CS550

# Reviewer: Evan Reviewee: Oscar

## Framework:

Code Review III

1. Is the framework up and running?
   * Yes
2. How well does the camera allow examination of the simulation?
   * Very well
3. What debugging tools are in place?
   * ImGui property editor, debug drawing, object selection, object editing, etc

## Collision detection:

1. Is collision detection functioning?
   * Yes
2. What algorithm is used for narrow phase collision?
   * GJK/EPA to find collisions and create contacts for the contact manifold
   * Manifold will keep up to four contact points to make it stable
   * 2D only needs like 2 points
3. What information is in the collision record?
   * Contact point on both colliders (world and local)
   * Normal, tangent, and bitangent of hyperplane
   * Penetration depth of the penetration
4. What, if any, broad phase collision detection is being used?
   * DynamicAABB Tree

## Collision resolution:

1. Is collision resolution functioning?
   * Yes
2. What algorithm is being used?
   * Sequential Impulses
3. How are stable situations modeled?
   * Small amount of interpenetration if you drag another object into other objects
   * Resting contact has a small amount of jittering but it’s not very noticeable. Could explode after a long time
4. How are multi-body situations modeled?
   * Resting contact can occur b/t multiple bodies
   * Friction is applied between multiple objects

## Additional Tasks:

1. Listen and understand current top problems.
   * Drawing isn’t working in release
   * Decrease time step for integrator to improve stability
   * Multiply previous (warm started) lambda with 0.5 to prevent explosions
   * Small jitter in resting contact
2. Examine code to look for solutions to current problems as well as efficiency, effectiveness, and good coding practices (may be continued after class).
   * Plug possible contacts into narrow phase to get resulting contact (which colliders are colliding – as id into hash map) -> aka which colliders are colliding
   * Loop through hash table to see if there is an entry. If no entry, init entry, else reduce invalid points. If an entry is no longer colliding, remove it.
   * Contact manifold keeps track of 4 contacts per pair
   * Go through contacts and apply impulses as needed.
   * In applying impulses, solves friction forces first then solves normal forces
   * Clamps friction b/t negative and positive max
   * Calculates velocity bias in initialization of constraint (combines baumgarte and restitution)
   * Calculates effective mass at initialization
   * New lambda is just -invEffectiveMass \* (deltaVinNormalDir + velocityBias)
   * Update impulse but clamp to prevent explosion
   * Apply impulses to current velocity
3. What did you learn from reviewing this code?
   * I didn’t know we should be keeping track of old contacts in the contact manifolds was important for sequential impulses

Reviewer signature:

Reviewee signature: