Physical Simulation

CS 4730 – Computer Game Design

First thing to realize…

- Physics (like, all of physics and doing all of physics) is HARD
- Physics in-game is HARD
- You (as beginning game designers… or as later pro game designers) probably do NOT want to do your own physics in many circumstances
- … well, it depends.

Kinematics vs. Dynamics

- Kinematics is the study of motion (position, velocity, acceleration)
- Dynamics is the full interactivity of all the forces in the environment
- You can handle kinematics
- You probably don't want to handle dynamics
- So, we'll use enough kinematics to make people believe it and "fake" the rest!
- Yes. Fake it.

Immersion in the World

- When a ball hits a paddle or brick in Breakout, how does it behave?

- http://www.roundgames.com/game/Xenocrate+2

Immersion in the World

- How does the ball react?
- Why does it react the way it does?
- How does it react differently if it hits in the middle of the paddle vs. the edge of the paddle?

Immersion in the World

- Why does the ball behave like this?
- Is this an “accurate” portrayal of the world?
- What’s wrong with the physics here?

Physics in Games

- So the question is not always "how do we model precise physics"
- But more often it's "how do we model a physics that makes sense in this world"
- "The correct physics" is often essential to a positive play experience and also fosters new dynamics in a game
- Remember: physics is the mathematical study of the physical world

The Role of Physics

Detect Collisions

- Responsive forces

Accumulate Forces

- Equations of Motion

Numerical Integration

- Updated world at Δt in the future

Enforce Constraints

- Adjust world to maintain user constraints

Basic Overall Idea

- Assume you have a simple particle
- You know it's current start position
- You know it's current velocity and acceleration
- You know the current forces working on it
- You know (or can adjust) the time until the next frame to draw
- Where's the particle when the next frame is drawn?

How do we pull this off?

- Roll your own
  - Pull out your physics book (or don't as the game requires) and go to town
  - Perhaps crazy physics is a mechanic in your game
  - Consider the physics of Breakout
  - What do you have to calculate?
  - How do you change from frame to frame?

How do we pull this off?

- Pull in a package
  - Unity
  - Havoc
  - PhysX (hardware acceleration)
  - Chipmunk
  - Farseer (what you would probably use)

Particle Kinematics - Position

- What is a Particle?
  - A sphere of finite radius with a perfectly smooth, frictionless surface
  - Experiences no rotational motion
- Particle Kinematics
  - Defines the basic properties of particle motion
  - Position, Velocity, Acceleration

$$\mathbf{p} = \langle p_x, p_y, p_z \rangle$$

Newton's Laws

- First Law: An object in motion will remain in motion unless acted upon by external force.
- Second Law: Force = Mass * Acceleration
- Third Law: Every action has an equal and opposite reaction.

Variables

- t_x The final time for the period we are examining
- t_0 The initial time
- Δt The change in time ($t_x - t_0$) for the period we are examining
- $\Delta \mathbf{v}$ The change in velocity ($v_x - v_0$) for the period we are examining
- $\Delta \mathbf{a}$ The change in acceleration ($a_x - a_0$) for the period we are examining

Change in Velocity

- Assume the acceleration (and mass and thus force) is constant
- The velocity at a given time is the sum of:
  - the initial velocity
  - the change due to the acceleration

$$\mathbf{v}_t = \mathbf{v}_0 + \frac{\mathbf{F}}{m}\Delta t$$

Change in Position

- Assume the acceleration (and mass and thus force) is constant
- The position at a given time is the sum of:
  - the initial position
  - the position change due to the initial velocity
  - The position change due to the acceleration

$$\mathbf{p}_t = \mathbf{p}_0 + \mathbf{v}_0 \Delta t + \frac{\mathbf{F}}{2m} \Delta t^2$$

The Problem With Integration

- It's all an approximation
- If the Δt is too big, then you get really imprecise physics
- If the Δt is too small, your processor can't handle it all
- If it happens in a single frame, all variables are constant
- How do curved trajectories turn out?

It Gets Worse…

- Rigid body physics adds in volume and mass
- So now, we are focusing our equations on the center of mass of the body… but other things could also happen, such as torque

Looking at Physics Code

- Let's look at Farseer physics again…
- How about another game?

Different Physics

- Why are these built differently?
- Why does this matter?
  - Mechanics perspective
  - Dynamics perspective
  - Aesthetics perspective