Building a Physics Engine

CS-116B: Computer Graphics Algorithms
Spring 2018

Components of a Physics Engine

A generic physics engine contains the following components:

- "Physics models"
- "Simulated objects manager"
- "Collision detection engine"
- "Collision response module"
- "Force effectors"
- "Numerical integrator"
- "Game engine interface"

Source: *Physics for Game Developers*, pp. 283

Physics Engine: Components we will use

- **Physics models**:
  We will be using simple spheres, line segments, and polygons for our simulations.

- **Collision detection and collsion response**:
  We will use this to determine when a hanging rope collides with an inanimate sphere. We will also be using collision detection to determine when an animated sphere is colliding with hanging cloth.

- **Force effectors**:
  We will be using this to apply wind forces and gravity to hanging rope and cloth.

- **Numerical integrator**:
  We will be using this algorithm to animate the rope and cloth and give them elastic (somewhat) properties.

Numerical integrator

- We will be using Verlet integration to compute the displacement of interconnected particles connected together via weightless springs (Mass Spring Model).
- For simulating rope, the particles will be connected linearly like a chain.
- For simulating cloth, the particles will be connected in a mesh-like array.
- For rope and cloth simulation we will also be using the Mass Spring Model.

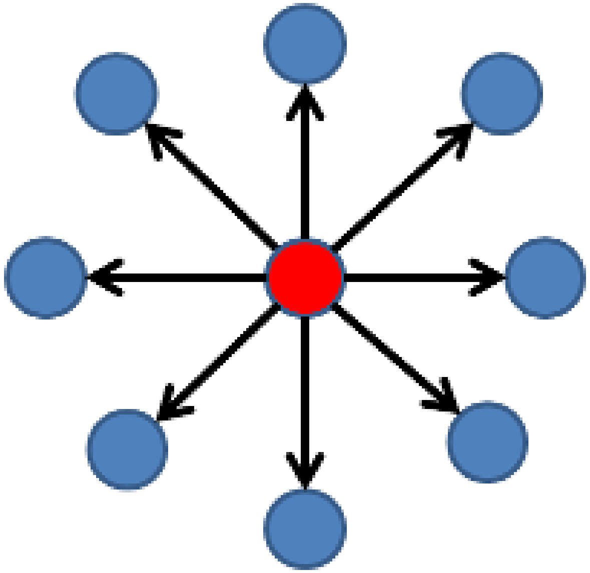
Particle Physics

- With the exception of Marching Squares, our assignments involve the displacement of particles.
- Note: For simplicity in developing our physics engine, you do not need to be concerned with transfer of momentum or conservation of energy in any of our projects.

Particle Physics: Assignment 2

- In assignment 2, you will be creating an explosion.
- The particles will emanate from a center point.
- The particles will move in a random direction but the motion will remain straight.
- There is no gravity or other forces to influence trajectory of each particle.
- You will compute the displacement over time. Each particle will be assigned a randomly generated, pre-defined acceleration. You will compute displacement for each particle at time, t using: $s=(1/2)*at^2$

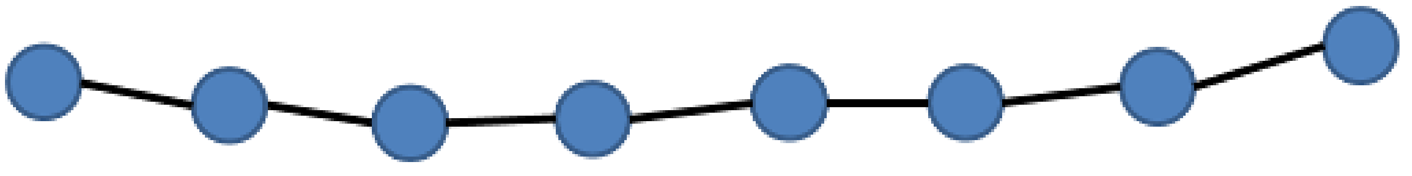
Particle Physics: Assignment 2



Particle Physics: assignment 3

- In assignment 3, you will be simulating rope.
- The particles will be constrained to each other via weightless springs.
- The particles will have a mass, m.
- The particles will be affected by gravity. The particles are also affected by spring forces.
- You will use Newton's second law of motion, $F=ma$ and Hooke's law: $F=kx$.
- Some of the particles will collide with a motionless and weightless sphere. *Note: You do not need to be concerned with transfer of momentum or conservation of energy during a collision.*

