Collision Avoidance
(this is actually an excuse to discuss Functional Animation)

Mike Bailey
mjb@cs.oregonstate.edu

Collision Avoidance Isn’t the Same as Collision Detection

Both recognize that objects cannot occupy the same space at the same time, but ...

Collision Detection lets the objects collide and bounce off each other

Collision Avoidance tries to get the objects to change their paths so that they don’t collide in the first place

How do Springs Work?

$x + F$

$k = \text{spring stiffness in Newtons/meter or pounds/inch}$

$x = \text{spring displacement}$

$x - D_0 = \text{unloaded spring length}$

$F = ma = -k(x - D_0)$

Or, as a differential equation,

$m\ddot{x} + k(x - D_0) = 0$

Functional Animation

In Functional Animation, we setup a fake force system (with fake springs and other mechanical components) to “convince” an object to go to a certain place without us having to actually animate it to go there.

If this was all we were going to do, then keyframe animation would get the object there just as well.

But, the big advantage of Functional Animation is that we can add other fake forces to make the objects behave in more complex ways, such as avoiding each other.

First Goal – Make the Free Body Move Towards its Final Position

$m\ddot{x} + c\dot{x} + kx = 0$

where:

$m$ is the mass

$c$ is the damping (≈ a shock absorber)

$k$ is the spring constant

Rearrange to get just the acceleration:

$\ddot{x} = \frac{-c\dot{x} - kx}{m}$

Second Goal – Make the Free Body Want to Move Away From All the Other Bodies

$m\ddot{x} = \sum F$

Rearrange to get just the acceleration:

$\ddot{x} = \frac{\sum F}{m}$
Repulsive Force

\[ F_{	ext{repulsive}} = \frac{k}{d^{n}} \]

Distance between the boundaries of the 2 bodies

Repulsion Coefficient

Repulsion Exponent

Total Goal – Make the Free Body Move Towards its Final Position While Being Repelled by the Other Bodies

\[ m\ddot{x} + c\dot{x} + kx = \sum F \]

\[ \dot{x} = \sum \frac{F - c\dot{x} - kx}{m} \]

Accelerating an Object Towards a Target, Under the Influence of Outside Forces

\[ m\ddot{x} + c\dot{x} + kx = \sum F \]

Fundamental equation of a second order system, if anchored at the origin

\[ m\ddot{x} + c\dot{x} + k(x - x_r) = \sum F \]

Fundamental equation of a second order system, if anchored at a target position. (We’re assuming that the target final velocity wants to be 0.)

\[ \ddot{x} + c\dot{x} + k(x - x_r) = \sum F \]

We’re not doing a real physics simulation, just going for an effect. Therefore, we can normalize the mass, and scale c, k, and the external forces appropriately.

\[ \ddot{x} = -c\ddot{x} - k(x - x_r) - \sum F \]

Solve for the acceleration that moving towards the target needs and the outside forces influence

Use that acceleration to compute the next position and velocity (1st order, 2nd order, 4th order, etc.)

An Increased k Gets the Object to the Target Faster (But Increases Target Overshoot)

An Increased c Reduces Target Overshoot (But the Object Moves More Sluggishly)

Collision Avoidance in Action – Time Animation
Varying the Parameters -- Start with This

Increasing Stiffness
- Stiffness = 3
- Stiffness = 6
- Stiffness = 9

Increasing Damping
- Damping = 7
- Damping = 10
- Damping = 13

Increasing the Repulsion Coefficient
- Repulse = 10
- Repulse = 30
- Repulse = 50
Increasing the Repulsion Coefficient

Repulse = 10, 30, 50

Increasing the Repulsion Exponent

Power = 2, 4, 6

Parameter Rules of Thumb

A larger $k$ value:
- Gets the object to its goal faster
- Possibly overshoots

A larger $c$ value:
- Gets the object to its goal slower
- Decreases spurious wiggles

A larger Repulsion Coefficient:
- Objects give each other a wider berth
- It’s can get unnecessarily wide

A larger Repulsion Exponent:
- Influence waits to start until the objects are closer
- Influence increases quickly as the objects get closer
- Sometimes the influence increases too quickly and the objects do less avoiding and more bouncing