

Otimização para partículas em Engines Gráficas e Suas Dificuldades

Computação Gráfica e o caminho para o real aprendizado

José Carlos de Almeida Machado

Motivação

Escopo

- Usar/Criar um ambiente gráfico para poder aplicar simulações com alto grau de controle
- Permitir aplicação de shaders em modelos para adicionar nas simulações
- Permitir interação entre objetos, focando em partículas
- Otimizar diversas funcionalidades utilizando CUDA
- Aprimorar UI e UX

Realidade

- Cadê documentação?
- Como funciona isso em C++?
- Como funciona isso em OpenGL?
- Como gerenciar isso com cmake?
- Como funcionava produto escalar de matriz mesmo?
- Como funcionava produto vetorial mesmo?
- VAO, VBO, EBO, UBO e quantos mais buffers??
- Como o CUDA não está rodando junto com o código pronto????
- Que que é Sol e sair de casa??????
- To com sono

Escopo

- Abrir a janela mais simples no OpenGL
- Fazer com que pontos sejam desenhados na tela aleatoriamente
- Atualizar a posição dos pontos sequencialmente e utilizando kernels

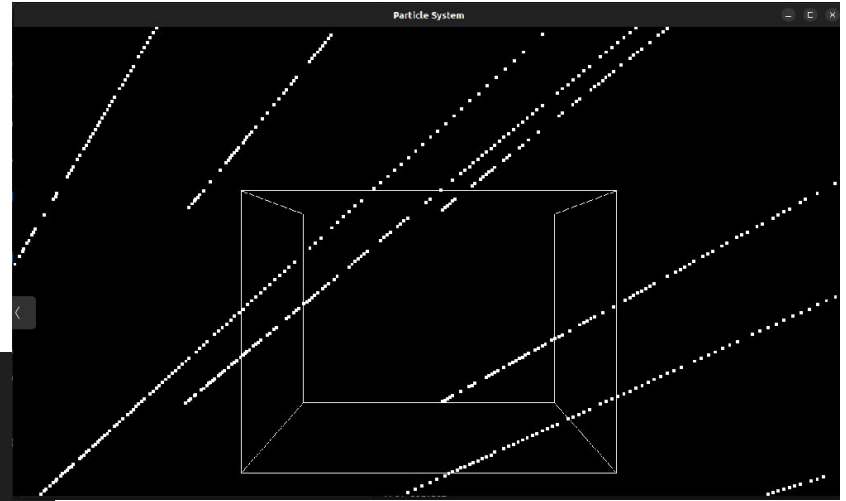
Simulação

Realizar simulações com partículas muitas vezes com:

- 100000 pontos
 - CPU sequencial
 - GPU paralelo

CPU - Sequential

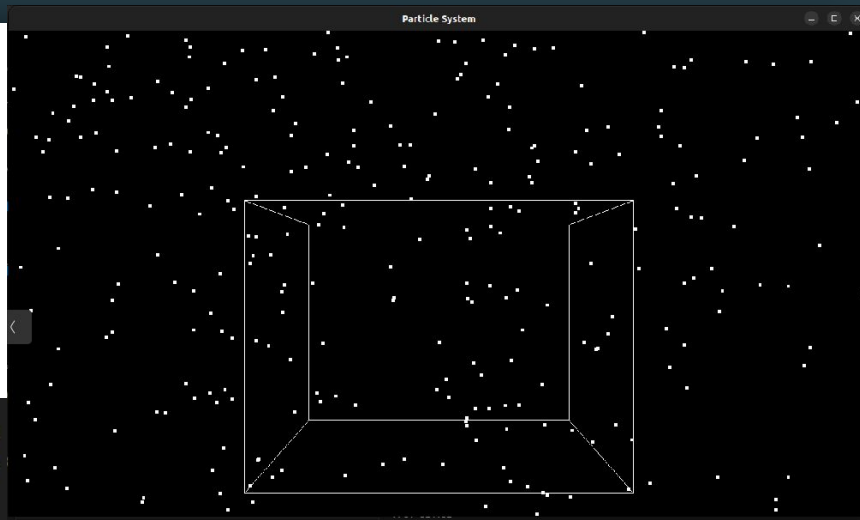
```
void update_particles_host(float *part, float dt) {  
    for (int i = 0; i < NUM_PARTICLES*9; i+=9) {  
        part[i+7] = -9.8f;;           // Apply gravity  
        // Update velocity  
  
        part[i+3] += part[i+6] * dt;  
        part[i+4] += part[i+7] * dt;  
        part[i+5] += part[i+8] * dt;  
        // Update position  
        part[i+0] += part[i+3] * dt;  
        part[i+1] += part[i+4] * dt;  
        part[i+2] += part[i+5] * dt;  
    }  
}
```



```
-----  
FPS: 506.096  
FPS: 580.75  
FPS: 466.873  
FPS: 516.05  
FPS: 561.692  
FPS: 491.32  
FPS: 451.564  
FPS: 576.949  
FPS: 451.824  
□
```

