COMPUTAÇÃO GRÁFICA

Geração procedural

Prof. Érick Oliveira Rodrigues

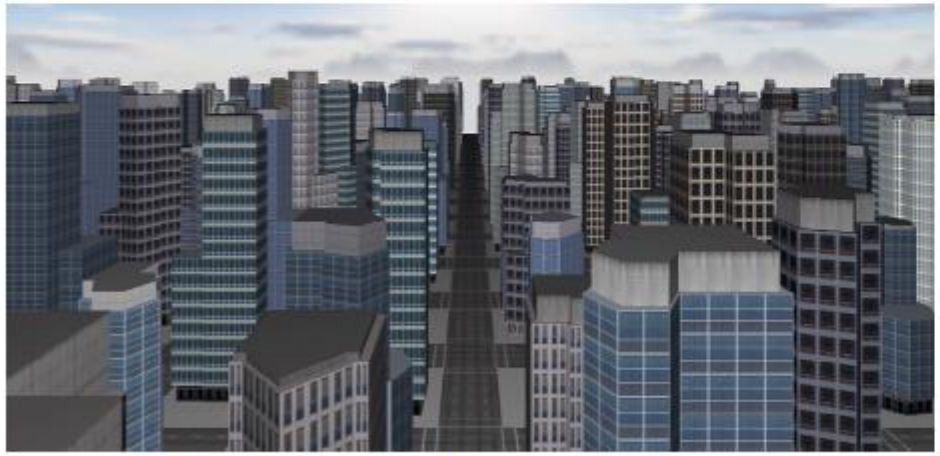
TIPOS DE GERAÇÃO

- A geração procedural está presente em jogos e simuladores, em todos os sentidos abaixo:
  - Geração de mapas;
    - Geração de montanhas;
    - Geração de terreno (mato, terra, água, etc);
    - Geração de árvores;
    - Geração de folliage (matinhos dentro do jogo por exemplo);
  - Geração de personagens;
  - Geração de cidades;
    - Geração de casas;
    - Geração de furniture (elementos dentro da casa, etc);
  - Geração / alteração de cores de assets;
  - Geração de partículas (efeitos gráficos como magia/spell por exemplo);
  - Geração de itens, armas, etc.

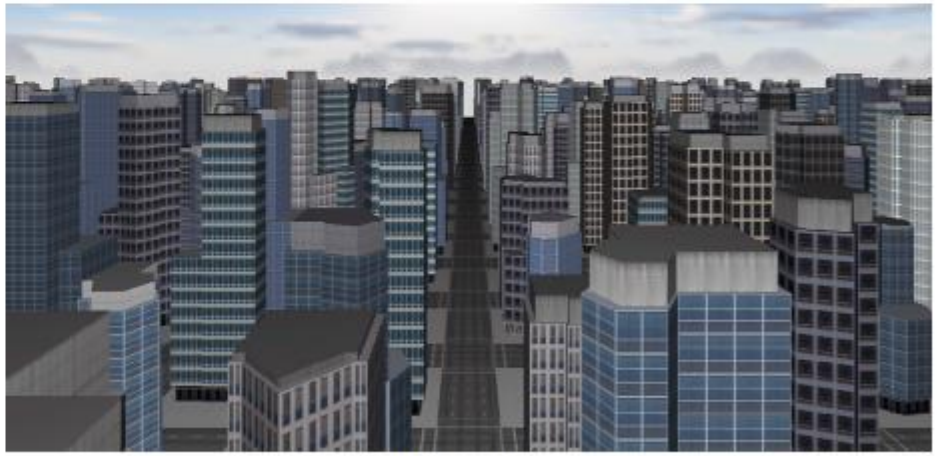
GERAÇÃO DE PRÉDIOS

(a) 100 buildings

(b) 200 buildings



(c) 500 buildings



(d) 1000 buildings

Figure 8: Procedural City Performance

[Link para o artigo]

GERAÇÃO PROCEDURAL DE ESTRADAS

Figure 16: *A mountain road following the path of a river.*

Figure 19: *Different road paths obtained by modifying the relative influence of the cost functions.*

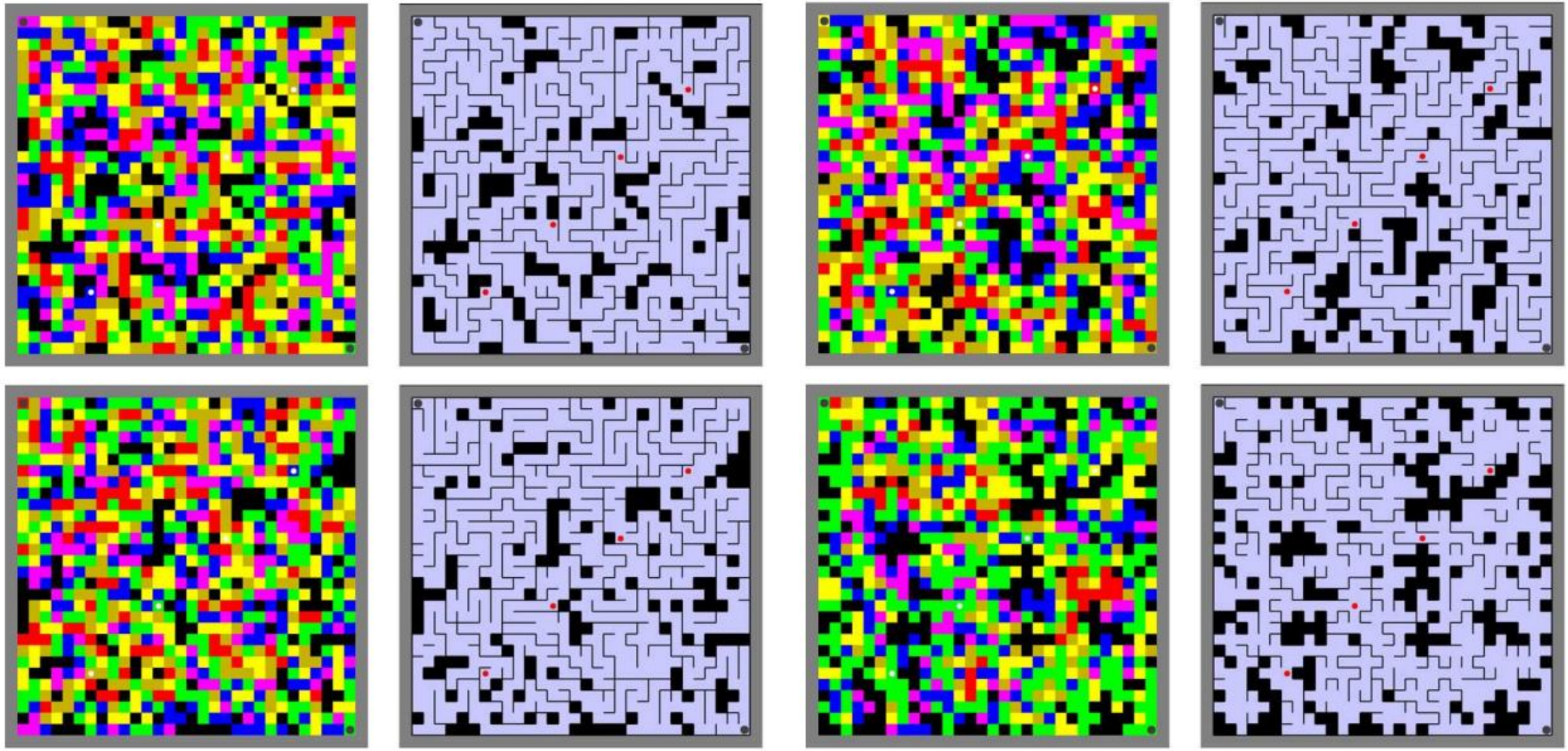
Link para o artigo

GERAÇÃO PROCEDURAL DE VILAREJOS

Fig. 14 A real (*top left*) and a procedurally generated highland hamlet (*top right*, *bottom*)

