A Star Wars-themed sunset scene. In the foreground, a Stormtrooper stands on a beach with palm trees. In the middle ground, a speeder bike drives along the water's edge. A TIE fighter flies in the sky above the horizon. The sky is a gradient from blue to orange.

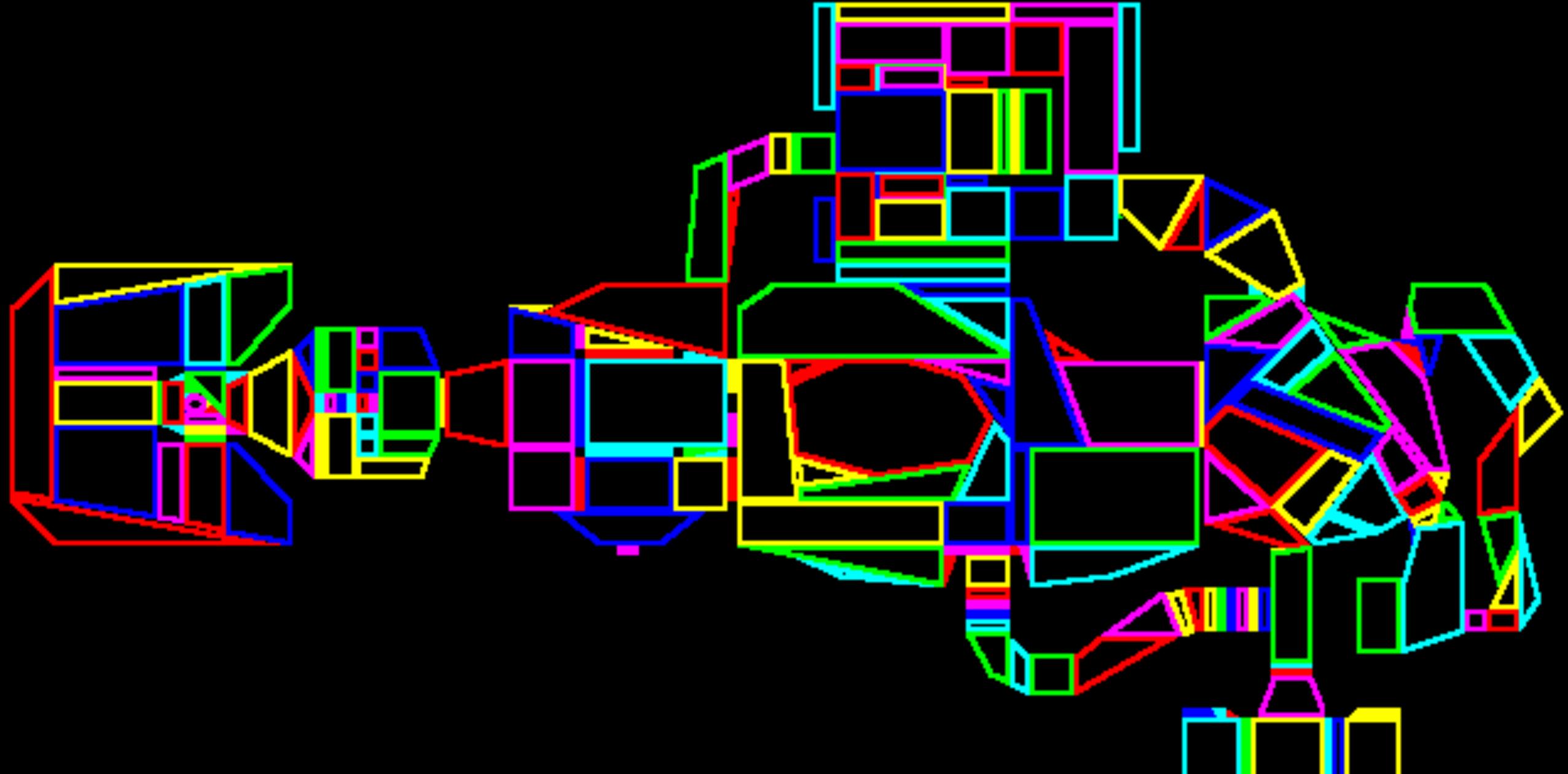
CSC317 Computer Graphics

... starts at 11:10am

Rob Katz

Some Slides/Images adapted from Marschner and Shirley

Today: Bounding Volume Hierarchies



Announcements

Assignment 1 grades out (please request remark in 1 week)

Assignment 4 is out today (due October 13th)

A4 is notoriously hard START EARLY

Any Questions ?

Today: Bounding Volume Hierarchy

Common Geometric Queries on Graphics

Bounding Volumes

Spheres

Boxes

Object-Partitioning Hierarchies

Sphere Trees

AABB Trees

Space-Partitioning Hierarchies

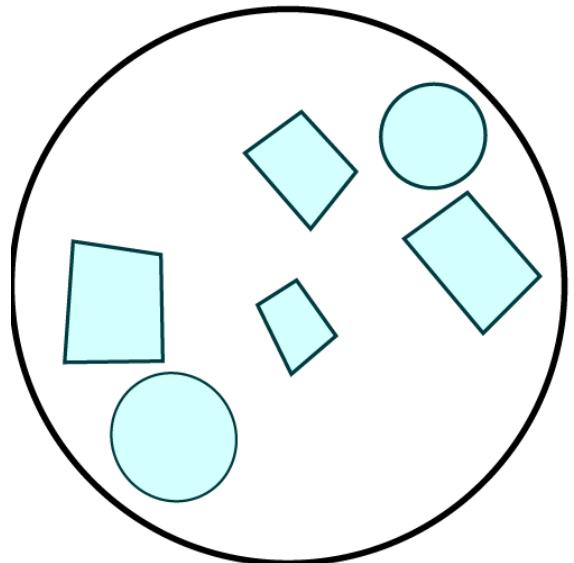
Uniform Spatial Subdivision

Axis-Aligned Spatial Subdivision

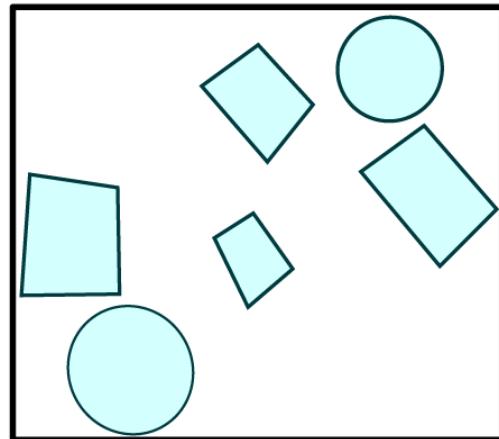
Bounding Volumes (BVs)

“Simple” geometry that fully encloses a **collection** of other geometry

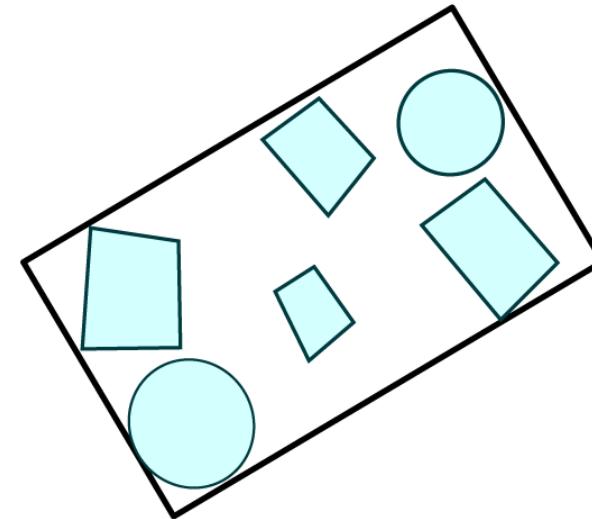
Sphere



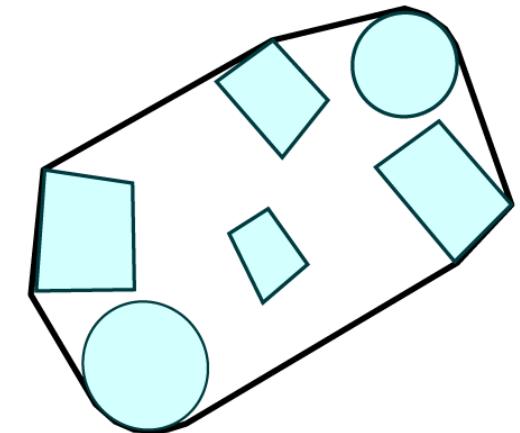
Object Oriented Bounding Box
(OOBB)



Axis-Aligned Bounding Box
(AABB)



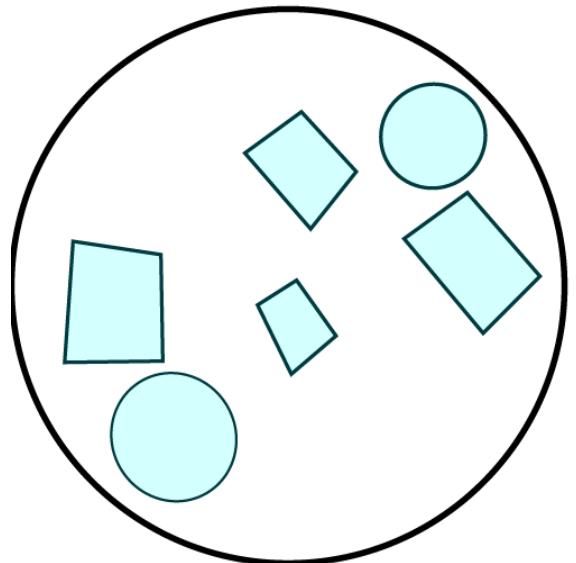
Convex Hull



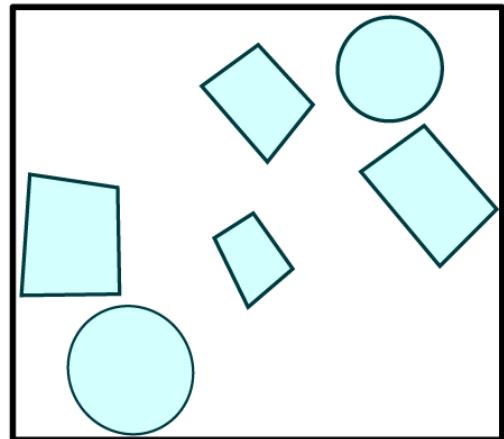
Bounding Volumes (BVs)

“Simple” geometry that fully encloses a **collection** of other geometry

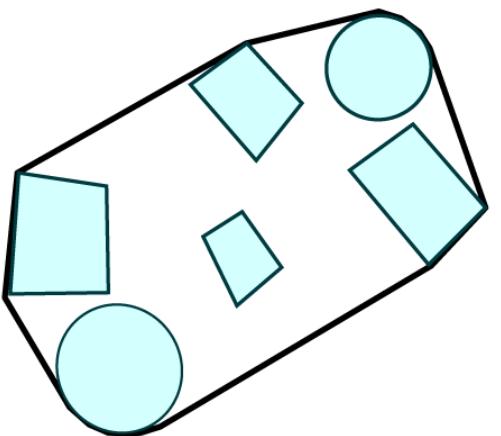
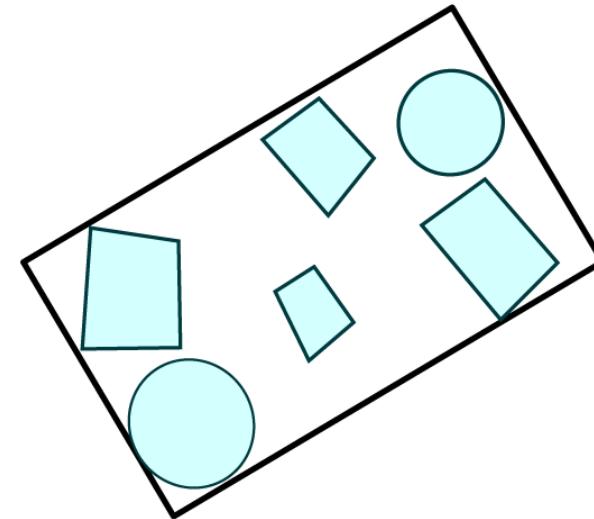
Sphere (A lot)



Object Oriented Bounding Box
(OOBB) (A little)

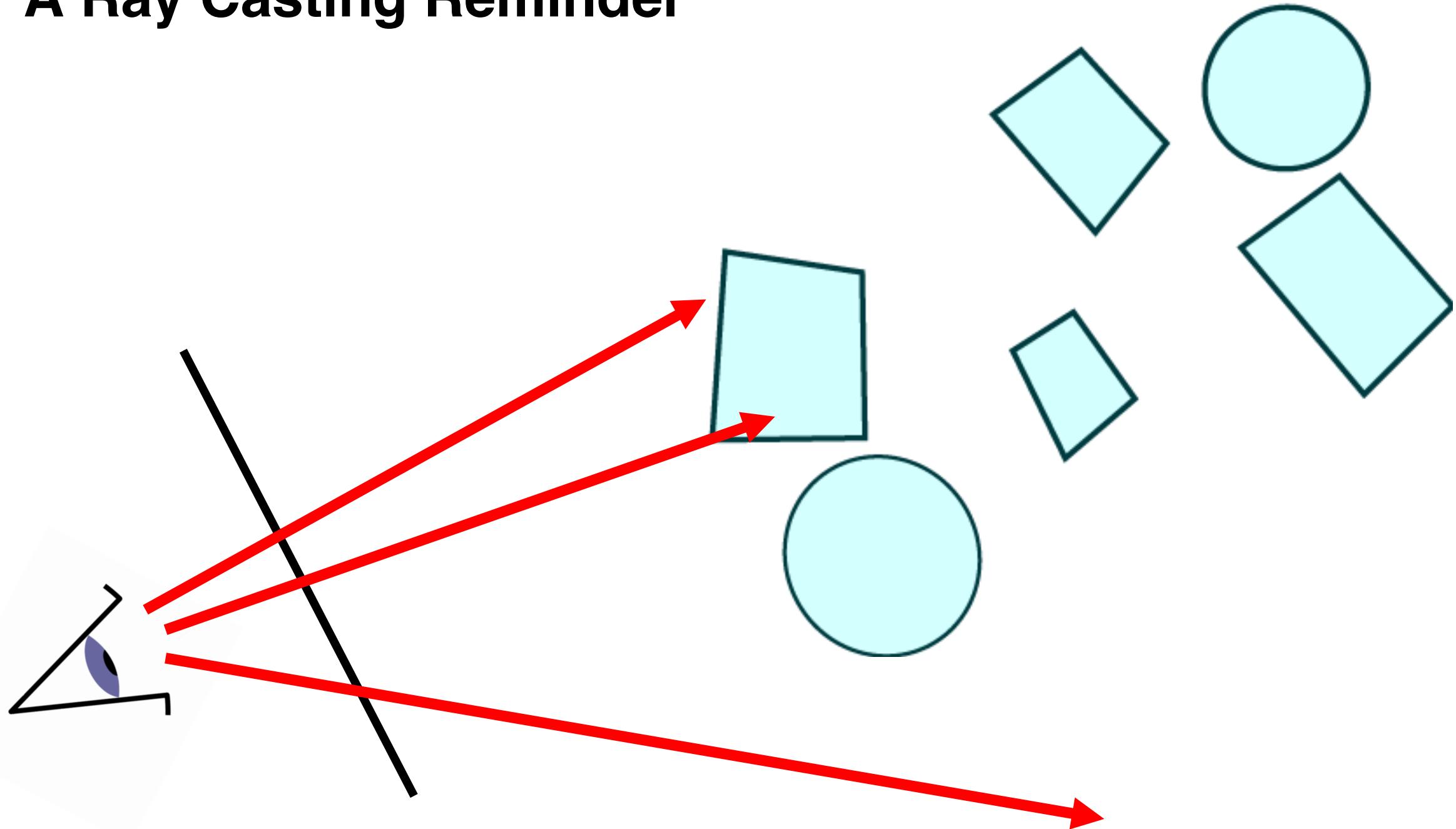


Axis-Aligned Bounding Box
(A lot)

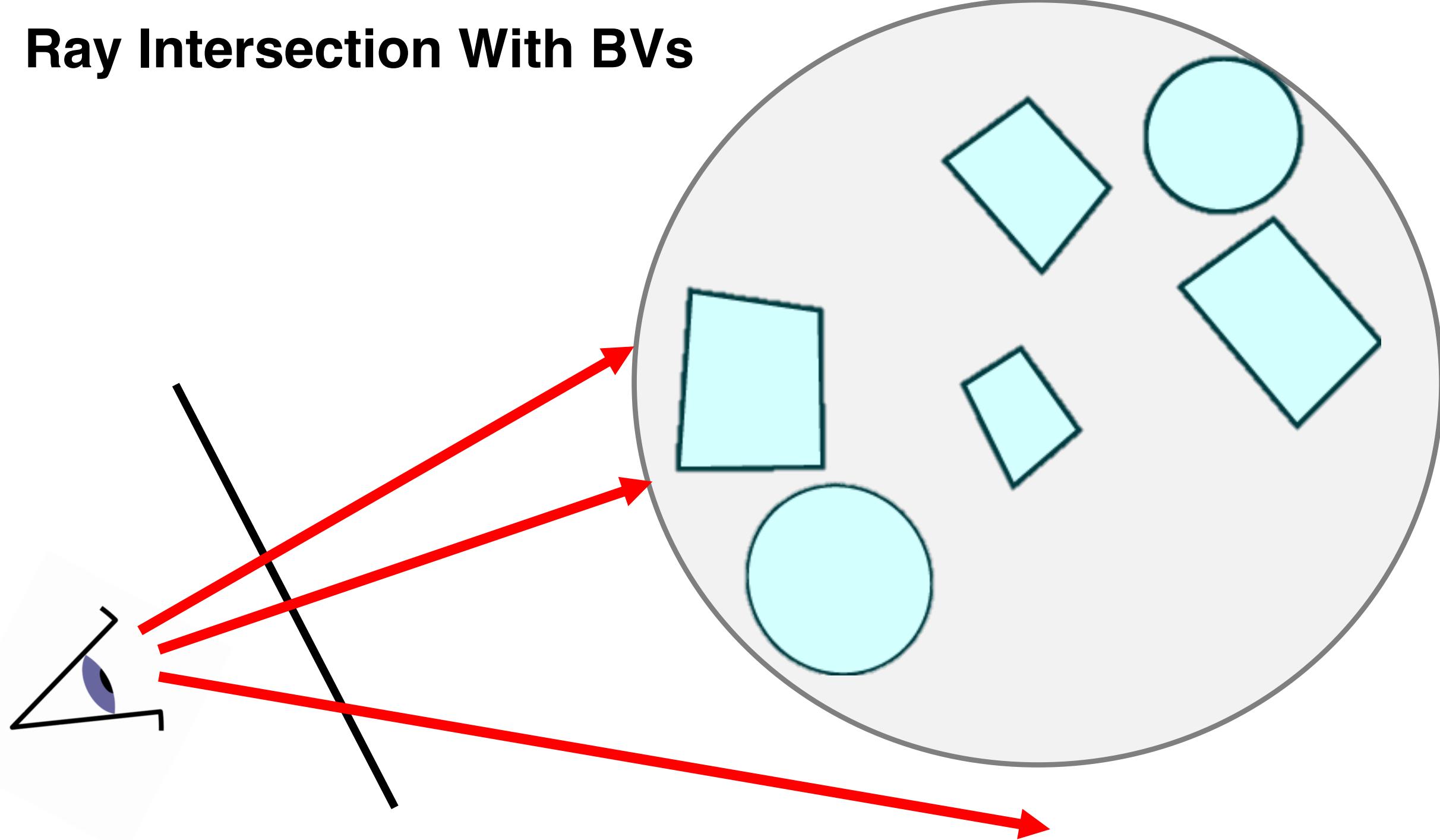


Convex Hull
(Nope)

