

# Physics Notes #3: Collisions

CSCI 321

WWU

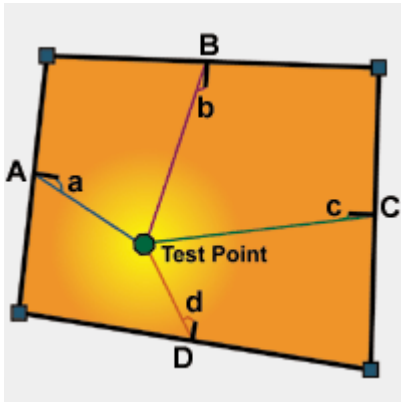
April 19, 2016

# Advanced collision techniques

## Reading:

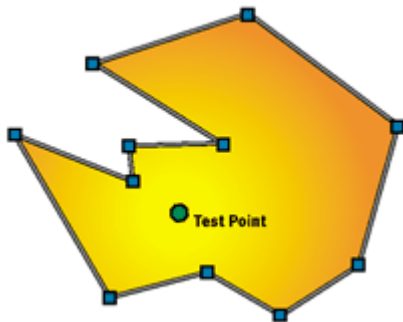
- [http://www.gamasutra.com/view/feature/3429/crashing\\_into\\_the\\_new\\_year\\_.php](http://www.gamasutra.com/view/feature/3429/crashing_into_the_new_year_.php)
- [http://www.gamasutra.com/view/feature/3426/when\\_two\\_hearts\\_collide\\_.php](http://www.gamasutra.com/view/feature/3426/when_two_hearts_collide_.php)
- [http://www.gamasutra.com/view/feature/3427/collision\\_response\\_bouncy\\_.php](http://www.gamasutra.com/view/feature/3427/collision_response_bouncy_.php)
- [http://www.gamasutra.com/view/feature/3190/advanced\\_collision\\_detection\\_.php](http://www.gamasutra.com/view/feature/3190/advanced_collision_detection_.php)

## Is a point inside a polygon?

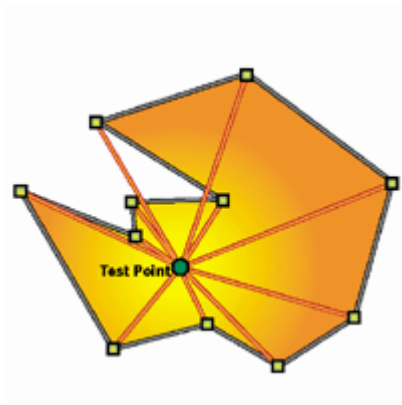


- Dot vector to point with inward pointing normal.
- Polygons usually have a consistent winding direction.
- How can you quickly find the inward pointing normal?

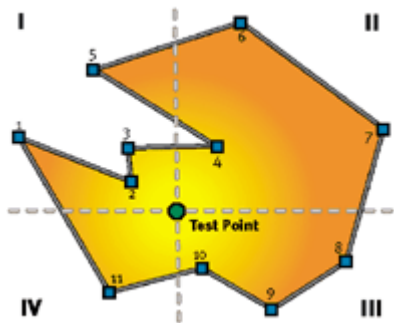
Does not work with concave polygons



Sum of all the angles = 360?

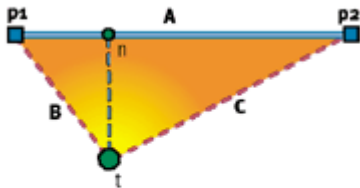


Quadrant crossing = 4?



Or check even/odd intercepts.

