

# 3D FRACTAL ENGINE

Computação Visual - 2024/2025

André Oliveira - 107637

Bernardo Figueiredo - 108073

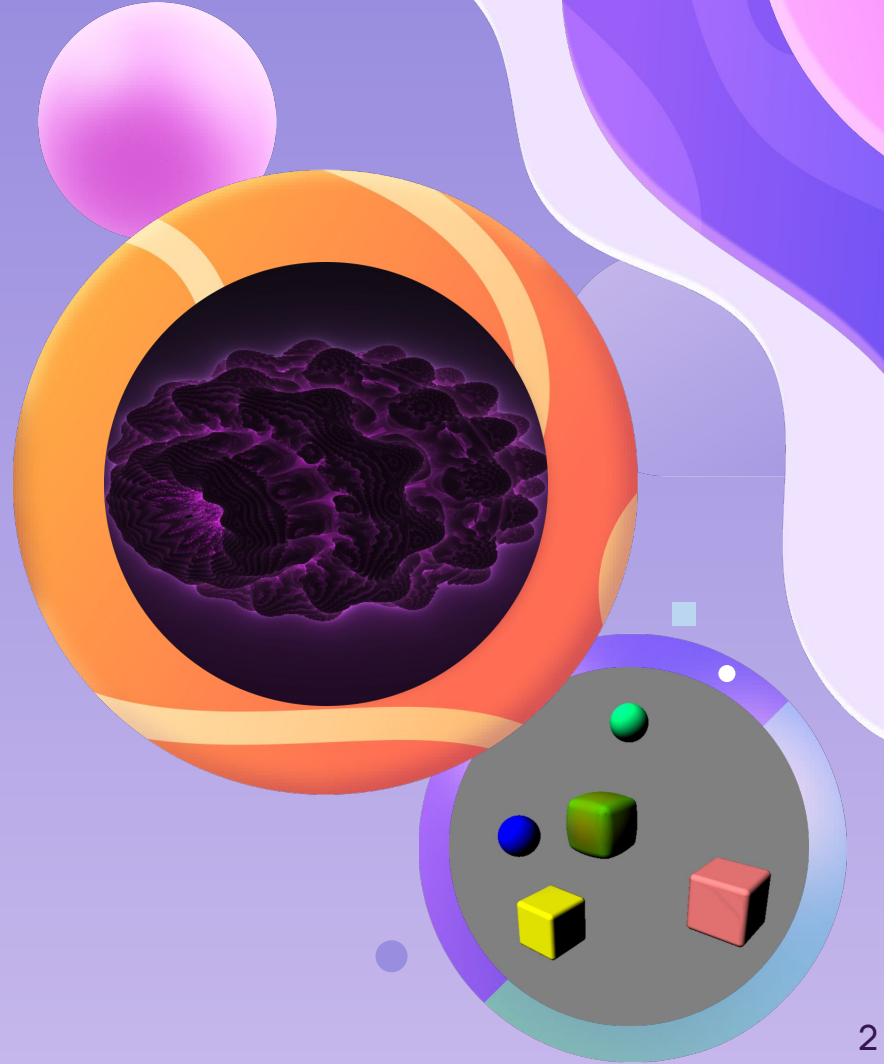
Joaquim Rosa - 109089

# CONTEXTO

1. Utilização de técnicas de Computação Visual para renderizar fractais 3D em tempo real.

- Ray Marching
- Realistic Illumination
- Soft e Hard Shadows
- ...

2. Criação de um ambiente interativo, para os utilizadores explorarem essas técnicas em objetos 3D mais básicos, permitindo o ajuste de variáveis e visualizando o impacto dessas alterações em tempo real.



# OBJETIVOS

## GERAIS

01

Explorar conceitos fundamentais de Computação Visual

02

Introduzir conceitos avançados de renderização - Ray Marching e Signed Distance Functions

## ESPECÍFICOS

01

Implementar fractais 3D

02

Explorar técnicas avançadas de iluminação

03

Criar um ambiente interativo básico

# ABORDAGEM ADOTADA

PYTHON



PYGAME



DEAR PYGUI

OPENGL



# FUNÇÕES DE DISTÂNCIA

A essência do Ray Marching está nas **Signed Distance Functions (SDFs)**.  
Uma SDF é uma função que, dado um ponto no espaço, retorna:

Uma **distância positiva**, se o ponto está fora do objeto.

