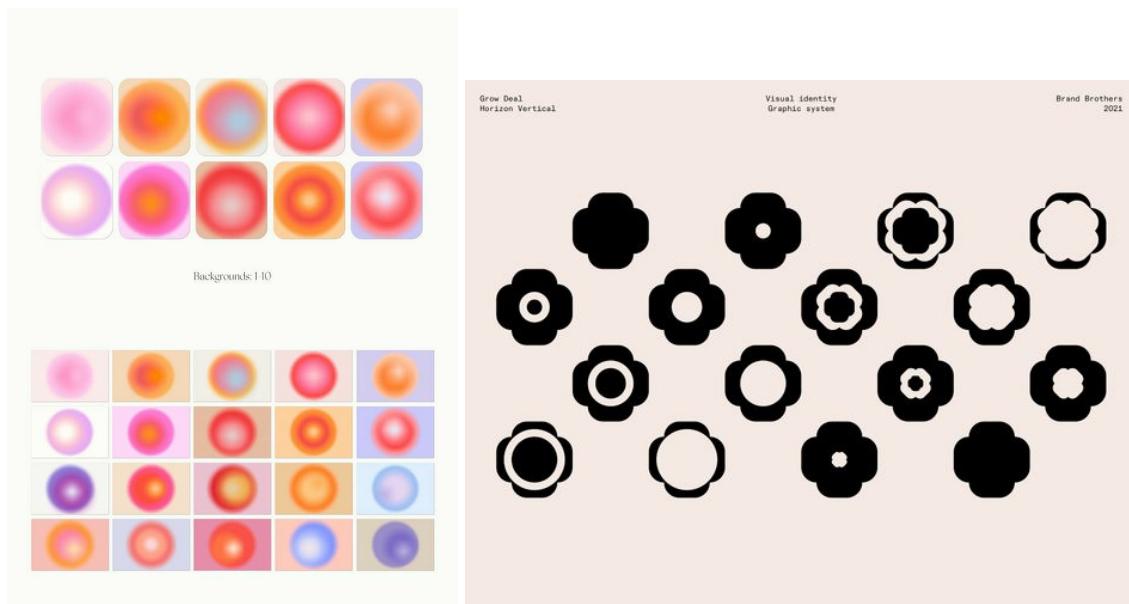
## Objectif d'exploration

Pour l'exploration no1, j'ai voulu approfondir la technique vue en classe de la mosaïque. J'aimais l'idée de répéter à l'infini plusieurs variantes d'un même objet, sous le format d'une grille.

Mon intention a été de créer un générateur de fleurs, soit un ensemble aléatoire de différentes formes géométriques évoquant des pétales de fleur. Voici mon esquisse de base en imaginant ce petit système :



Voici mes inspirations :



## Réflexion sur les étapes à entreprendre

1. déterminer les composantes d'une seule fleur et ses paramètres qui seront variables.
2. créer des palettes de couleurs aléatoires
3. composer le tout en une grille

## Chemin pour s'y rendre

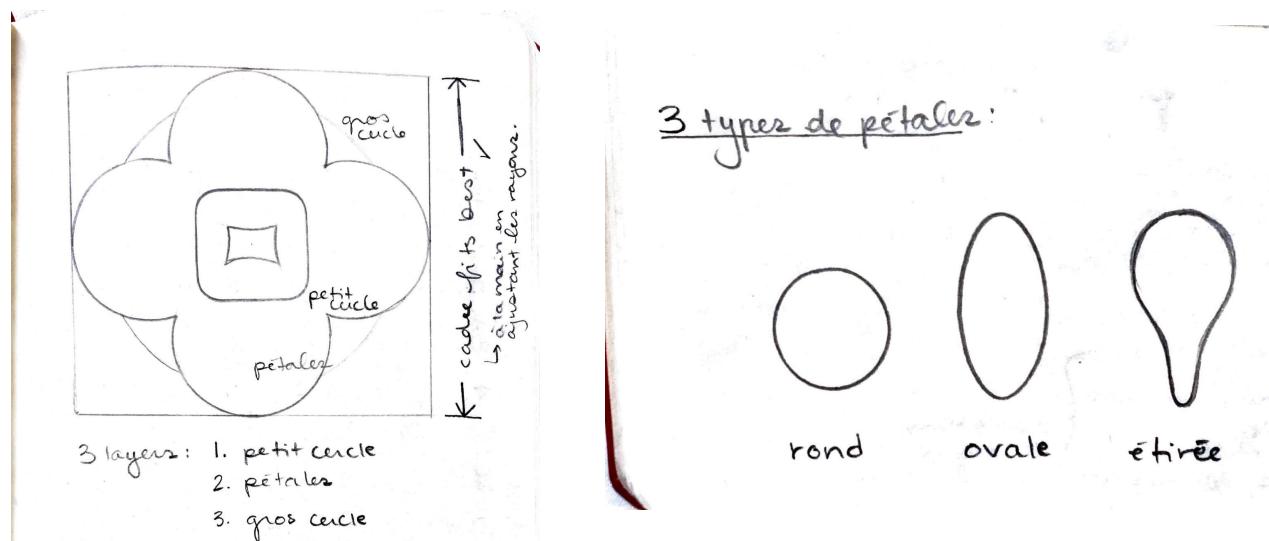
### 1. La fleur

#### Composition de base

J'ai décidé de commencer le tout en créant la fleur de base.

