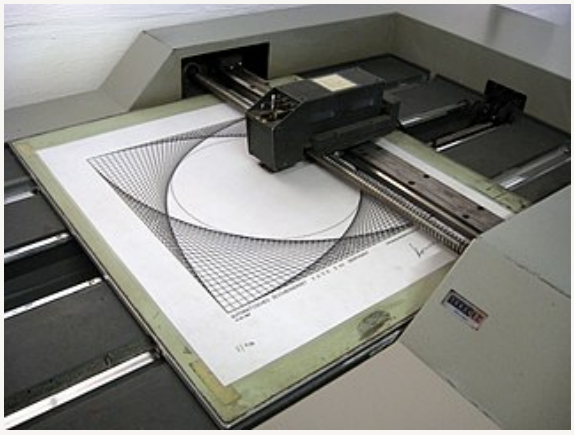
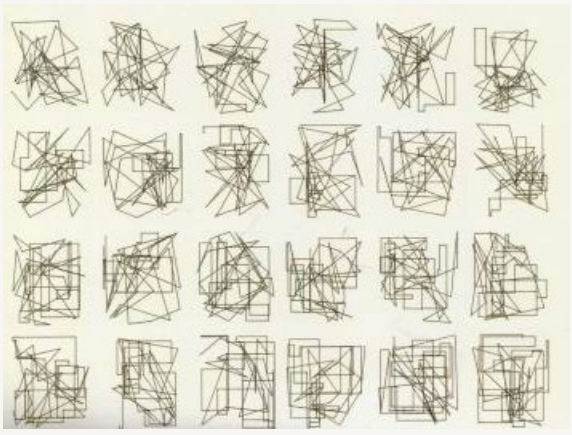
Gabriel Melo Araujo - 536093

George Nees foi um teórico alemão da Arte Digital, que produziu desde de 1965 diversas esculturas, filmes e gráficos por computador, tais como:

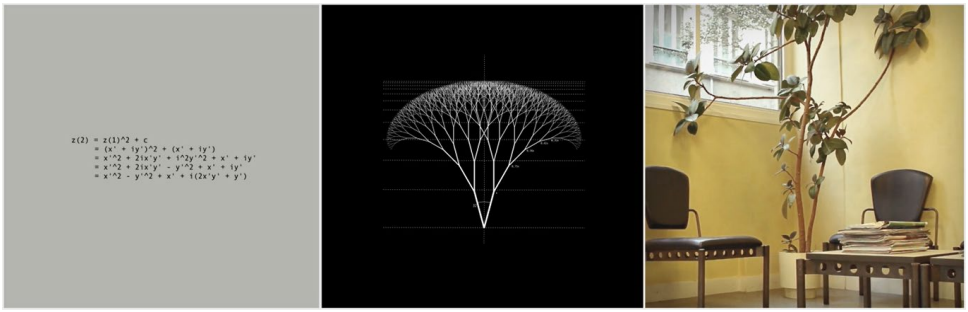




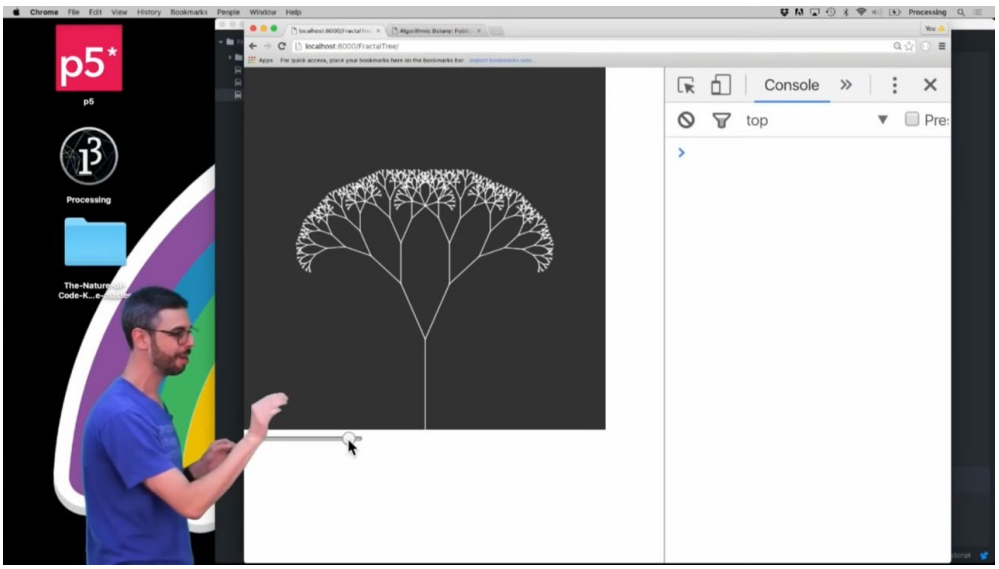
É possível encontrar em suas obras elementos e formas geométricas básicas, repetidas, alteradas e/ou distorcidas a partir da delimitação de um padrão, regra e/ou restrição.