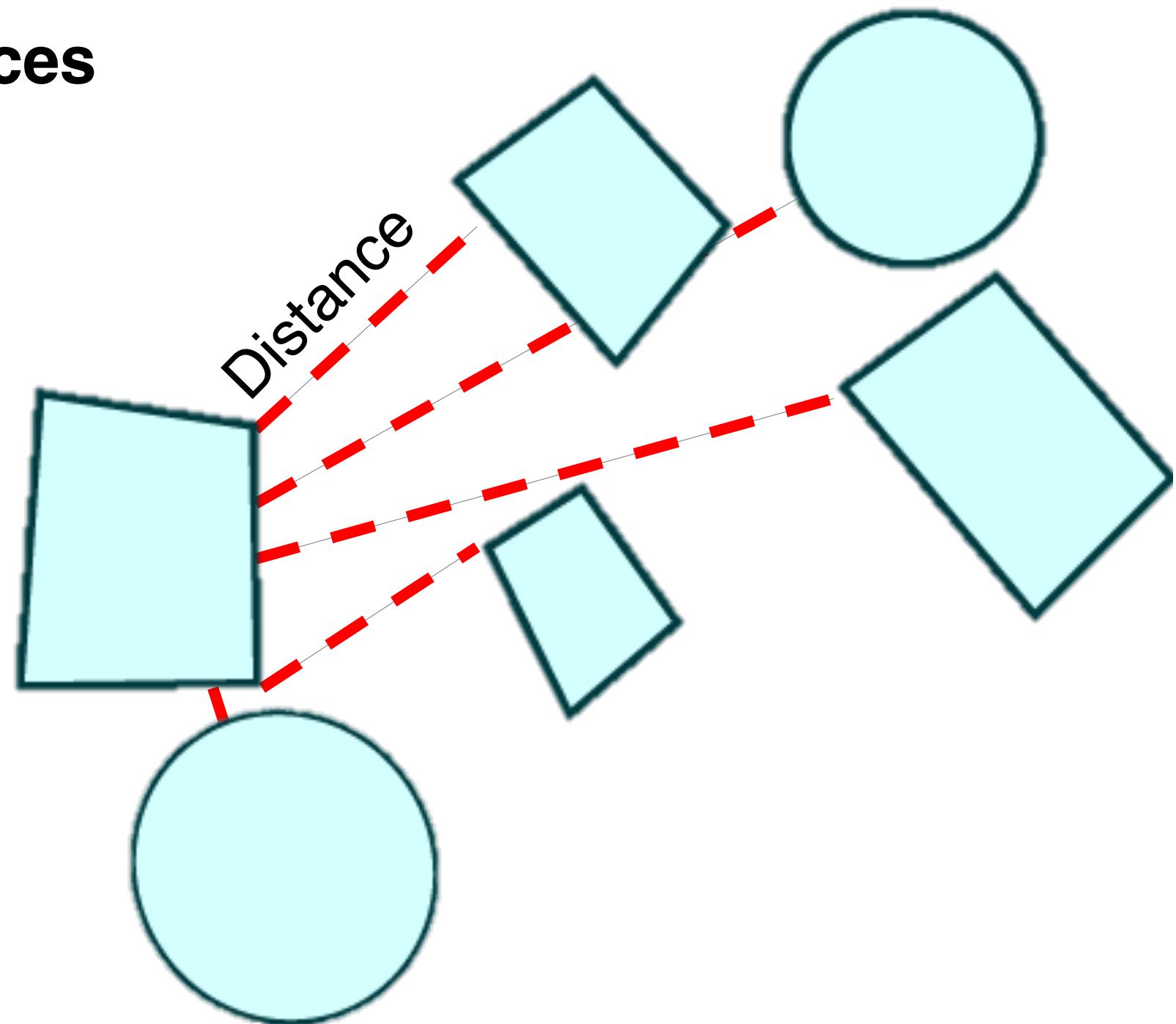
A Ray Casting Reminder



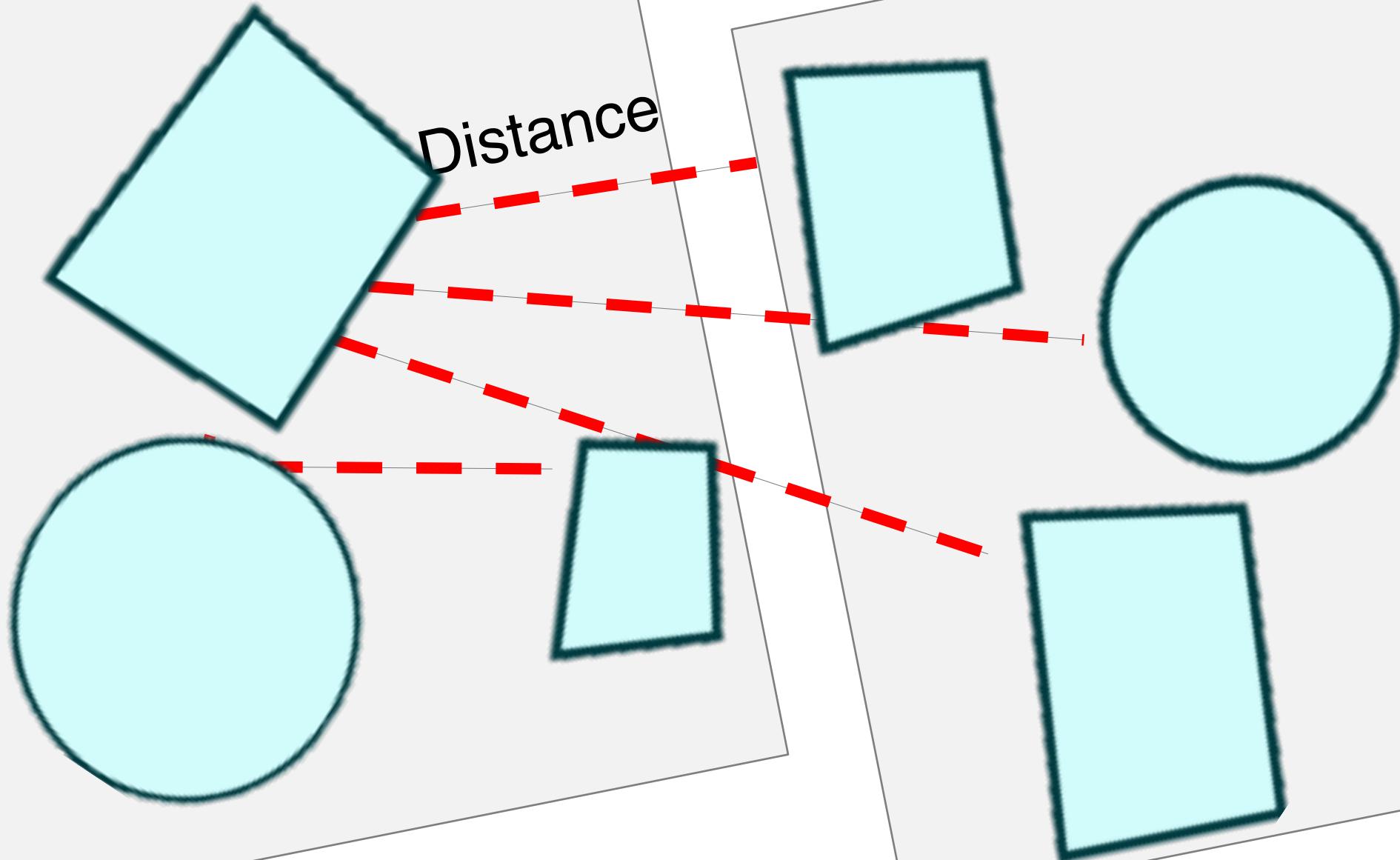
Ray Intersection With BVs



Closest Distances



Closest Distances with BVs

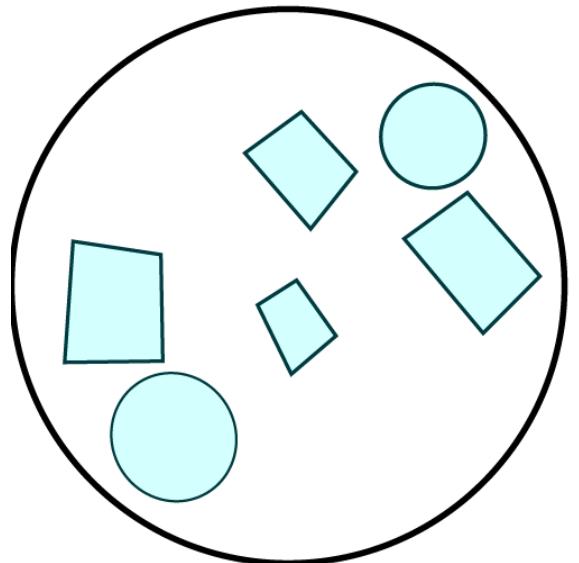


Bounding Volumes (BVs)

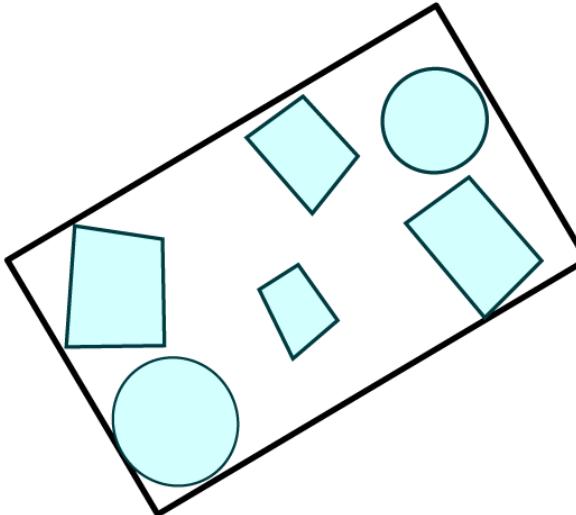
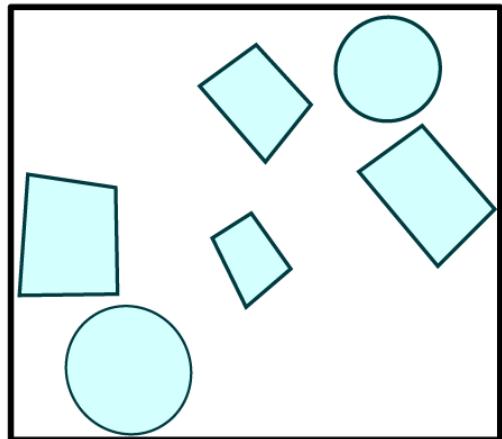
“Simple” geometry that fully encloses a **collection** of other geometry

Should fit geometry tightly

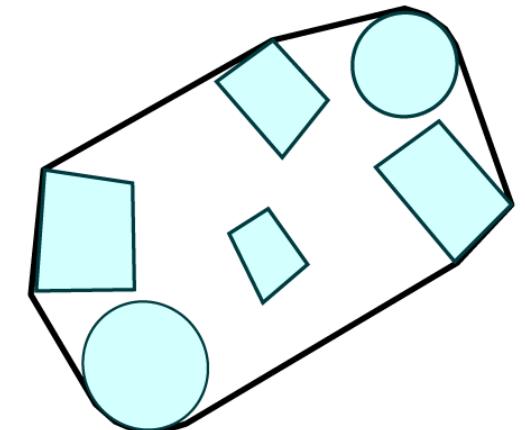
Sphere



Object Oriented Bounding Box
(OOBB)



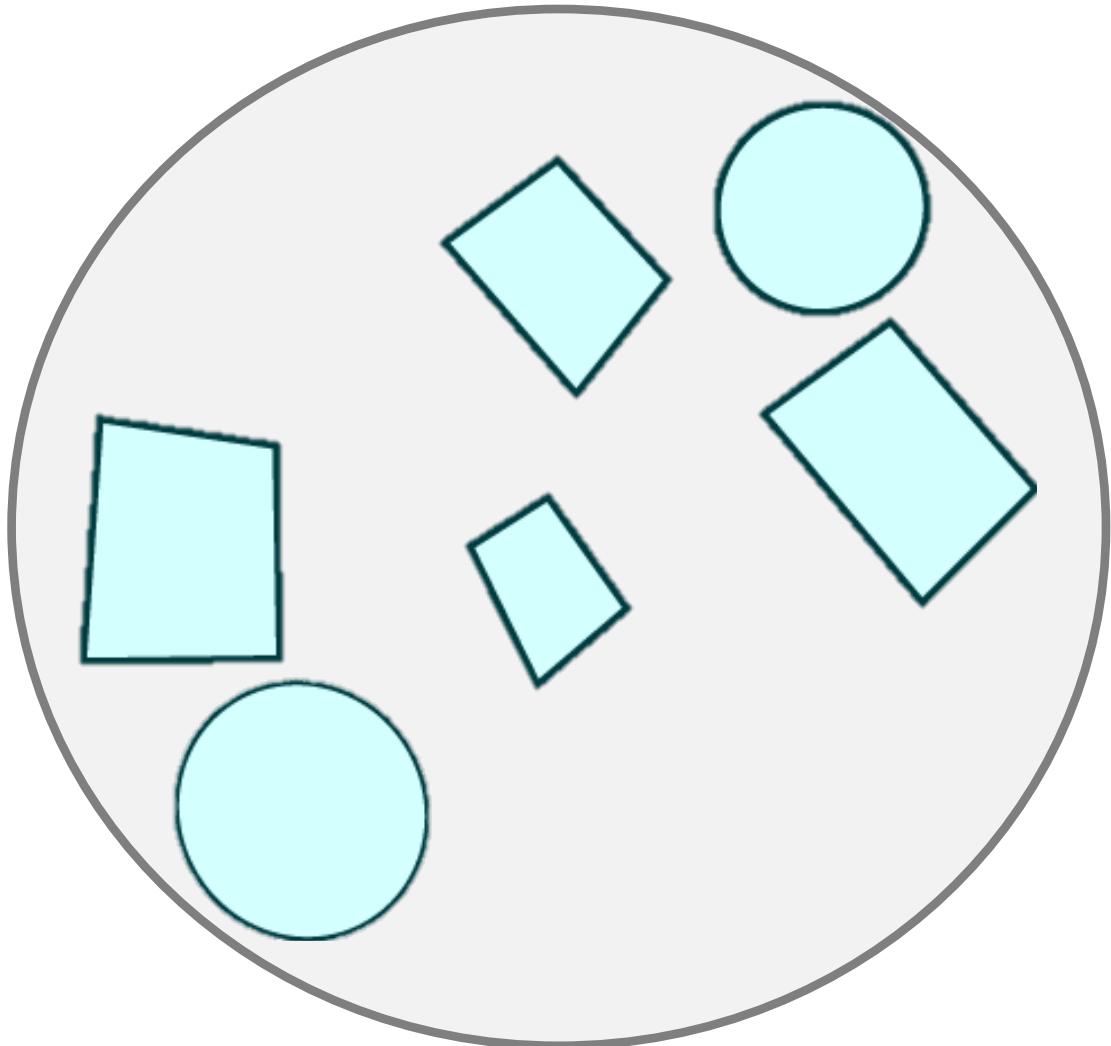
Axis-Aligned Bounding Box
(AABB)



Convex Hull

Building a Bounding Sphere

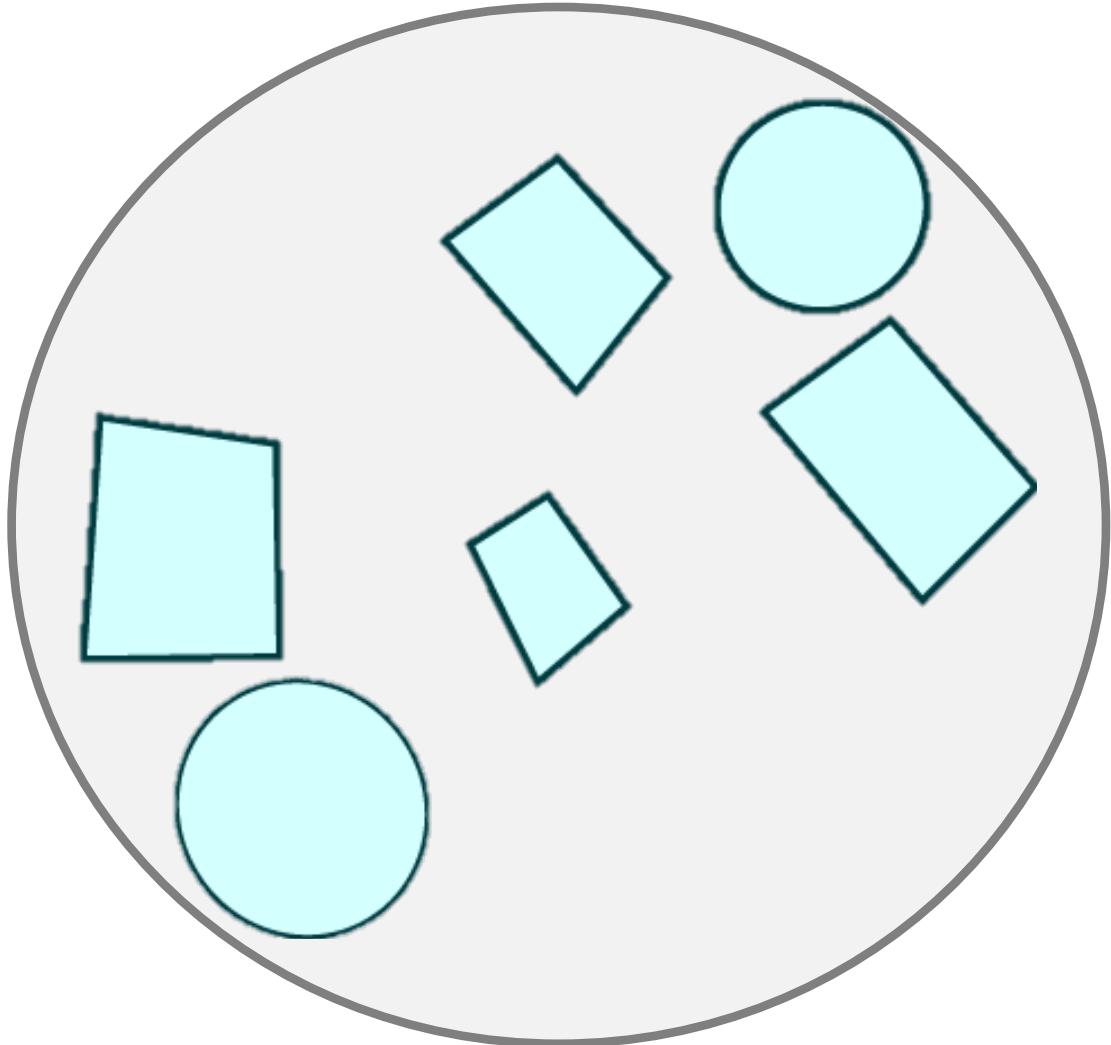
Parameters of a Sphere ?



Building a Bounding Sphere

Parameters of a Sphere:

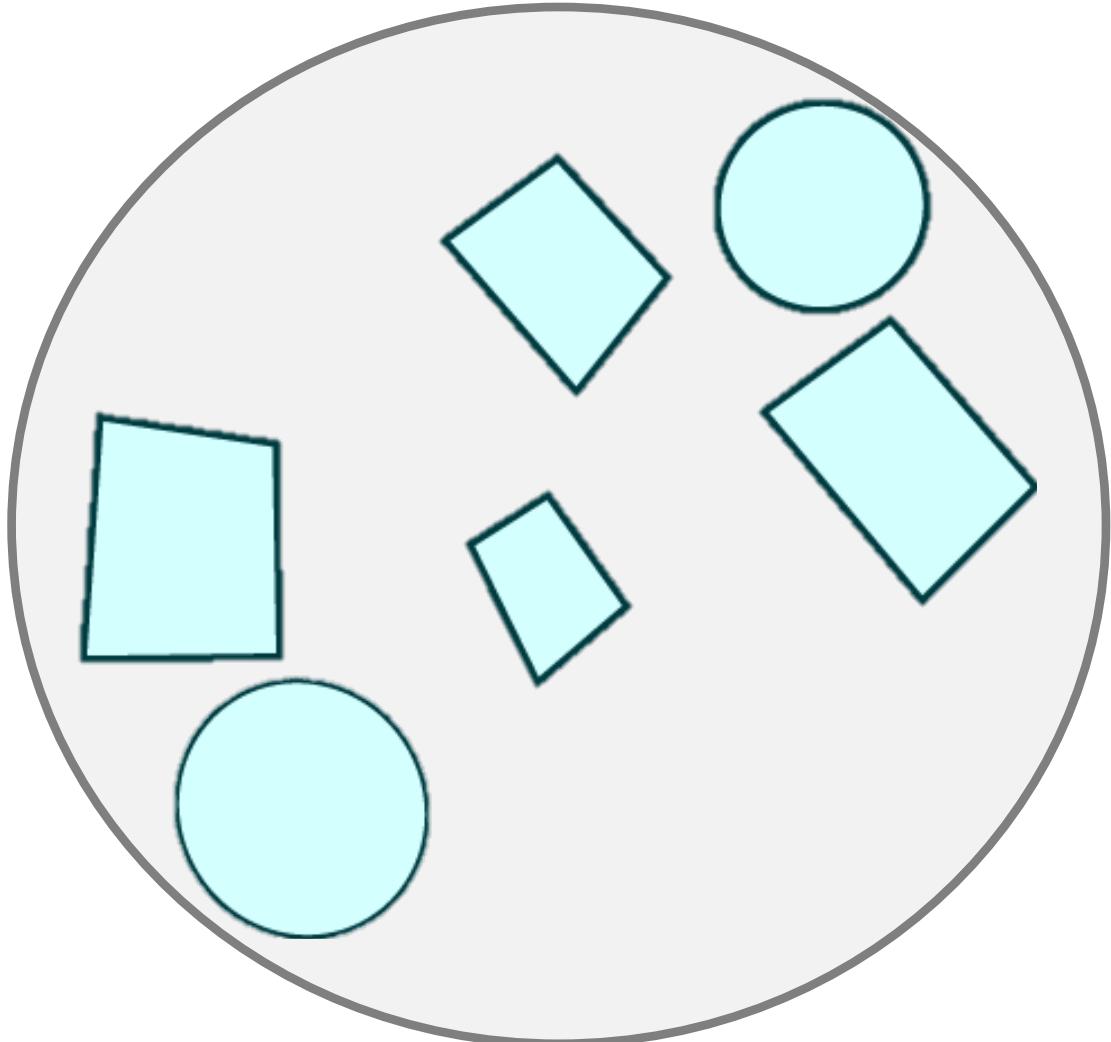
1. Center = ?
2. Radius = ?