En jouant avec des cercles, des sphères, des polygones et géométries, j'ai exploré ce qu'étaient mes possibilités graphiques pour dessiner une fleur géométrique.

Mes fleurs sont travaillées en 3 parties/3 layers, soit une série de pétales, un petit rectangle arroondi au centre, puis un gros cercle en arrière.

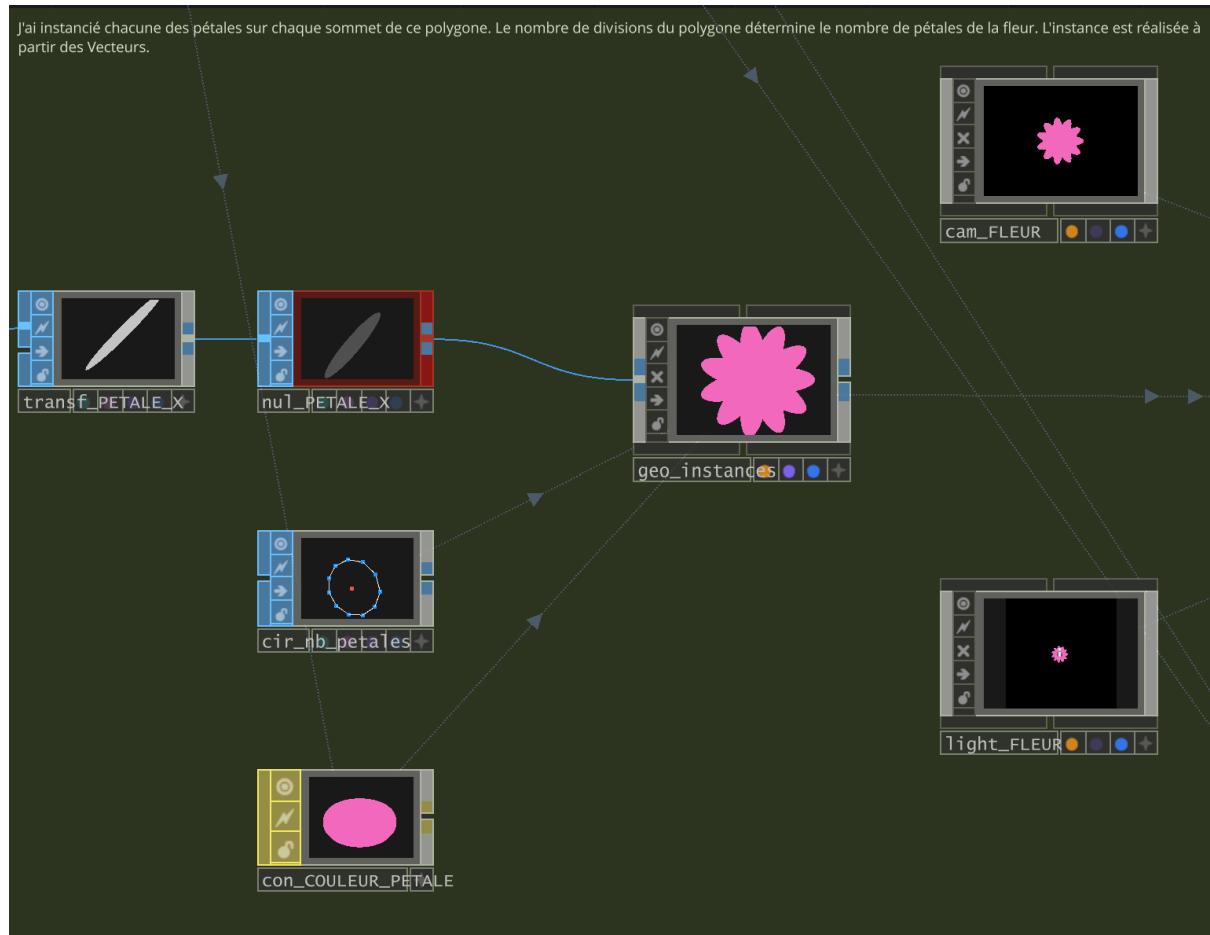


Ensuite pour diversifier le tout, j'ai composé 3 types de pétales différentes, qui seront choisies aléatoirement grâce à un opérateur switch.



## Pétales

Le nombre de pétales sera déterminé par le nombre de divisions d'un cercle, puis leur disposition sera exécutée par une instanciation afin de les répéter à la position des sommets du polygone créé.