## Keeping at arm's length



$$A = p_2 - p_1$$

$$B = t - p_1$$

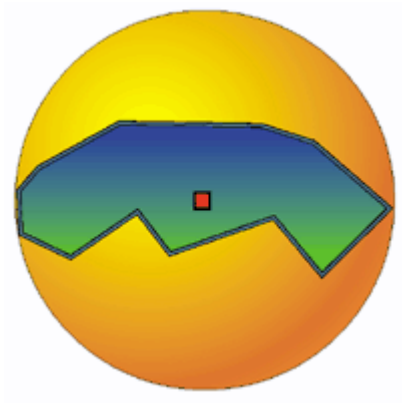
$$C = t - p_2$$

$$n = p_1 + A \frac{B \cdot A}{(B \cdot A) + (C \cdot A)}$$

$$A_{norm} = \frac{A}{|A|}$$

$$n = p_1 + A_{norm}(B \cdot A_{norm})$$

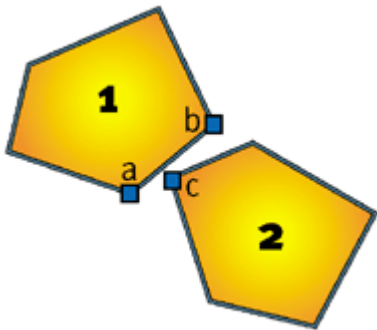
## Bounding spheres



- Collision between spheres?
- Collision sphere with polygon?



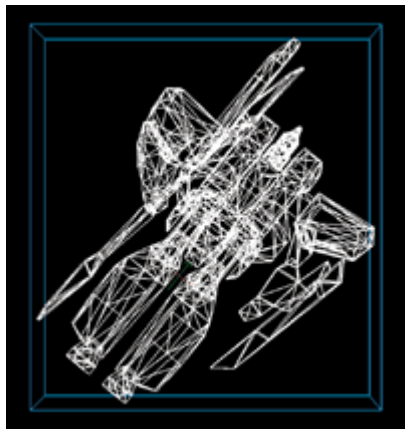
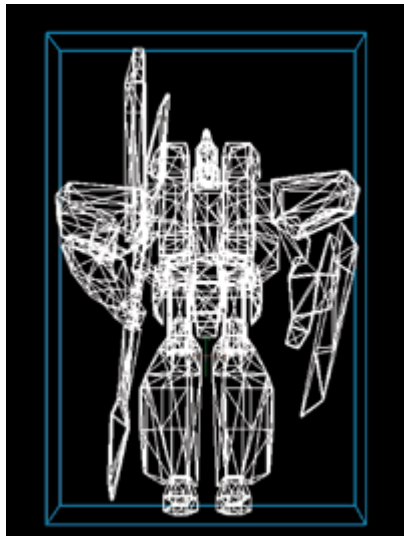
## Find a separating plane



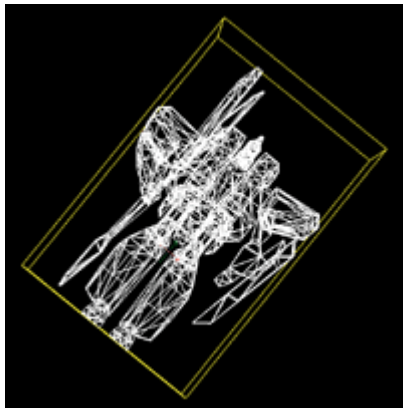
- See notes on Verlet integration:

[http://www.gamedev.net/page/resources/\\_/technical/math-and-physics/a-verlet-based-approach-for-2d-game-physics-r2714](http://www.gamedev.net/page/resources/_/technical/math-and-physics/a-verlet-based-approach-for-2d-game-physics-r2714)

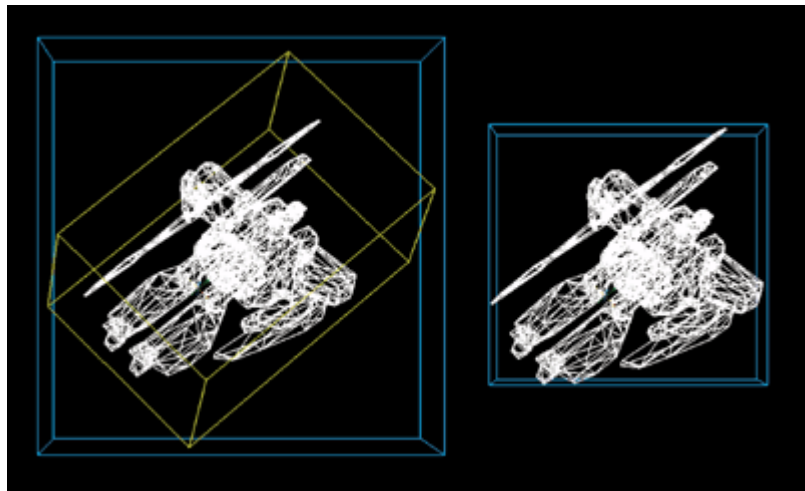
## Axis-aligned bounding box (AABB)