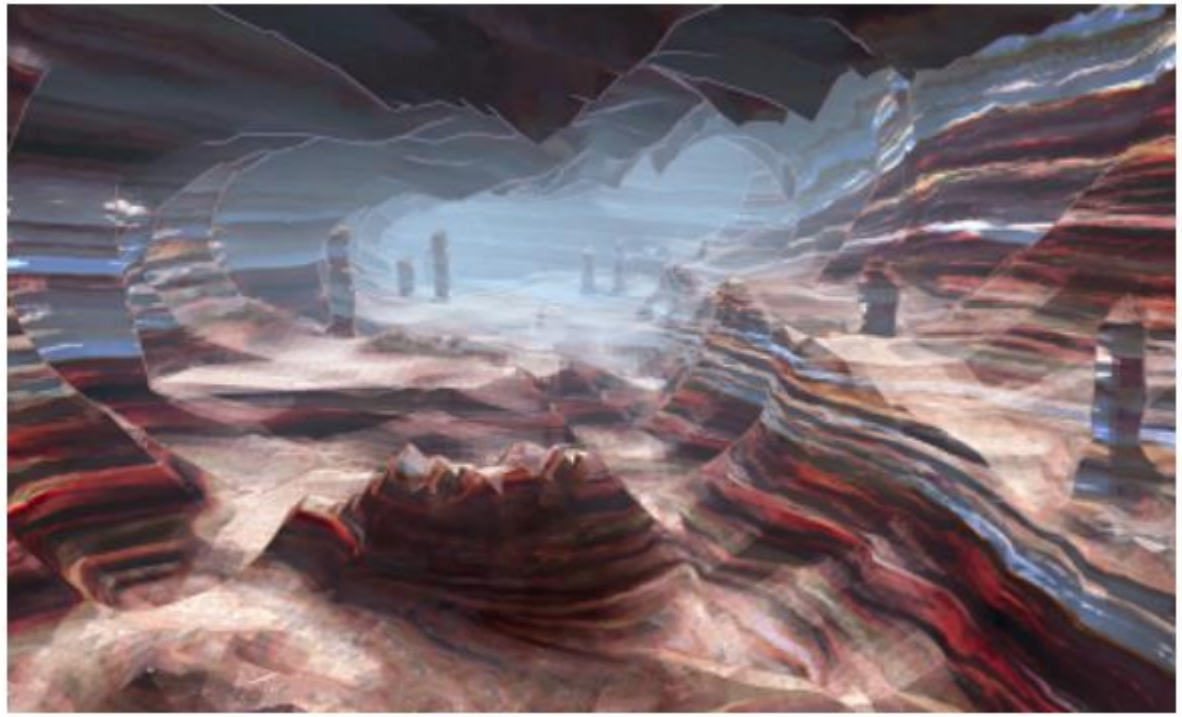
Link para o artigo

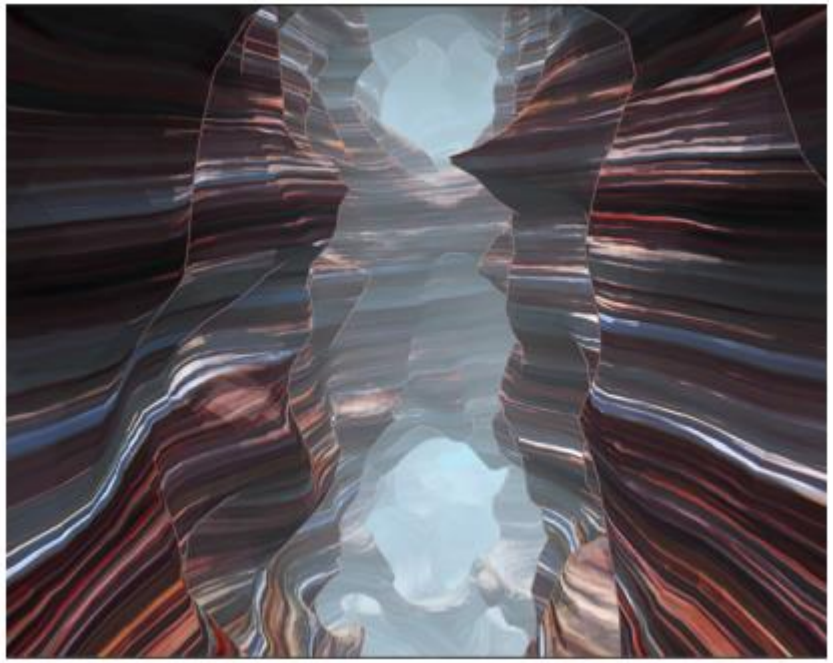
GERAÇÃO PROCEDURAL DE LABIRINTOS



Link para o artigo

GERAÇÃO PROCEDURAL DE CAVERNAS



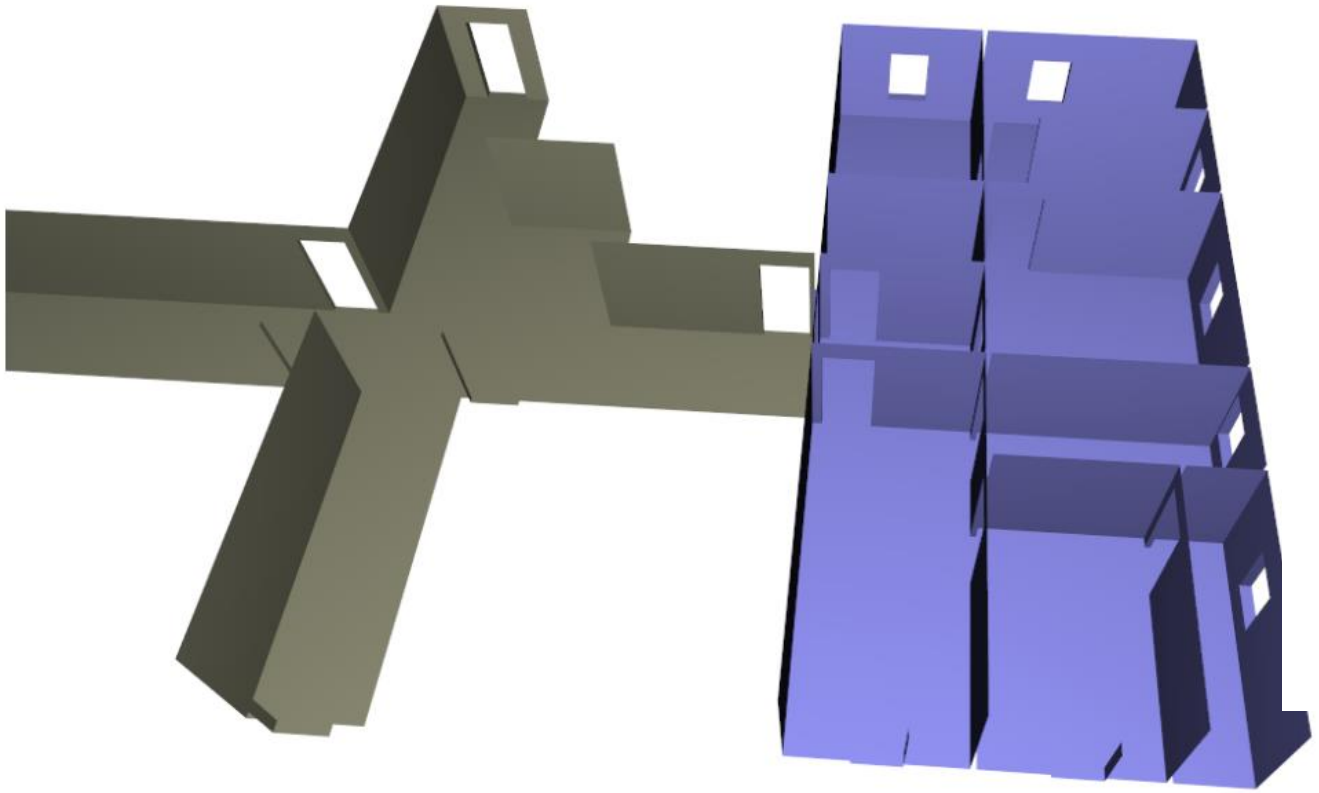


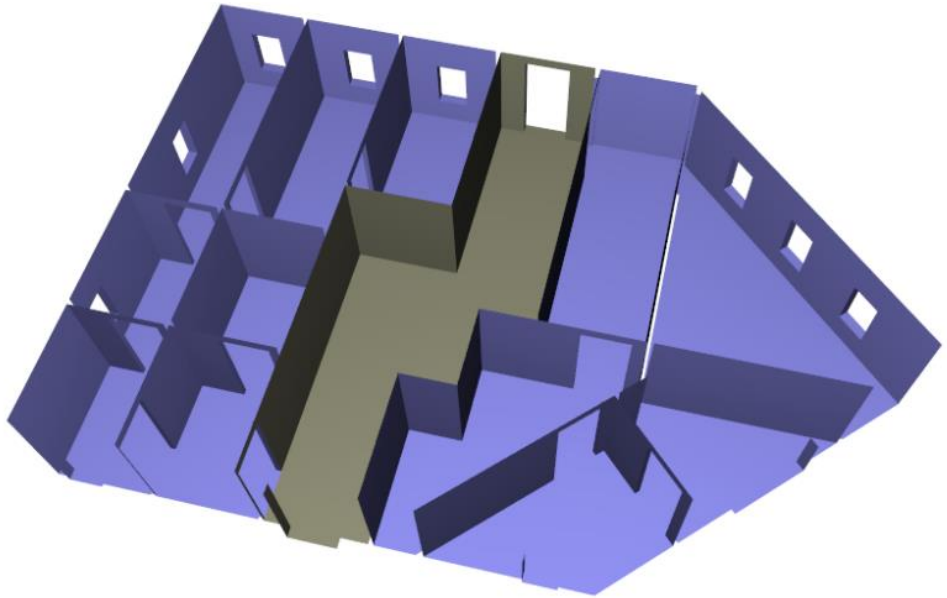
Link para o artigo

GERAÇÃO PROCEDURAL DE PILHAS DE PEDRA

[Link para o artigo]

ESPAÇOS INTERNOS





[Link para o artigo]

GERAÇÃO PROCEDURAL DE FASES

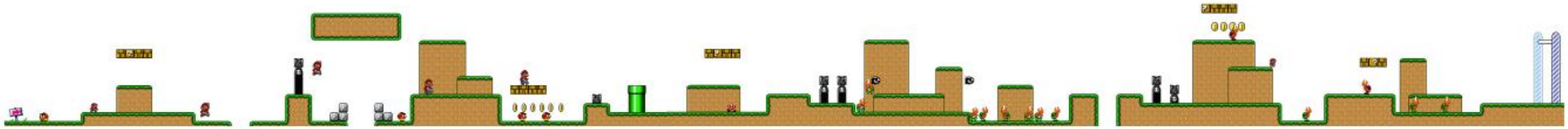
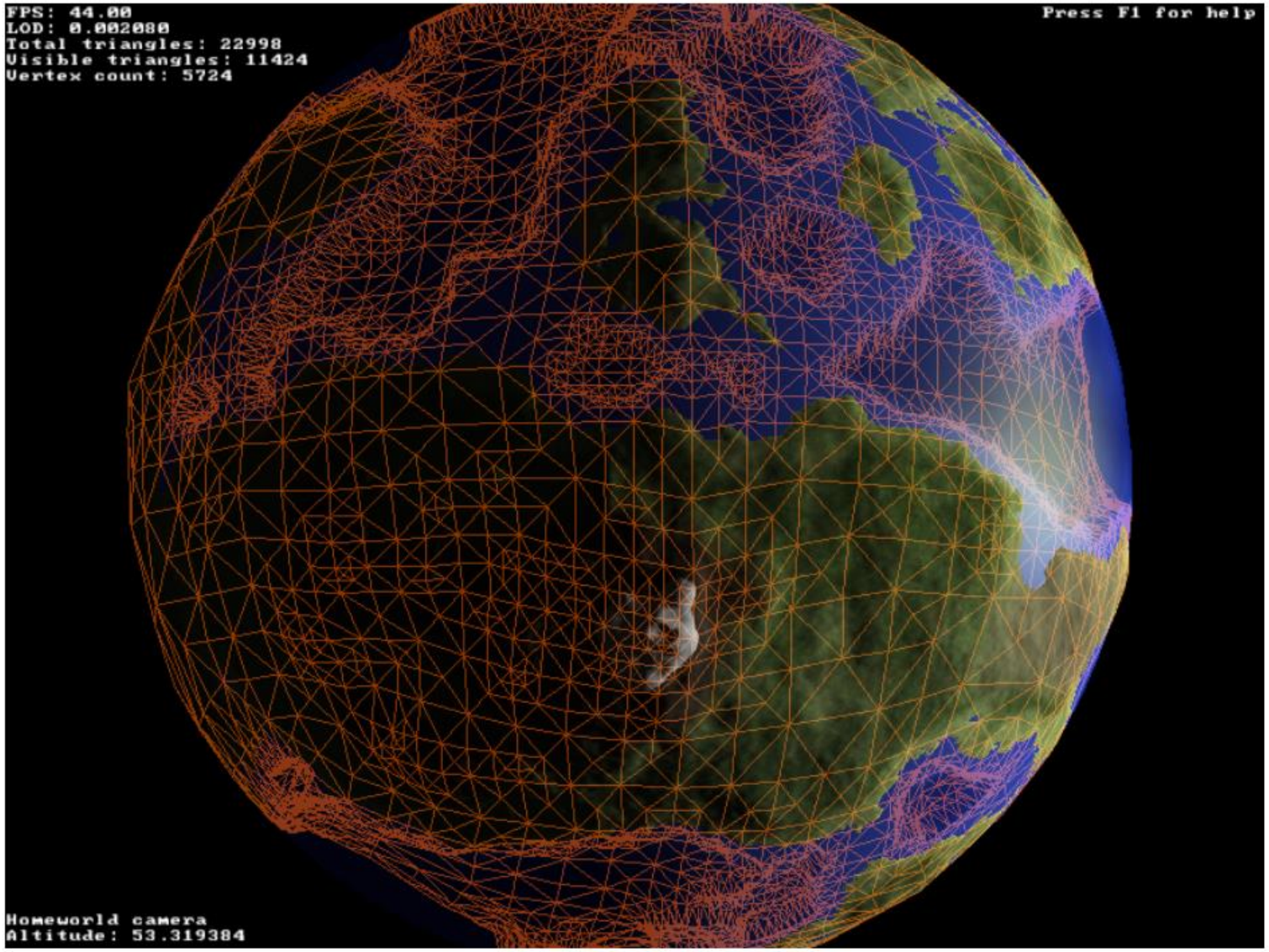


Fig. 6: A representative full-sized level generated by HCS using the parabolic progression arc.

Link para o artigo 1

Link para o artigo 2

GERAÇÃO PROCEDURAL DE PLANETAS



Link para o artigo

GERAÇÃO PROCEDURAL DE ITENS

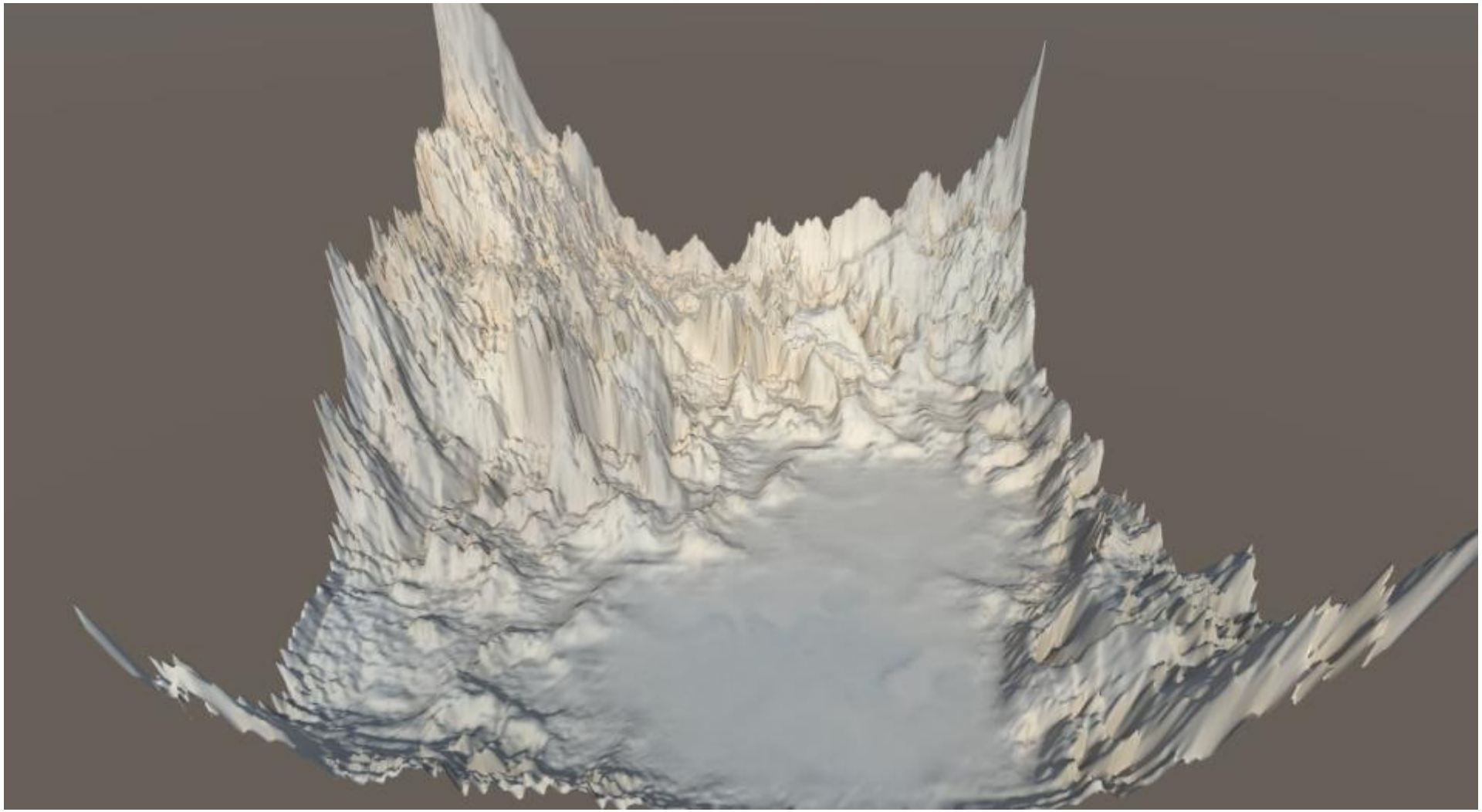
[Link para o artigo]

GENERATIVE ADVERSARIAL NETWORKS

- Atualmente, está em alta a utilização de Generative Adversarial Networks (GANs) para a geração procedural.
  - Por exemplo, o a seguinte implementação chamada Pix2Pix (ou o CycleGAN – por exemplo, na mesma pasta):
    - https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

- Em seguida, mostro alguns trabalhos que utilizaram GANs na geração procedural.
  - Um ponto positivo é que são frameworks que não são difíceis de usar, normalmente precisando apenas de uma base de dados adequada.