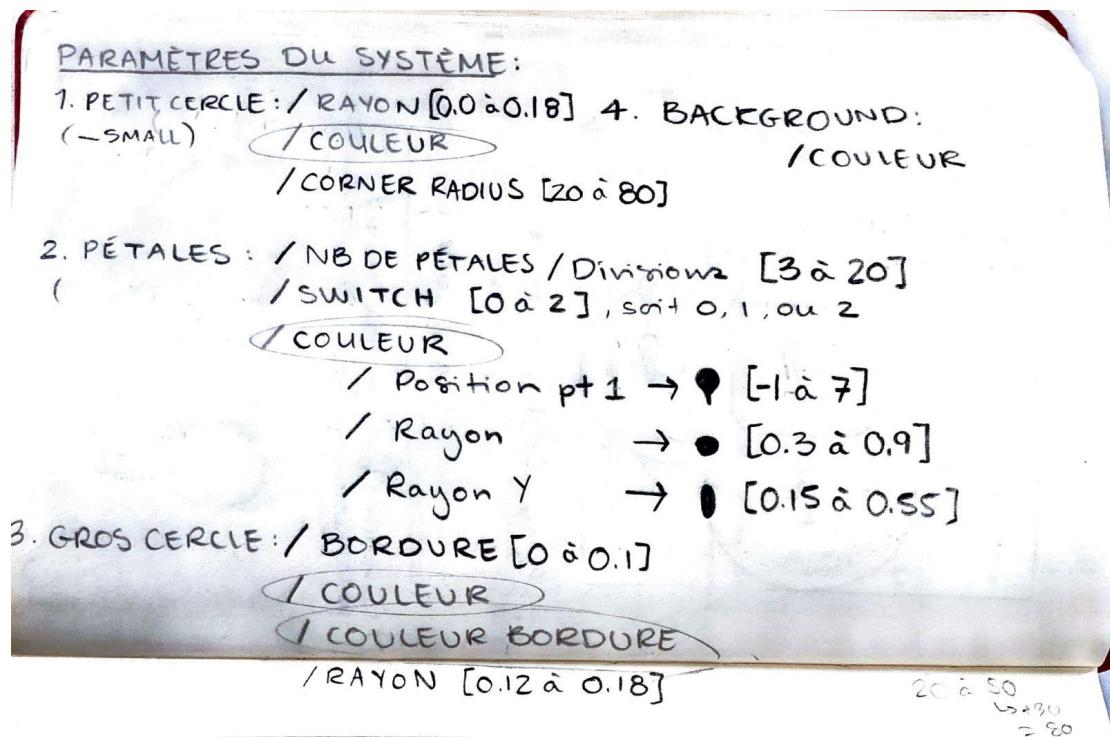


Afin que les pétales soient disposées en étoiles, l'instanciation s'est faite par rapport aux vecteurs du polygone `cir_nb_petales`. J'ai effectué quelques rotations afin que les pétales soient bien disposées et vues de face.

## Paramètres variables

Afin de créer des variantes de fleurs, je me suis questionnée sur quels paramètres pourraient varier afin de créer des itérations nombreuses mais aussi cohérentes et esthétiques.

Voici les paramètres que j'ai décidé de faire varier, ainsi que leur range respectif minimum et maximum afin que la fleur ne devienne pas hors de proportion :



