

Aplicação para comparar os algoritmos de reta e círculo

Arthur Naves Pedroso

RA00275136

A API gráfica usada para o desenvolvimento da aplicação foi o DirectX11. O DirectX11 é composto de diversas APIs que possuem diferentes funcionalidades como Desenho 2D, Desenho 3D, I/O, “machine learning”, etc; no caso da atividade só foram utilizadas as APIs Direct3D e DXGI. A API Direct3D foi utilizada para desenhar os gráficos da atividade, ela é uma API destinada para desenhos 3D, ou seja, não é possível desenhar imagens “rasterizadas” ou acender pixels, ao invés disso, planos colocados sobre um espaço tridimensional são utilizados, e a posição desses elementos no eixo de profundidade do sistema de coordenadas é fixado para 0 em todos os objetos. Isso é exemplificado na imagem abaixo que demonstra um modelo 3D de um triângulo usado no projeto.

```
const Vertex Triangle[] =  
{  
    { 0.0f, 1.0f, 0.0f },  
    { 1.0f, -1.0f, 0.0f },  
    { -1.0f, -1.0f, 0.0f },  
};
```

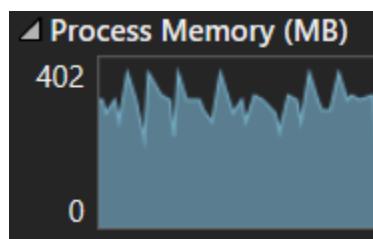
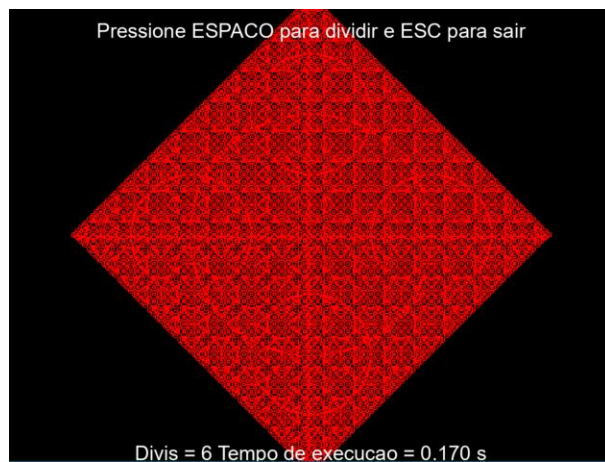
Para conseguir simular um pixel colorido dentro do Direct3D, um quadrado tridimensional é enviado para a placa de vídeo. Esse processo necessita de uma estrutura de dados que contenha no mínimo 6 vértices, cada um contendo 3 pontos flutuantes para representar a posição do elemento sobre o espaço tridimensional. Cada vértice também contém 4 pontos flutuantes adicionais que representam a cor (RGBA) aplicada no “Vertex Shader”.

Com base nesse uso da API são implementados 2 sistemas de renderização, o primeiro, apelidado de “DrawSoftware”, faz os desenhos dos exercícios com os Pixels acima, logo, uma reta traçada na tela será composta por múltiplos quadrados tridimensionais que juntos assumem a forma de uma reta. Já o segundo sistema, apelidado de “DrawHardware”, faz os desenhos com base em geometrias já montadas (a imagem a direita demonstra um círculo usado nesse sistema), assim, uma reta traçada por esse sistema será composta por apenas 1 quadrado tridimensional escalado e rotacionado para assumir a forma necessária para o exercício.

```
{ 0.9750000000f, 0.0000000000f, 0.0000000000f },
{ 0.7071068000f, 0.7071068000f, 0.0000000000f },
{ 1.0000000000f, 0.0000000000f, 0.0000000000f },
{ 0.9750000000f, 0.0000000000f, 0.0000000000f },
{ 0.6894291000f, 0.6894291000f, 0.0000000000f },
{ 0.7071068000f, 0.7071068000f, 0.0000000000f },
{ 0.6894291000f, 0.6894291000f, 0.0000000000f },
{ -0.00000004371f, 1.0000000000f, 0.0000000000f },
{ 0.7071068000f, 0.7071068000f, 0.0000000000f },
{ 0.6894291000f, 0.6894291000f, 0.0000000000f },
{ -0.00000004262f, 0.9750000000f, 0.0000000000f },
{ -0.00000004371f, 1.0000000000f, 0.0000000000f },
{ -0.00000004262f, 0.9750000000f, 0.0000000000f },
{ -0.7071068000f, 0.7071068000f, 0.0000000000f },
{ -0.00000004371f, 1.0000000000f, 0.0000000000f },
{ -0.00000004262f, 0.9750000000f, 0.0000000000f },
{ -0.6894291000f, 0.6894291000f, 0.0000000000f },
{ -0.7071068000f, 0.7071068000f, 0.0000000000f },
{ -0.6894291000f, 0.6894291000f, 0.0000000000f },
{ -1.0000000000f, -0.00000008742f, 0.0000000000f },
{ -0.7071068000f, 0.7071068000f, 0.0000000000f },
{ -0.6894291000f, 0.6894291000f, 0.0000000000f },
{ -0.9750000000f, -0.00000008524f, 0.0000000000f },
{ -1.0000000000f, -0.00000008742f, 0.0000000000f },
{ -0.9750000000f, -0.00000008524f, 0.0000000000f },
{ -0.7071066000f, -0.7071069000f, 0.0000000000f },
{ -1.0000000000f, -0.00000008742f, 0.0000000000f },
{ -0.9750000000f, -0.00000008524f, 0.0000000000f },
{ -0.6894290000f, -0.6894292000f, 0.0000000000f },
{ -0.7071066000f, -0.7071069000f, 0.0000000000f },
{ -0.6894290000f, -0.6894292000f, 0.0000000000f },
{ 0.00000001192f, -1.0000000000f, 0.0000000000f },
{ -0.7071066000f, -0.7071069000f, 0.0000000000f },
{ -0.6894290000f, -0.6894292000f, 0.0000000000f },
{ 0.00000001163f, -0.9750000000f, 0.0000000000f },
{ 0.00000001192f, -1.0000000000f, 0.0000000000f },
{ 0.00000001163f, -0.9750000000f, 0.0000000000f },
{ 0.7071070000f, -0.7071065000f, 0.0000000000f },
{ 0.00000001192f, -1.0000000000f, 0.0000000000f },
{ 0.00000001163f, -0.9750000000f, 0.0000000000f },
{ 0.6894293000f, -0.6894289000f, 0.0000000000f },
{ 0.7071070000f, -0.7071065000f, 0.0000000000f },
{ 0.6894293000f, -0.6894289000f, 0.0000000000f },
{ 1.0000000000f, 0.00000017485f, 0.0000000000f },
{ 0.7071070000f, -0.7071065000f, 0.0000000000f },
{ 0.6894293000f, -0.6894289000f, 0.0000000000f },
{ 0.9750000000f, 0.00000017047f, 0.0000000000f },
{ 1.0000000000f, 0.00000017485f, 0.0000000000f },
```

Como resultado das características dos dois sistemas de renderização, o sistema “DrawHardware” acaba sendo mais rápido e muito mais eficiente no uso da memória do que o “DrawSoftware”. Os testes realizados são apresentados nas imagens abaixo:

Sistema "DrawSoftware"



Sistema "DrawHardware"