GERAÇÃO PROCEDURAL USANDO GAN



Link para o artigo

GERAÇÃO PROCEDURAL DE SOMBRAS

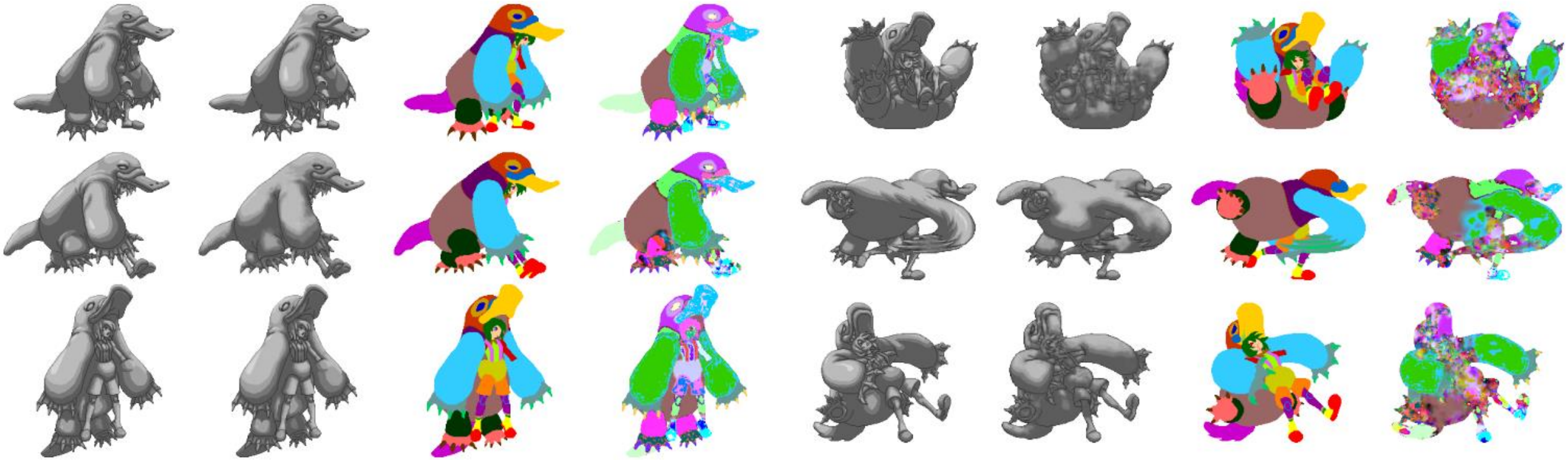


Figure 6. Results for six different sprites taken from *Sarah* animations. From top to bottom and left to right, each sprite is shown in its respective ground truth *gray*, generated *gray*, ground truth *color* and generated *color* forms.

Link para o artigo

GERAÇÃO PROCEDURAL DE TEXTURAS

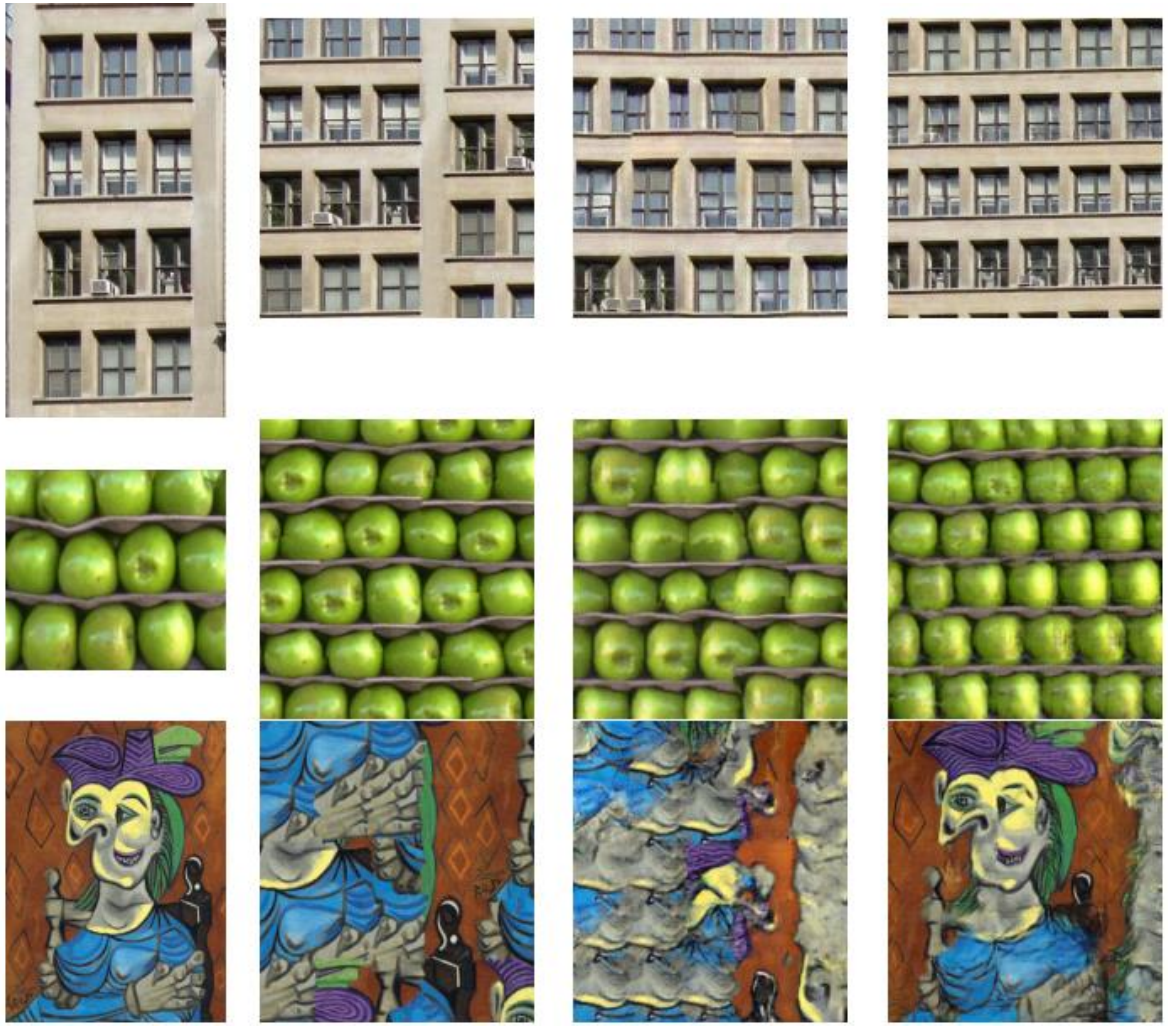


Fig. 6: **Experimental results.** Column one is the source image. Column two is IQ generated. Column three is IQ + GAN, and column four is cDCGAN

Link para o artigo

MAPAS DE SATÉLITE

Fig. 1. Procedurally generated samples of satellite images.