Building a Bounding Sphere

Parameters of a Sphere:

1. Center = $\mathbf{c} = \frac{1}{n} \sum_{i=1}^n \mathbf{v}^i$
2. Radius = $r = \max (\mathbf{v}^i - \mathbf{c})$
 $\mathbf{v}^i \in \text{Vertices}$



Ray-Sphere Intersection



Ray-Sphere Intersection

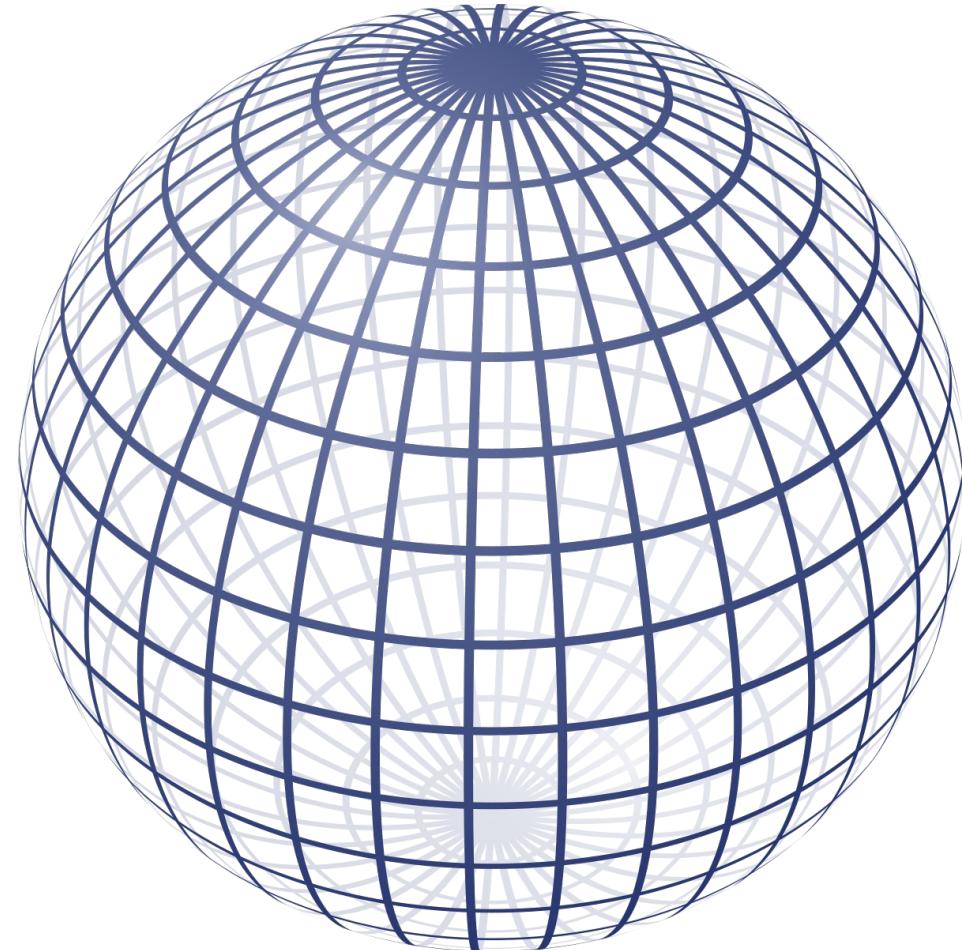
$$\mathbf{p}(t)^T \mathbf{p}(t) - r^2 = 0$$

$$a \cdot t^2 + b \cdot t + c = 0$$

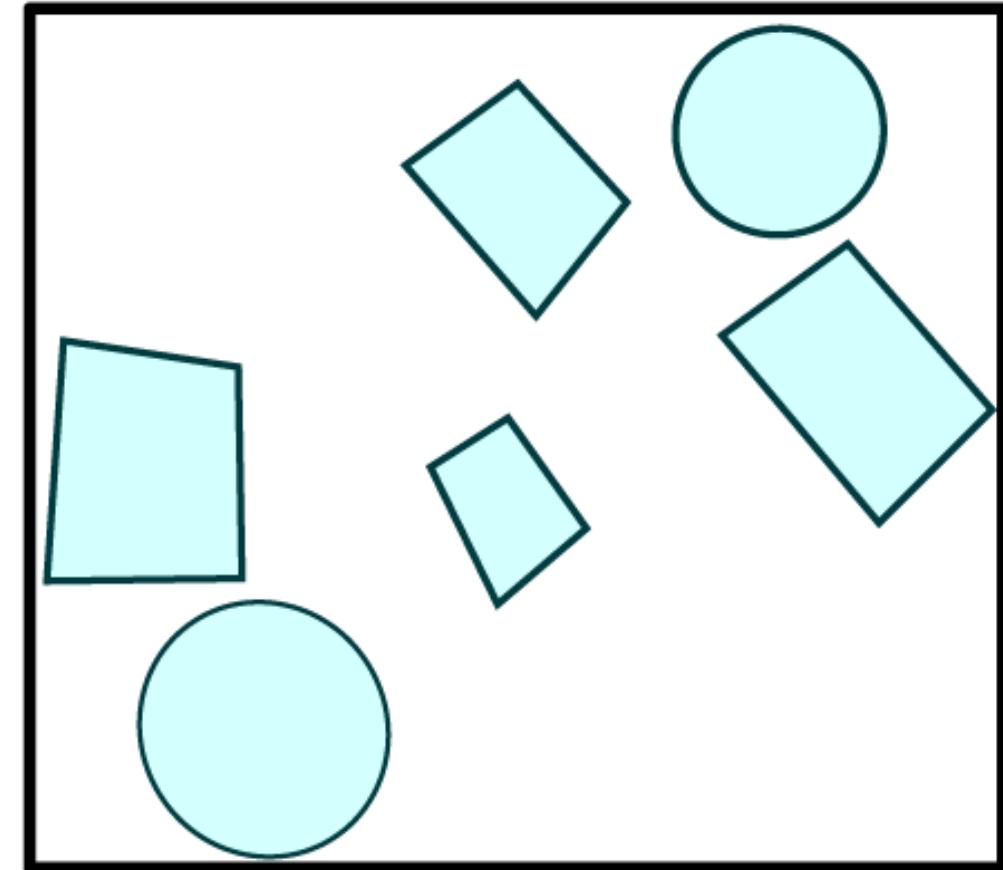
$$a = (\mathbf{s} - \mathbf{e})^T (\mathbf{s} - \mathbf{e})$$

$$b = 2\mathbf{e}^T (\mathbf{s} - \mathbf{e})$$

$$c = \mathbf{e}^T \mathbf{e} - r^2$$



Building and Axis-Aligned Bounding Box (AABB)



Building and Axis-Aligned Bounding Box (AABB)

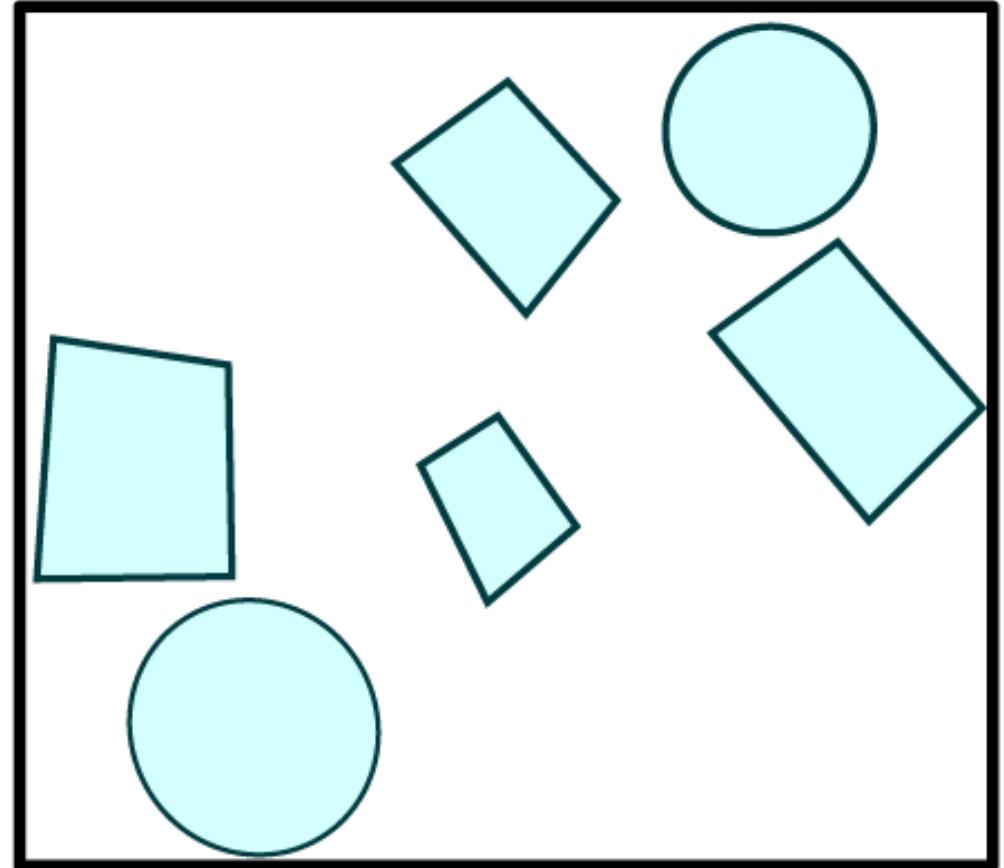
$$x_{min} = \min(v_x^i)$$

$$x_{max} = \max(v_x^i)$$

$$y_{min} = \min(v_y^i)$$

$$y_{max} = \max(v_y^i)$$

$\mathbf{v}^i \in \text{Vertices}$



Ray-AABB Intersection

$$t_{\text{xmin}} = (x_{\text{min}} - x_e) / x_d$$

$$t_{\text{xmax}} = (x_{\text{max}} - x_e) / x_d$$

$$t_{\text{ymin}} = (y_{\text{min}} - y_e) / y_d$$

$$t_{\text{ymax}} = (y_{\text{max}} - y_e) / y_d$$

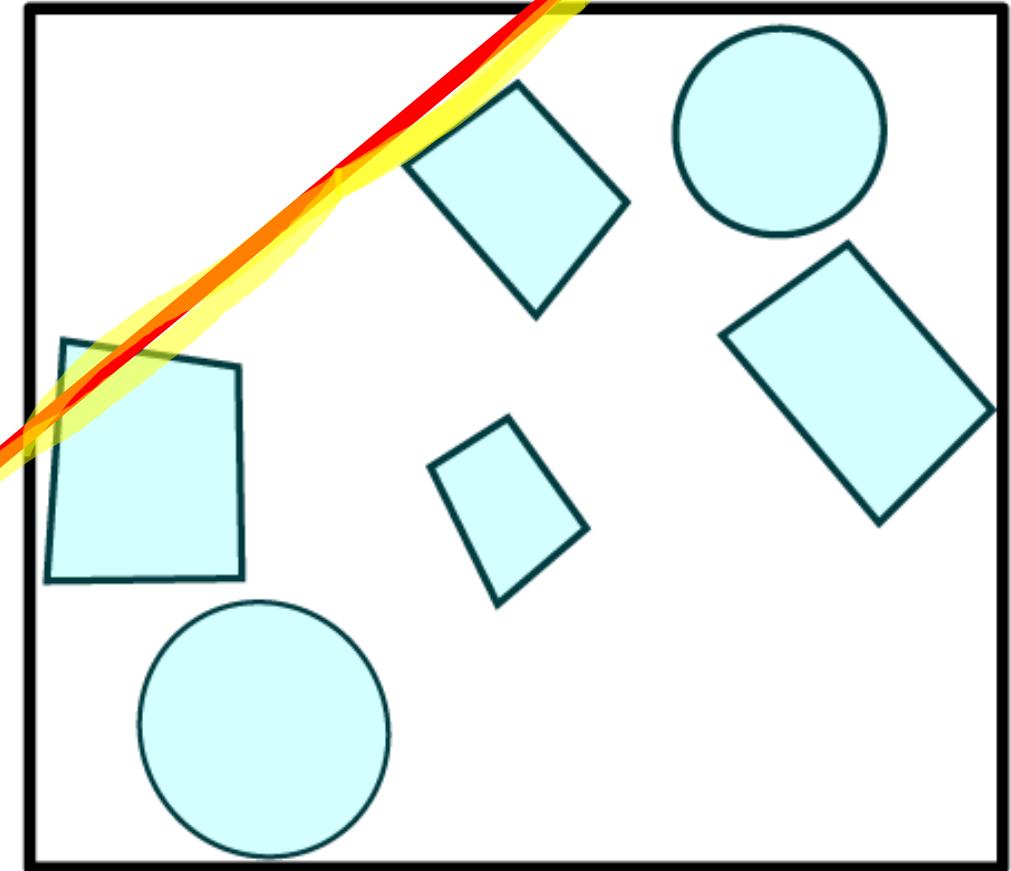
Ray-AABB Intersection

$$t_{xmin} < t_{ymax}$$

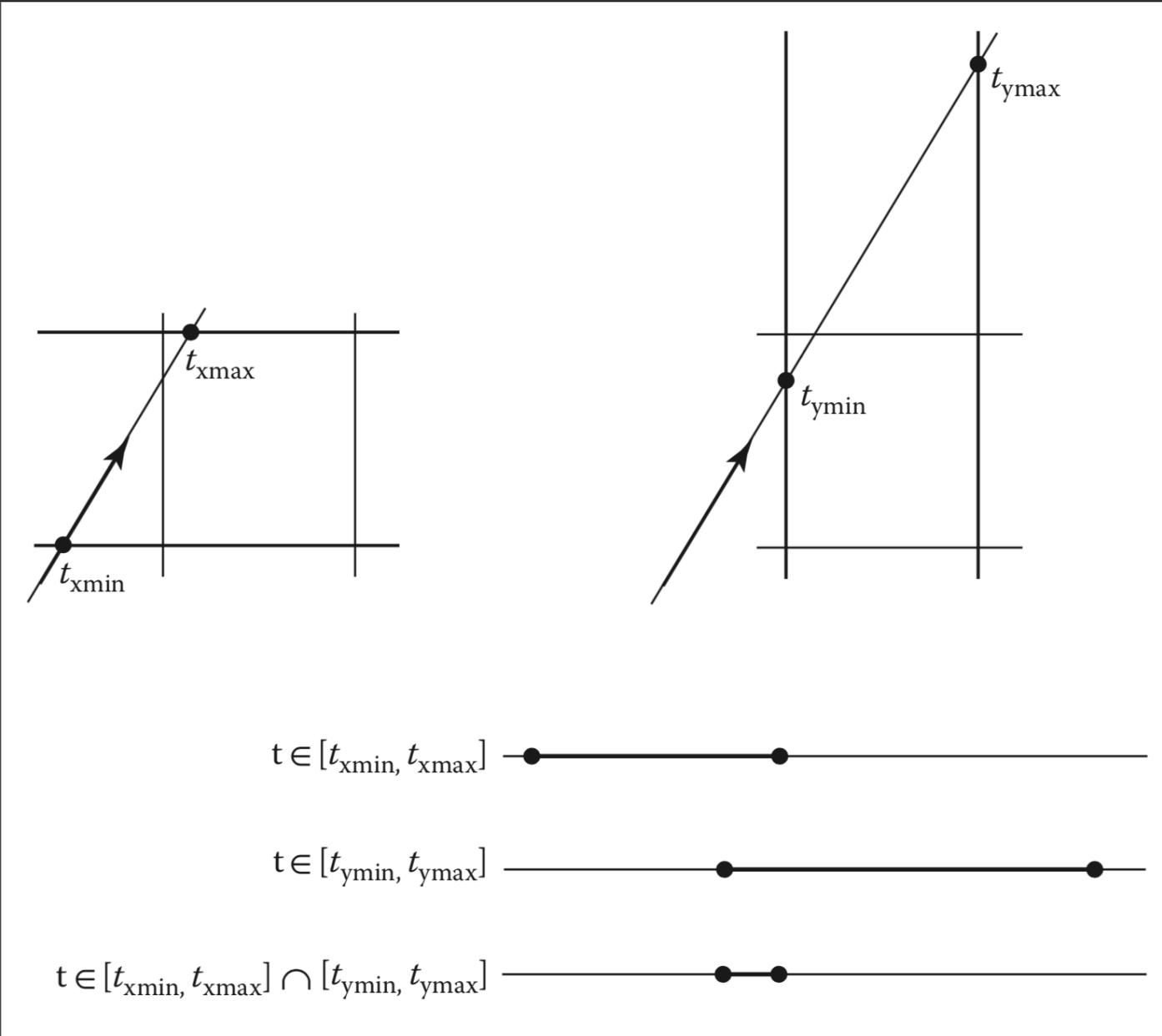


$$t_{ymax}$$

$$t_{xmin}$$



Ray-AABB Intersection



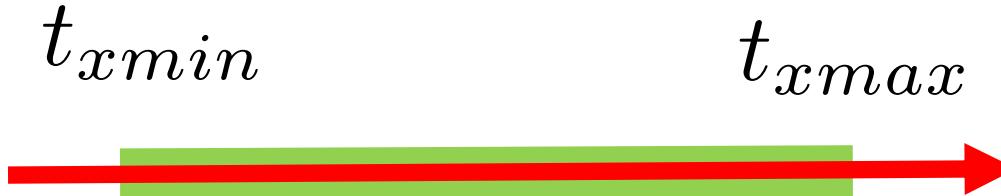
Ray-AABB Intersection



Intersection of Intervals ?



Ray-AABB Intersection



Intersection of Intervals



$$\max(t_{xmin}, t_{ymin}) \quad ? \quad \min(t_{xmax}, t_{ymax})$$

Ray-AABB Intersection

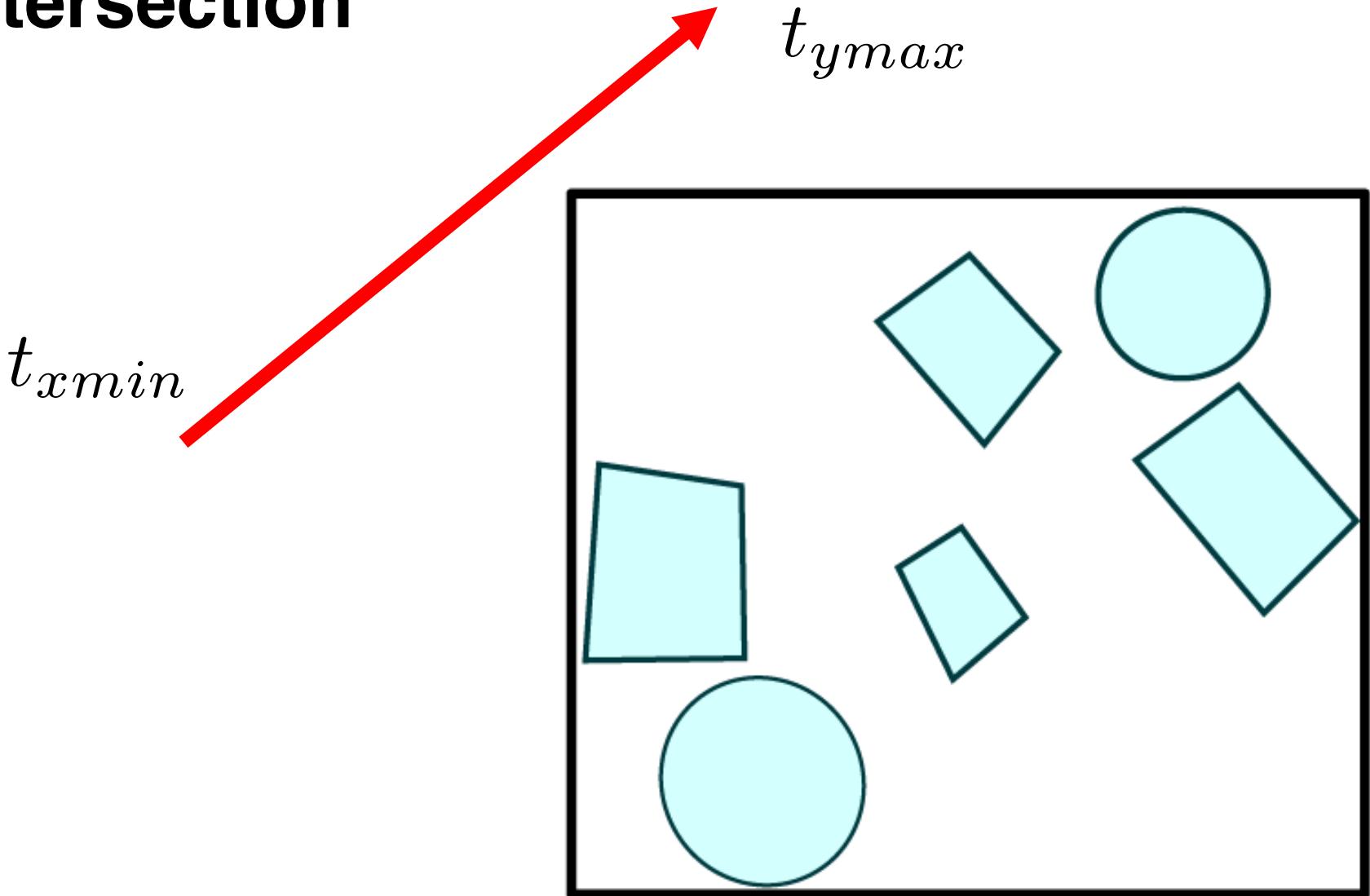


Intersection of Intervals

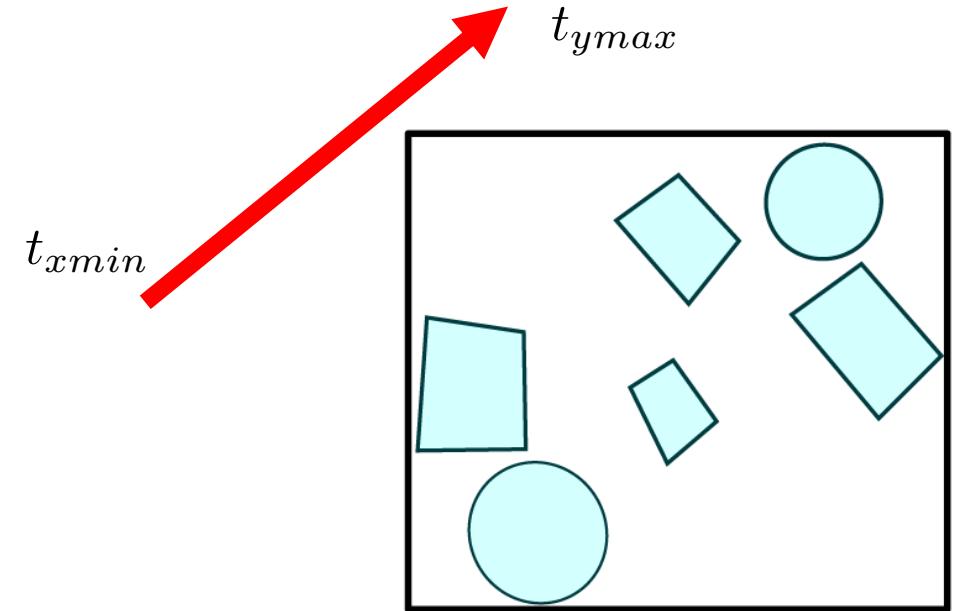
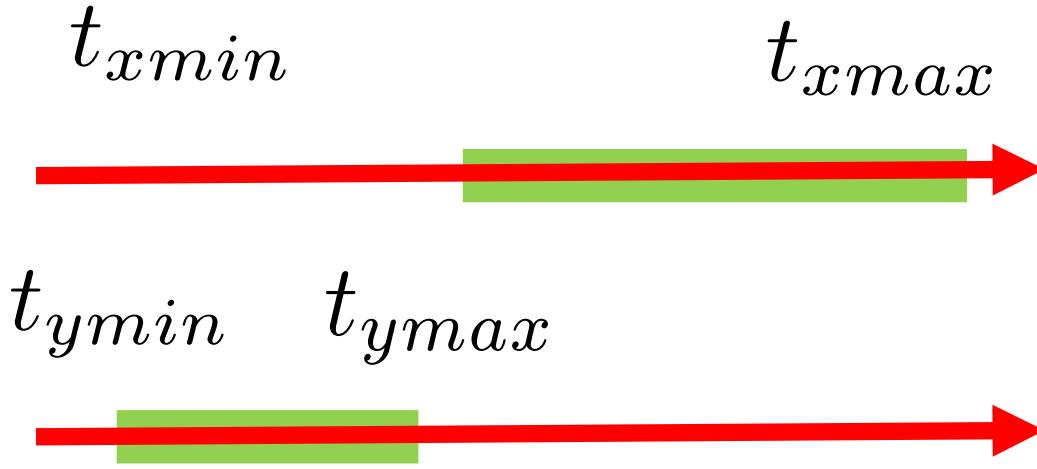