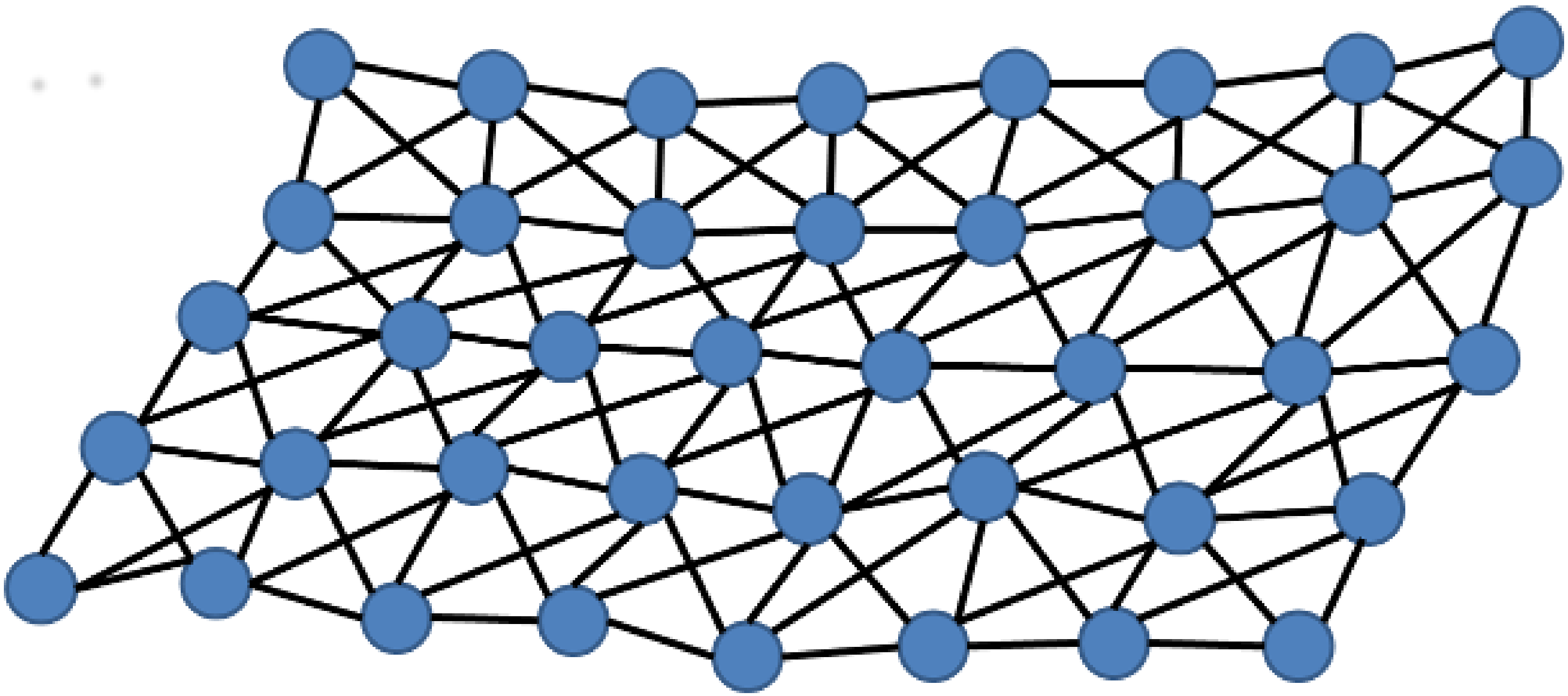
Particle Physics: assignment 3



Particle Physics: assignment 4

- In assignment 4, you will be simulating cloth.
- This is very similar to rope simulation except the particles are now connected to each other via a two-dimensional matrix of weightless springs.
- The springs are connected to particles horizontally, vertically, and diagonally.
- The springs are weightless. The particles have a mass, m.
- The particles are subject to a gravitational force. The particles are also subject to spring forces.
- You will use Newton's second law of motion, $F=ma$ and Hooke's law: $F=kx$.

Particle Physics: assignment 4



Particle Physics: assignment 5

- In assignment 5, you will be simulating cloth with wind, gravity, and collision detection. This assignment builds upon your previous work in cloth simulation.
- The springs are connected to particles horizontally, vertically, and diagonally.
- You will use Newton's second law of motion, $F=ma$ and Hooke's law: $F=kx$.
- The cloth will be subject to gravitational forces, random wind forces, and spring forces.
- The cloth will collide with a weightless, moving sphere.

Next lecture: Vector operations

- In our next lecture, we will review vectors and vector operations such as:
- vector dot product
- vector cross product
- vector magnitude
- normalizing a vector
- These vector operations will be very important in developing our physics engine. We need to develop a robust library of vector math operations.