Rigid Body Simulation

CSE 78814
Motivation

- 500rings
- Bones
- New_bones
- Domino
- 300 cubes
- Handle
  - collision
  - contact
  - stacking
Background

- Simulation loop:
  - Update position and velocity.
  - Process collision.
  - Process contact.

- Problem: *e.g.* Block sliding down inclined plane.
  - Initially sliding down.
  - Update position and velocity $\Rightarrow$ interpenetrating plane.
  - Process collision $\Rightarrow$ velocity reflected.
  - No contact to process.
  - Next iteration $\Rightarrow$ object bounces.
Fixing Problem

- Velocity threshold
  - [Mirtich & Canny ’95 – 2 papers]

- Ordering of the simulation loop:
  - Process collision.
  - Update velocity.
  - Process contact.
  - Update position.

- Velocity update integrates forces and contact processing resolves forces.
Collision Processing: Algorithm Overview

- Compute candidate positions of bodies.
- For each intersecting pair of bodies:
  - Determine interpenetrating points.
  - Sort points by penetration depth (deepest first).
- For each point in order:
  - Apply frictional impulse (unless bodies receding).
Collision Processing: Algorithm Overview

- Repeat above a number of times.
  - Resolving one collision might create new ones.
- Applies series of impulses rather than simultaneously resolving all collisions.
Contact Processing

- Determine contacts and prevent penetration.
- Approaches:
  - Simultaneously solve for all contact forces. *e.g.* [Baraff ‘94]
  - Penalty method (repulsion forces). *e.g.* [Moore & Wilhelms ‘88]
  - Approximate continuous contact by a series of inelastic impulses. *e.g.* [Guendelman ‘2003]
- Gradually slow down bodies.
  - Transition impulses from $\varepsilon = -1$ to $\varepsilon = 0$.
  - Use more (smaller) impulses for contact.
Computing Stack Ordering

- Order bodies in stacks into increasing levels:
  - Compute directed graph of “resting on” relation.
  - Group together cycles (get a DAG).
  - Find levels consistent with DAG.
Implementation

- 1) Reset forces

- 2) Save the current linear velocity, angular velocity, orientation, position and the vertices

- 3) Apply the external forces e.g. Gravity

- 4) Create a predicted velocity (Angular & Linear) integrated from the forces

- 5) Predicted position (Centre of mass, Vertex positions & Orientation) is integrated from the predicted velocity
Check for collisions

6) Check for collisions and store contact data in a list
   - A pair is said to be non-separating if \( \text{dot}( V_{\text{rel}}, n) < 0 \), that is the bodies are moving towards each other at the collision point
   - 6-1) The list is sorted by penetration depth
   - 6-2) Using the PREDICTED positions and CURRENT velocity values calculate the elastic collision impulse and friction impulse
   - 6-3) Update the CURRENT velocities of the Rigid bodies referenced in the contact data
   - 6-4) Update the relative velocities in the contact data
   - 6-5) Remove contacts which now have a Separating velocity
   - 6-6) If the list is not empty go back to step 6-2).
7) Repeat 5) - 6) to improve the simulation but stop if there are no collisions.
   - Use the current velocity to integrate predicted positions.

8) Apply the external forces e.g. Gravity and Air Resistance

9) Integrate CURRENT velocities from the forces
Contact

- 10) Create a predicted position using the current velocity

- 11) Check for contacts and store the contacts in a Contact Graph ordered in the direction of gravity. All immovable objects are placed at the root node. Only store contacts if they are non-separating.
  - 11-1): Using the current positions, predicted contact point and current velocity values calculate the in-elastic (e=-1) collision impulse and friction impulse
  - 11-2): Update the current velocities of the rigid bodies referenced in the contact data
  - 11-3): Update the relative velocities in the contact data
  - 11-4): Remove contacts which now have a Separating velocity
  - 11-5): If the list is not empty go back to step 11-2).
Contact

12) Repeat stages 10) - 11) to get an improved visual result. Reduce \( e \) by 0.1 each time i.e. -0.9, -0.8 etc. Then perform a shock propagation phase, after each level on the contact graph is resolved temporarily set their status to immovable.

13) POSITION UPDATE

Update the current position using the current velocity
References