$$\max(t_{xmin}, t_{ymin}) < \min(t_{xmax}, t_{ymax})$$

Ray-AABB Intersection



Ray-AABB Intersection

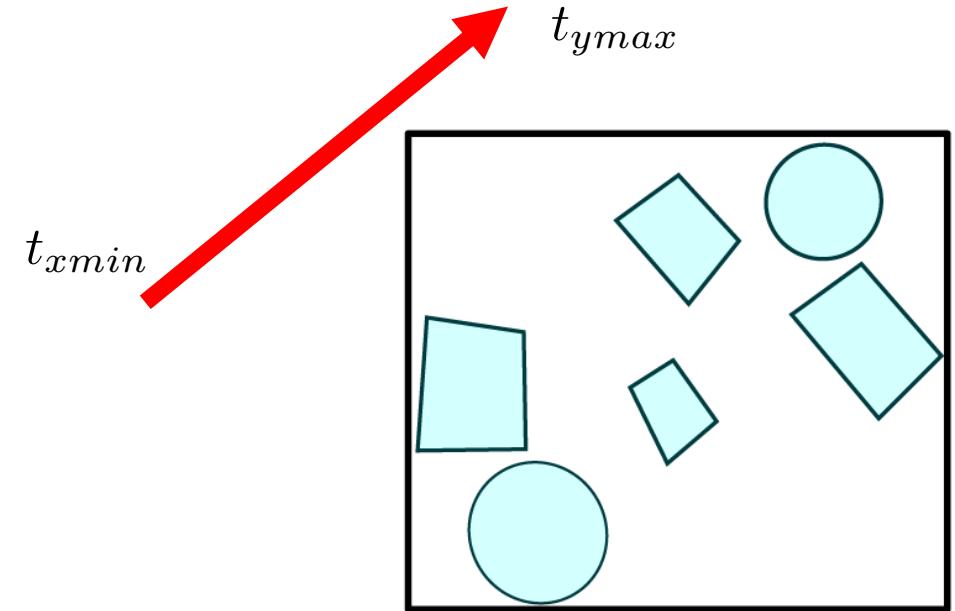
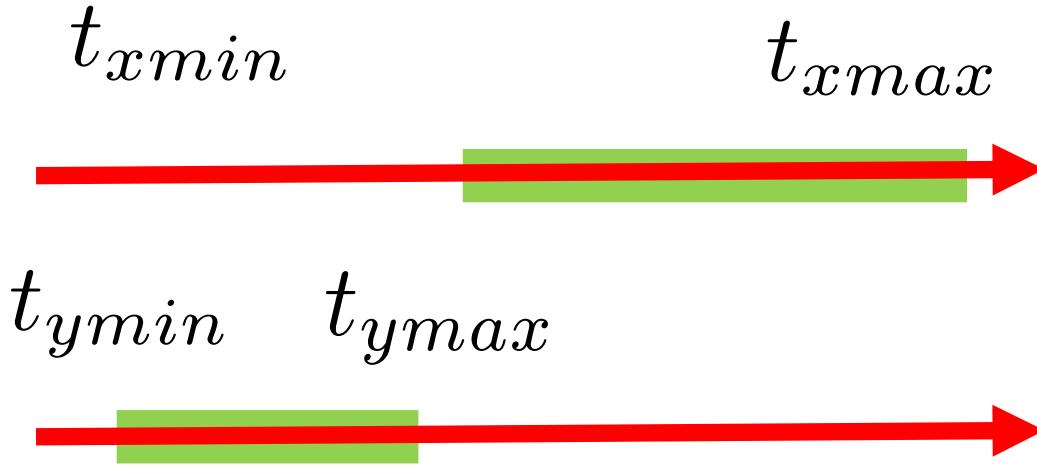


Intersection of Intervals



$$\max(t_{xmin}, t_{ymin}) \quad ? \quad \min(t_{xmax}, t_{ymax})$$

Ray-AABB Intersection

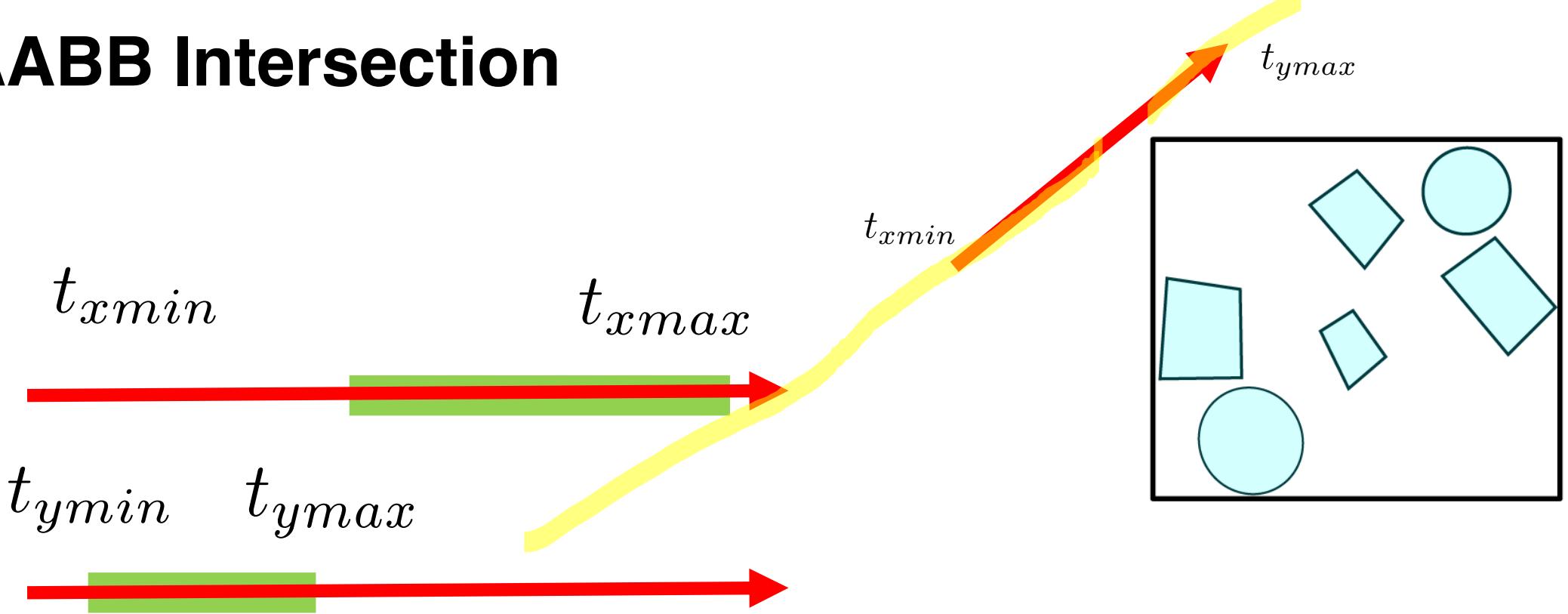


Intersection of Intervals



$$\max(t_{xmin}, t_{ymin}) > \min(t_{xmax}, t_{ymax})$$

Ray-AABB Intersection



Intersection of Intervals



$$\max(t_{xmin}, t_{ymin}) > \min(t_{xmax}, t_{ymax}) \quad \text{--> CHECK}$$

Building and Axis-Aligned Bounding Box (AABB)

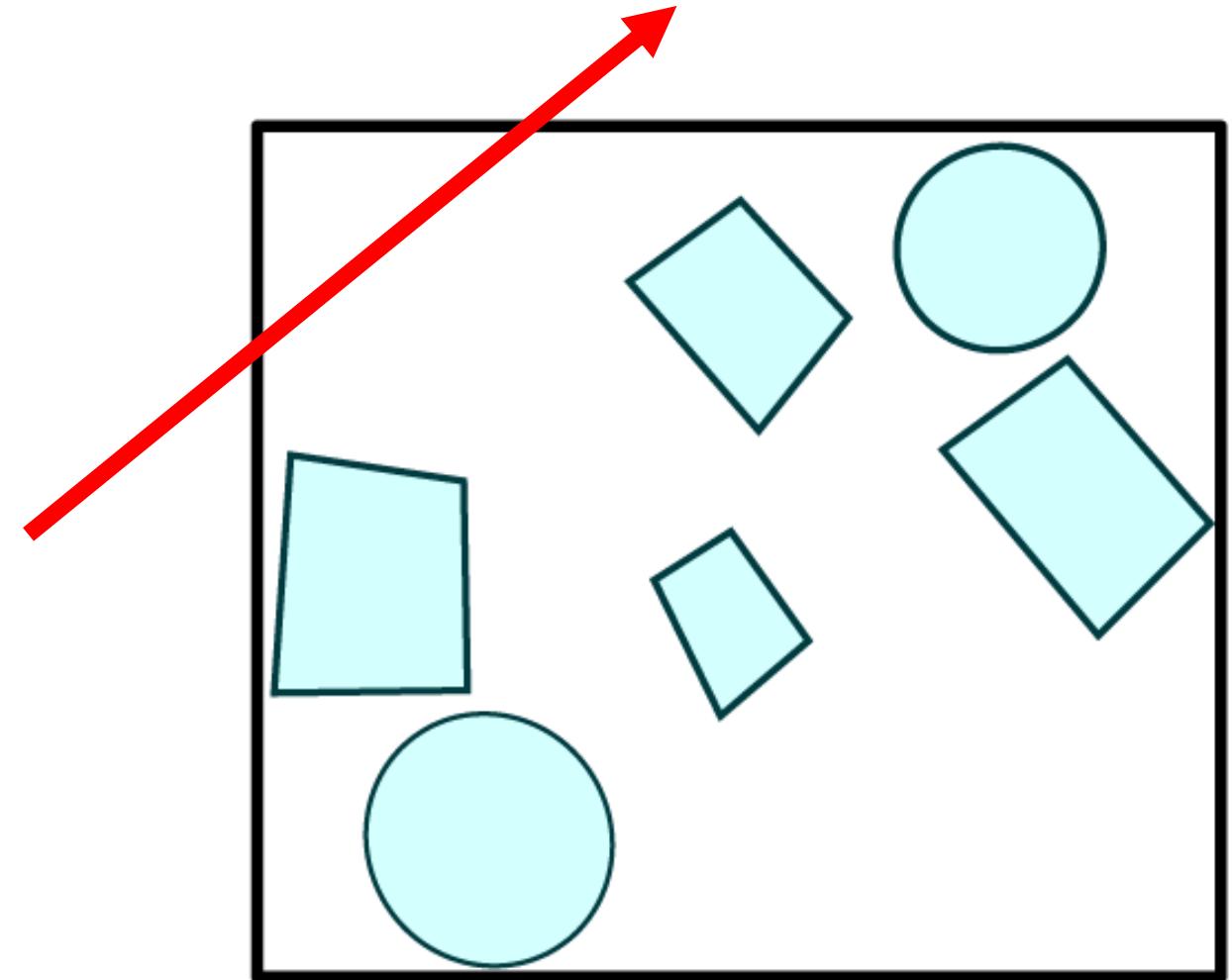
$$x_{min} = \min(v_x^i)$$

$$x_{max} = \max(v_x^i)$$

$$y_{min} = \min(v_y^i)$$

$$y_{max} = \max(v_y^i)$$

$\mathbf{v}^i \in$ Vertices



Building and Axis-Aligned Bounding Box (AABB)

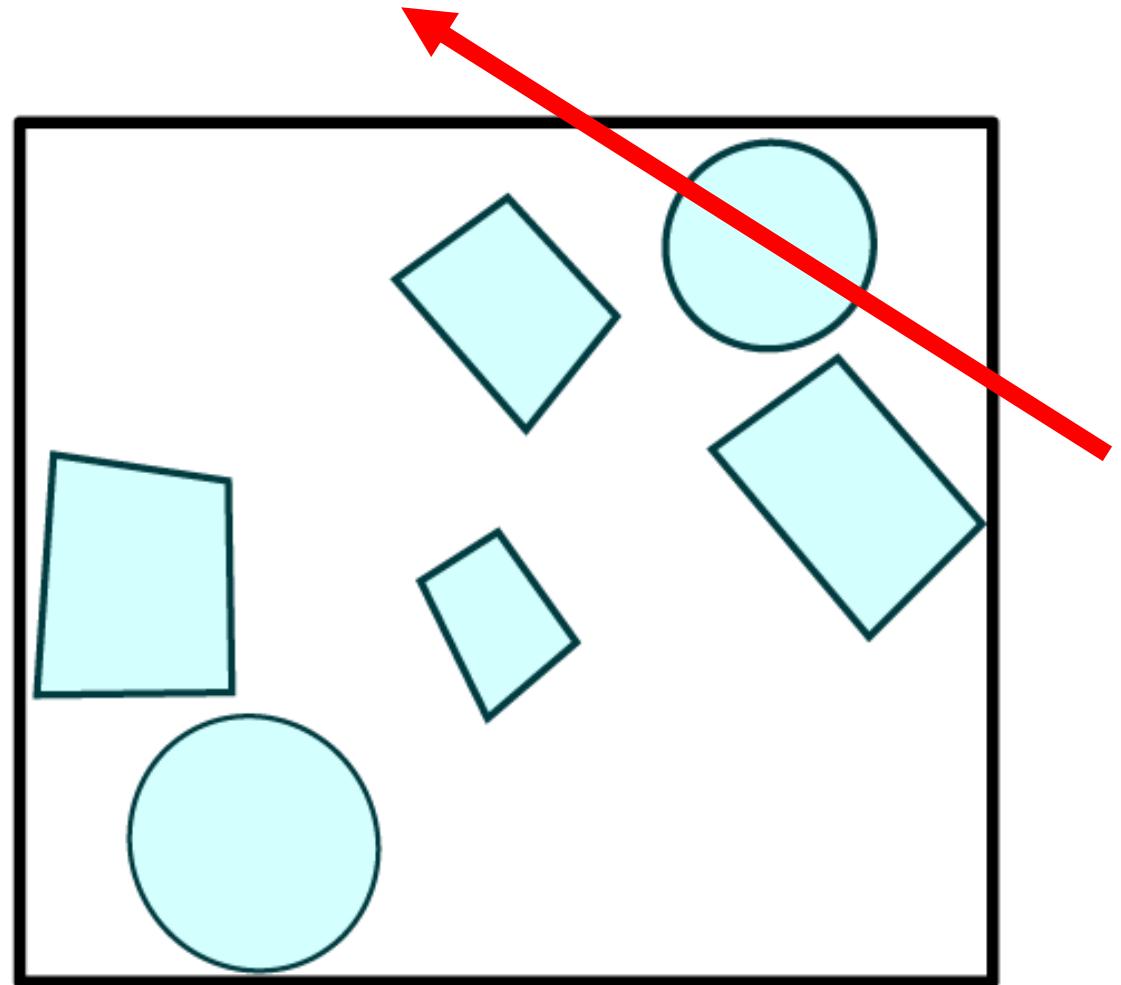
$$x_{min} = \min(v_x^i)$$

$$x_{max} = \max(v_x^i)$$

$$y_{min} = \min(v_y^i)$$

$$y_{max} = \max(v_y^i)$$

$\mathbf{v}^i \in$ Vertices



Building and Axis-Aligned Bounding Box (AABB)