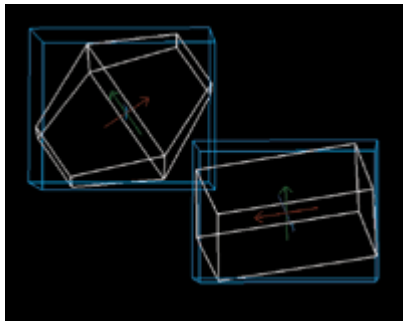
## Oriented bounding box (OBB)



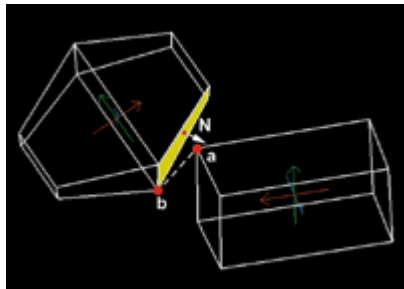
## Fast and slow AABB calculation



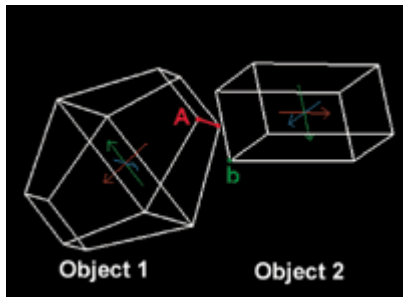
Two objects that might be colliding



Find a separating plane: first try faces



In 3D may not be a separating face



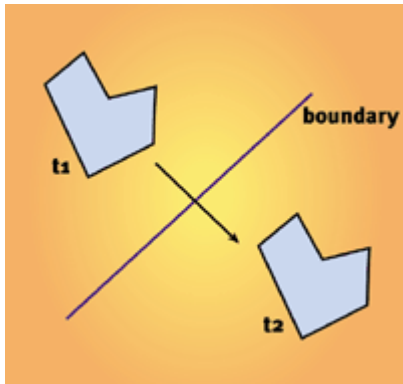
- Need to check point-edge combinations as well.

## Other points

- Cache separating planes.
- Separating plane only works for convex objects.

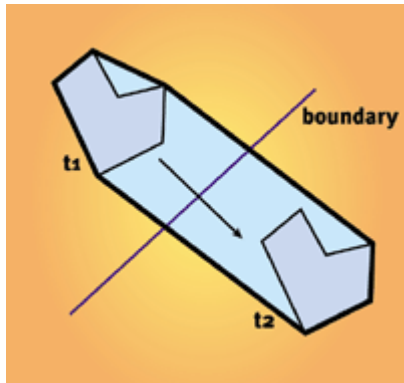


## Collision may happen between frames

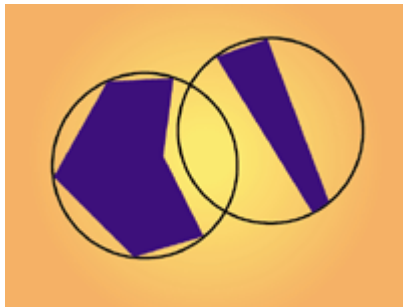


- Smaller  $\Delta t$  in the physics.
- Don't make really thin walls.