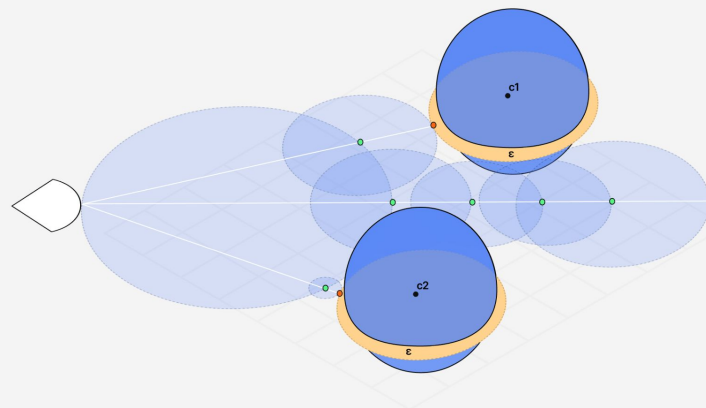
Uma **distância zero**, se o ponto está na superfície do objeto.

Uma **distância negativa**, se o ponto está dentro do objeto.

# RAY MARCHING

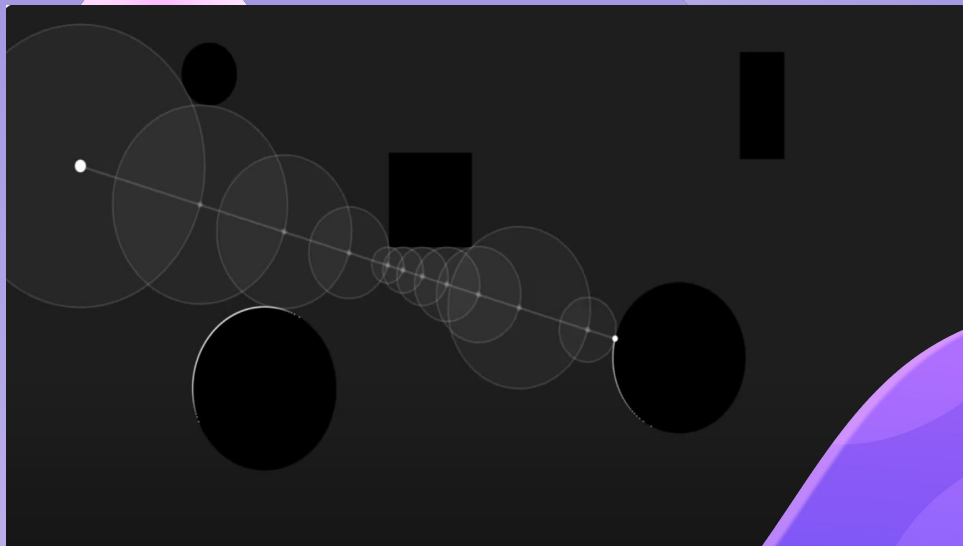
É uma técnica de renderização que se destaca pela sua capacidade de criar cenas tridimensionais altamente detalhadas, especialmente aquelas que não podem ser representadas por coleções de vértices, como fractais.

Utiliza **funções de distância (Distance Functions)** para determinar a proximidade de um ponto no espaço tridimensional a um objeto ou superfície.



