

UNIVERSIDADE FEDERAL DE PELOTAS

FACULDADE DE COMPUTAÇÃO



LUCAS SCHNEIDER LUDWIG

SAMUEL STARKE

RELATÓRIO

APLICAÇÃO GRÁFICA DE SIMULAÇÃO DE PARTÍCULAS

PELOTAS – RS

2024

Descrição da Aplicação

O projeto consiste em um simulador de partículas 2D, que utiliza uma combinação de linguagens de programação para implementar a física das partículas e para visualização gráfica. A física de interação entre as partículas e a aplicação da força gravitacional são implementadas em C, enquanto a visualização e animação das partículas são realizadas em Python.

Linguagens Utilizadas e Implementação

O código é dividido em duas partes principais, cada uma sendo implementada em uma linguagem diferente:

- **C:**
 - Responsável pela lógica de física do simulador. As funções que calculam as interações entre as partículas, incluindo atualização de posições, aplicação de força gravitacional, e verificação de colisões, estão todas implementadas em C. Essas funções são definidas no arquivo `particles.c`, com suas declarações em `particles.h`.
- **Python:**
 - Responsável pela visualização e animação do comportamento das partículas. O script Python (`particles_app.py`) utiliza as funções implementadas em C para calcular a dinâmica das partículas e, em seguida, exibe esses cálculos em uma animação 2D. O Python faz isso usando bibliotecas como `numpy` e `matplotlib`.

Interface entre C e Python

A interface entre as duas linguagens é realizada através da biblioteca dinâmica (`particles.so`), que é compilada a partir do código C. Essa biblioteca é carregada no script Python usando o módulo `ctypes`, que permite a chamada de funções C diretamente a partir do código Python.

- **Módulo `ctypes`:**
 - No script Python, `ctypes` é usado para carregar a biblioteca C e para definir a estrutura `Particle`, que corresponde à estrutura definida em C. A função `apply_gravity`, por exemplo, é mapeada em Python para a função correspondente em C, permitindo que o Python manipule as partículas simuladas pela lógica implementada em C.

Processo de Compilação:

O `Makefile` incluído no projeto automatiza a compilação do código C, gerando a biblioteca `particles.so`, que pode ser utilizada no ambiente Linux. A compilação é feita com o GCC (GNU Compiler Collection), e o arquivo resultante é um objeto compartilhado (`.so`) que pode ser carregado e executado por Python.

Métodos Utilizados para a Interface

Os métodos utilizados para permitir a interface entre as duas linguagens incluem:

- **Compilação de Código C com biblioteca compartilhada:**
 - O código C é compilado em uma biblioteca compartilhada (`.so`), que pode ser carregada em tempo de execução pelo script Python.
- **Uso de `ctypes` em Python:**
 - `ctypes` é utilizado para definir a interface entre Python e C, mapeando as funções e estruturas definidas em C para que possam ser usadas no ambiente Python.

Exemplos gráficos da aplicação:



