Physics

Tuesday, November 16, 2021 12:38 PM

Big Picture: Two major approaches to animation

Key-framed

Procedural (e.g. algorithm based)
- algorithm sin/cos
- simulation (ex. physics)

Rigid Body Dynamics

→ perspective based on a "physics engine"

→ model of how objects move in response to forces

→ "passive phenomena": move only according to external forces (as opposed to animals which can move of their own accord)

→ objects are non-deformable, don't change size or shape

→ uses Newton's laws of motion

Two categories (this class)

1. Particle Systems
   → rigid bodies are points/spheres
   → only translational dynamics matter

2. 3D object simulation
   → both translation & rotation dynamics

Particles Systems:

→ simple & versatile

→ used for snow, rain, dust, fire, fireworks, confetti, hair, cloth

Simulating a particle system:

Each simulation step

1. Accumulate forces (compute net force: $F = \Sigma f_i$)
2. Take derivatives
   → compute how quantities change over time
   → compute acceleration a from net force
   $a = F/$ ← compute

→ compute how quantities change over time

$a = F/m$ ← compute acceleration a from net force F and particle mass m

$$\left.\begin{aligned} v &= f(a) \\ p &= f(v) \end{aligned}\right] \leftarrow \text{approximation called "Euler's Method"}$$

(3a) Get particle state (pos, mass, vel)

(3b) Set particle state

Typically, particles will be stored together in big array

$\boxed{\text{EX}}$ Suppose we have position, velocity, mass, and forces for each particle

$$\begin{bmatrix} x \\ v \\ f \\ m \end{bmatrix} \xrightarrow{\text{entire system looks like}}$$

particle 1	particle 2	. . . .	particle n

particle state (one particle)

<u>note</u>: x is position here

p refers to whole particle

f = forces

If $p = \begin{bmatrix} x \\ v \end{bmatrix}$, then the derivative $\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ a \end{bmatrix} = \begin{bmatrix} v \\ f/m \end{bmatrix}$

To update p each frame

$$p^{t+1} = p^{t} + \Delta t \, \dot{p}^{t}$$

where t is current frame

Δt is the time since the last frame (dt())

$$\Rightarrow \begin{bmatrix} x \\ v \end{bmatrix}^{t+1} = \begin{bmatrix} x \\ v \end{bmatrix}^{t} + \begin{bmatrix} v \\ f/m \end{bmatrix}^{t} \Delta t$$

This process of updating based on derivatives is called <u>numerical</u> <u>integration</u>

We are using <u>Euler's Method</u> (first order approximation) in this class but more accurate method exist (Midpoint Method, Runge-Kutta)

⋮

<u>Examples of Force Types</u>

Examples of Force Types:

1. Constant: ex gravity, wind
2. Pos / time dependent: flow fields, force fields (such as a wall), penalty forces (which prevents particles from intersecting)
3. Velocity dependent: drag, e.g. $f = -cv$ ($c \in \mathbb{R}$ is a constant)
4. <u>N-ary</u>: ex. springs, forces where nearby particles exert forces on each othere

$\boxed{\text{EX}}$ Particle under gravity

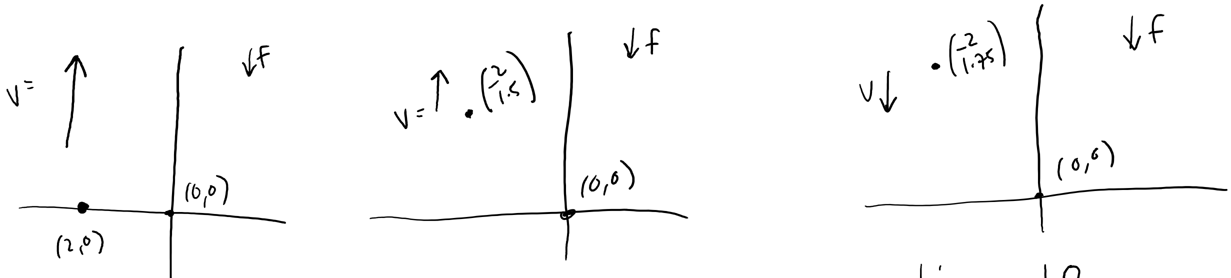
Suppose we have a particle w/ initial position $x = (-2, 0)$ & mass $= 1.0$ vel $= (0, 3)$. Suppose gravity is $(0, -5)$ and

What is the state $p = \begin{bmatrix} x \\ v \end{bmatrix}$ if $\Delta t = 0.5s$?