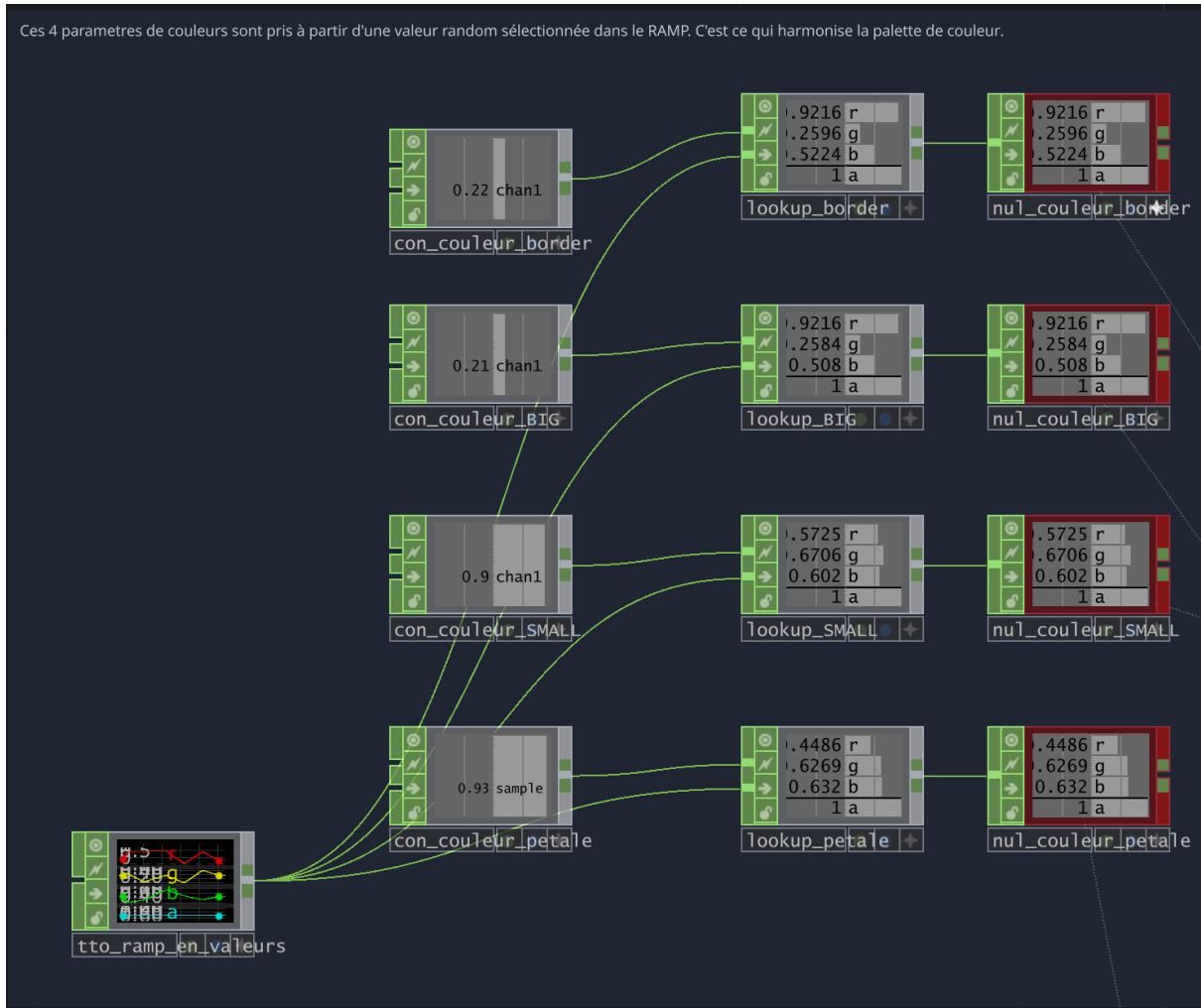
Enfin, voici le code du Chop execute qui fait modifier ces paramètres :

```
21
22 def onvaluechange(channel, sampleIndex, val, prev):
23
24     import random
25
26     op('cir_nb_petals').par.divs = random.randrange(3., 20.)
27     op('rct_SMALL').par.cornerradius = random.randrange(20., 80.)
28     op('switch').par.input = random.randrange(0., 2.)
29     op('pnt_no1').par.tx = random.randrange(-100., 700.)/100
30     op('cir_oval_1').par.rady = random.randrange(15., 55.)/100
31     op('cir_BIG').par.borderwidth = random.randrange(0., 100)/1000
32     op('cir_BIG').par.radiusy = op('cir_BIG').par.radiusx = random.randrange(0., 180.)/1000
33     op('cir_3').par.radx = op('cir_3').par.rady = random.randrange(30., 90.)/100
34     op('rct_SMALL').par.sizex = op('rct_SMALL').par.sizey = random.randrange(0., 180.)/1000
35     op('con_couleur_petale').par.value0 = random.randrange(0., 100.)/100
36     op('con_couleur_SMALL').par.value0 = random.randrange(0., 100.)/100
37     op('con_couleur_BIG').par.value0 = random.randrange(0., 100.)/100
38     op('con_couleur_border').par.value0 = random.randrange(0., 100.)/100
39
40     return
41
```

## Couleurs des fleurs

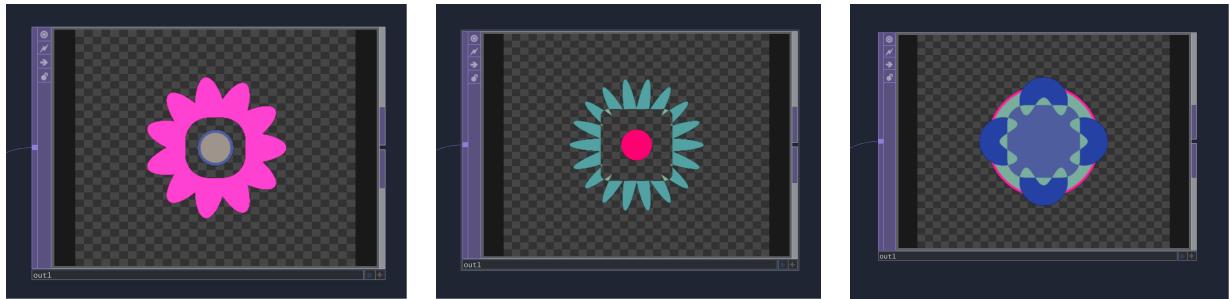
Les couleurs des fleurs font toutes parties de la même gamme de couleur puisqu'elles sont sélectionnées à travers un RAMP.

Les paramètres LOOKUP permettent de prendre une couleur du dégradé après avoir reçu une constante générée aléatoirement entre 0 et 1.