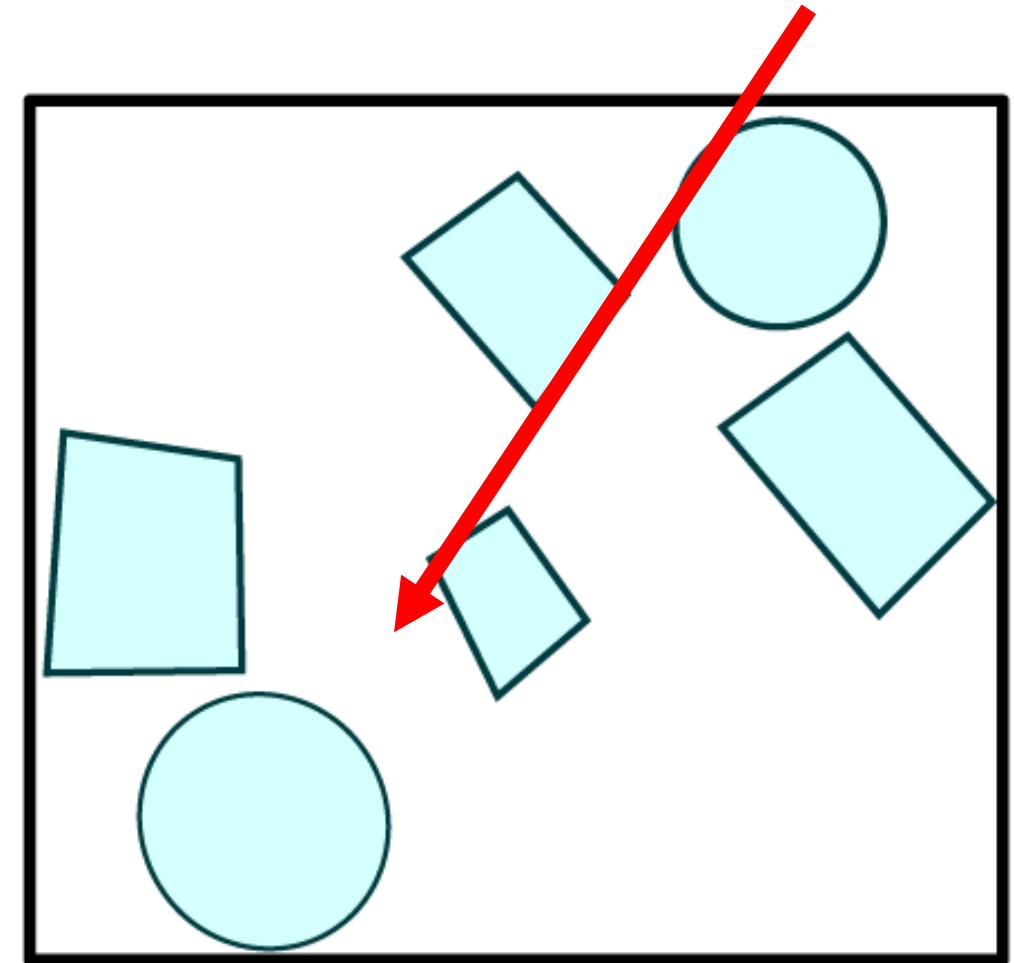
$$x_{min} = \min(v_x^i)$$

$$x_{max} = \max(v_x^i)$$

$$y_{min} = \min(v_y^i)$$

$$y_{max} = \max(v_y^i)$$

$\mathbf{v}^i \in$ Vertices



Ray-AABB Intersection

if ($x_d \geq 0$) **then**

$$t_{\text{xmin}} = (x_{\text{min}} - x_e) / x_d$$

$$t_{\text{xmax}} = (x_{\text{max}} - x_e) / x_d$$

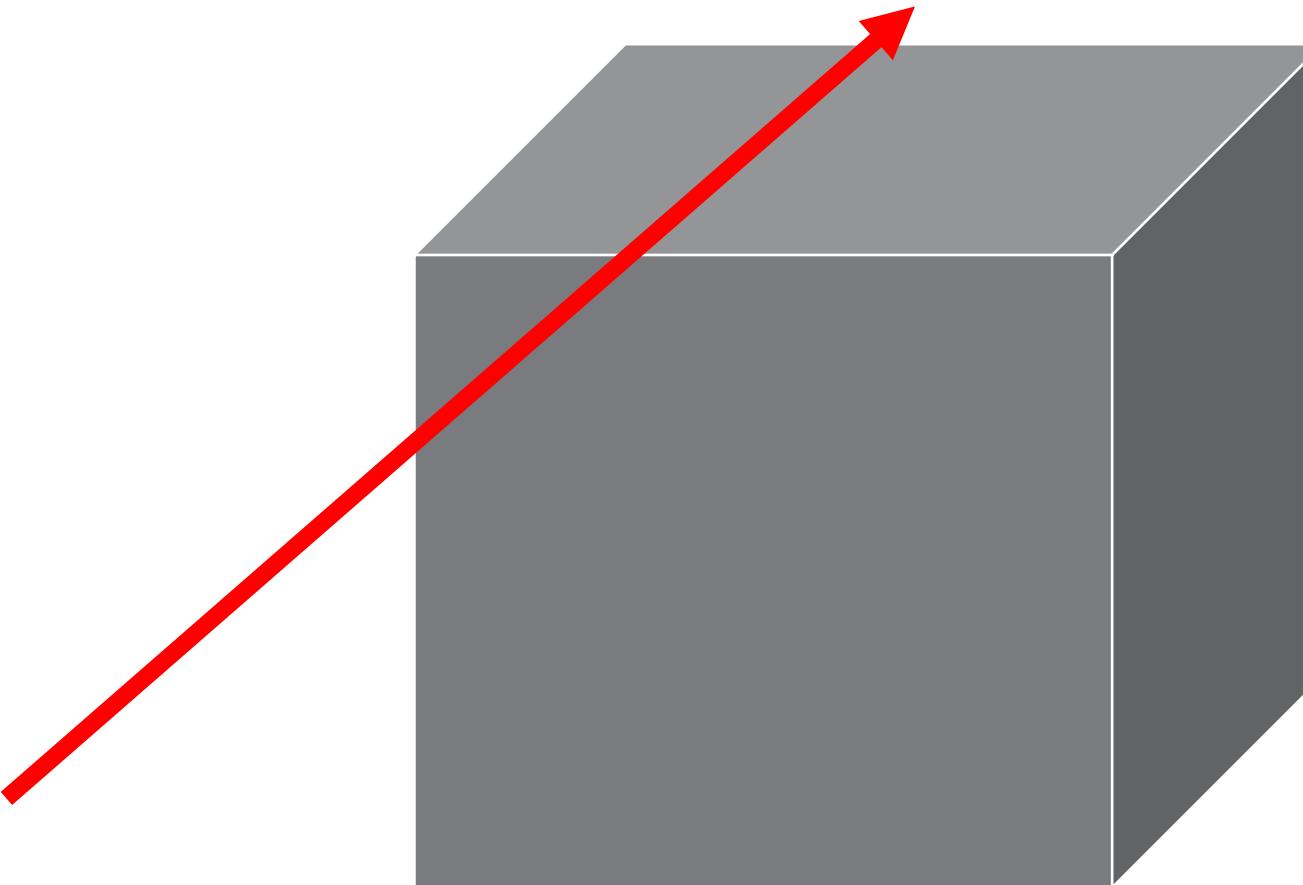
else

$$t_{\text{xmin}} = (x_{\text{max}} - x_e) / x_d$$

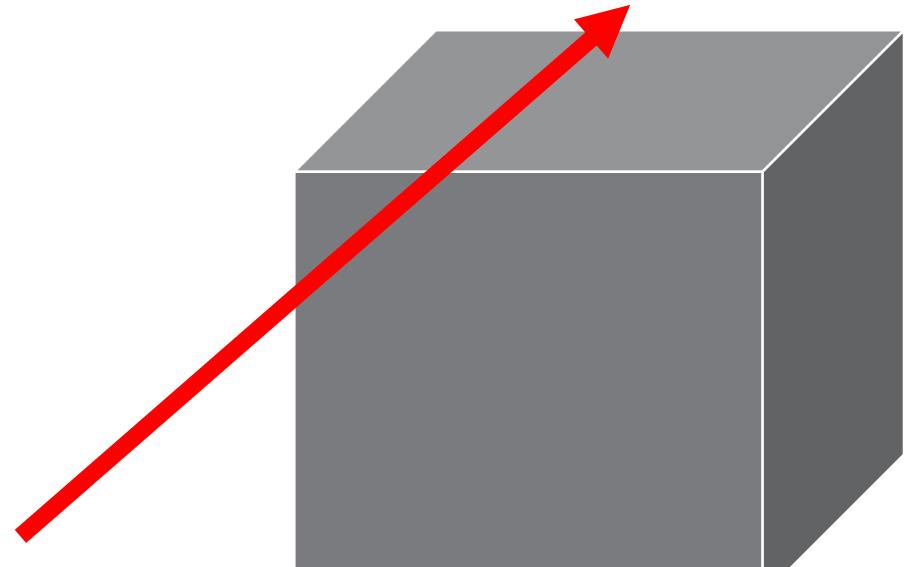
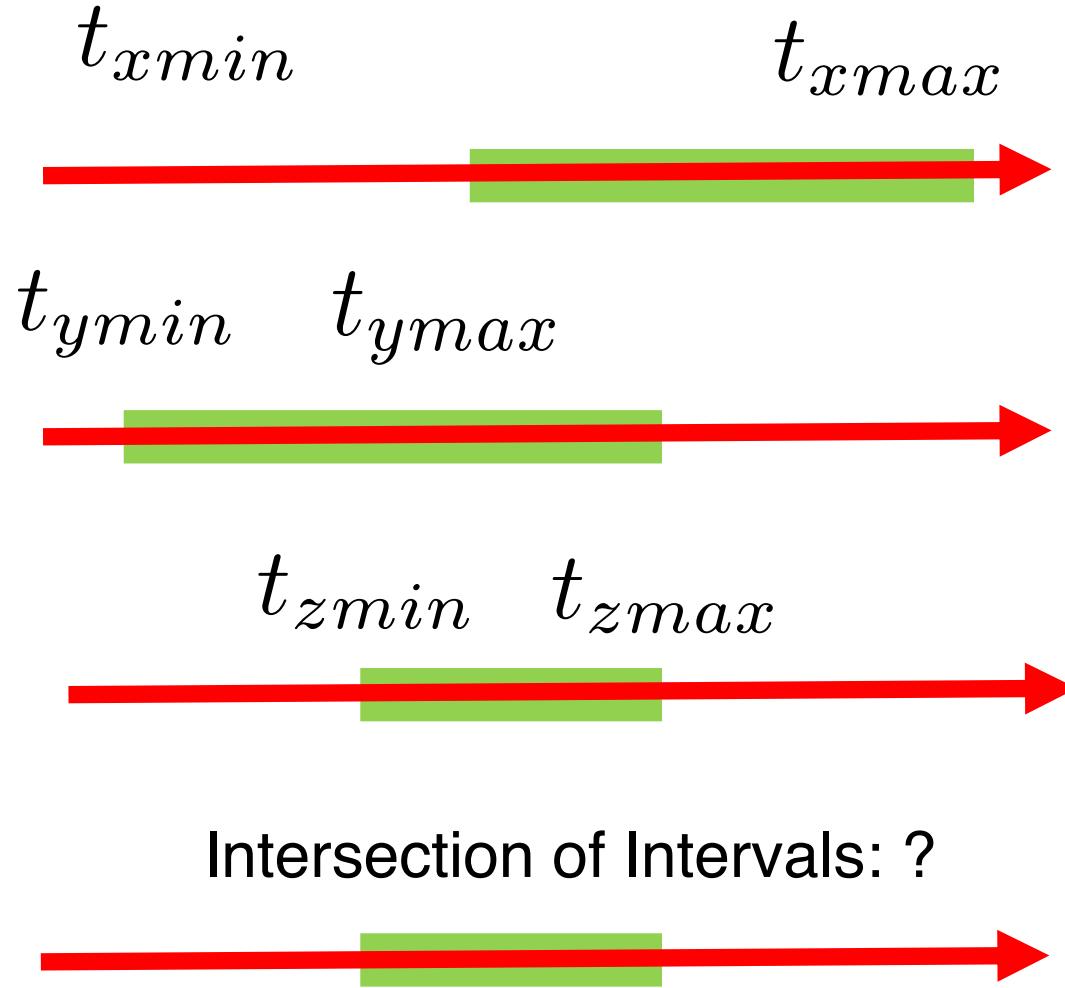
$$t_{\text{xmax}} = (x_{\text{min}} - x_e) / x_d$$

When does this fail ?

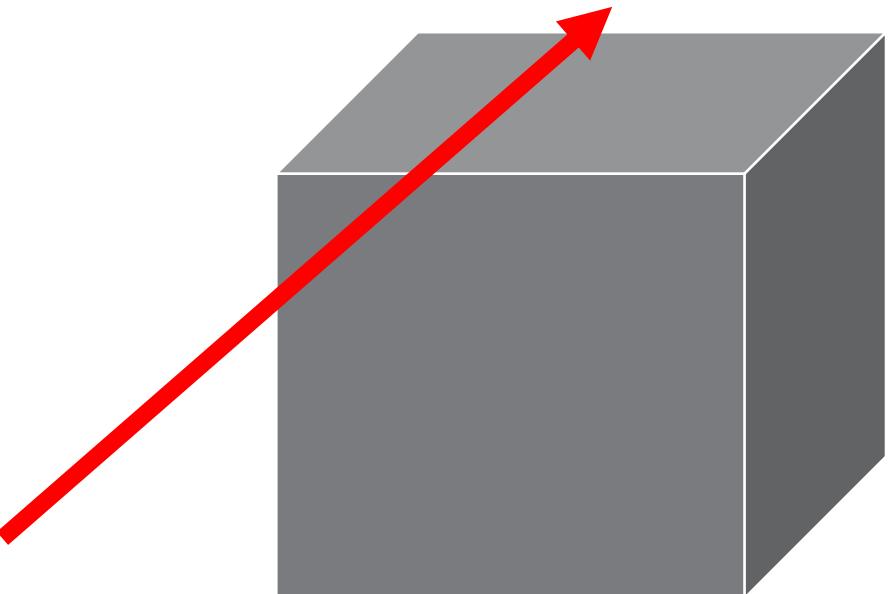
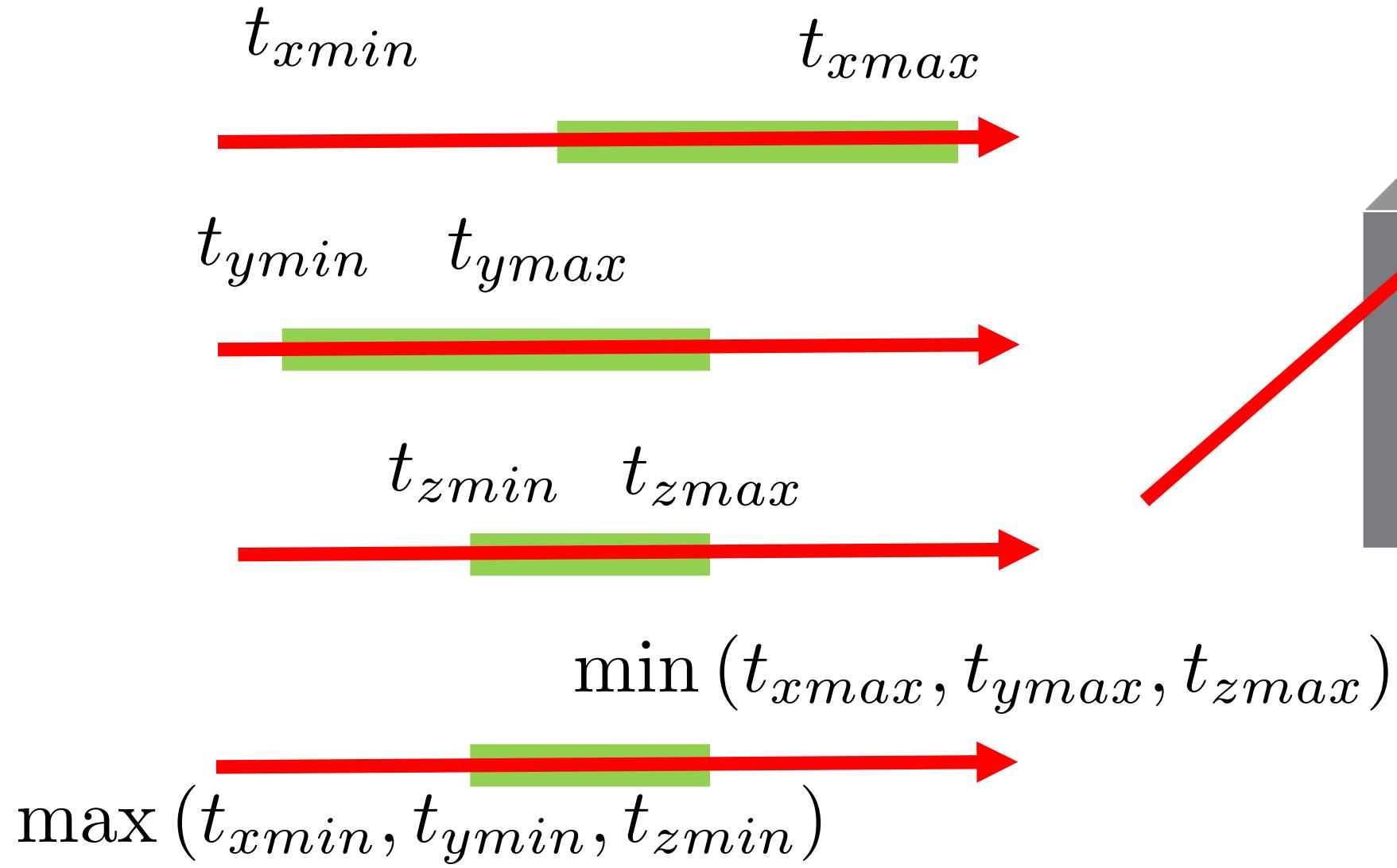
What happens in 3D ?



Ray-AABB Intersection



Ray-AABB Intersection



Ray-AABB Intersection

if ($x_d \geq 0$) **then**

$$t_{\text{xmin}} = (x_{\text{min}} - x_e) / x_d$$

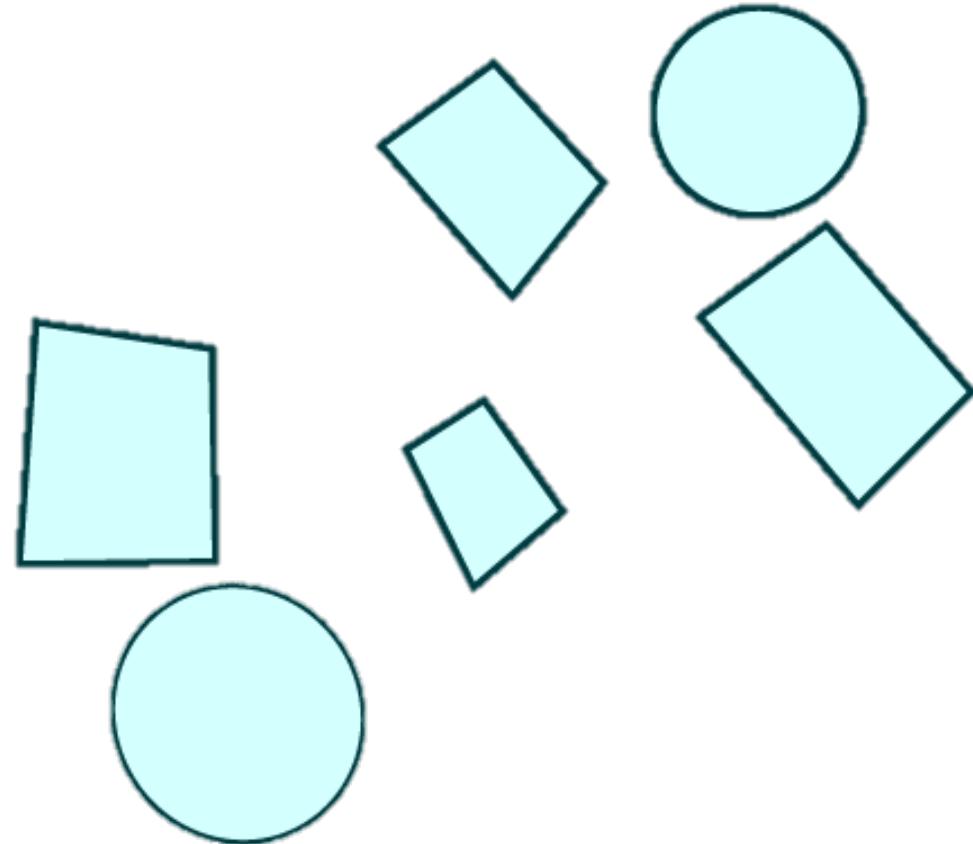
$$t_{\text{xmax}} = (x_{\text{max}} - x_e) / x_d$$

else

$$t_{\text{xmin}} = (x_{\text{max}} - x_e) / x_d$$

$$t_{\text{xmax}} = (x_{\text{min}} - x_e) / x_d$$

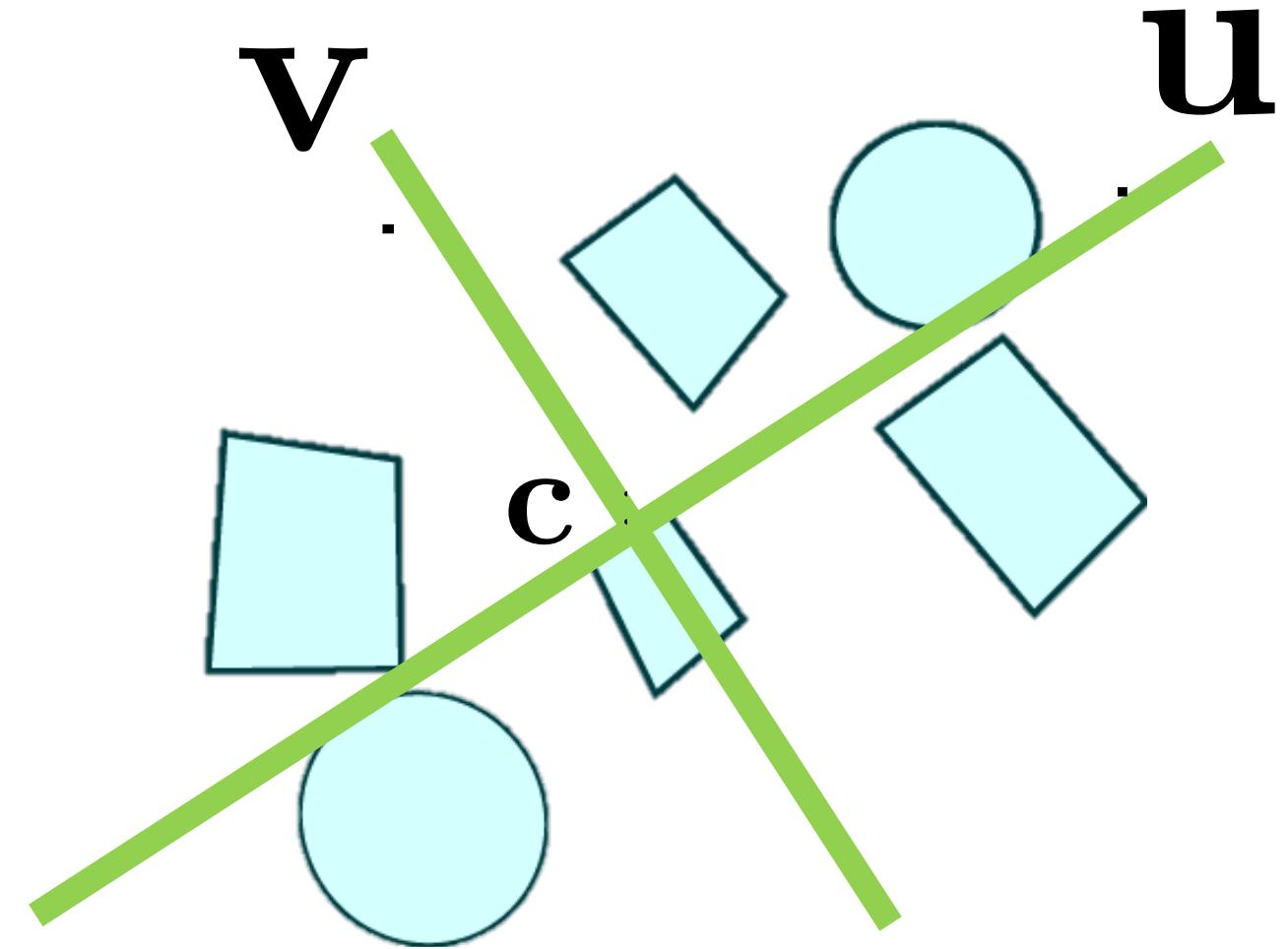
Building an Object-Oriented Bounding Box (OOBB)



Building an Object-Oriented Bounding Box (OOBB)