$$\text{time} = 0, \quad p = \begin{bmatrix} \begin{pmatrix} -2 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 3 \end{pmatrix} \end{bmatrix}$$

$$\text{time} = 0.5, \quad \dot{p} = \begin{bmatrix} \overbrace{\begin{pmatrix} 0 \\ 3 \end{pmatrix}}^{\text{old } v} \\ \underbrace{\begin{pmatrix} 0 \\ -5 \end{pmatrix}}_{f/m} \end{bmatrix} \Rightarrow p = p + \dot{p}\,\Delta t = \begin{bmatrix} \begin{pmatrix} -2 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 3 \end{pmatrix} \end{bmatrix} + \begin{bmatrix} \begin{pmatrix} 0 \\ 3 \end{pmatrix} \\ \begin{pmatrix} 0 \\ -5 \end{pmatrix} \end{bmatrix} 0.5 = \begin{bmatrix} \begin{pmatrix} -2 \\ 1.5 \end{pmatrix} \\ \begin{pmatrix} 0 \\ .5 \end{pmatrix} \end{bmatrix}$$

$$\text{time} = 1.0, \quad \dot{p} = \begin{bmatrix} \begin{pmatrix} 0 \\ .5 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 3 \end{pmatrix} \end{bmatrix}, \quad p = \begin{bmatrix} \begin{pmatrix} -2 \\ 1.5 \end{pmatrix} \\ \begin{pmatrix} 0 \\ .5 \end{pmatrix} \end{bmatrix} + (0.5) \begin{bmatrix} \begin{pmatrix} 0 \\ 0.5 \end{pmatrix} \\ \begin{pmatrix} 0 \\ -5 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} -2 \\ 1.75 \end{pmatrix} \\ \begin{pmatrix} 0 \\ -2 \end{pmatrix} \end{bmatrix}$$



time = 1.0

time = 0 time = 0.5 time

Spring forces (N-ary):

$$f_i = -k_s \left(\|\Delta x\| - r \right) \frac{\Delta x}{\|\Delta x\|}$$

$$f_j = -f_i$$

forces exerted between particles p_i & p_j

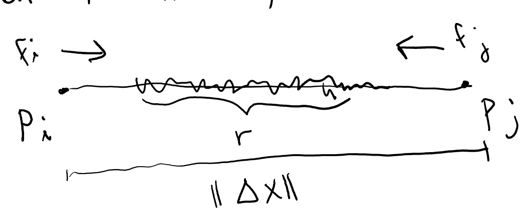
where k_s is a spring constant & controls how "stiff" the spring is.

r is a rest length

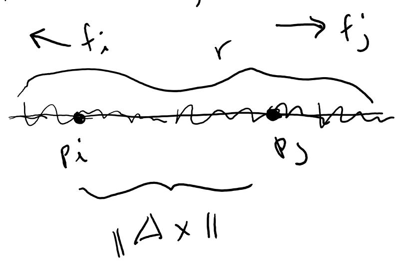
$\|\Delta x\|$ is the distance between p_i & p_j

$\|\Delta x\| = \|p_i - p_j\|$

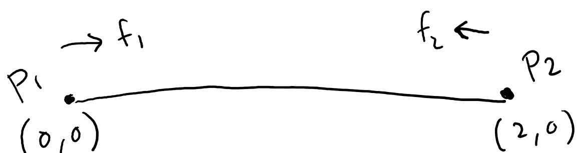
When $\|\Delta x\| > r$, we push inwards



When $\|\Delta x\| < r$, we push outwards



$\boxed{EX}$ Two particles connected by a spring



Let $r = 1$ and $k_s = 1$

Compute f_1 & f_2

where $f_i = -\left[k_s \left(\|\Delta x\| - r \right) \right] \frac{\Delta x}{\|\Delta x\|}$ where $\Delta x_i = p_i - p_j$

$\Delta x_1 = p_1 - p_2 = (-2, 0)^T$

$\|\Delta x_1\| = \sqrt{(-2)^2} = 2 \Rightarrow \frac{\Delta x_1}{\|\Delta x_1\|} = (-1, 0)^T$

$$\|\Delta x_1\| = \sqrt{(-2)^2} = 2 \quad \Rightarrow \quad \|\Delta x_1\|$$

$$f_1 = -\left[1 \cdot (2-1)\right]\begin{pmatrix}-1\\0\end{pmatrix} = (1,0)^T$$

$$f_2 = -f_1 = (-1,0)^T$$