Para a composição de suas obras, o conceito da recursividade é fundamental, sendo também a base desta atividade.

Aproveitando-se da atividade anterior, em que foquei na representação de uma árvore através dos 3 modelos, nota-se que os modelos matemático e computacional usam a repetição de um mesmo processo, para gerar resultado, processo conhecido como “recursividade”.



Me baseando num vídeo explicativo muito bem produzido do programador Daniel Shiffman, que além de ser um imenso contribuidor das diversas artes disponibilizados para a comunidade de usuários do Processing, disponibiliza seus códigos em seu diretório de aulas no Github, que pode ser acessado através de seus vídeos no canal The Coding Train



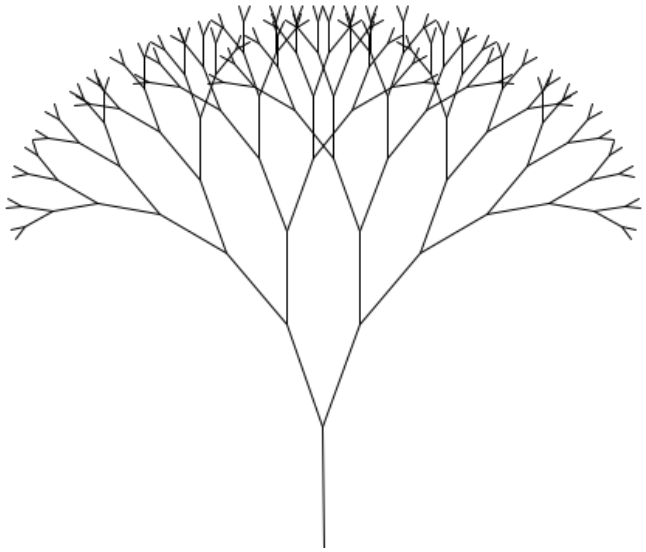
Esse código usa uma função recursiva arvore para desenhar os galhos de uma árvore.

```
void setup() {
  size(500, 500);
}

void draw() {
  background(255); // define um plano de fundo na cor branca.
  translate(width/2, height); // move a 'origem' para o meio da tela.
  arvore(0, 0, -90, 8); // desenha a árvore, a partir de sua origem.
}

void arvore(float x1, float y1, float angulo, float profundidade) {
  if (profundidade != 0) { // determina se a quantidade pré-determinada
de ramos da árvore foram desenhados.
    float x2 = x1 + (cos(radians(angulo)) * profundidade * 10.0);
    float y2 = y1 + (sin(radians(angulo)) * profundidade * 10.0);
    line(x1, y1, x2, y2);
    /* abaixo há uma chamada recursiva para determinar a posição dos
galhos. Realizando uma soma do valor dos ângulos para os galhos que
divergem para a direita, e uma subtração dos valores para os galhos que
divergem para a esquerda.
     */
    arvore(x2, y2, angulo - 20, profundidade - 1);
    arvore(x2, y2, angulo + 20, profundidade - 1);
  }
}
```

A função recebe quatro argumentos: As coordenadas iniciais x & y do galho (`x1 & y1`), o ângulo do galho (`angulo`), e a "profundidade" de repetições da recursão (`profundidade`).

A função calcula os valores finais das coordenadas de x & y do galho (x2 & y2) usando trigonometria e desenha uma linha que vai dos pontos iniciais até os pontos finais adquiridos.

Após este procedimento, a função realiza duas chamadas a si mesma, para desenhar dois novos galhos ao final das coordenadas finais anteriores, com uma diferença angular e uma "profundidade" reduzida. Este processo se repete até que a profundidade seja zerada, gerando a seguinte figura:



A função `setup` determina o tamanho do canvas, já a função `draw` determina a cor do plano de fundo e a chamada da função `arvore` para o desenho do figura.

Ao realizar a chamada da função, é possível determinar os pontos de coordenada de onde a árvore começará a ser desenhada, o ângulo de visualização do desenho e a quantidade de vezes que novos galhos ainda serão desenhados. O ângulo de distanciamento dos galhos pode ser alterado na chamada recursiva da função `arvore`.