

## ◆ Entendendo o Algoritmo Diamond-Square

 Objetivo

Gerar um terreno proceduralmente que pareça **natural** - com montanhas, vales e transições suaves entre altitudes.

## Conceito fundamental

O algoritmo trabalha em **níveis de detalhe**:

1. Começa com apenas 4 pontos (cantos)
  2. Calcula pontos intermediários usando **média dos vizinhos**
  3. Adiciona **variação aleatória** para naturalidade
  4. Repete o processo em escalas cada vez menores

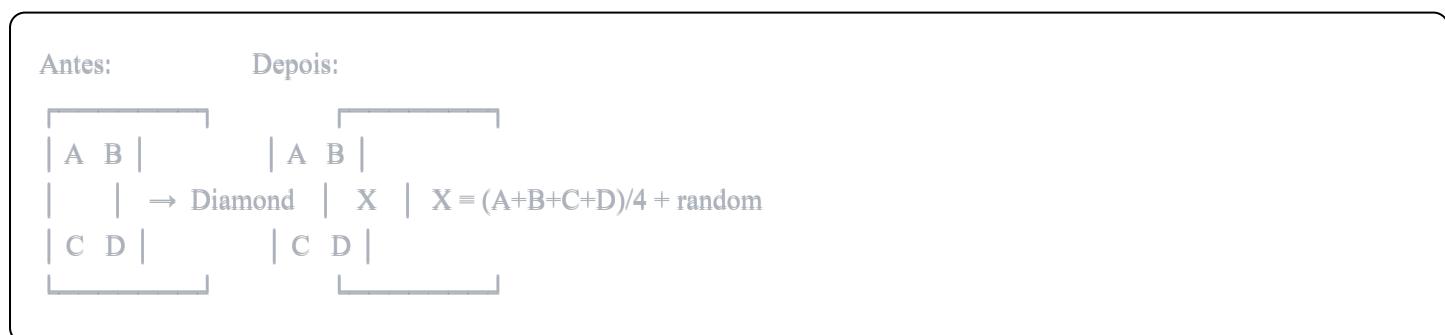
**Analogia:** É como esculpir uma montanha:

- Primeiro faz o formato geral (grosso)
  - Depois adiciona detalhes médios (encostas)
  - Por fim, adiciona texturas finas (rochas pequenas)

## As duas etapas

## **Etapa 1: Diamond (Losango)**

Calcula o **centro** de cada quadrado usando os **4 cantos**:



Código:

cpp

```

double media = (cantoA + cantoB + cantoC + cantoD) / 4.0;
double variacao = random() * amplitude;
centro = media + variacao;

```

## Etapa 2: Square (Quadrado)

Calcula os **pontos médios das bordas** usando vizinhos ortogonais:

Antes:	Depois:
A B	A 1 B   1 = (A+B+X)/3 + random
X   → Square   2 X 3   2 = (A+C+X)/3 + random	
C D	C 4 D   3 = (B+D+X)/3 + random

| 4 = (C+D+X)/3 + random

## Código:

```

cpp

double media = (vizinho1 + vizinho2 + centro) / 3.0;
double variacao = random() * amplitude;
ponto = media + variacao;

```

## ⟳ Iterações sucessivas

### Iteração 0 (inicial): 5×5