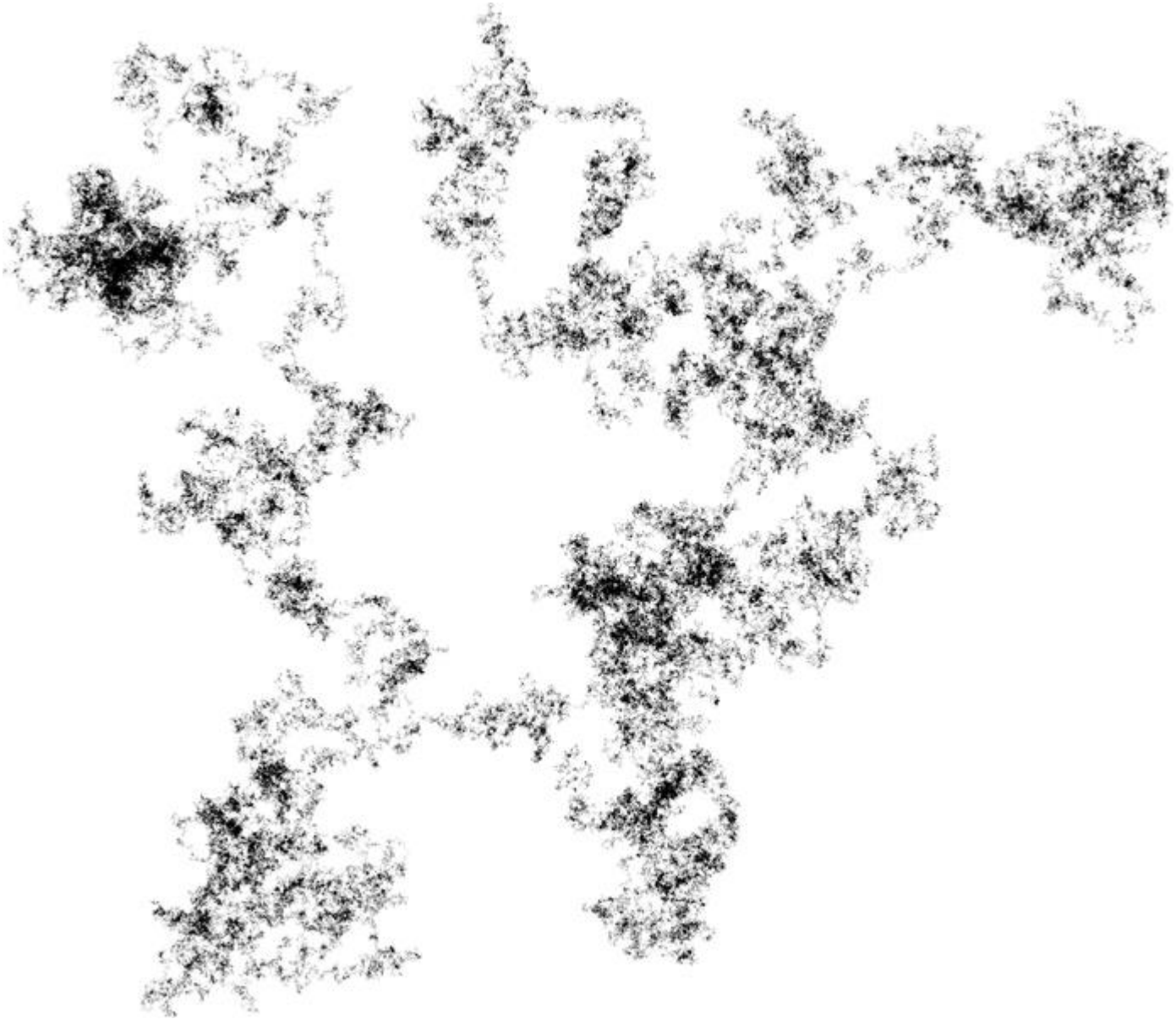
Link para o artigo

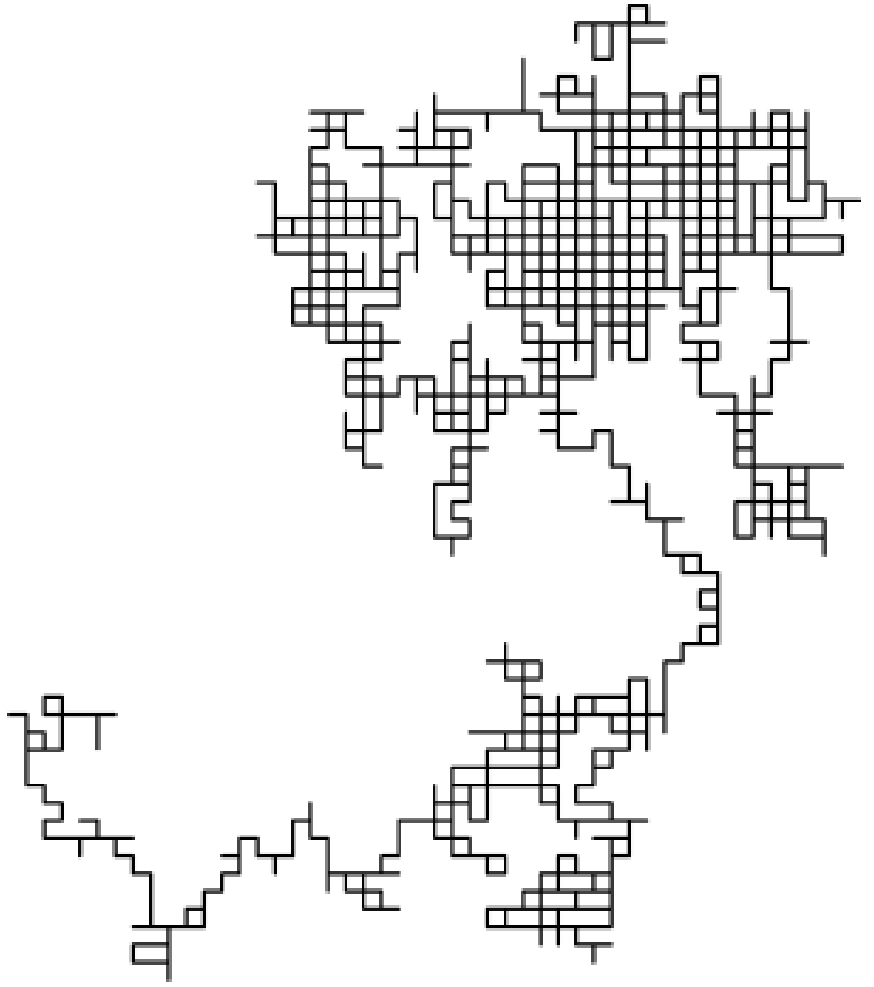
COLORIZAÇÃO DE LINEART

OUTROS

- Ainda, outras opções envolvem:
  - Geração procedural de cabelo
  - Geração procedural para jogo matemático
  - Geração procedural de histórias
  - Geração procedural de quests
  - Survey de geração procedural de uma forma geral
  - Geração procedural de música

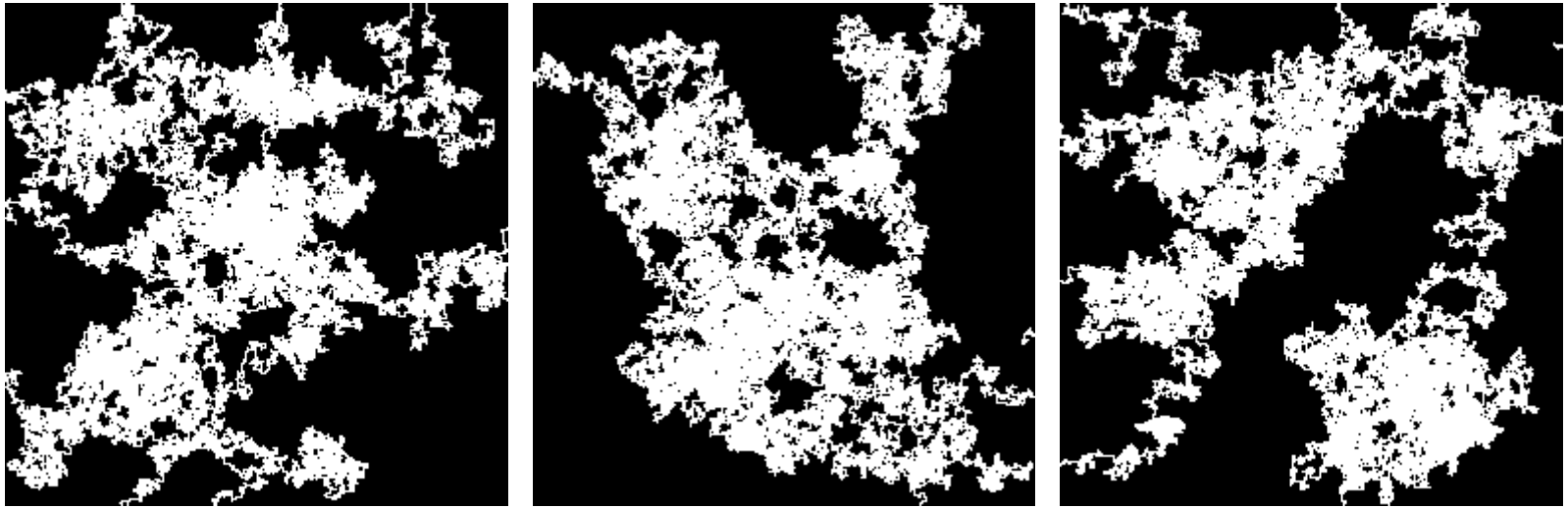
RANDOM WALK





RANDOM WALK

1. Andar com o pixel de forma randômica pelo mapa;
    1. Para cada movimento, marcar como “branco” o pixel que foi andado;
2. Verificar se o pixel saiu dos limites da imagem;
    1. Se sim, mover ele pra uma nova posição randômica dentro da imagem;
3. Voltar ao passo 1;



Para baixar um código exemplo:

M3P-JKC

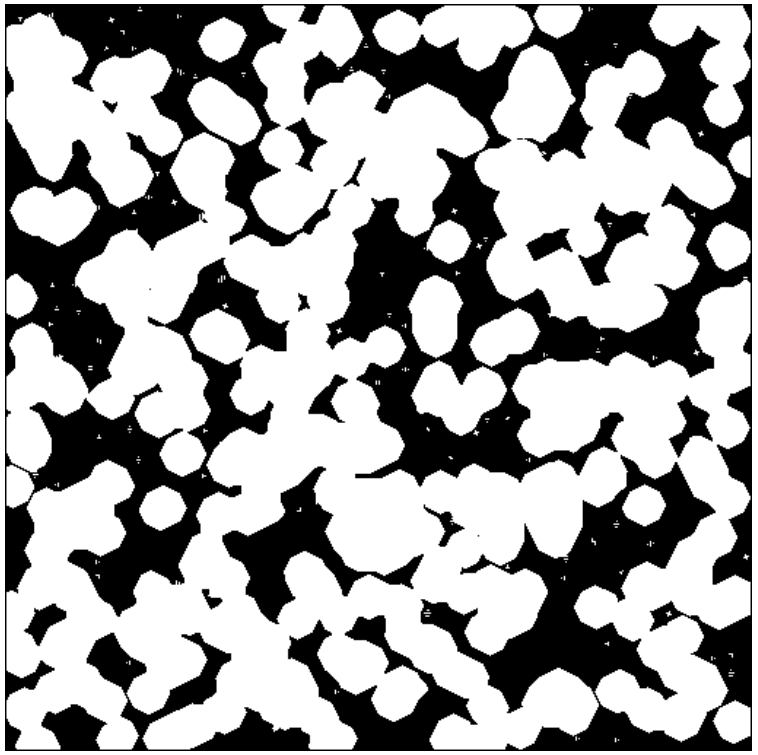
CELLULAR AUTOMATA

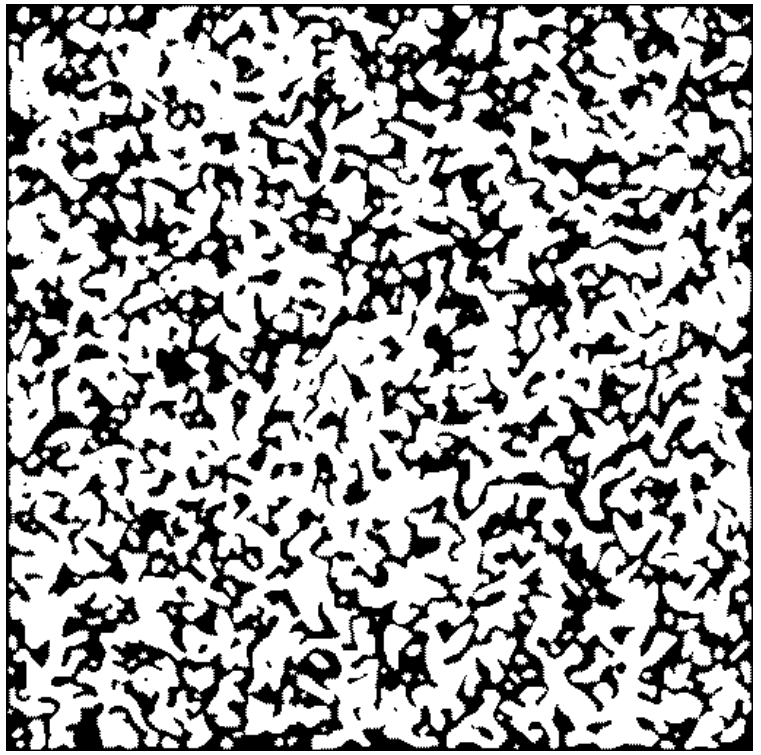
1. Gerar uma imagem binária base com ruído;
2. Verificar se cada ponto gruda no ponto vizinho (quantidade de vizinhos do ponto precisa ser maior que um valor, em uma vizinhança de 8, usar o valor > 4 para grudar, por exemplo);
    1. Se não estiver grudado ainda a outros pontos, andar randomicamente com o ponto;
3. Voltar ao passo 2;

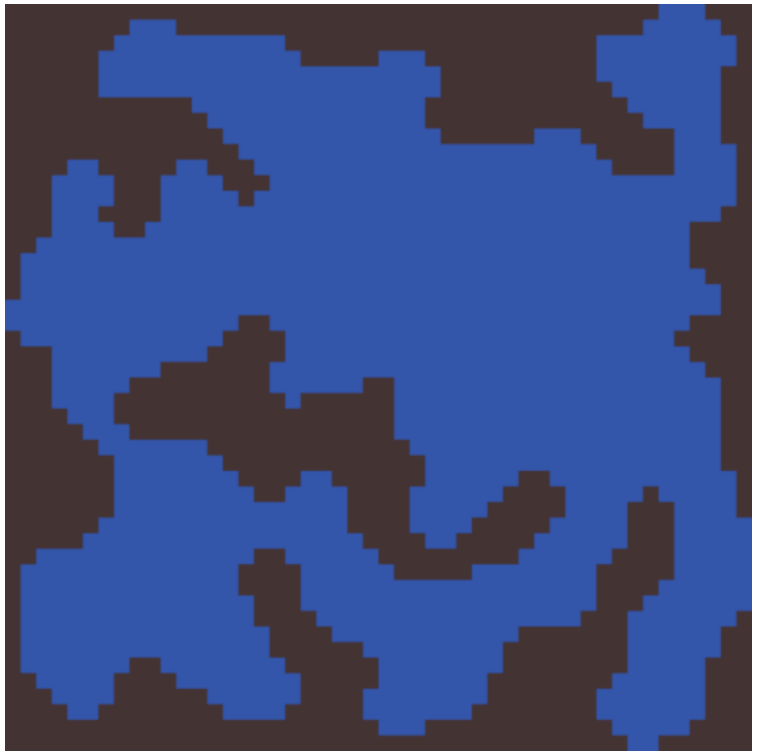
Simples assim! Um ponto negativo desse algoritmo é que ele não gera "tons de cinza".

- Código para baixar um exemplo: ZGN-NE3

CELLULAR AUTOMATA

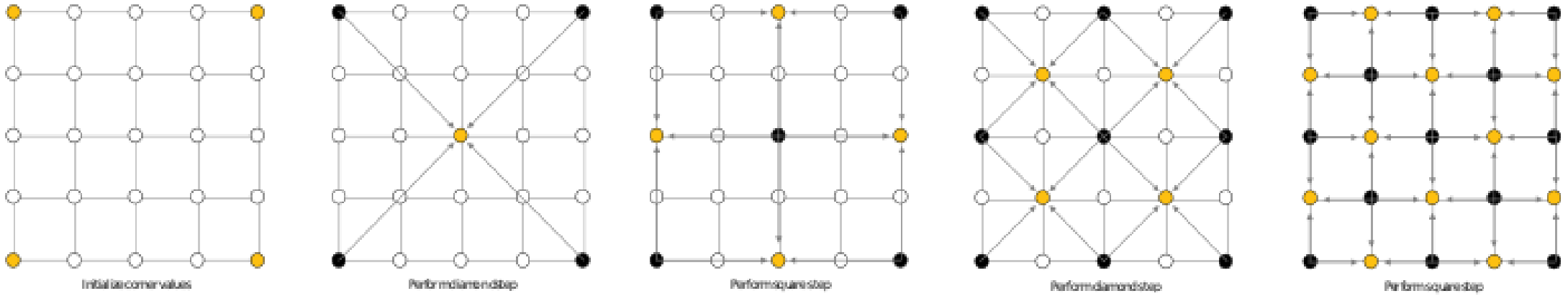






DIAMOND-SQUARE ALGORITHM

1. Gere uma imagem em tom de cinza com alguns ruídos;
2. Passo quadrado: para cada quadrado sete o ponto do meio sendo a média dois 4 pontos mais um valor randômico.
3. Passo diamante: para cada diamante sete o ponto do meio como a média dos 4 pontos mais um valor randômico.



