CMSC425 Lecture Outline March 4th

References:
Mount, Lecture 7 (or 8): Geometric Programming: Sample Solutions
Mount, Lecture 9: Geometric Data Structures: Enclosures and Spatial Indices
See also previous midterm practice questions

Outline:
I. Administrivia
- A. Hw1 due. Questions?
- B. Final project proposal. Questions?
- C. Project 1b submission issues. Will be understanding
- D. Mini-lecture videos coming on some topics

II. Instant HW1 and "professional" solutions to a geometric/graphics problem
- A. Many solutions given for Instant Hw1 were close, not quite right
  - Could have found error through simple test (but only one student tested!)
  - Ok in context of the class- misleading material in lecture
  - Not ok in professional context
- B. For courses, typical homework solutions
  - Focus on getting the "answer"
  - Stop when the original question is answered
- C. Professional solutions
  - Focus on getting an answer suitable for professional implementation
  - "Prove" the answer through some argument/demonstration
  - Test the answer on enough cases to be sure it works
  - Show that it's the most efficient of reasonable solutions (efficient enough)
  - Stop when the team feels they have the best solution
- D. A difference: student plagiarism vs. professional copyright/patent violations
  - Students can steal stuff (fair use) but must not plagiarize
  - Professionals can plagiarize but can't steal stuff (violate copyright/patent)
- E. Objective of class: you can read and access literature on this material
  - http://www.realtimerendering.com/intersections.html
  - http://www.math.kit.edu/ianm2/lehre/am22016s/media/distance-harvard.pdf

II. Applications of geometric principles to game programming problems
- A. Sources of examples
  - i. Mount Lecture 7: Geometric Programming: Sample Solutions
  - ii. Mount Practice Problems for the First Midterm (spring 2018)
  - iii. Also see previous courses, in one larger Handout pdf
- B. Observation
  - The problem may look complex but have a relatively simple solution

B. Today

i. Shot gun simulator
ii. Projectile shooting
iii. Projectile direction (arrow)
iv. Evasive action

III. Geometric Data Structures: Colliders and collisions

Two problems:

Colliding two objects efficiently

Efficiently finding collisions in large group of objects (better than n^2)

A. Approximate complex shapes with approximating colliders

i. Want accurate and fast approximation

ii. If inaccurate

a. Ghost collisions

b. Bad physics

ii. If too accurate then slow

B. Bounding enclosures – single bodies

i. Axis-aligned bounding boxes (AABB) (also called rectilinear)

ii. General bounding boxes (rotated)

iii. Bounding spheres

iv. Bounding ellipsoids

v. Capsules

vi. k-DOPs (k-discrete oriented polytope)

vii. And – don't forget an object represented by a single point

C. Collisions

i. The more types of enclosures you support, the more type*type collisions you must support

ii. AABB-AABB

iii. General box-box

iv. Sphere-sphere

v. Capsule-Capsule

D. Compound objects and hierarchical representations

R-trees

E. Data structures for many objects

A. Grid

i. Store in array

ii. In hash map

iii Linear allocation

a. Morton order

b. Hibert order

B. Spatial trees

i. Quadtree (Octree

ii. K-d tree