

c3particles: Modeling a Particle System in C++

Rosalie Kletzander

Practical Course "Advanced Software Development with Modern C++"
Summer Semester 2018
MSc Tobias Fuchs
Institute for Computer Science
Ludwig-Maximilians-Universität München

Abstract. Particle systems are used in many different areas, ..animation, simulation, research, No matter the area of application, the basic rules governing these systems are the same: the laws of physics. C3particles (cpp particles) implements a model of a particle system in C++ that separates the physical concepts and laws from the underlying graphics library. This enables a mathematical formulation of the forces influencing the particles.

1 Introduction

2 Short Recap of Physics

In order to be able to model the physics of a particle system, it is necessary to first understand the basic rules of motion.

Newton's First Law of Motion states that

Every object in a state of uniform motion tends to remain in that state of motion unless an external force is applied to it.

This means that an object will not move unless it is accelerated by a force, which brings us to Newton's Second Law of Motion:

The relationship between an object's mass m , its acceleration a , and the applied force F is $F = m * a$.

With this information it is possible to calculate the acceleration of an object by dividing the applied force by the object's mass. The next step is deriving the velocity and location of the object per time step by integration.

The velocity of an object can be calculated by integrating the acceleration over time t .

$$\vec{v}(t) = \int (\vec{a}) dt = \vec{a} * t + C_v$$

C is the integration constant, in this case it is equal to the velocity of $t-1$ for discrete time steps. Integrating the velocity over t yields the location.

$$\vec{s}(t) = \int (\vec{v}) dt = \int (\vec{a} * t + C_v) dt = \frac{\vec{a} * t^2}{2} + C_s$$

Analogous to C_v , C_s is equal to the location at $t-1$.

These formulas serve as the foundations of the physical model of the particle system.

3 Modeling the Particle System

- elements of the system: particle (newtonian object)
- force
- \rightarrow concepts
- particle is passive, it doesn't "care" where the force is coming from \rightarrow need some way to apply a force to a particle

4 Implementation

- system diagram
- particle system would be boring without visualization, however, this is not part of the formal model and so it is also a separate module in the code
- OpenGL is used, however any graphics library could be utilized with an appropriate interface (Particle Renderer)