Physics
Physics Engines

1. Terrain Collisions
2. Broad phase
3. Narrow phase
4. GJK
Collision Detection

Terrain, Static Object Collision
Multiple Objects
Naive Collision Detection

- Test one object against another
  - Assuming N moving objects, N*(N-1)/2 tests. (Why?)
  - If M static objects, M*N+N*(N-1)/2 tests.

- Better CD strategies
  - Use bounding boxes to find potential collisions.
  - Use appropriate tree representation of objects (bounding volume hierarchy) to test potential collision for an actual collision.
Rays

- Imagine a car is driving on a road sloping upwards.
- Could test all triangles of all wheels against road geometry.
- For certain applications, we can approximate, and still get a good result.
CD with rays, cont’d

- Put a ray at each wheel.
- Compute the closest intersection distance, \( t \), between ray and road geometry.
- If \( t=0 \), then car is on the road.
- If \( t>0 \), then car is flying above road.
- If \( t<0 \), then car is ploughing deep in the road.
- Use values of \( t \) to compute a simple collision response.
Use spatial data structures for the road.
E.g. BVH (discussed latter) or BSP tree.
The distance along ray can be negative.
Therefore, either search ray in both positive and negative direction
void Actor::clampToTerrain()
{
    TerrainGroup* tg = TerrainEngine::singleton()->getTerrainGroup();
    baseNode->setPosition(baseNode->getPosition().x, tg->getHeightAtWorldPosition(baseNode->getPosition()) + 1, baseNode->getPosition().z);
}

- Ogre::TerrainGroup Class Reference
- Ogre::Ray Class Reference
Bounding Volumes

Ogre Scene Node Bounding Volume
Ogre Scene Query
Bounding Volume Heirarchy  

e.g. OCTrees

- Use a separate BVH for the two objects
- Test BVH against other BVH for overlap
- When triangles overlap, compute exact intersection, if needed
CD between many objects

- Why needed?
- Consider several hundreds of rocks tumbling down a slope…
- This system is often called ”First-Level CD or Broad Phase CD”
- We execute this system because we want to execute the 2\textsuperscript{nd} system less frequently
- Assume high frame-to-frame coherency
  - Means that object is close to where it was previous frame
  - Reasonable
Bounding volumes are sorted. Now sorting is expensive: $O(n \times \log n)$. But this need only apply to starting position.

Exploit frame-to-frame coherency in subsequent frames

The list is not expected to change much

Therefore, "resort" with bubble-sort, or insertion-sort Expected: $O(n)$, but there can be exception.

From [Moller and Haines 2002]
Sweep and Prune

- Assume objects may translate and rotate.
- Then we can find a minimal AABB, which is guaranteed to contain object for all rotations.
- Do collision overlap three times.
  - One for x, y, and z-axes
  - Arrange by bounding box intervals on each axis. Overlapping intervals means possible collision.
Sweep and Prune

- Assume objects may translate and rotate.
- Then we can find a minimal AABB, which is guaranteed to contain object for all rotations.
- Do collision overlap three times.
  - One for x, y, and z-axes
  - Arrange by bounding box intervals on each axis. Overlapping intervals means possible collision.
Sweep and Prune 2D
Sweep and Prune

- Keep a boolean for each pair of intervals
- Invert end points when sort order changes
- If all boolean for all three axes are true, \(\rightarrow\) overlap

From [Moller and Haines 2002]