Physics Engines

Jeff Wilson
jeff@bitc.gatech.edu

Maribeth Gandy
maribeth@cc.gatech.edu

Clint Doriot
clint.doriot@gtri.gatech.edu
What is a Physics Engine?

• Provides physics simulation in a virtual environment
• High Precision vs. Real Time
• Real Time requires a lot of approximations
• Can be used in creative ways

http://www.youtube.com/watch?v=2FMtQuFzjAc
Real Time Physic Engines

• Havok (commercial), Newton (closed source), Open Dynamics Engine (ODE) (open source), Aegia PhysX (accelerator board available)
• Unity
  o PhysX integration
    ▪ Rigidbodies
    ▪ Soft Bodies
    ▪ Joints
    ▪ Ragdoll Physics
    ▪ Cloth
    ▪ Cars

http://docs.unity3d.com/Documentation/Manual/Physics.html
Concepts

- Forces
- Rigid Bodies
  - Box, sphere, capsule, character mesh etc.
- Constraints
- Collision Detection
  - Can be used by itself with no dynamics
Bodies

- Rigid
- Movable (Dynamic)
  - Kinematic
  - Unity’s IsKinematic = false
    - You are not controlling the velocity & position, Unity is doing that
- Unmovable (Static)
  - Infinite mass
- Properties
  - Mass, dynamic/static friction, restitution (bounciness), softness
    - Anisotropic friction (skateboard)
  - Mesh shapes with per triangle materials (terrain)
Bodies

- Position
- Orientation
- Velocity
- Angular Velocity
- These values are a result of forces
Collisions

- Bounding boxes (collision hulls) reduce complexity of collision calculation
  - Made from the primitives mentioned before
    - Boxes, capsules etc.
- How your model is encapsulated determines accuracy, and computational requirements
- Collision groups
  - Tweak simulation, game play, path planning
Forces

• Force
  o constant
• Impulse
  o Instantaneous
• Vector and magnitude
• Acceleration and smooth options available
• Torque (spin)
Connectors

• Joints
  o Restrict motion between actors, rotation and translation
  o Constraints
  o Projection mode (amount of joint separation allowed)
  o Actors collidable or not (bendable)
  o Restitution
  o Revolute (hinge)
  o Spherical (three degrees of freedom)
  o Prismatic (shock absorber), cylindrical joints
  o Point on line (shower curtain)
  o Pulley
Connectors

• Spring
  o Joint with natural resting angle
  o Force
  o Damping
• Joint Motor (apply relative torque)
• Breakable joints (max force, max torque)
Constraints

- **Hard**
  - Never violated
  - In reality will be violated by errors in simulation
- **Soft**
  - Designed to be violated
- **Joint constraints**
  - Degrees of freedom, linear/angular amounts
- **Freeze flags (position and rotation)**
- **Linear and angular damping**
  - In absence of friction and collision (wind resistance)
Deactivation

- Limit actors awake in simulation
- Sleep linear and angular velocity
- Bounce threshold (stops vibration)
Ragdoll Physics

- Create human and other figures that move realistically
- Simplified skeleton
  - Collection of rigid bodies (bones)
  - Connected by hinges or springs (joints)
  - Joints have no stiffness
    - Hence “ragdoll”
  - Kinematic objects useful
- “Trespasser” first game to use
- Complex to combine this with animations
  - Blended ragdoll
- Other complex constructions made from components
  - Rope, grass, cloth, particles systems, vehicles

Putting it all in motion

- Set gravity vector
- Apply forces to bodies
- Adjust joint parameters as necessary
- Call collision detection
- Step the simulation based on time
  - Tradeoff of speed and accuracy
  - Substeps
    - Fixed update
- Keep the graphics object and physics object in synch
- Deactivate objects (manually or automatically)
Why does my simulation look wrong?

- Scale of your objects and world
  - May look wrong
  - May cause anomalies in simulation
- Disconnect between graphics and physics world
- Slow downs due to number of active objects
- Properties and their interactions also may result in strange results
  - Mass, Friction, magnitude of forces etc.
  - Velocities too fast for timestep
  - Whips
- Collision detection may break down
  - interpenetration
- Must move objects using the functionality of the physics engine. No “hand of god” behavior
Things you can do with a Physics Engine

- Detect collisions
- Simulate rigid bodies under the influence of forces
- Spring-mass systems
- Destructible buildings and structures
- Ray / Shape casting
- Trigger Volumes
- Complex machines (cranes, moving platforms, pulleys, etc)
- Traps (avalanche of boulders)
- Drivable vehicles
- Rag doll characters
- Powered rag dolls
- Dangling props
- Cloth
- Fluid Simulations
- Water surface simulations and buoyancy
- Audio propagation
Making it fun

• Factors
  o Quality of the simulation
  o Integration with other systems
  o Selection of physics driven gameplay elements

• Genres
  o Simulations
  o Physics Puzzle Game
  o Sandbox Games
  o Goal-based or Story-driven games?
    ▪ Trade off in control and realism

• Design Impacts
  o Predictability
  o Turning and control
  o Emergent behaviors
  o Non-static game environments