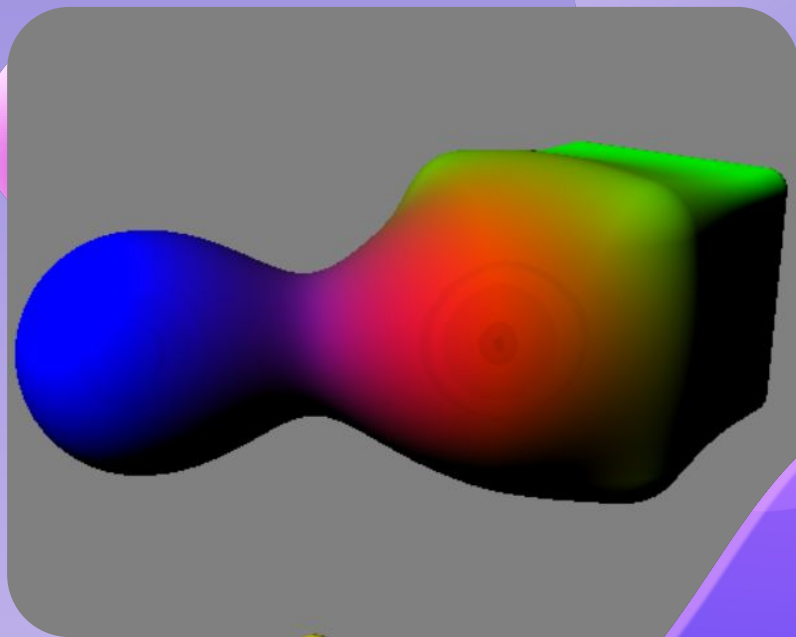
# RAY MARCHING

- O Ray Marching avança incrementalmente ao longo de um raio.
- A cada passo, o algoritmo consulta todas as funções de distância e escolhe o menor valor até a superfície mais próxima.
- Esse valor é usado para determinar o tamanho do próximo passo, “marchando” o raio até que:
  - Encontre uma superfície
  - Saia da cena
  - Ultrapasse um limite de iterações



# BLEND

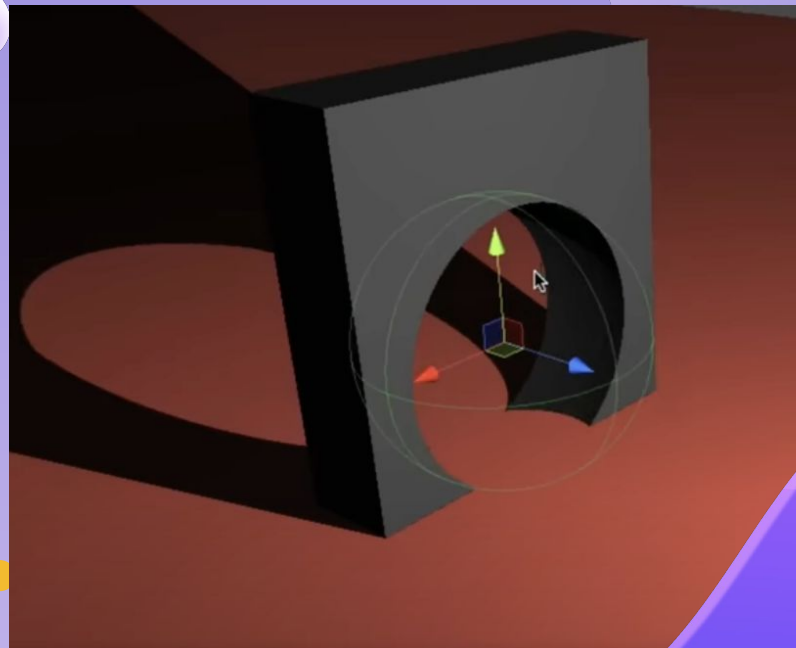
Ray Marching permite combinar suavemente diferentes formas geométricas, criando transições fluidas entre elas. Isso é alcançado modificando as **funções de distância** das geometrias individuais para que as suas bordas se sobreponham e se misturem gradualmente.





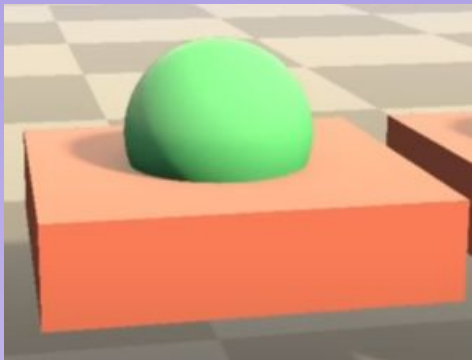
# SUBTRACTION MASK

Operação que permite remover partes onde dois (ou mais) objetos se intersectam, normalmente usado com uma primitiva visível e outra invisível.  $(\max(-a, b))$



# INTERSECTION MASK

Operação inversa do subtract mask, normalmente usada com dois (ou mais) objetos invisíveis, o resultado é que apenas a interseção é visível. ( $\max(a,b)$ )



The background is a solid light purple color. It is decorated with various abstract elements: large organic shapes in shades of pink and purple in the top-left and bottom-right corners; a large pink sphere in the top-left; a large teal and purple circle in the top-right; stylized leaves in purple and blue in the bottom-left; and several small geometric shapes (squares, circles, dots) in pink, yellow, blue, and white scattered across the canvas.

# DEMO

# ANÁLISE CRÍTICA

## Objetivos Alcançados

- Aprender Ray Marching e fazer um visualizador de fractais 3d

## Trabalho Futuro

- Subtraction Mask
- Intersection Mask
- Reflexos
- Implementar o projeto em Vulkan para troca de biblioteca gráfica em tempo real
- Estatísticas do desempenho para comparação
- Exploração da cena com um jogador móvel