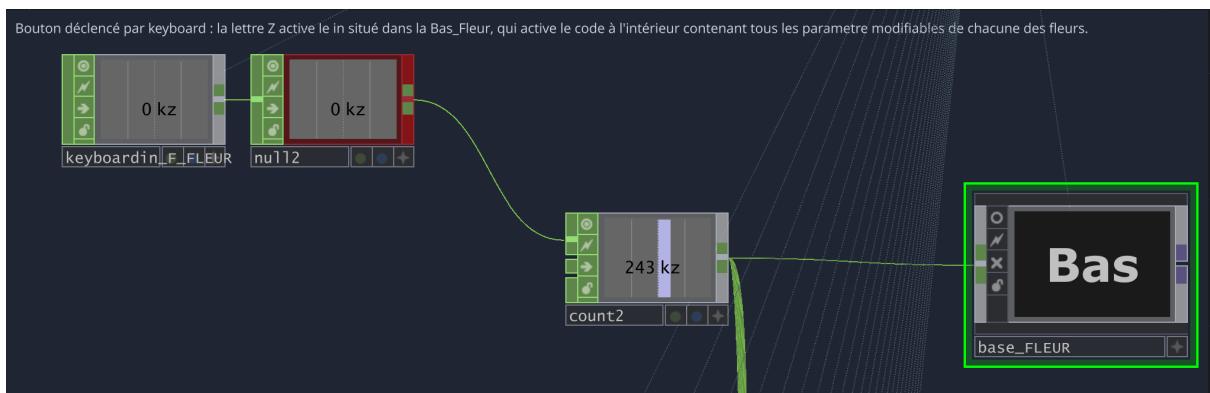


## Résultat de la fleur

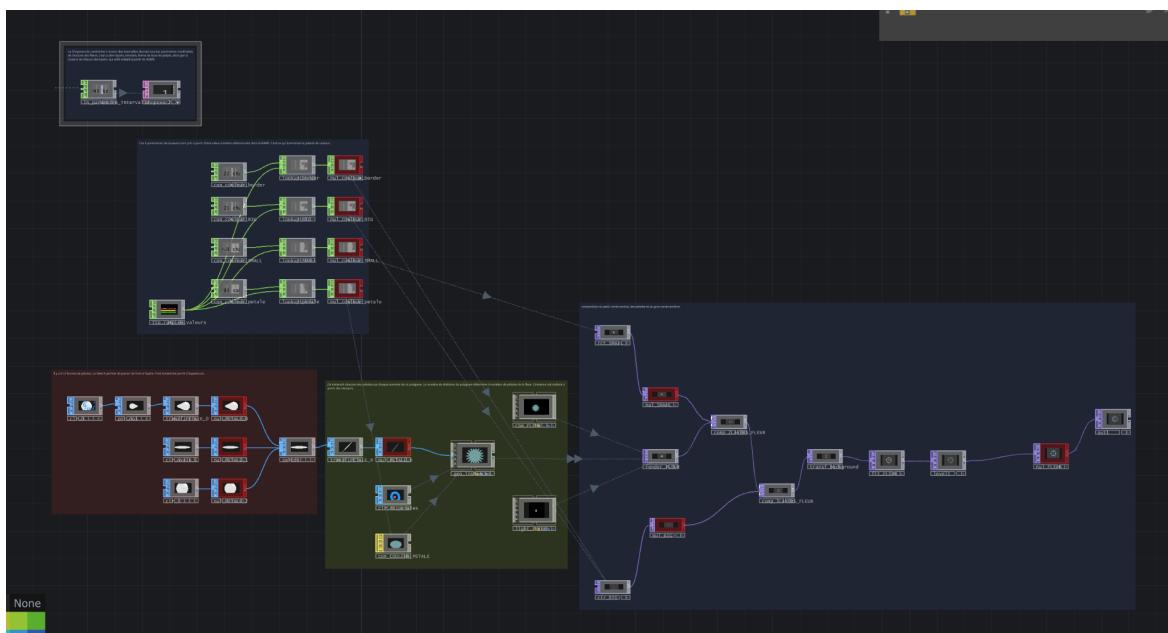
Voici le résultat de la fleur générée à trois reprises. J'aime vraiment la surprise que ça procure à chaque fois, ça devient addictif de vouloir savoir quelle forme aura la prochaine ! J'avais imaginé un peu de quoi chaque fleur avait l'air en testant les range des paramètres, mais l'addition des 3 formes est ce qui rend le tout surprenant à chaque fois. La première image ici représente quelque chose auquel je m'attendais. La deuxième est une variante funky, dans le sens ou je m'y attendais mais il y a une petite surprise. Puis la troisième je ne l'aurais pas imaginé, elle est plutôt inattendue !



Pour modifier la forme de la fleur, il faut appuyer sur la touche Z.