- A diferença é que esse algoritmo, diferente do anterior, consegue gerar heightmaps com variações de cinza.

FRACTAIS E CÁLCULO

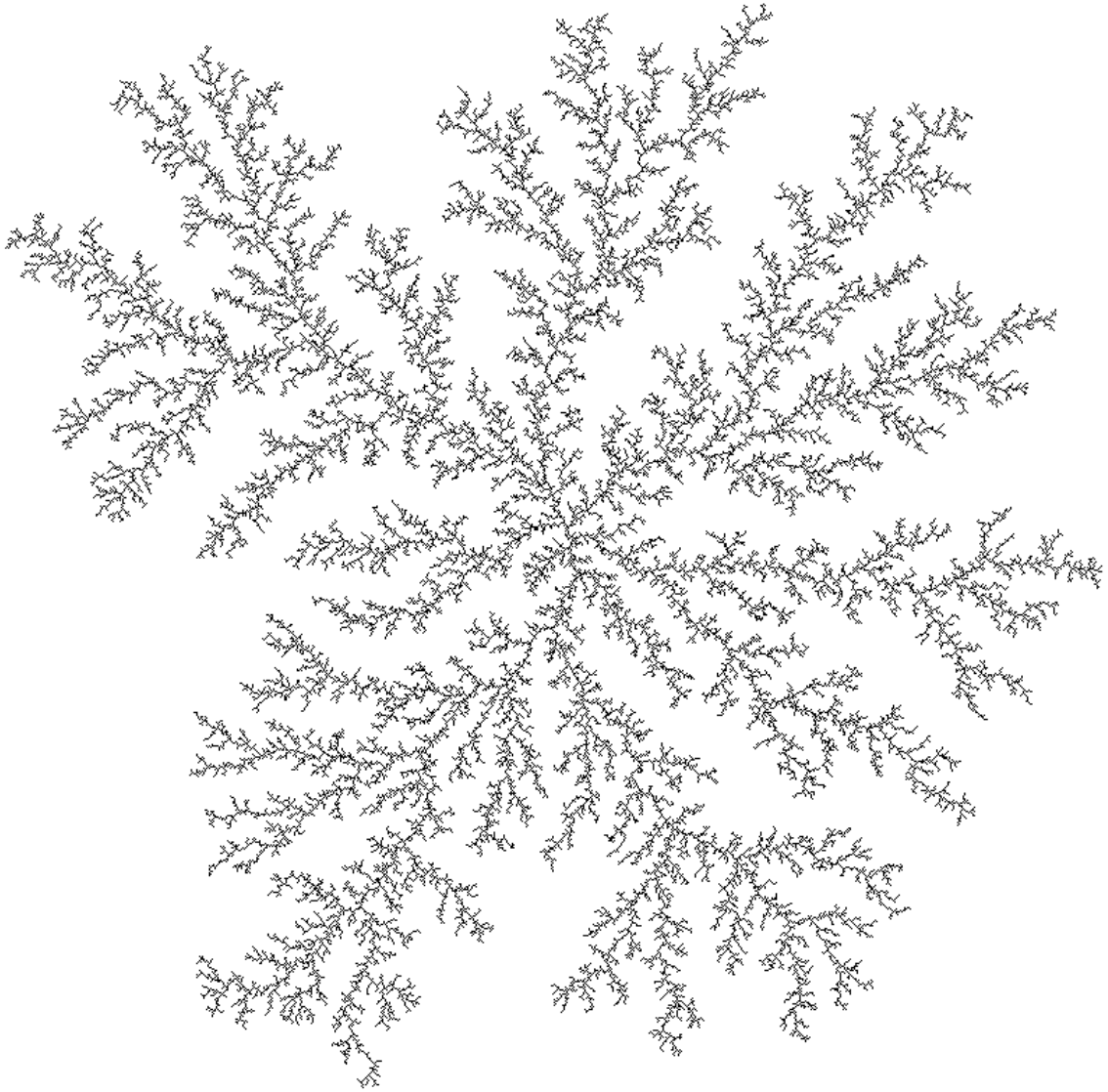
- Existem outras abordagens baseadas em fractais pra geração, entre outros, de:
  - Terrenos;
  - Árvores;
  - Texturas;
  - Caminhos (path); etc;

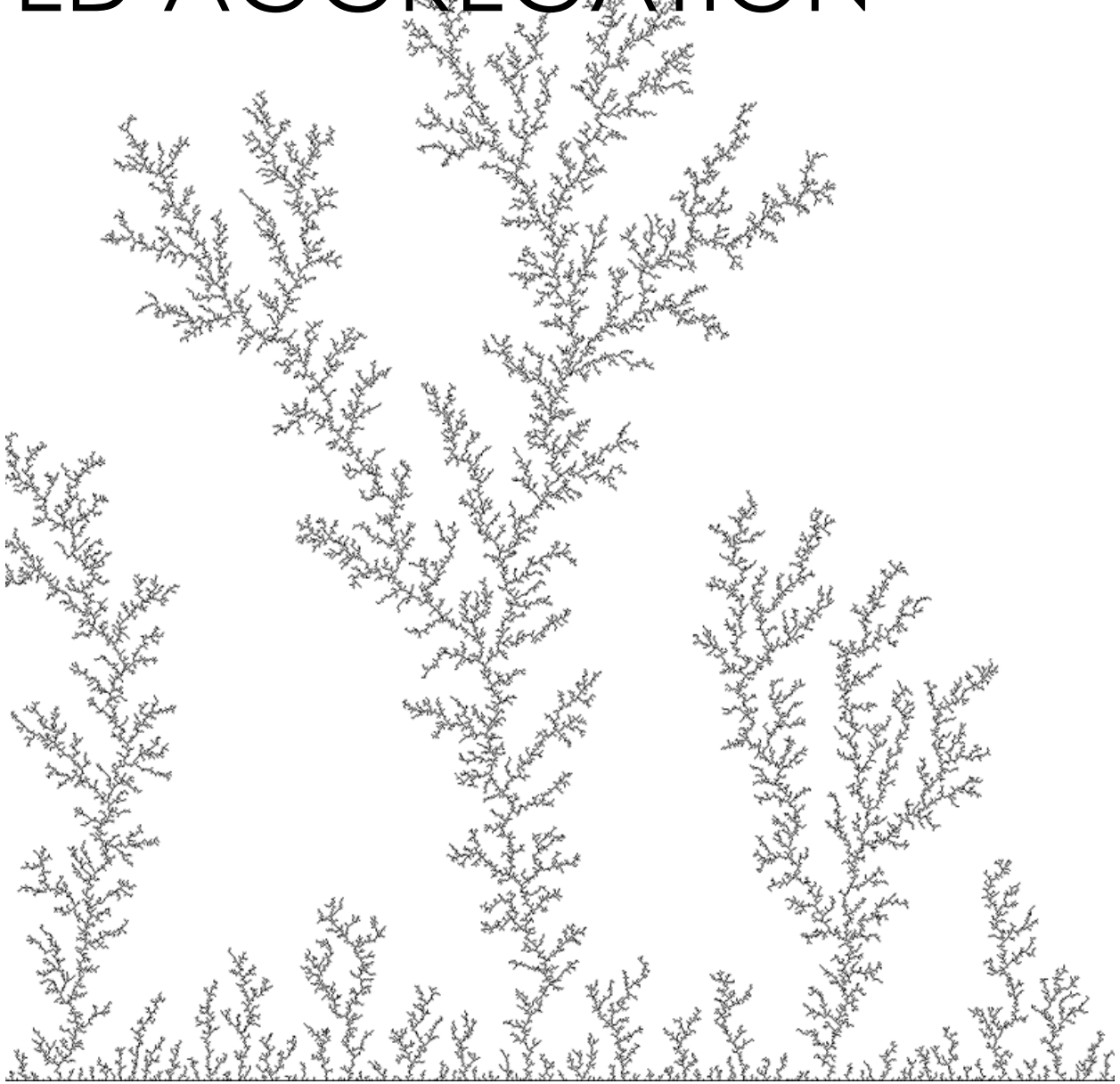
- Uma abordagem bem famosa, mas que usa cálculo e é um pouco mais complexa, porém bastante utilizada para geração procedural de terrenos, é a Perlin Noise.
  - Dizem que o minecraft usa uma modificação do Perlin Noise.
  - O photoshop também tem o algoritmo ou uma modificação do mesmo implementado.

DIFFUSION LIMITED AGGREGATION

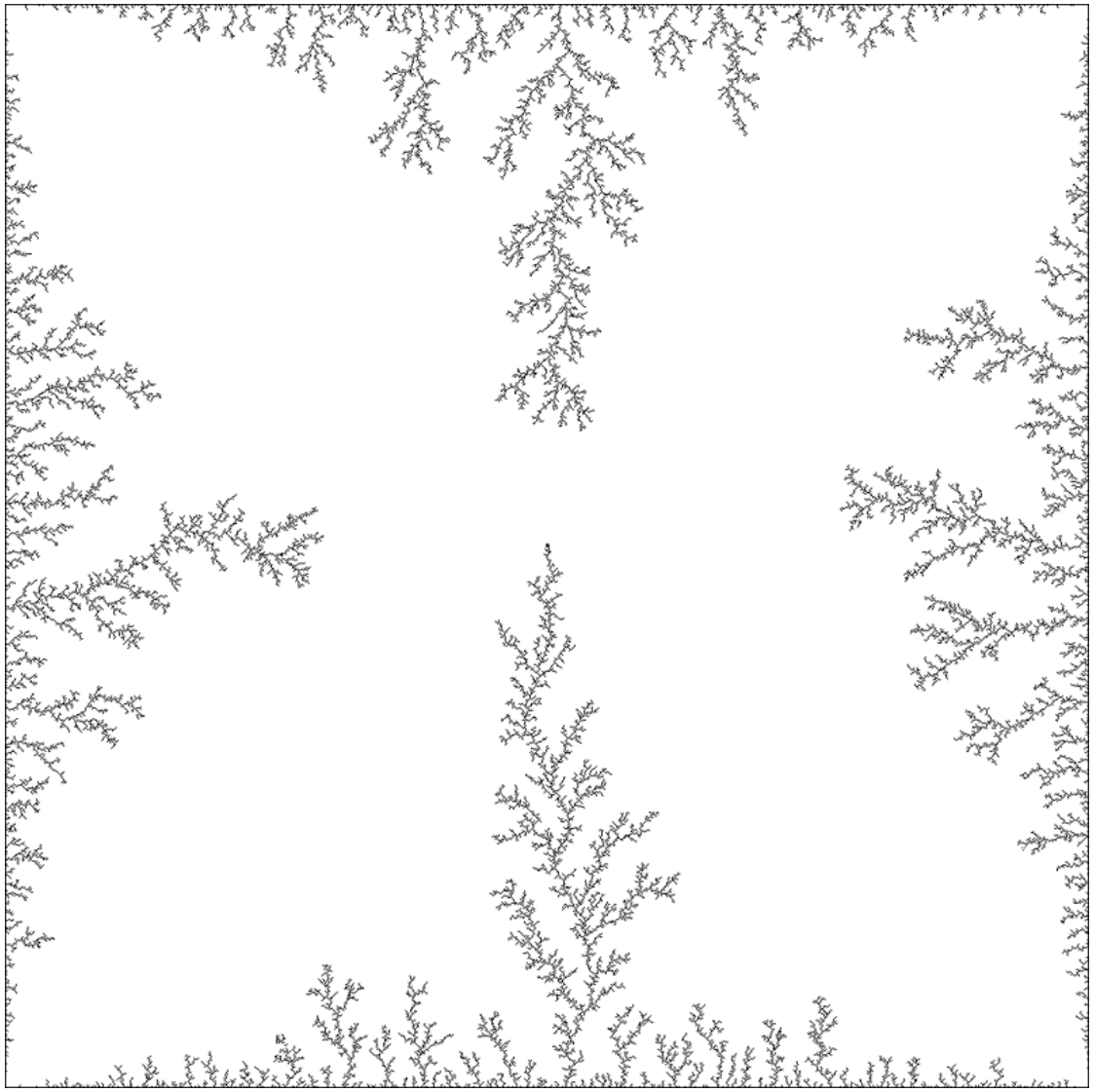
1. Estabeleça alguns pixels de barreira (já existentes);
2. Lance um pixel andando como random walk pela tela;
3. A cada movimentação do pixel, cheque se o pixel está próximo de/conectado a algum outro pixel;
    1. Se estiver, grude ele no outro pixel (é possível colocar uma probabilidade para que grude também – o que gera estruturas mais "grossas");
4. Volte ao passo 2;