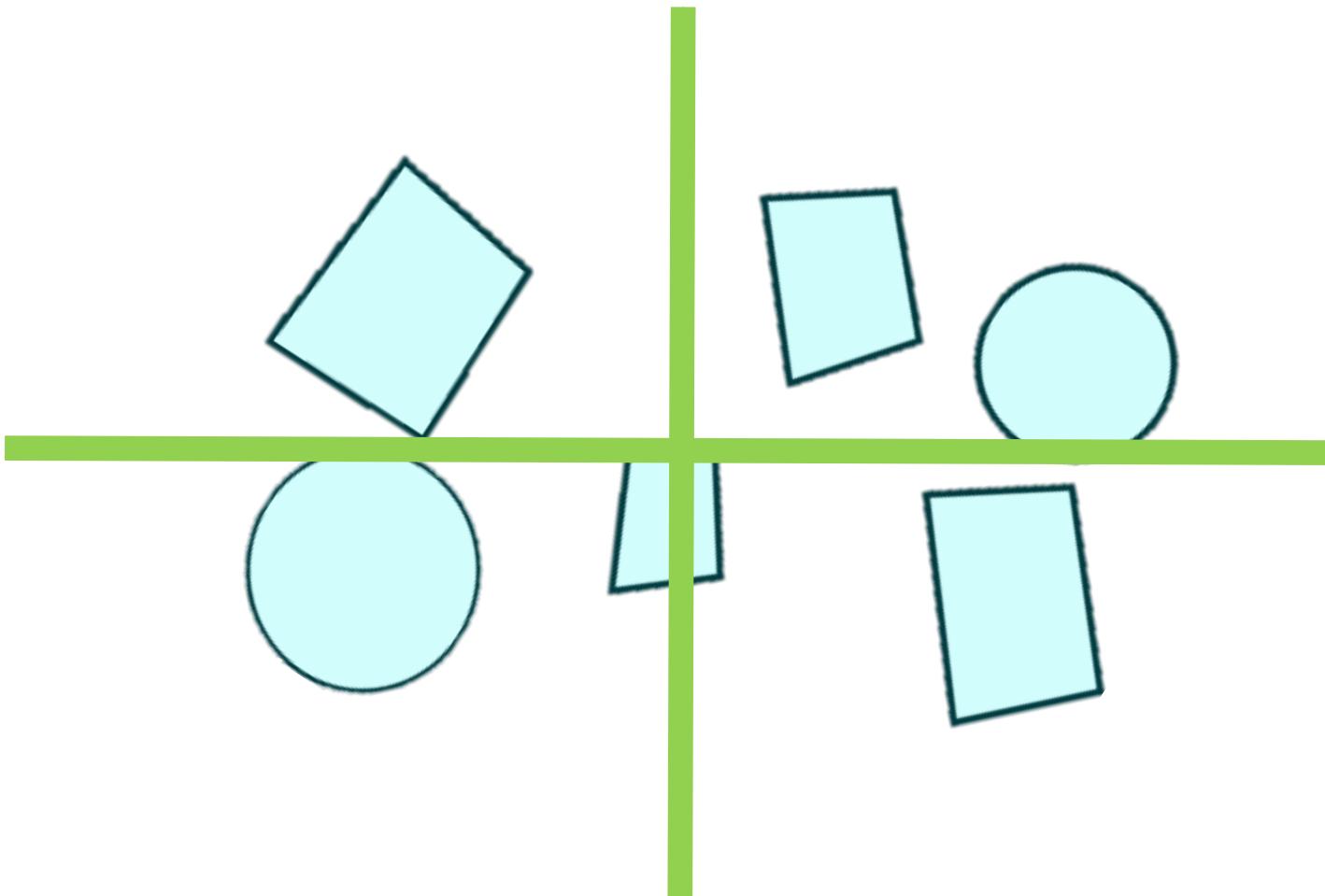
$$c = \frac{1}{n} \sum_{i=1}^n v^i$$

$$[u \quad v]$$



Find directions of maximum and minimum variance

Building an Object-Oriented Bounding Box (OOBB)



Build Rotation Matrix

Collision Query with Object-Oriented Bounding Box



Spatial Data Structures

Basic Idea – asymptotic improvement in spatial queries by subdividing

Two types of subdivisions – *object-based* and *spatial*

Spatial Data Structures

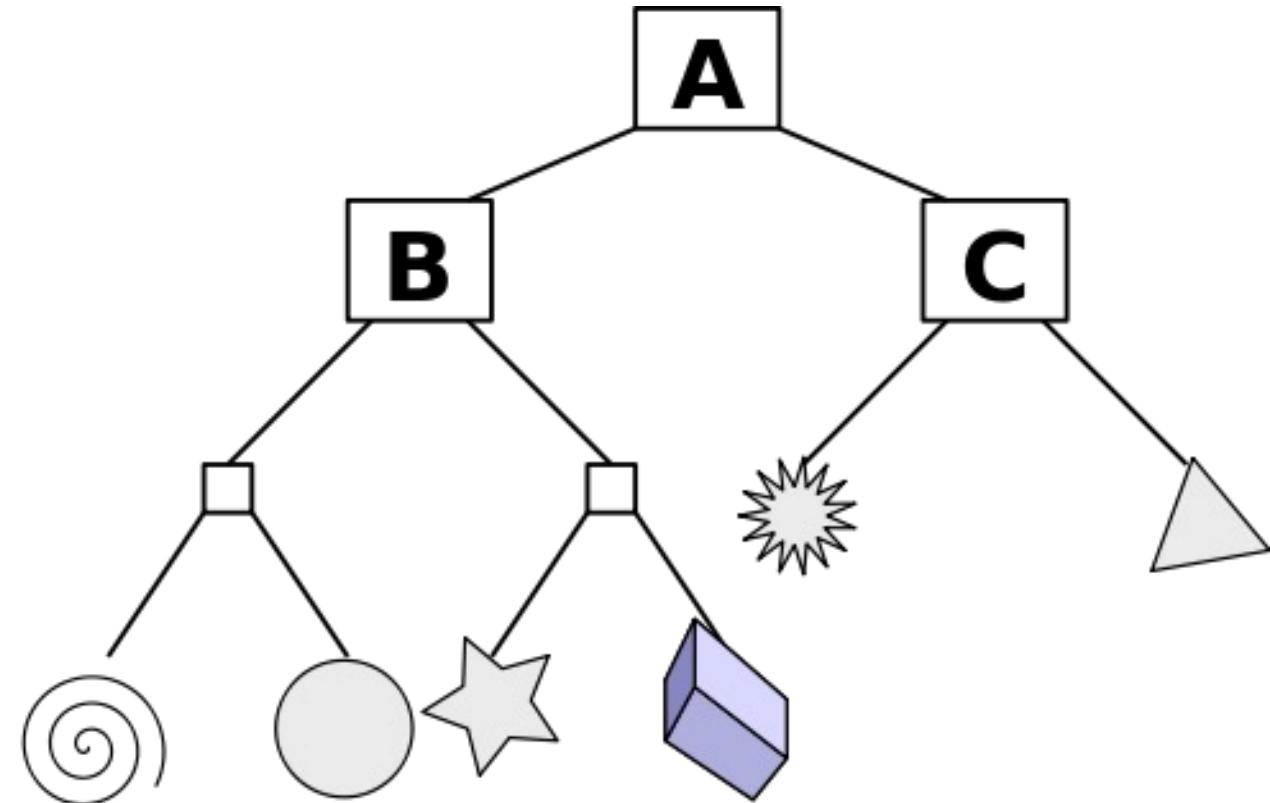
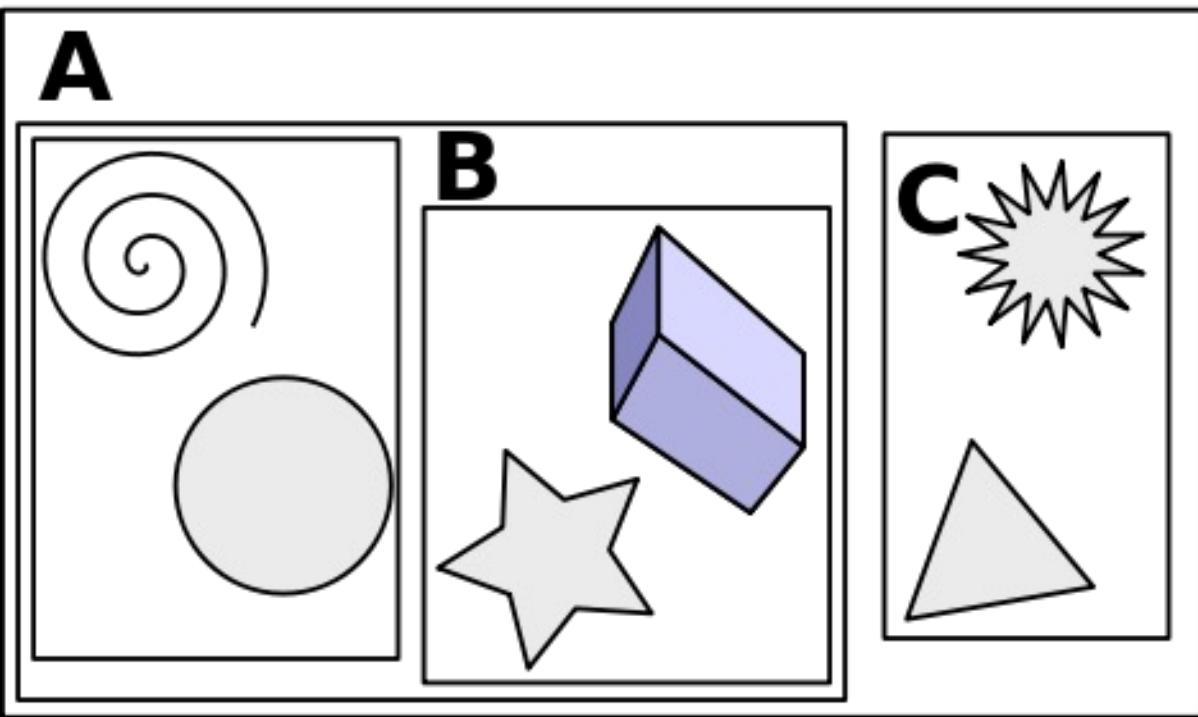
Basic Idea – asymptotic improvement in spatial queries by subdividing

Two types of subdivisions – ***object-based*** and ***spatial***

Our object-based data structures will be boundary volume hierarchies or BVHs.

BVHs are hierarchies of BVs represented by trees

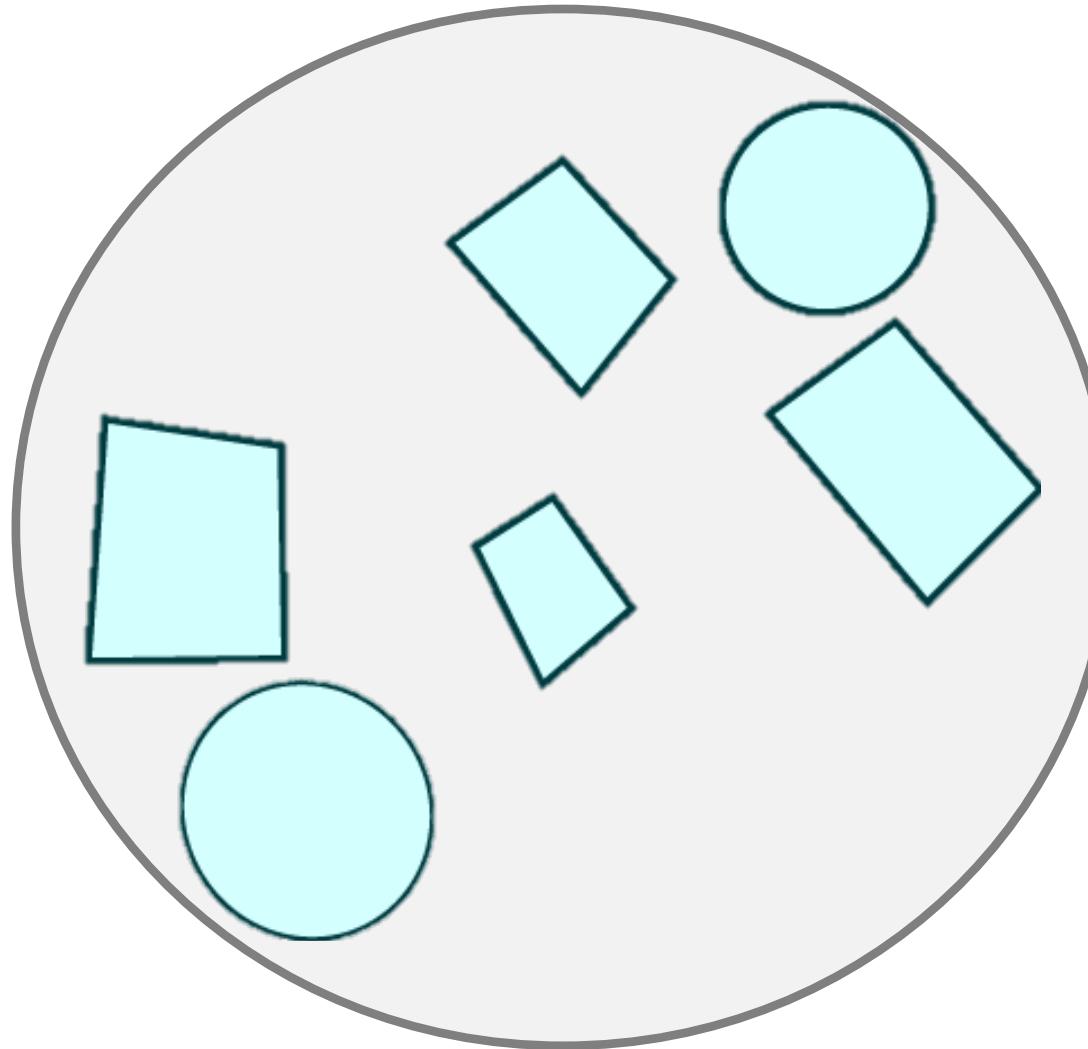
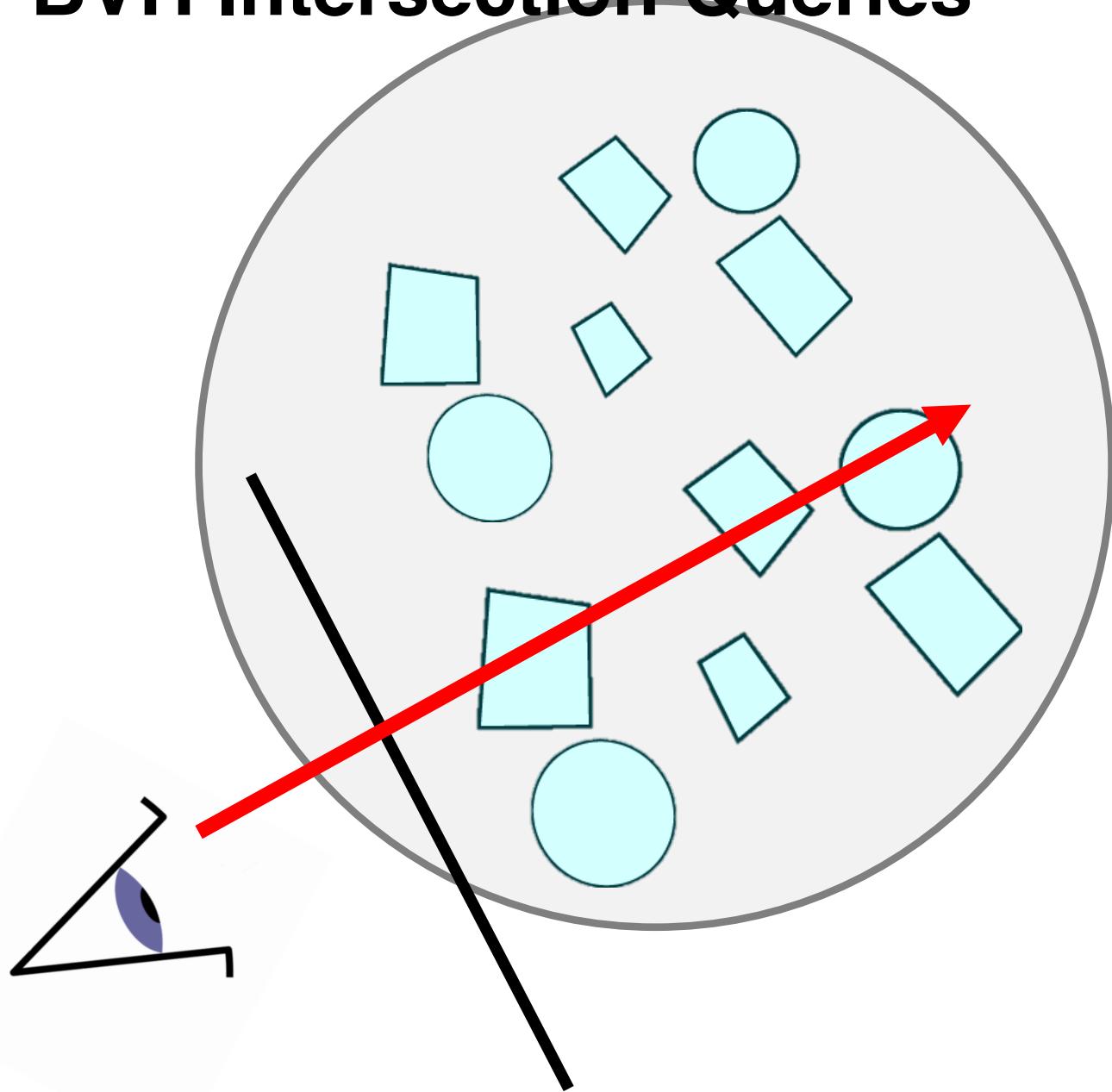
Bounding Volume Hierarchy