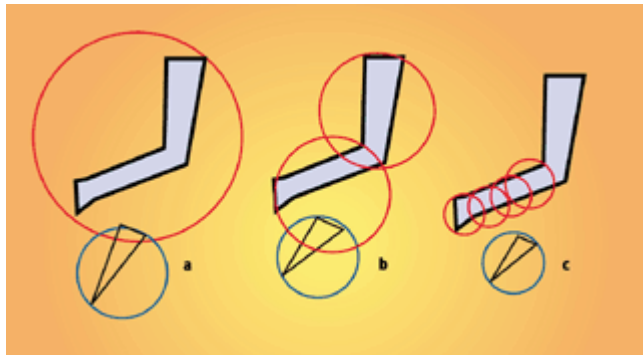
Create convex hull from object in two different frames



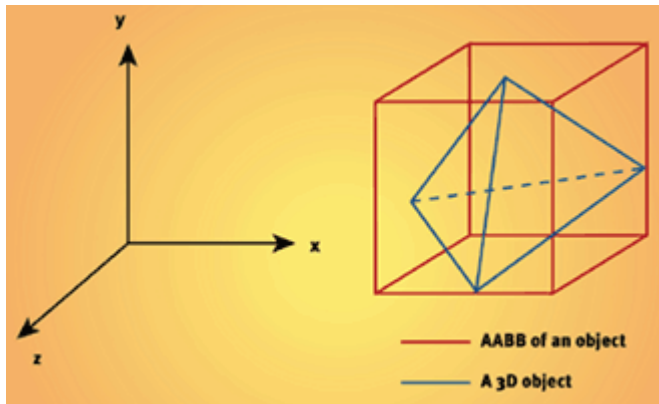
## Bounding spheres



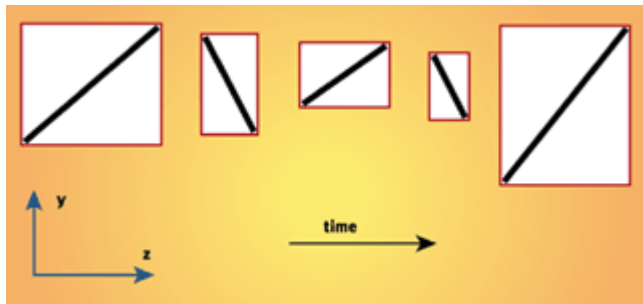
## Create tree of bounding spheres



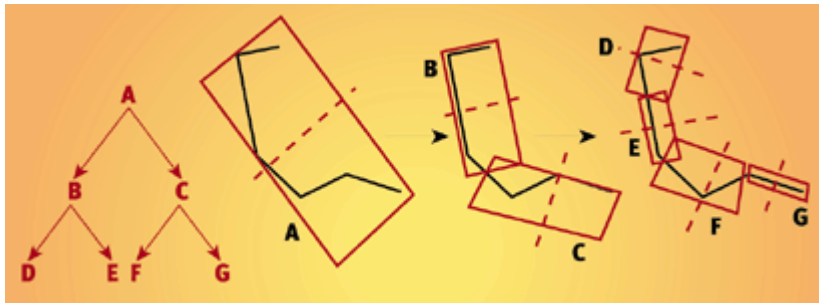
# AABB



## AABBs for rotating objects

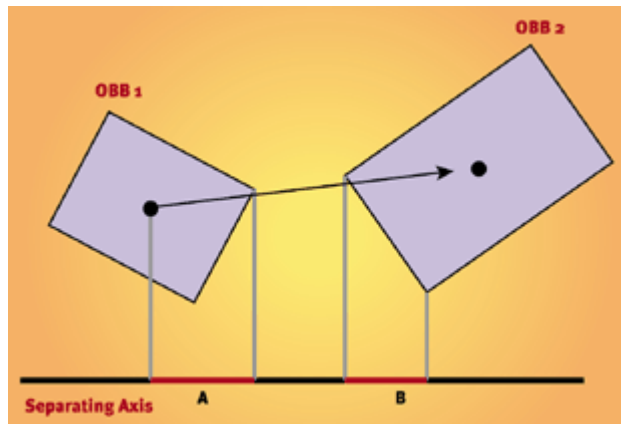


## Recursive OBB



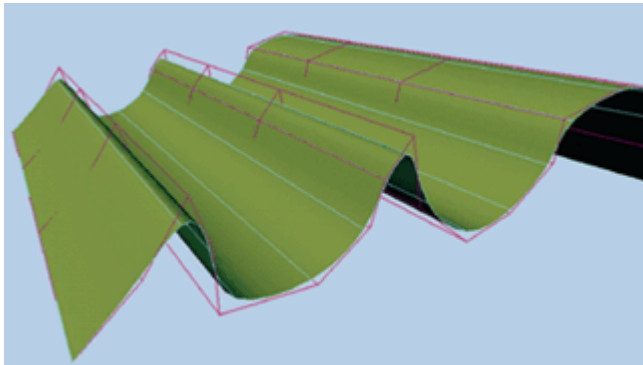
- Check for collision at top level, if exists, recurse on both.
- Note: Require artists to specify OBBs, convex hulls, etc. in advance.