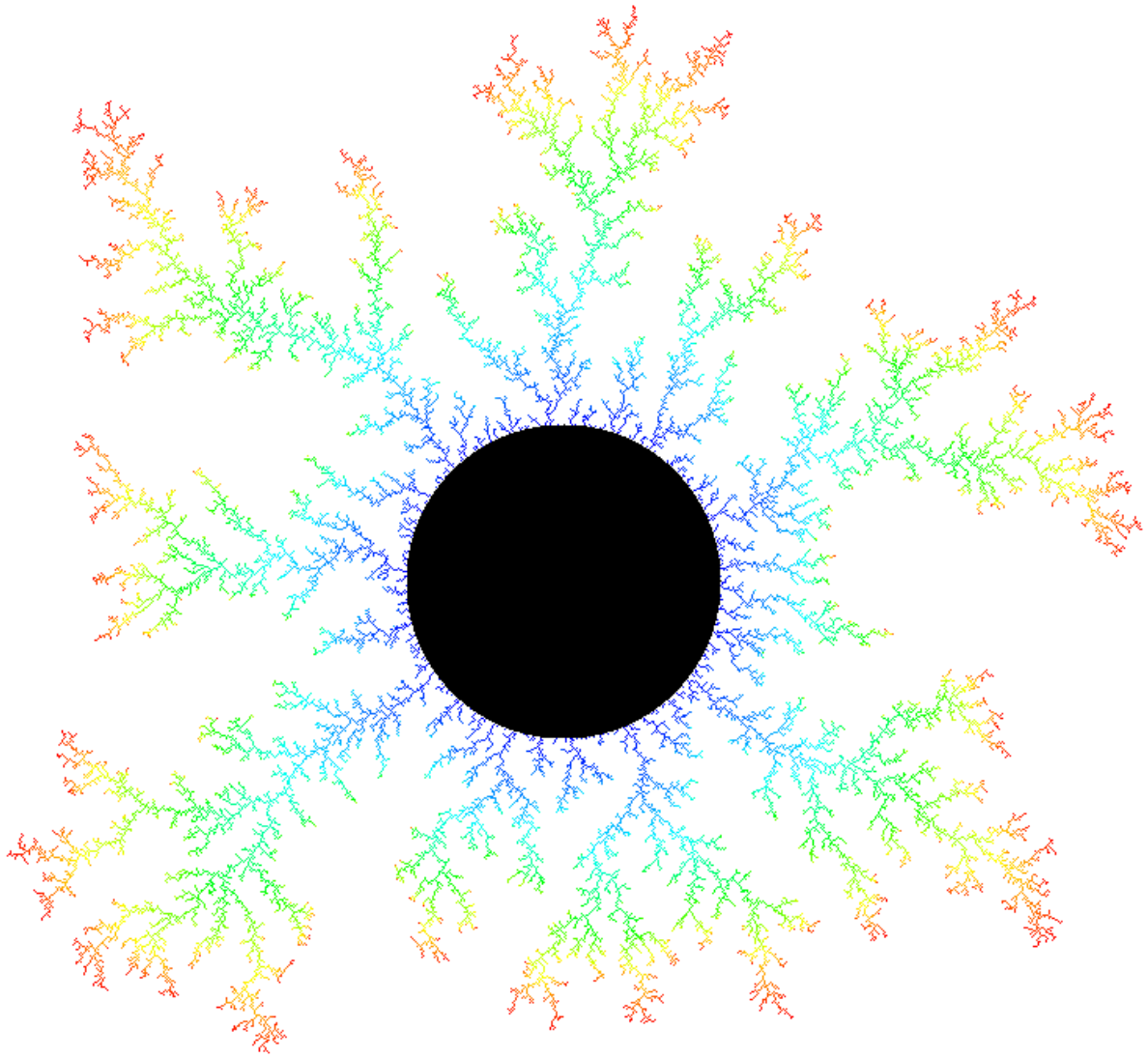
DIFFUSION LIMITED AGGREGATION



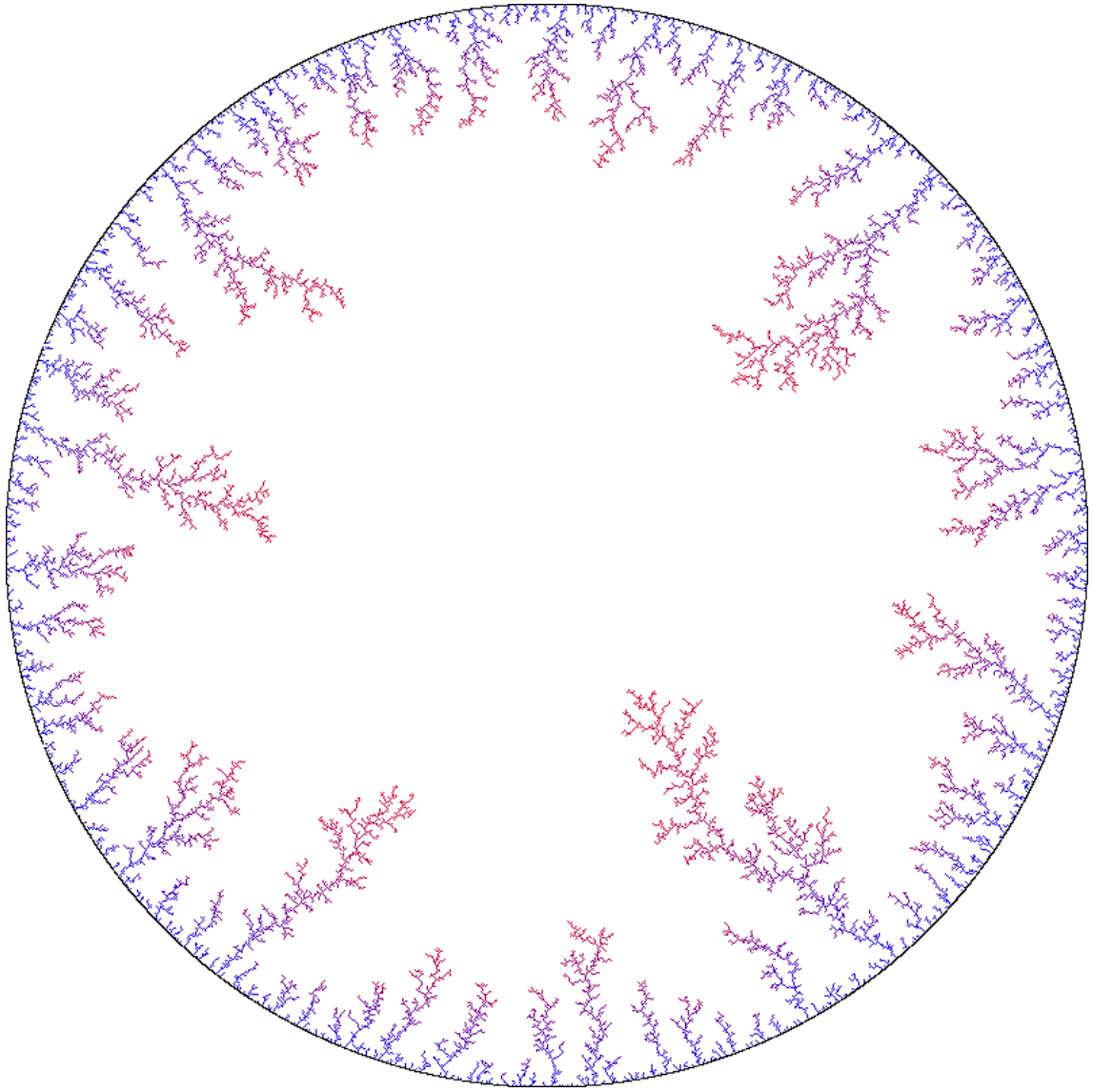


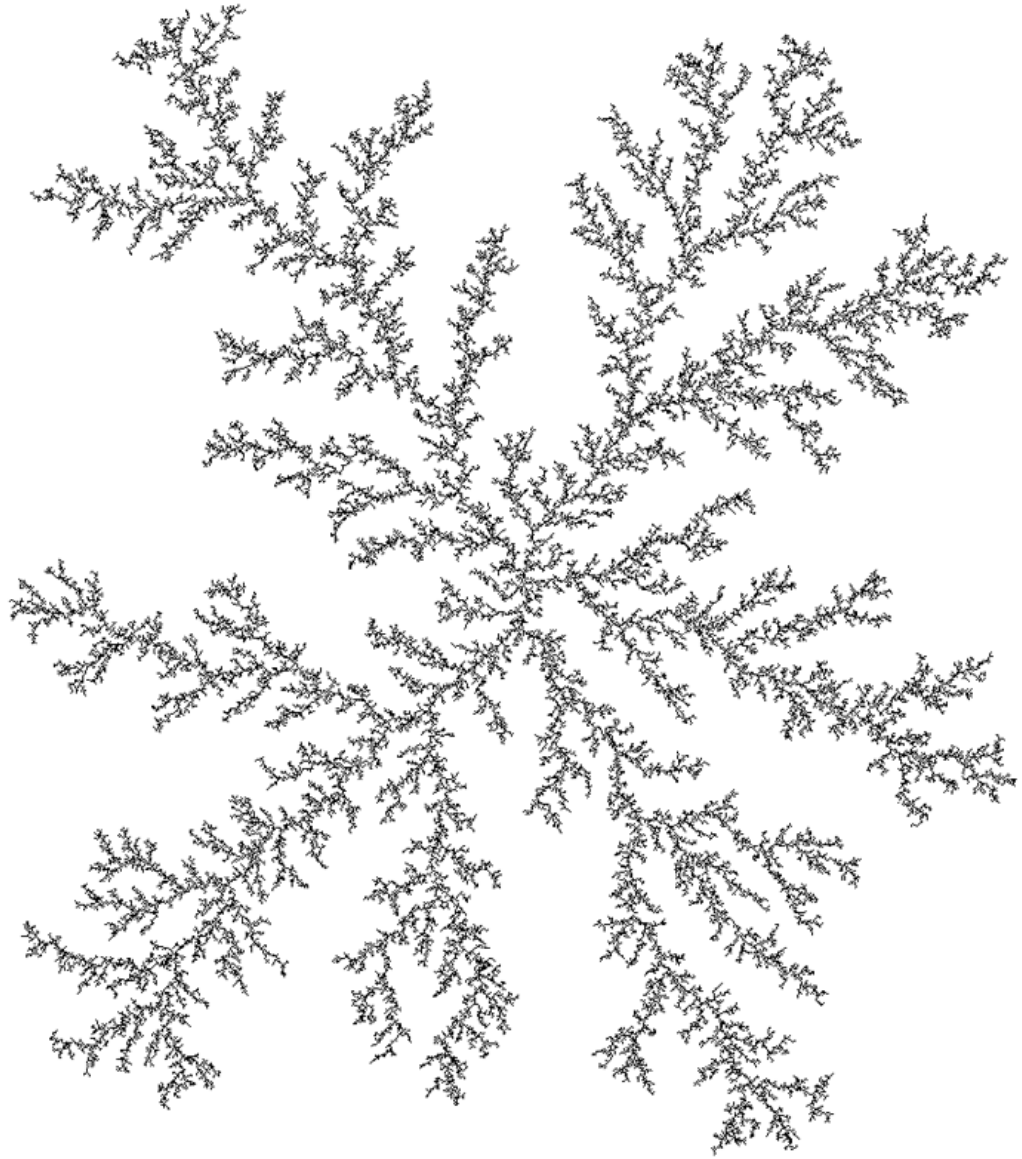
DIFFUSION LIMITED AGGREGATION



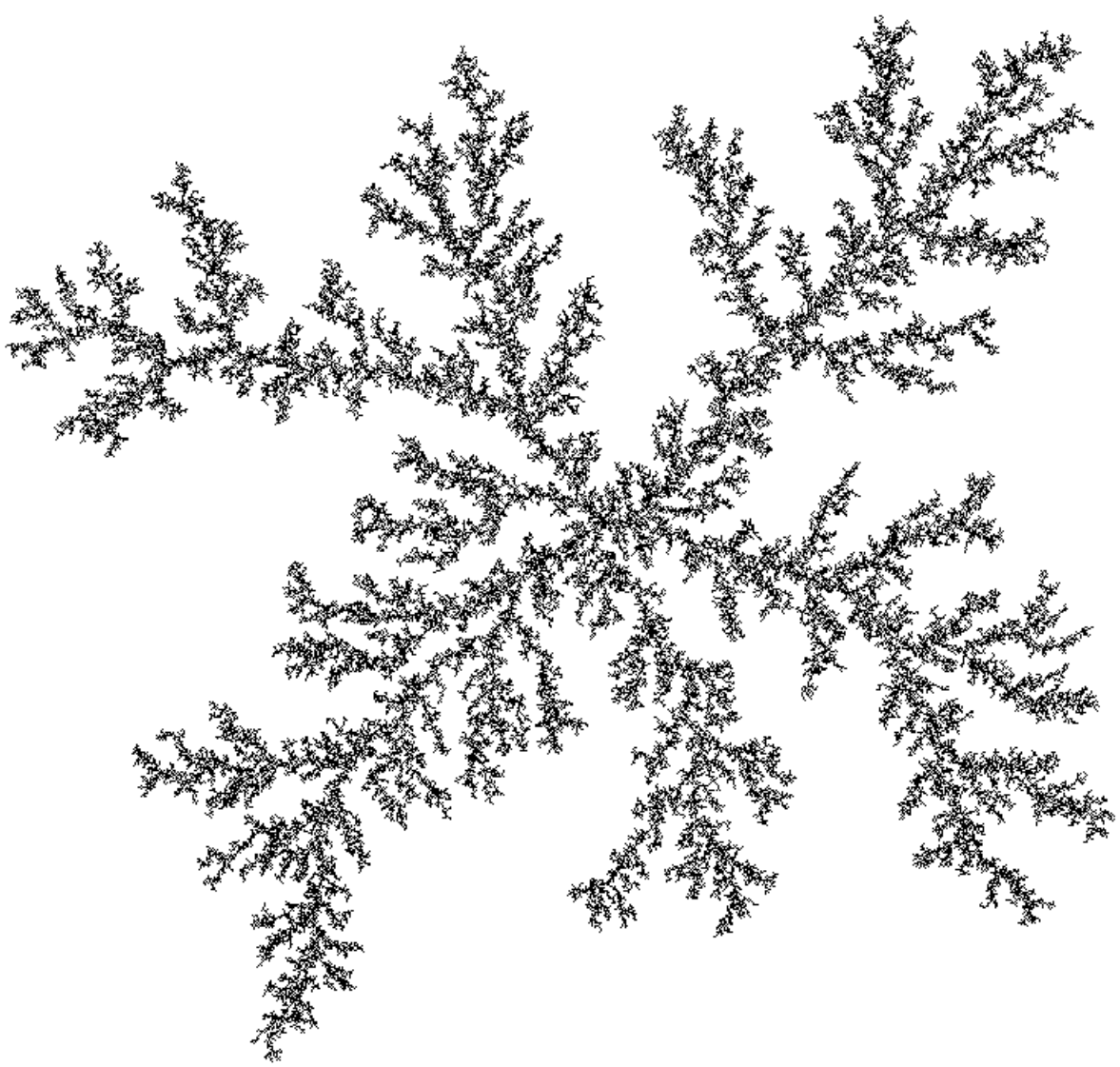


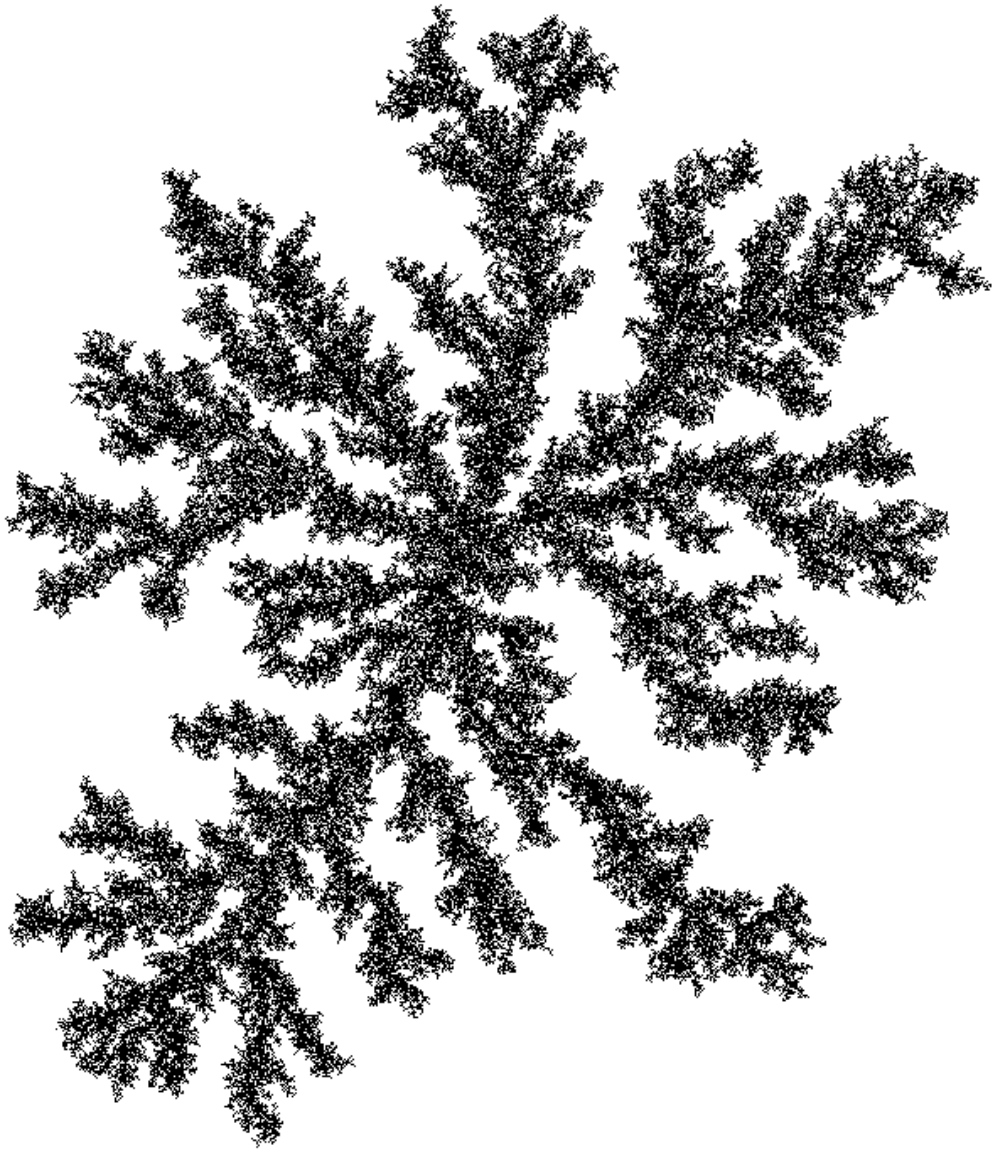
DIFFUSION LIMITED AGGREGATION



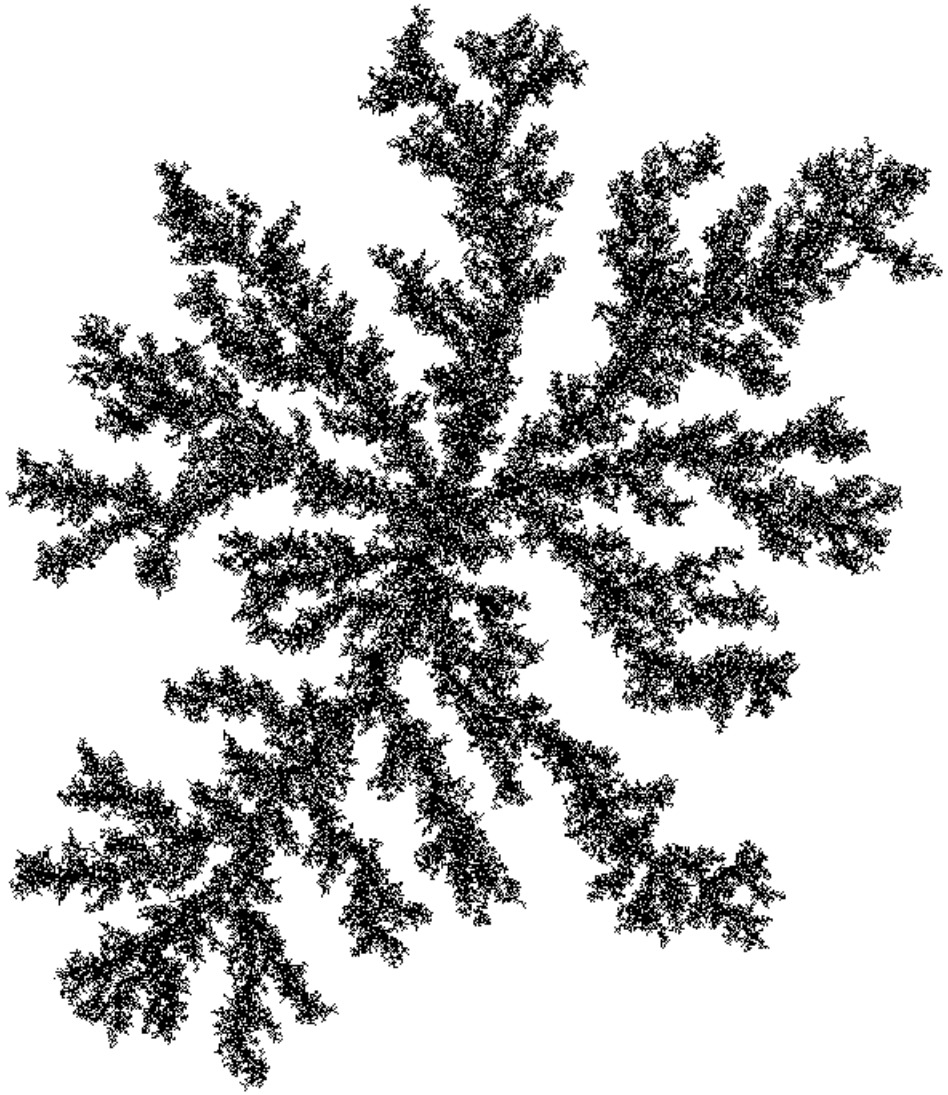


DIFFUSION LIMITED AGGREGATION

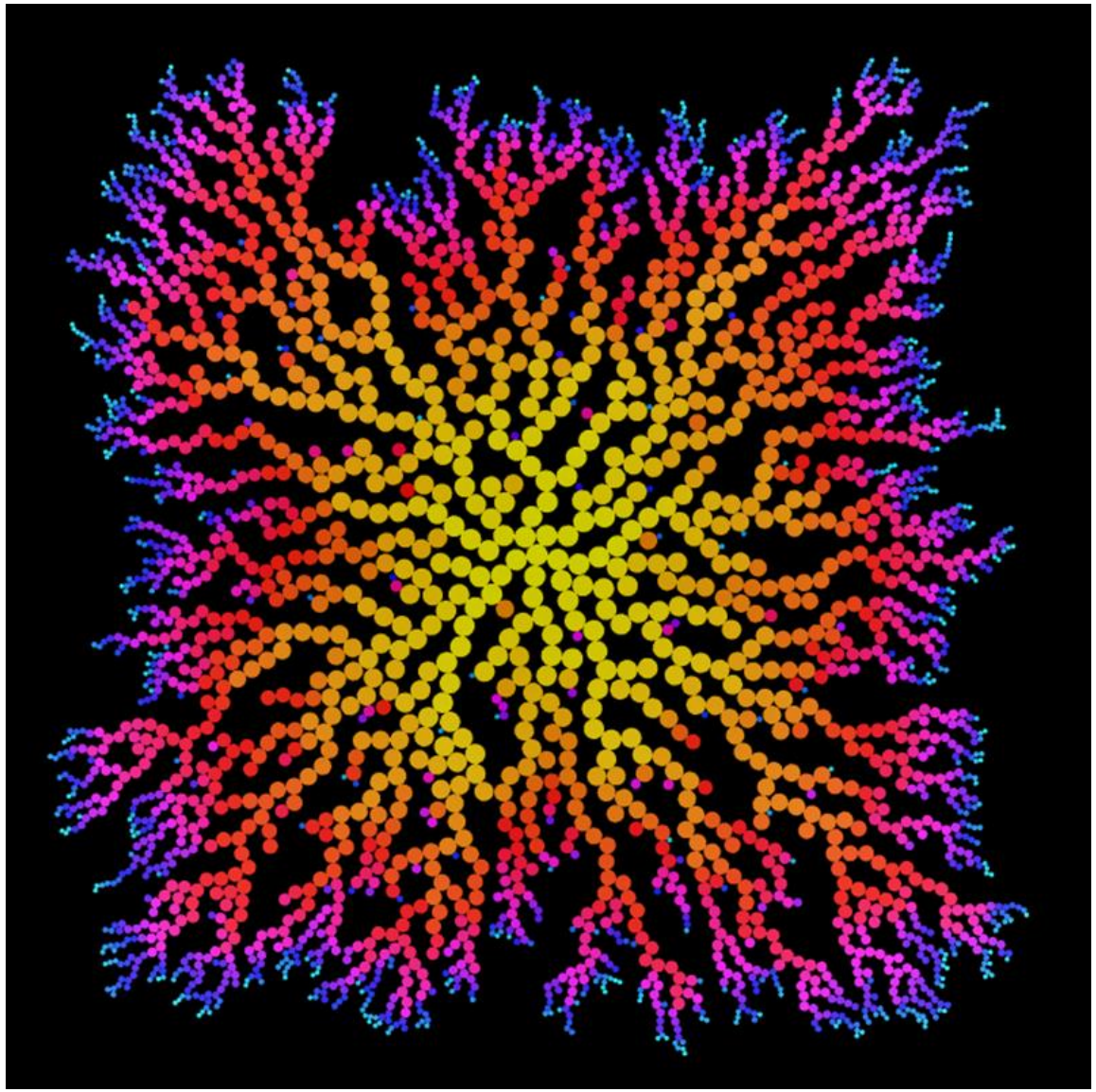


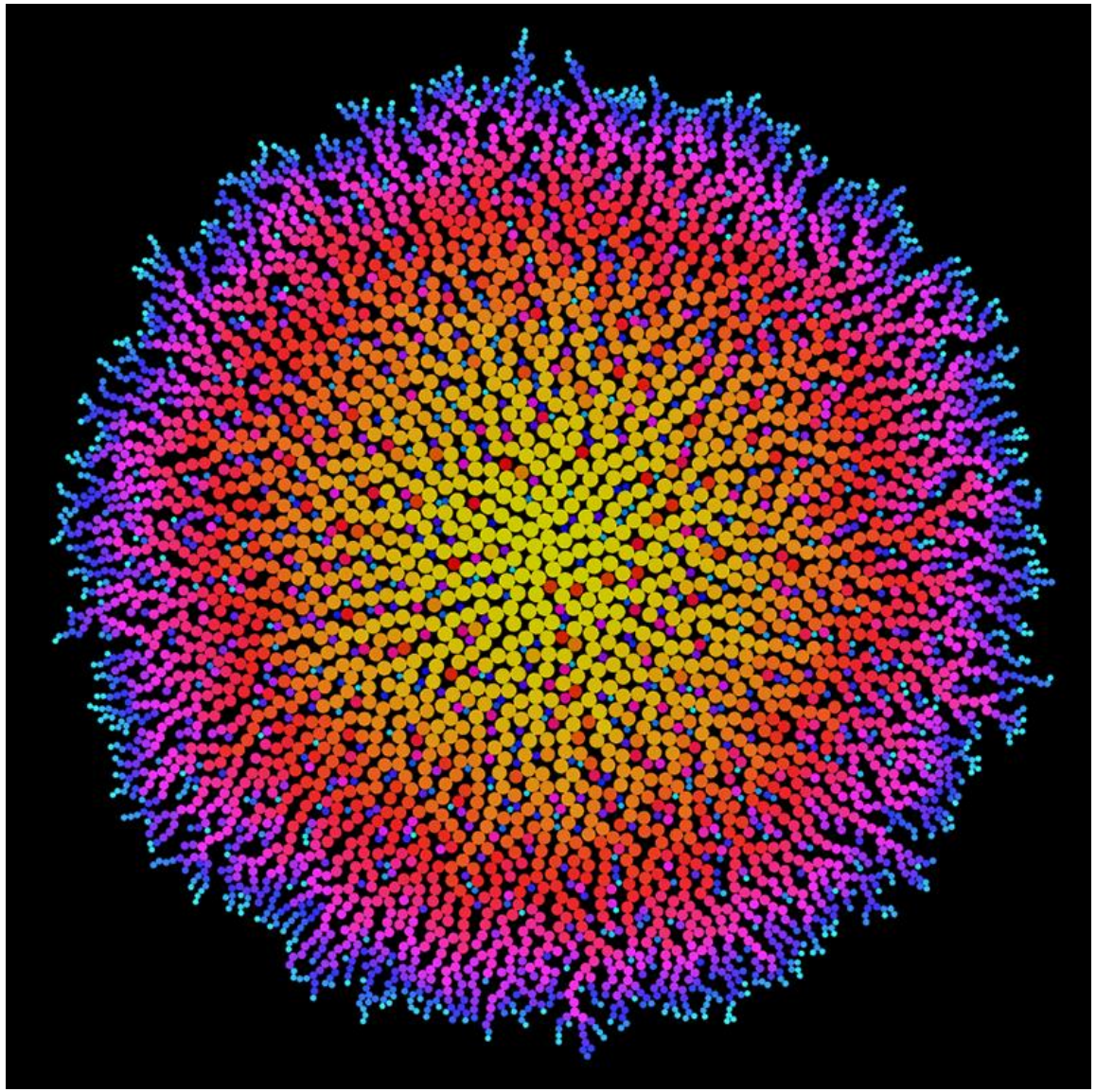


DIFFUSION LIMITED AGGREGATION