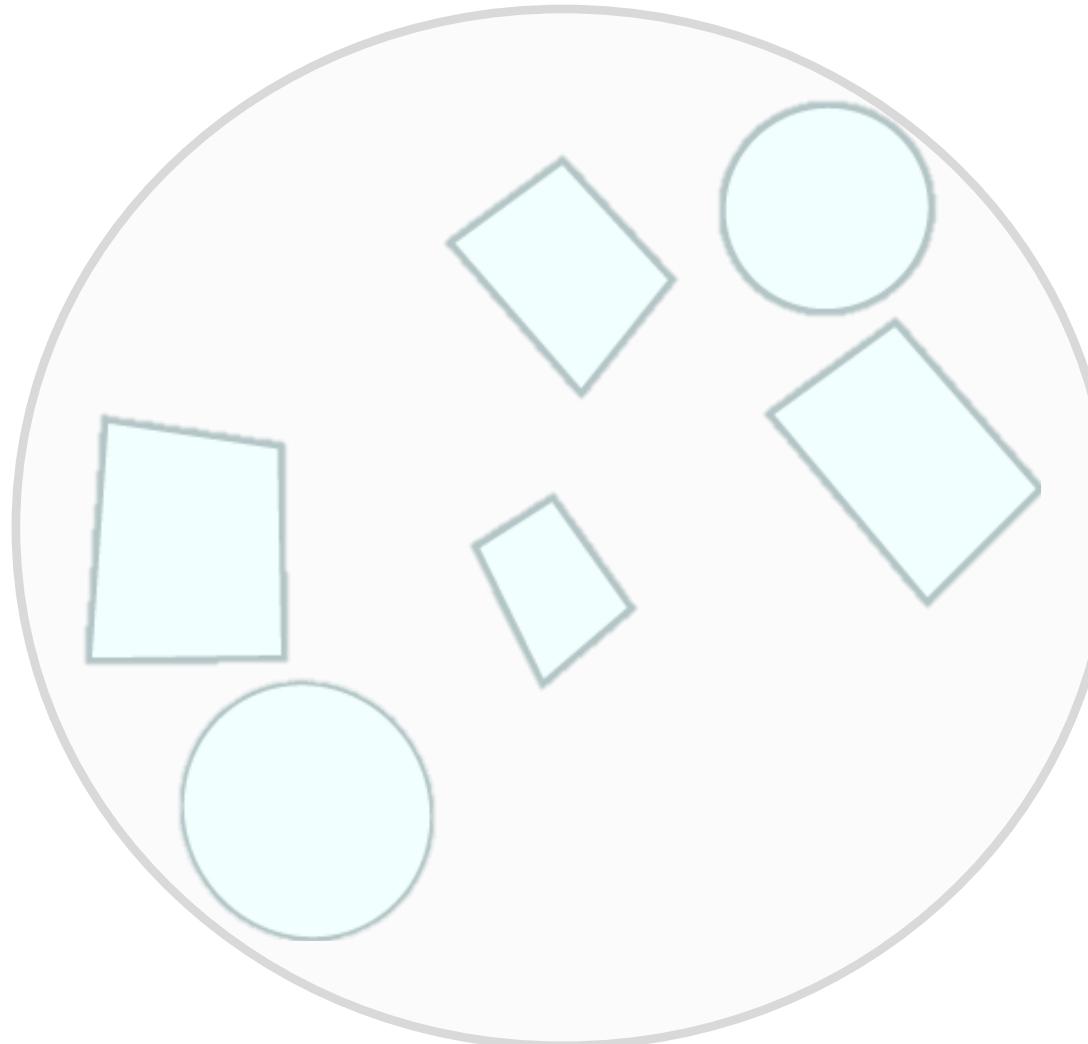
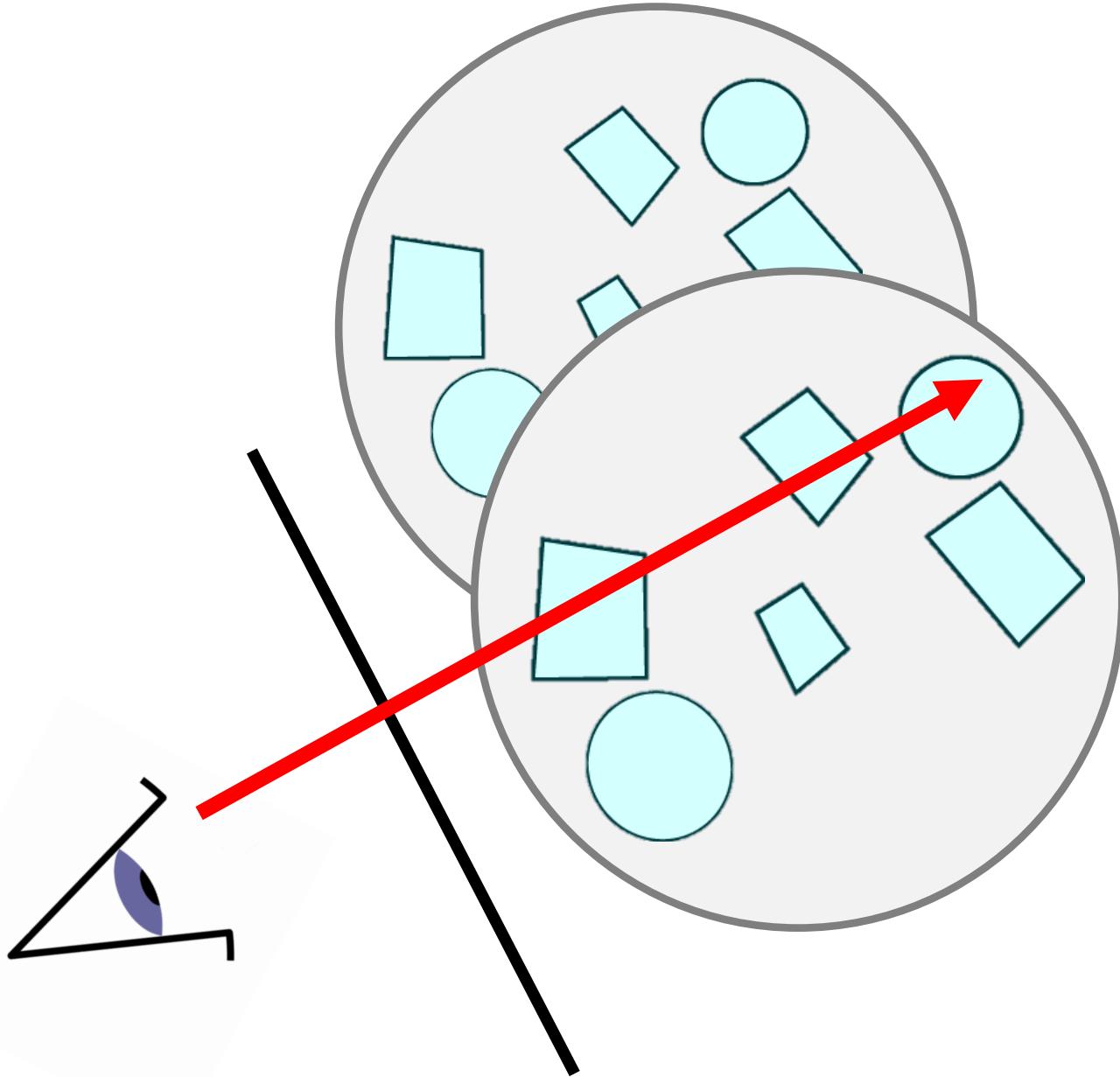
BVH Intersection Queries

```
intersect(bvNode, ray,t)
{
    if (bvNode== null || !bvNode.intersect(ray,t))
        return false;
    else
    {
        i1=intersect(bvNode.left, ray,t1); //check left BV
        i2=intersect(bvNode.right ray,t2); //check right BC
        if (i1 && i2) { t=min(t1,t2); return true; }
        if (i1) { t=t1; return true; }
        if (i2) { t=t2; return true; }
        return false;
    }
}
```

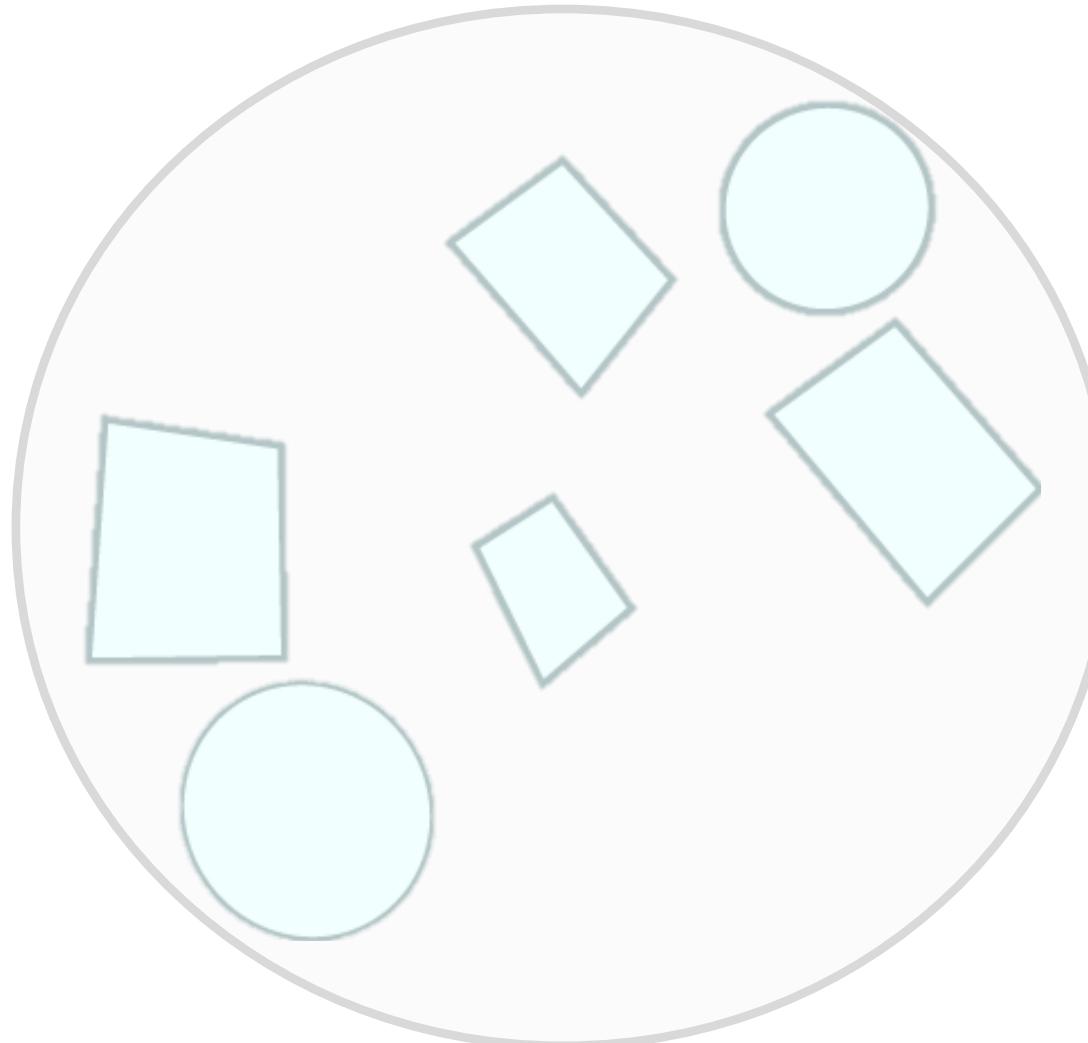
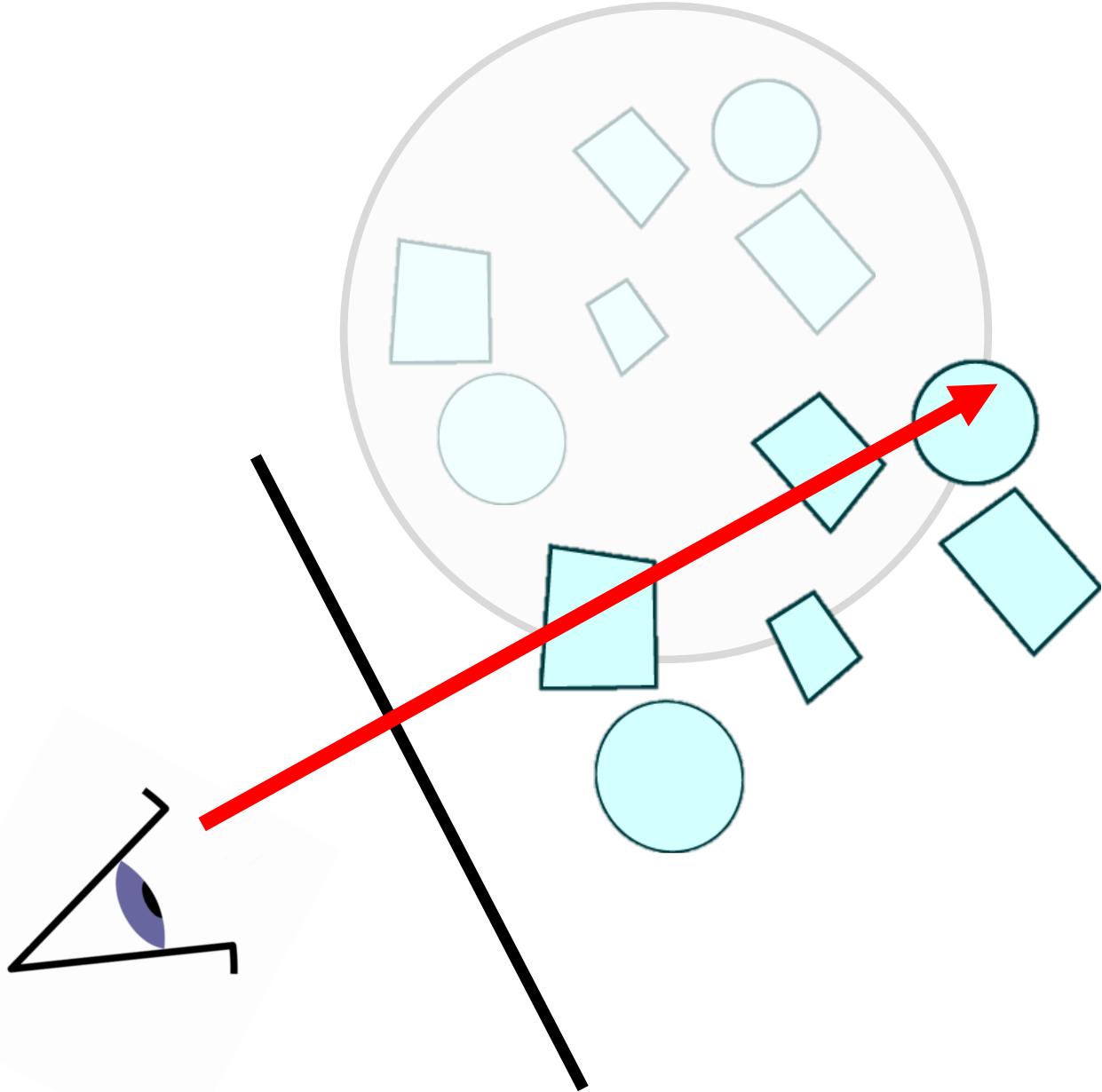
BVH Intersection Queries



BVH Intersection Queries



BVH Intersection Queries



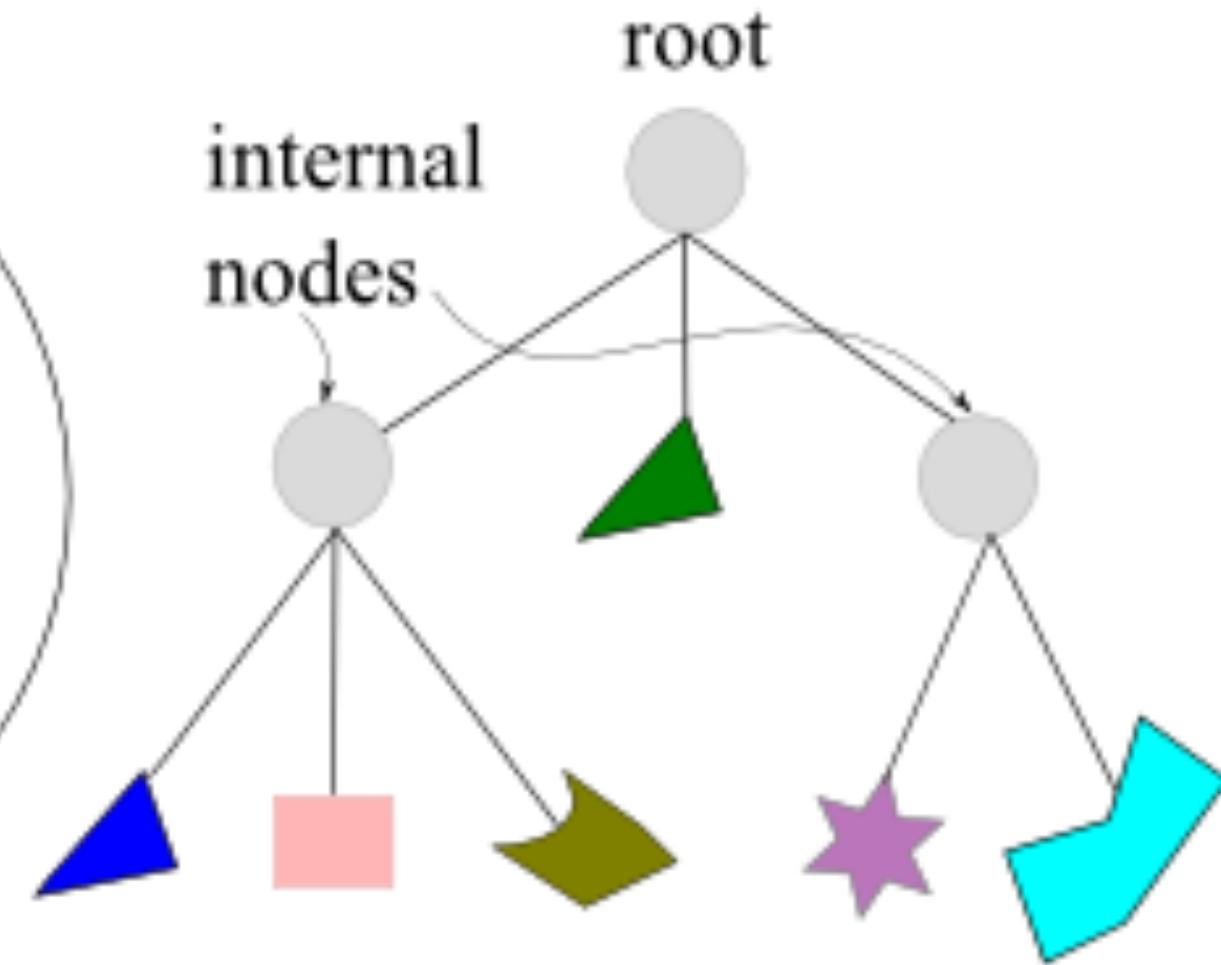
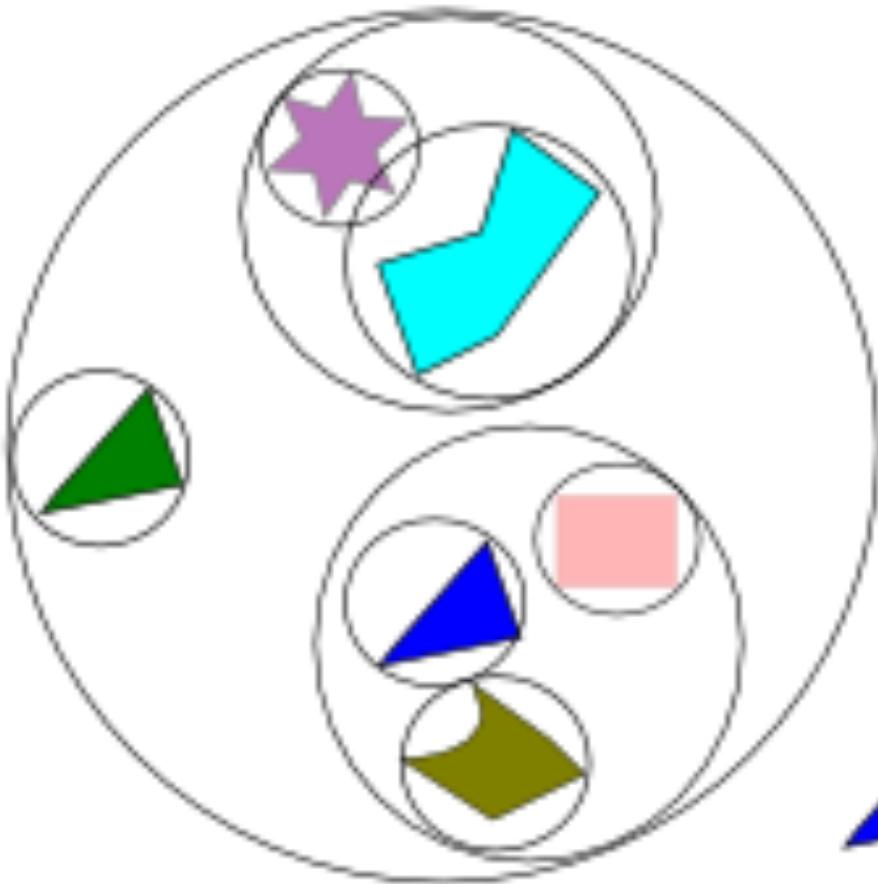
BVH Distance Queries

```
minDistance(bvNode, point, currentMin)
{
    d1=minDistance(bvNode.left, point, currentMin);
    d2=minDistance(bvNode.right, point, currentMin);

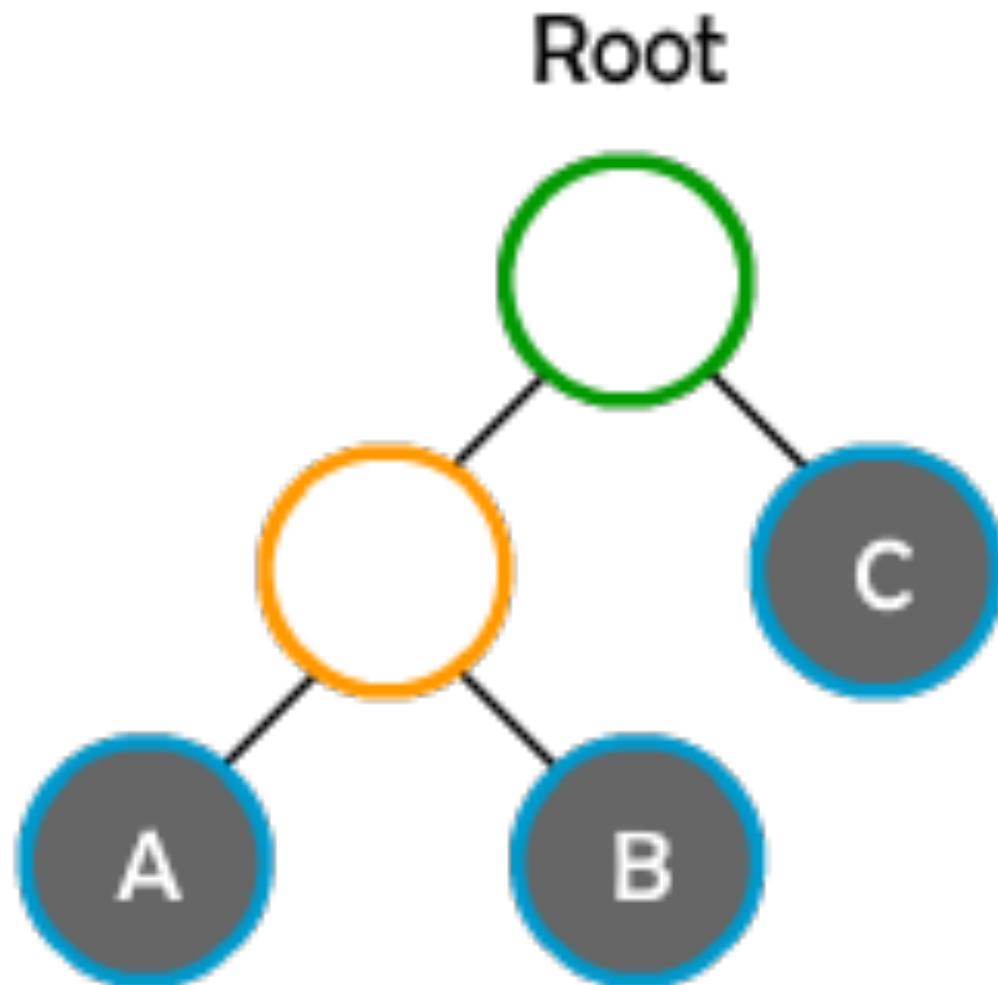
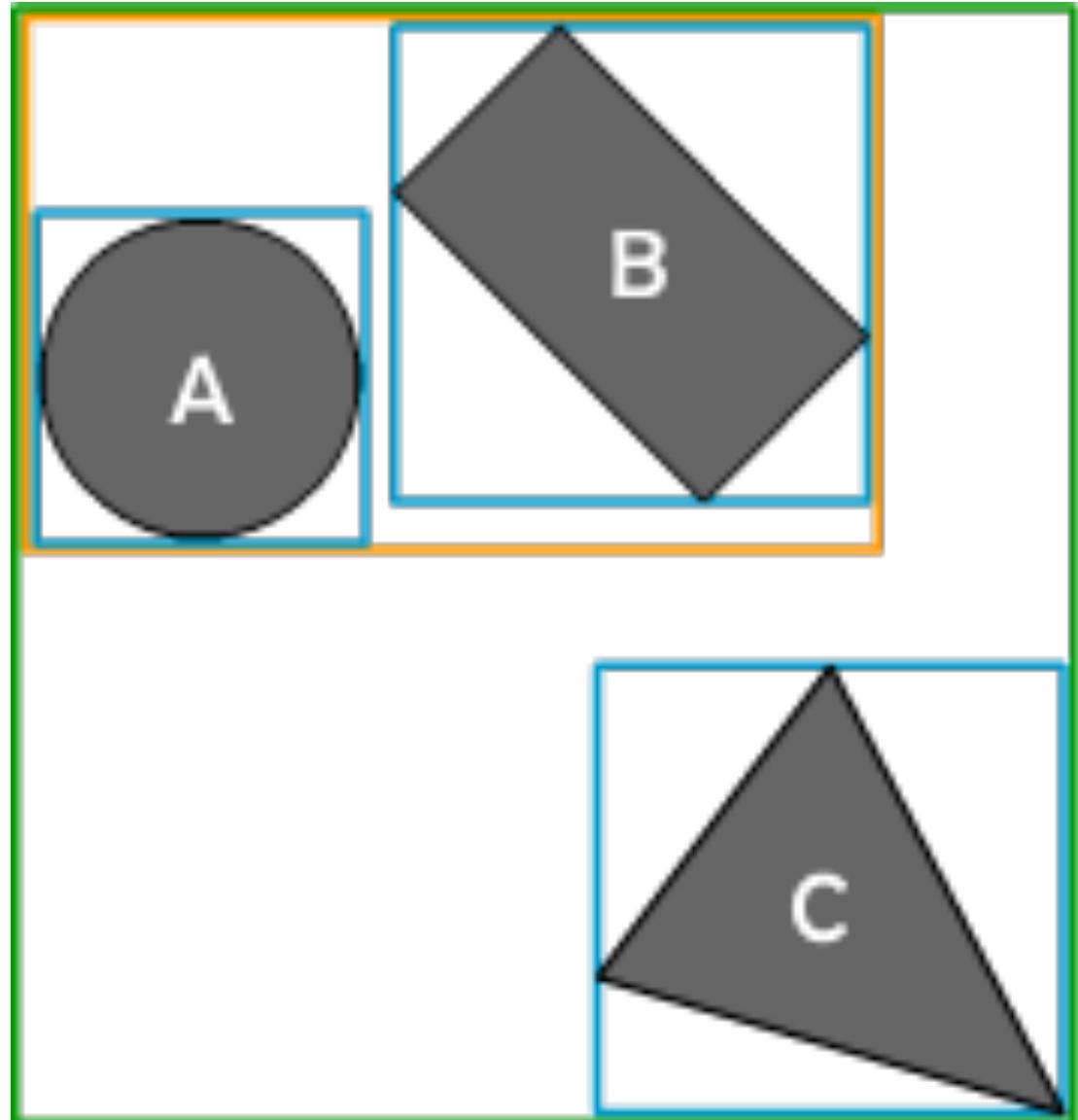
    if(min(d1,d2) > currentMin) {
        return currentMin
    }

    return min(d1,d2)
}
```