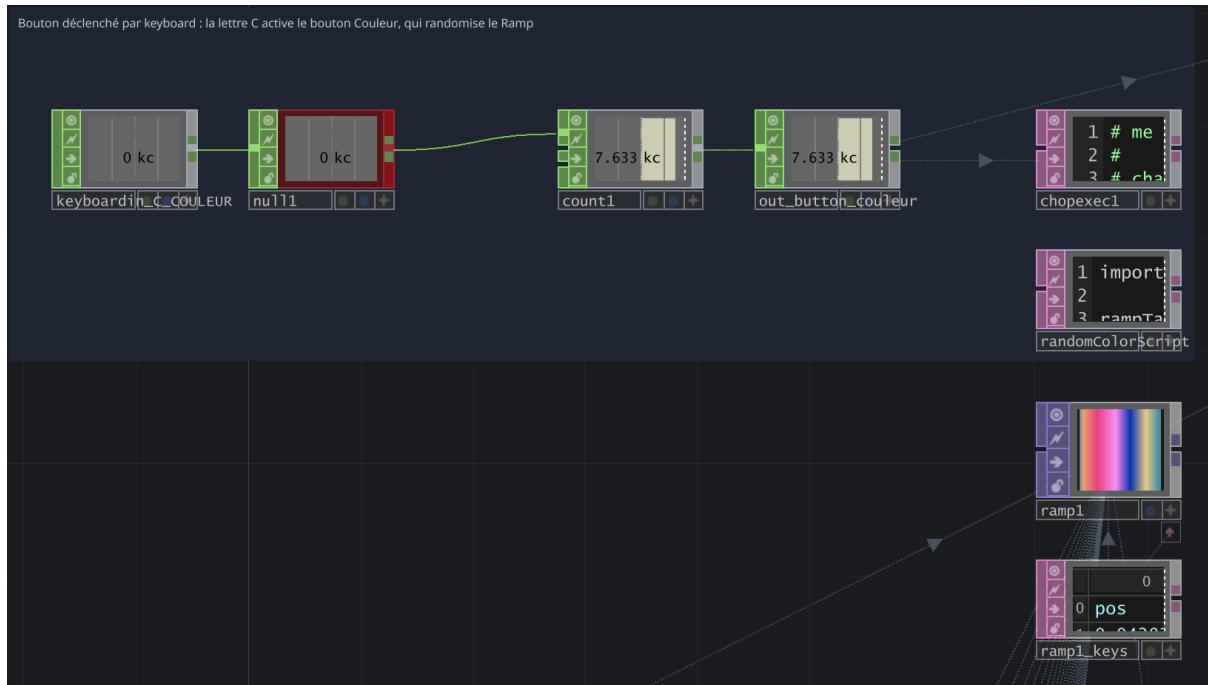
Voici donc le réseau final qui se trouve dans la Base Comp Fleur :



## 2. Palette de couleur aléatoire

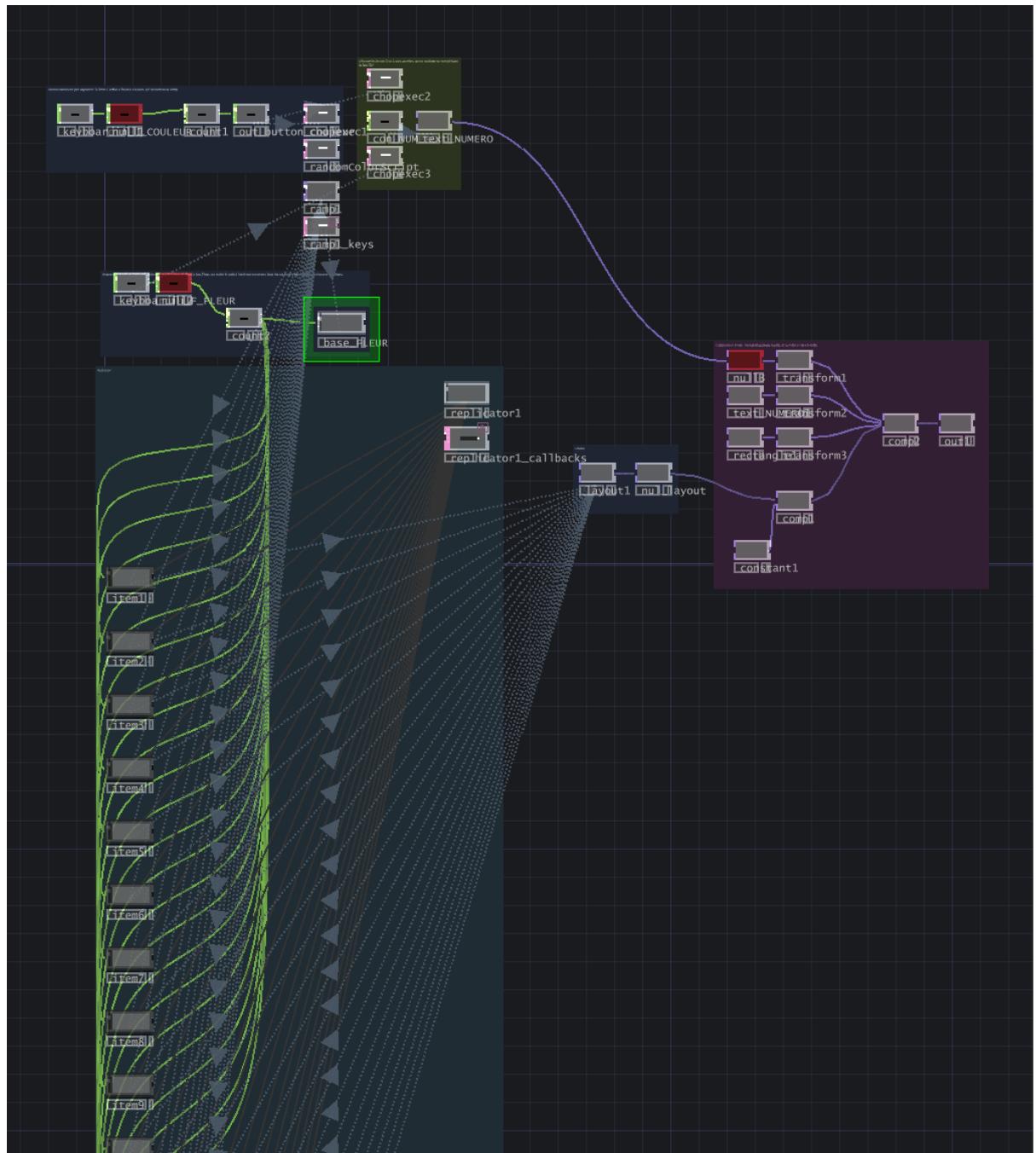
Comme mentionné plus tôt, les couleurs des fleurs sont des échantillons d'un dégradé. Ce dégradé est généré de manière aléatoire. Grâce à l'opérateur Keyboard in, quand la touche C est appuyée, le code s'exécute afin de générer une nouvelle table de valeur qui représente le dégradé.