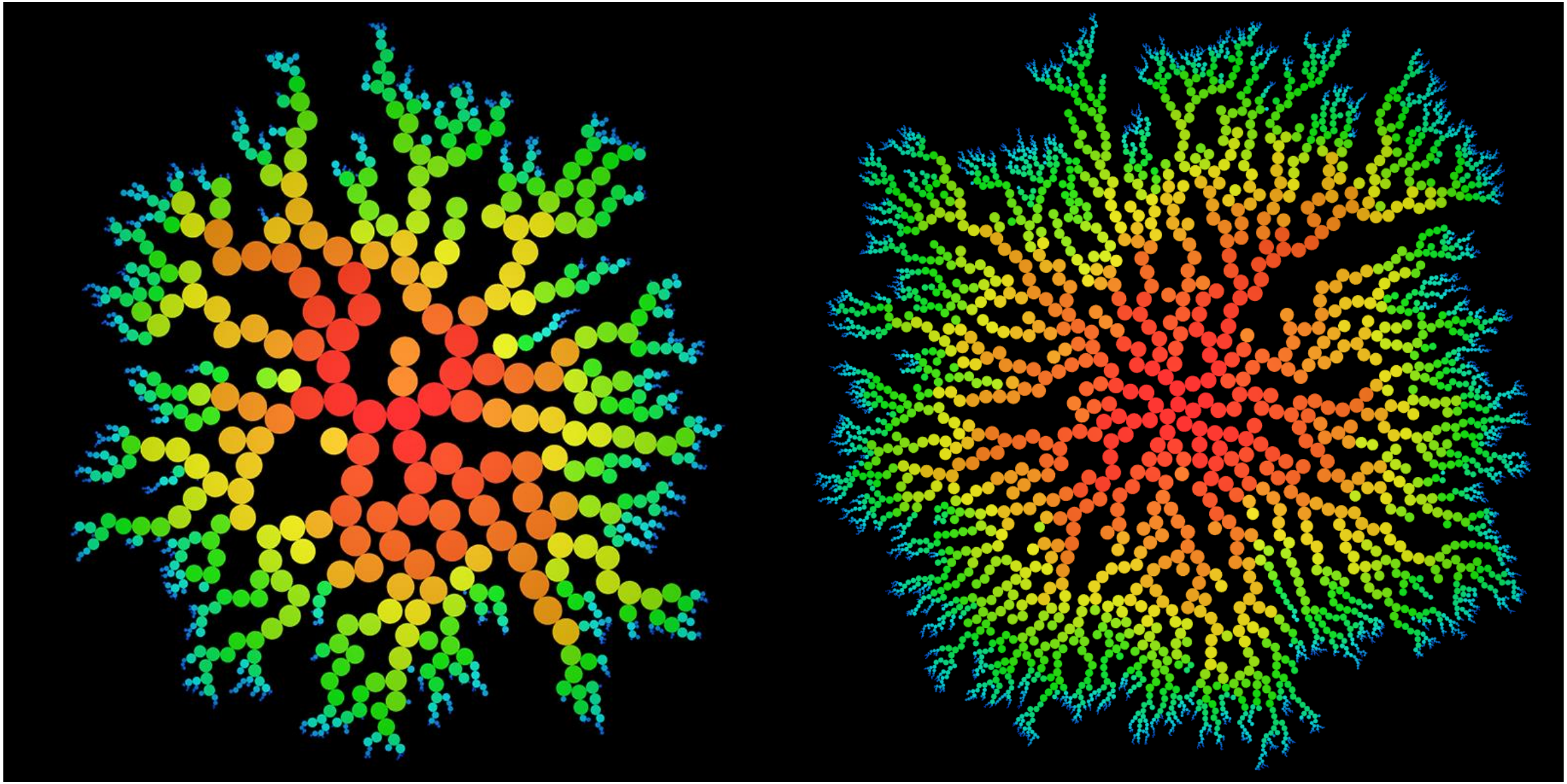


DIFFUSION LIMITED AGGREGATION

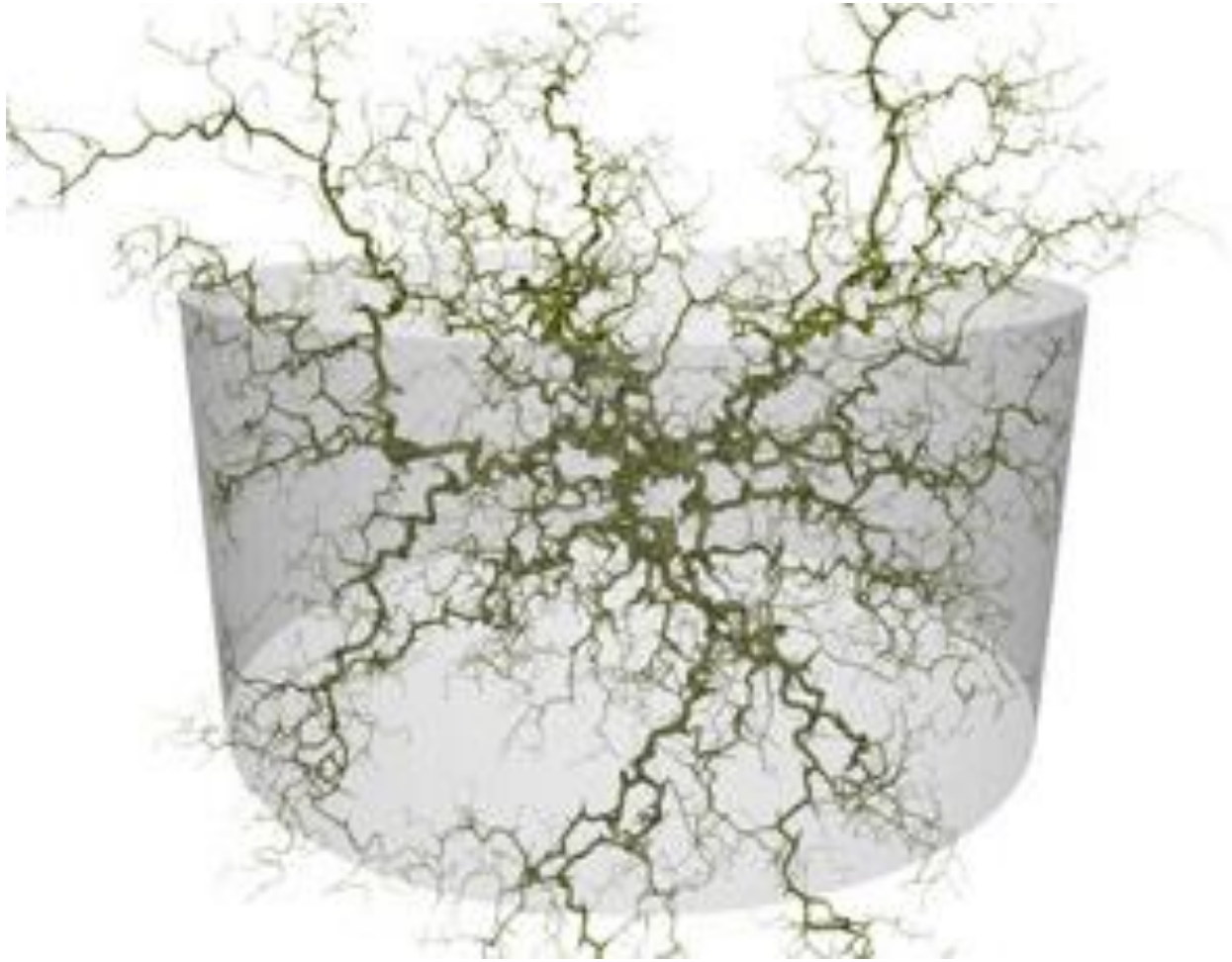




DIFFUSION LIMITED AGGREGATION



DIFFUSION LIMITED AGGREGATION



DIMENSÃO FRACTAL

- Existem dimensões entre 1 e 2?
  - Teoricamente e hipoteticamente, sim, é possível existir dimensão entre 1 e 2, a partir de uma teoria fractal.
  - Fica a recomendação do vídeo sobre cálculo de dimensão fractal, que é bastante usado na parte de análise de imagens: https://www.youtube.com/watch?v=gB9n2gHsHN4&t=613s

EXERCÍCIO

- Reimplemente os algoritmos vistos em sua linguagem de programação favorita.

- É possível fazer variações para gerar mapas diferenciados.
  - Também seria interessante (e possível) para o projeto da disciplina, reprogramar algum desses algoritmos em CUDA ou OpenCL.

- Desafio:
  - Fazer uma versão do RandomWalk com variações de cinza.
  - Alterar os algoritmos aprendidos para considerar diferentes elementos como, por exemplo, areia, grama, mar, etc. Fazer mais de uma camada de crescimento.