GPU - Paralelo

```
global__ void update_particles_kernel(float *part, float dt) {  
    int index = blockIdx.x * blockDim.x + threadIdx.x;  
    int i = index * 9;  
    int stride = blockDim.x * gridDim.x;  
    for (; i < NUM_PARTICLES; i += stride){  
        part[i+7] = -9.8f;           // Apply gravity  
        // Update velocity  
        part[i+3] += part[i+6] * dt;  
        part[i+4] += part[i+7] * dt;  
        part[i+5] += part[i+8] * dt;  
  
        part[i+0] += part[i+3] * dt;  
        part[i+1] += part[i+4] * dt;  
        part[i+2] += part[i+5] * dt;  
    }  
}
```

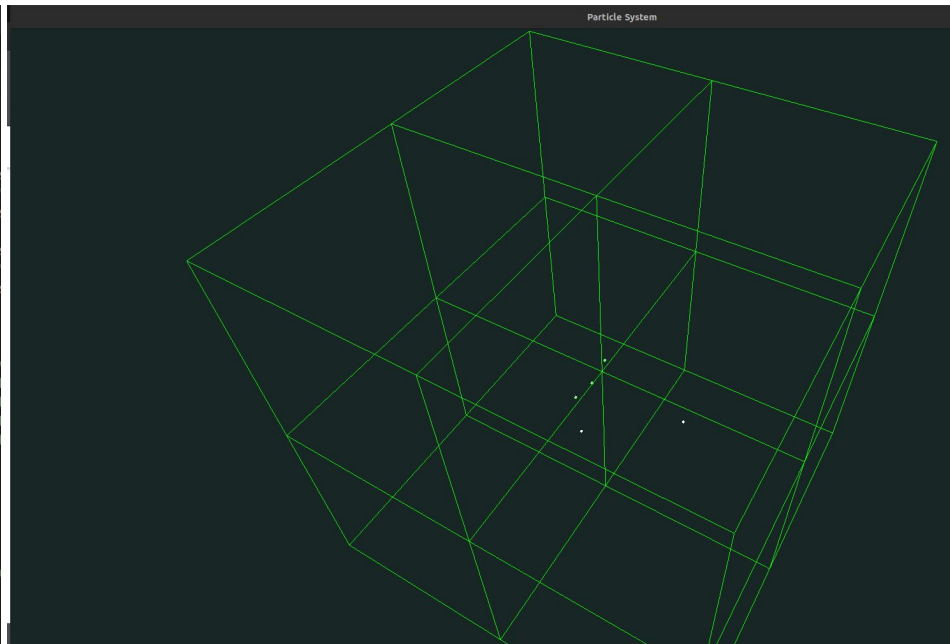
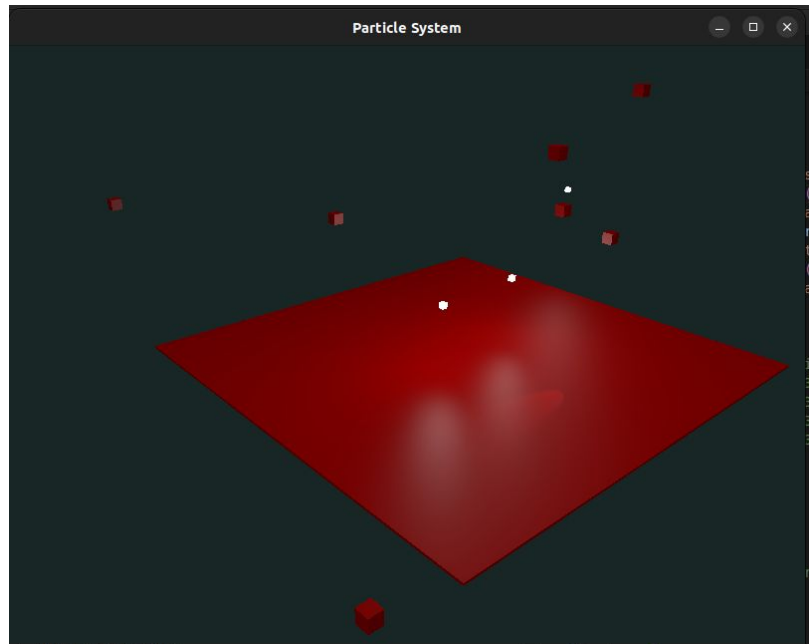


```
FPS: 253.024  
FPS: 241.168  
FPS: 185.873  
FPS: 271.022  
FPS: 318.676  
FPS: 314.985  
FPS: 331.131  
FPS: 260.685
```



Ou seja...

Frutos gerados



Obrigado pela atenção!

In memoriam



Mika

~2012 - 17/12/2023