Check boxes for separating planes



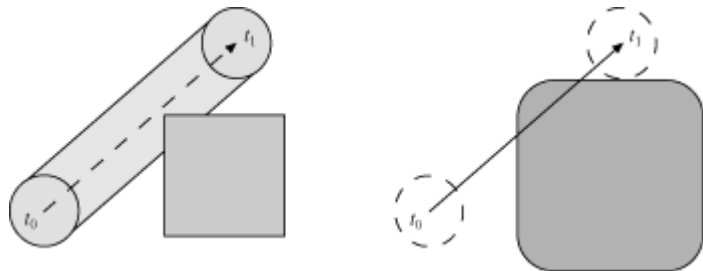
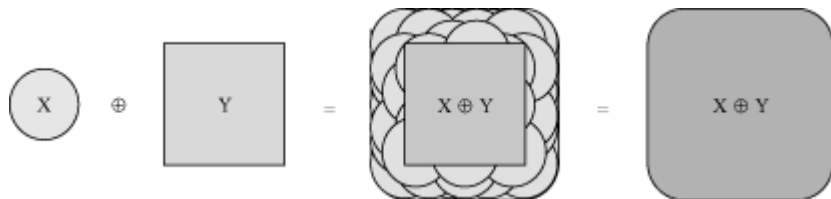


## Curved objects



- Approximate with linear objects

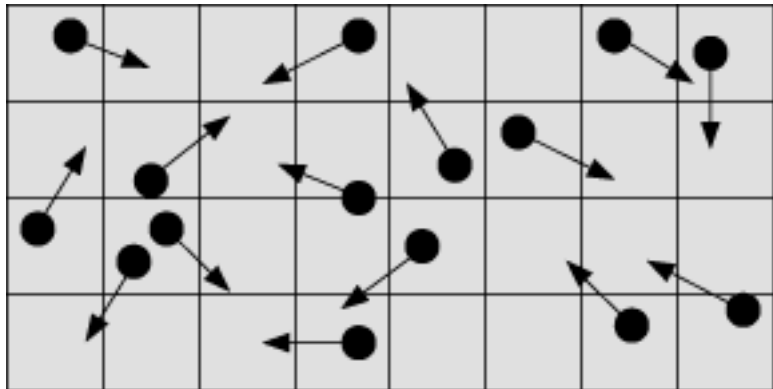
## Minkowski sums



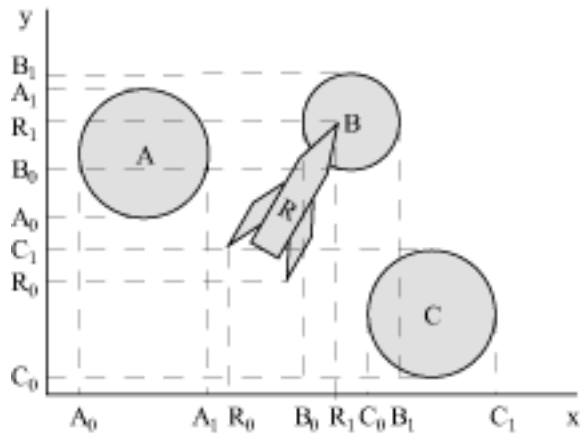
## Complexity with many objects

$$\binom{N}{2} = O(N^2)$$

Partitioning:  $O(N)$



Sweep and prune:  $O(N \log N)$



## Use a Library

- <http://www.box2d.org/>
- <http://www.pymunk.org/>
- <http://bulletphysics.org/>