```cpp
#include <iostream>
#include "Body.hpp"
#include <SFML/Graphics/CircleShape.hpp>
#include <SFML/Graphics/RectangleShape.hpp>
#include <sstream>
#include <vector>
#include <cmath>

using namespace std;

int main(int argc, const char * argv[])
{
    if(argc != 3)
    {
        cout << "Wrong number of arguments" << endl;
        return -1;
    }

    double total_t = atof(argv[1]);
    double delta_t = atof(argv[2]);
    double t = 0;

    int num_body;
    double universe_radius;

    cin >> num_body >> universe_radius;

    int window_size = 500;
    sf::RenderWindow window(sf::VideoMode(window_size, window_size), "Universe");
    window.setFramerateLimit(1);

    vector<Body*> body;

    for(int i = 0; i < num_body; i++)
    {
        body.push_back(new Body);
        cin >> (*body[i]);
        (*body[i]).setWindowSize(window_size);
        (*body[i]).setUnivSize(universe_radius);
    }

    double gravity = 6.67e-11;

    while (window.isOpen())
    {
        sf::Event event;
        while (window.pollEvent(event))
        {
            if (event.type == sf::Event::Closed){
                window.close();
            }
        }

        if (t < total_t)
        {
            for(int i = 0; i < num_body; i++)
            {
                double force_x = 0;
                double force_y = 0;

                for(int j = 0; j < num_body; j++)
                {
                    if(i != j)
                    {
                        double delta_x = body[j]->getPosX() - body[i]->getPosX();
                        double delta_y = body[j]->getPosY() - body[i]->getPosY();
                        double r = sqrt(pow(delta_x, 2) + pow(delta_y, 2));
                        double force = ((gravity * body[i]->getMass() * body[j]->getMass())
                         / pow(r, 2));

                        force_x = force_x + force * delta_x / r;
                        force_y = force_y + force * delta_y / r;
                    }
                }
                body[i]->step(delta_t, force_x, force_y);
            }
            t += delta_t;
        }

        window.clear(sf::Color::Black);
        for(int i = 0; i < num_body; i++)
        {
            window.draw(*body[i]);
        }
        window.display();
    }

    for (int i = 0; i < num_body; i++)
    {
        cout << (*body[i]) << endl;
    }

    return 0;
}

```