Cette opération se déroule à l'extérieur de la Base\_Comp\_Fleur car sinon le code s'exécutera à chaque fois que la Base est répétée. J'ai passé beaucoup de temps à tenter de résoudre ce problème.



### **3. Grille**

Pour répliquer les fleurs, j'ai utilisé un replicator, ainsi qu'un Layout Top afin de disposer les fleurs dans une composition en grille. (comme dans l'exemple vu en classe de Warhol)

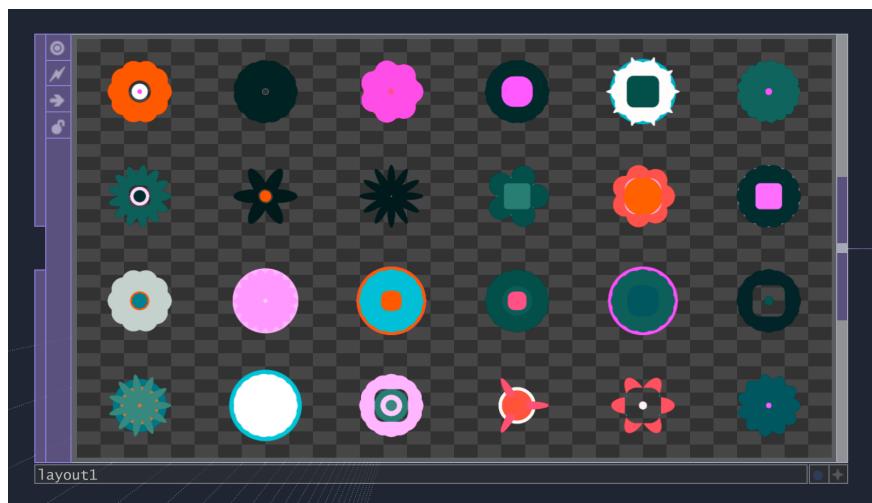


#### 4. Résultat

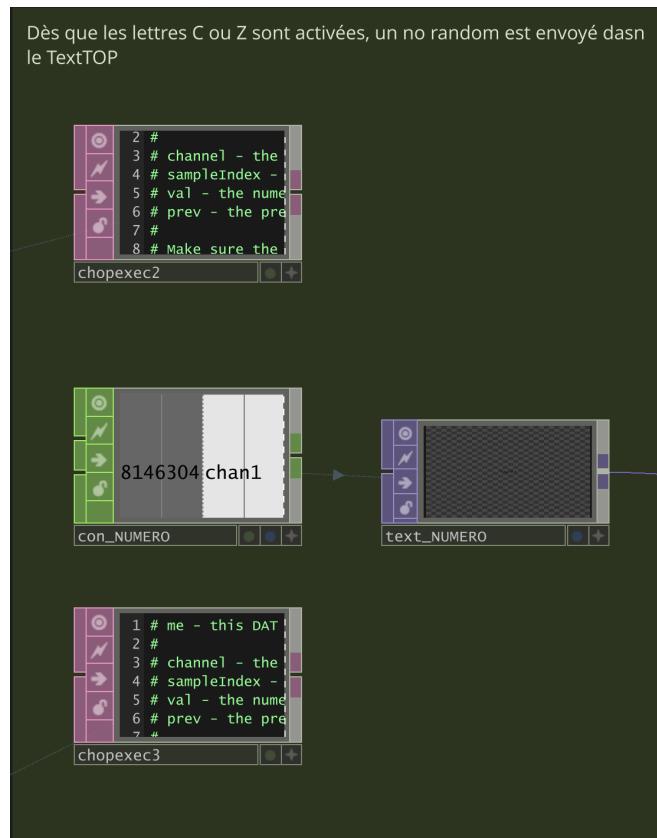
Voici le résultat de la grille de fleur : mêmes formes, différentes palettes de couleurs.



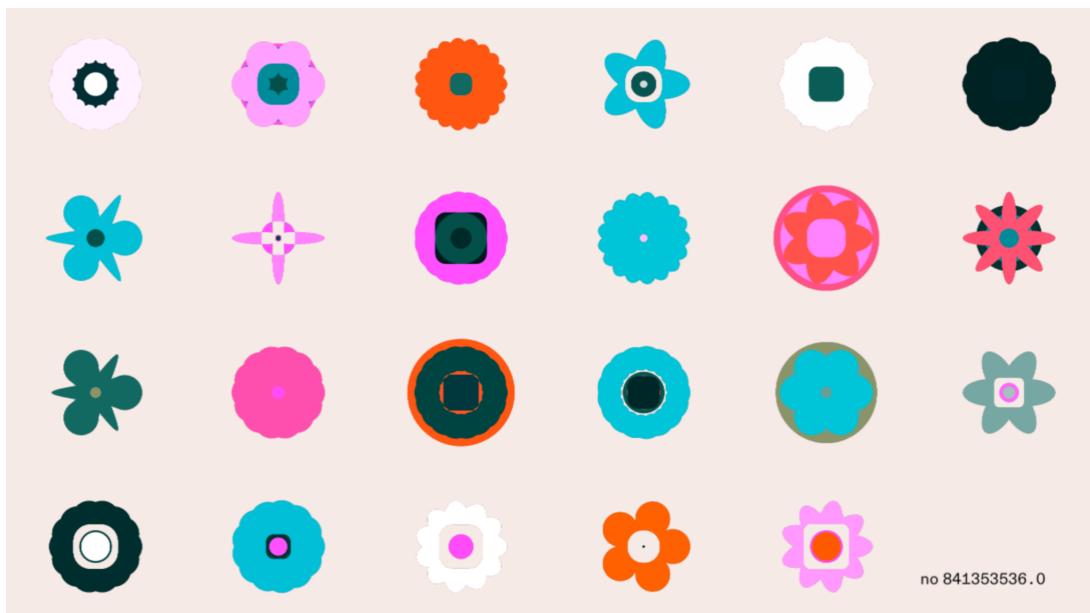
Et enfin le résultat de la grille pour une même palette, mais des formes différentes :



Dans la composition finale, j'ai ajouté au bas à droite un no aléatoire jusqu'à 999 999 999 afin de simuler un no de série du « jardin » ainsi créé.



Voilà ! :)



## **Conclusion :**

En conclusion, cette exploration m'a permis d'intégrer les techniques d'instanciation, du rendu d'une géométrie, du replicator, du layout Top, de la génération aléatoire de Ramp et de paramètres, ainsi que de l'utilisation du keyboard.

Mes plus grands défis ont été l'instanciation en étoiles (rotation) des pétales, ainsi que les codes des Chop Execute. Pour mes ranges, je voulais qu'ils soient entre des nombres décimaux. Ça a pris un petit moment avant que je comprenne que random.randrange prenait que des nombres entiers. Pour résoudre le problème, j'ai multiplié mes bornes d'intervalle par 100, pour ensuite diviser le nombre aléatoire donné par 100. Ensuite les couleurs m'ont apporté leur lot de défis aussi. Je ne comprenais pas pourquoi les Ramp étaient différents pour chaque réplique de Base\_Fleur alors que le bouton pour changer la palette se trouvait à l'extérieur des bases. J'ai fini par comprendre que le dégradé devait en sortir lui aussi.

Finalement, c'est la première fois que j'étais confronté à ce type de conception. J'ai apprécié de devoir travailler en déterminant des barèmes, soit les paramètres des formes. Le résultat final m'a plutôt épater. Je trouve cela fascinant de voir les possibilités infinies qui en résultent. J'y retrouve la magie que la génération d'image par AI offre, mais avec la satisfaction que les itérations proviennent de ma réflexion et non d'un model pour intelligence artificielle.

J'ai l'impression d'entrer dans un nouveau monde et j'y éprouve du plaisir puisqu'après 5 semaines et bon nombre de tutoriels youtube, je peux maintenant affirmer être à l'aise avec l'interface de TouchDesigner.