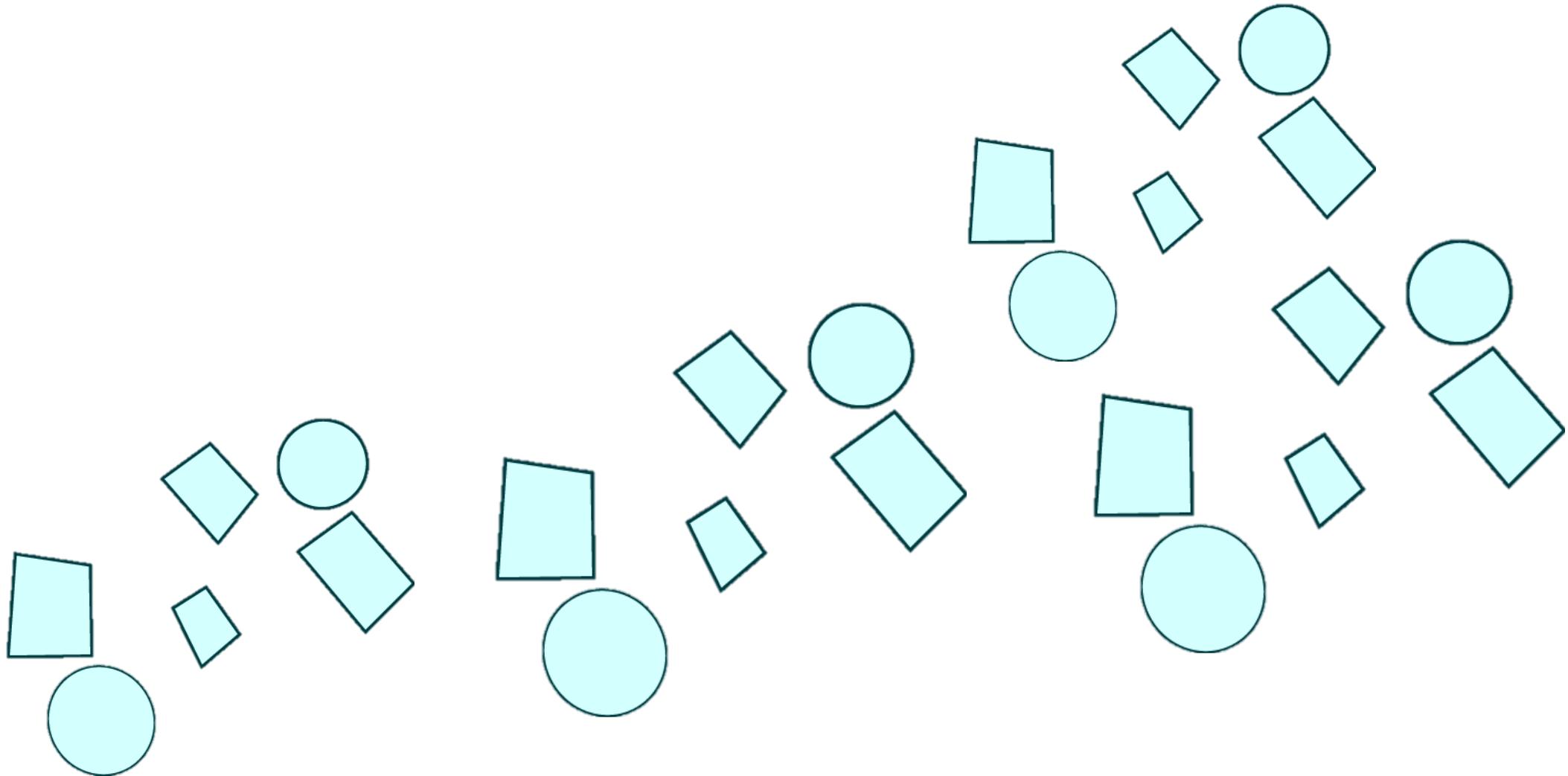
Sphere Trees



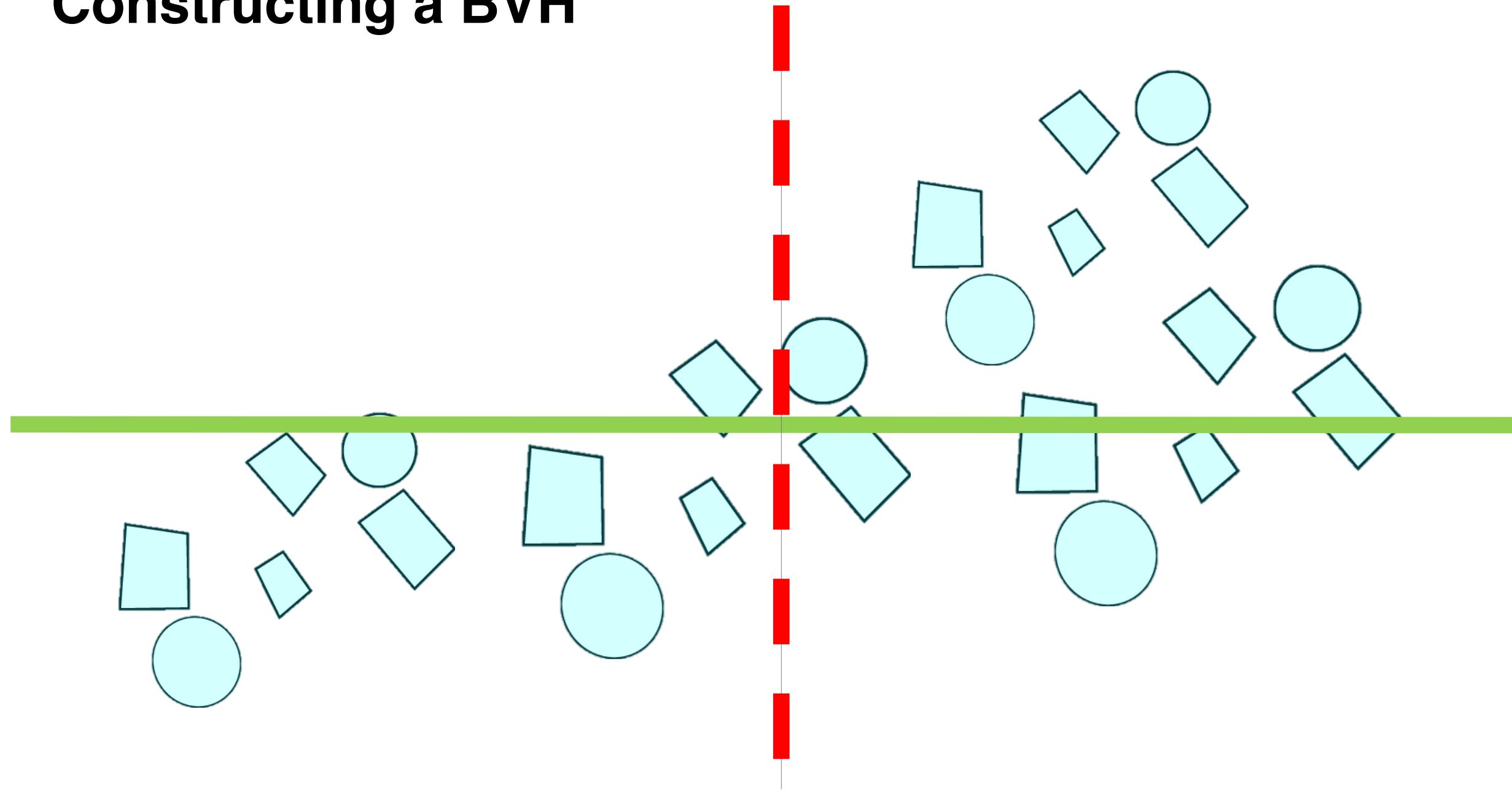
AABB Trees



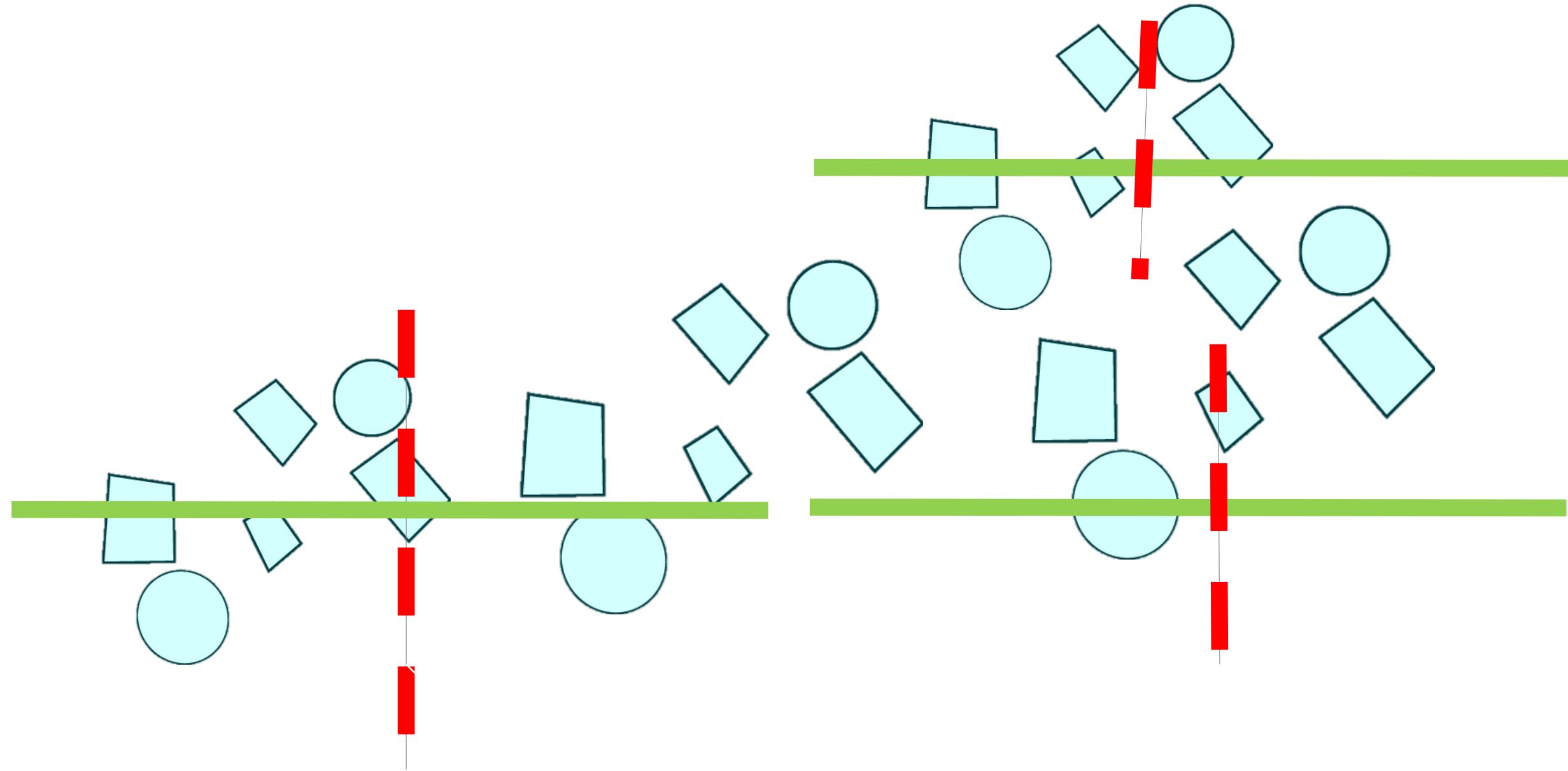
Constructing a BVH



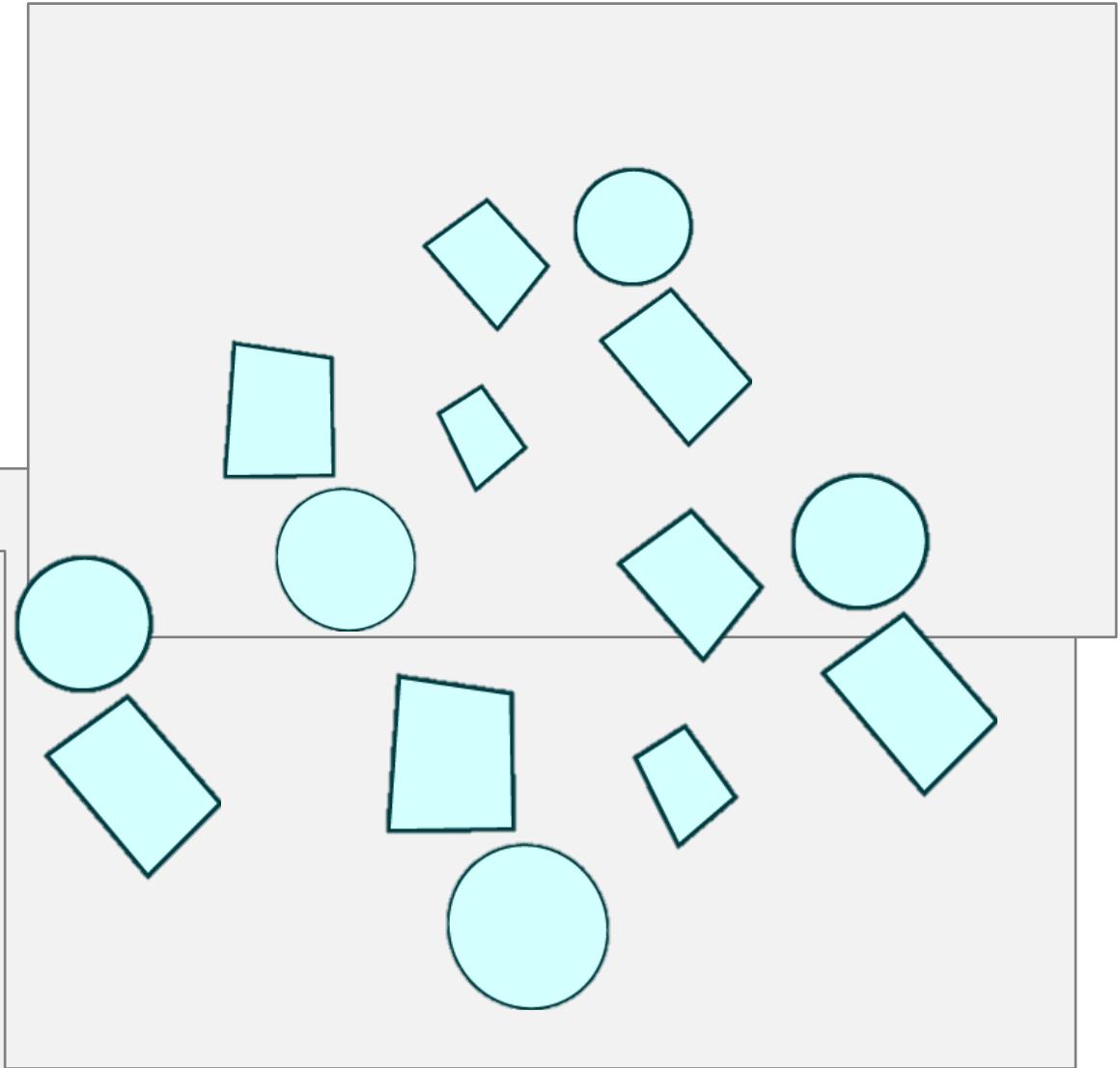
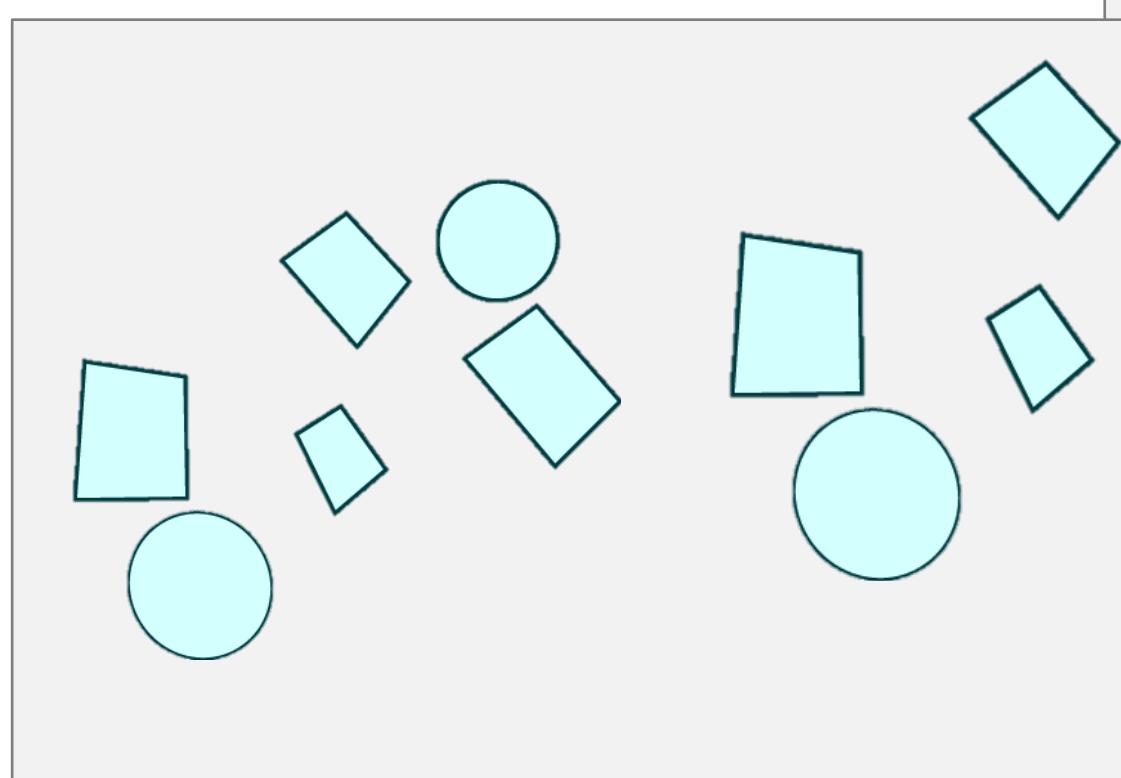
Constructing a BVH



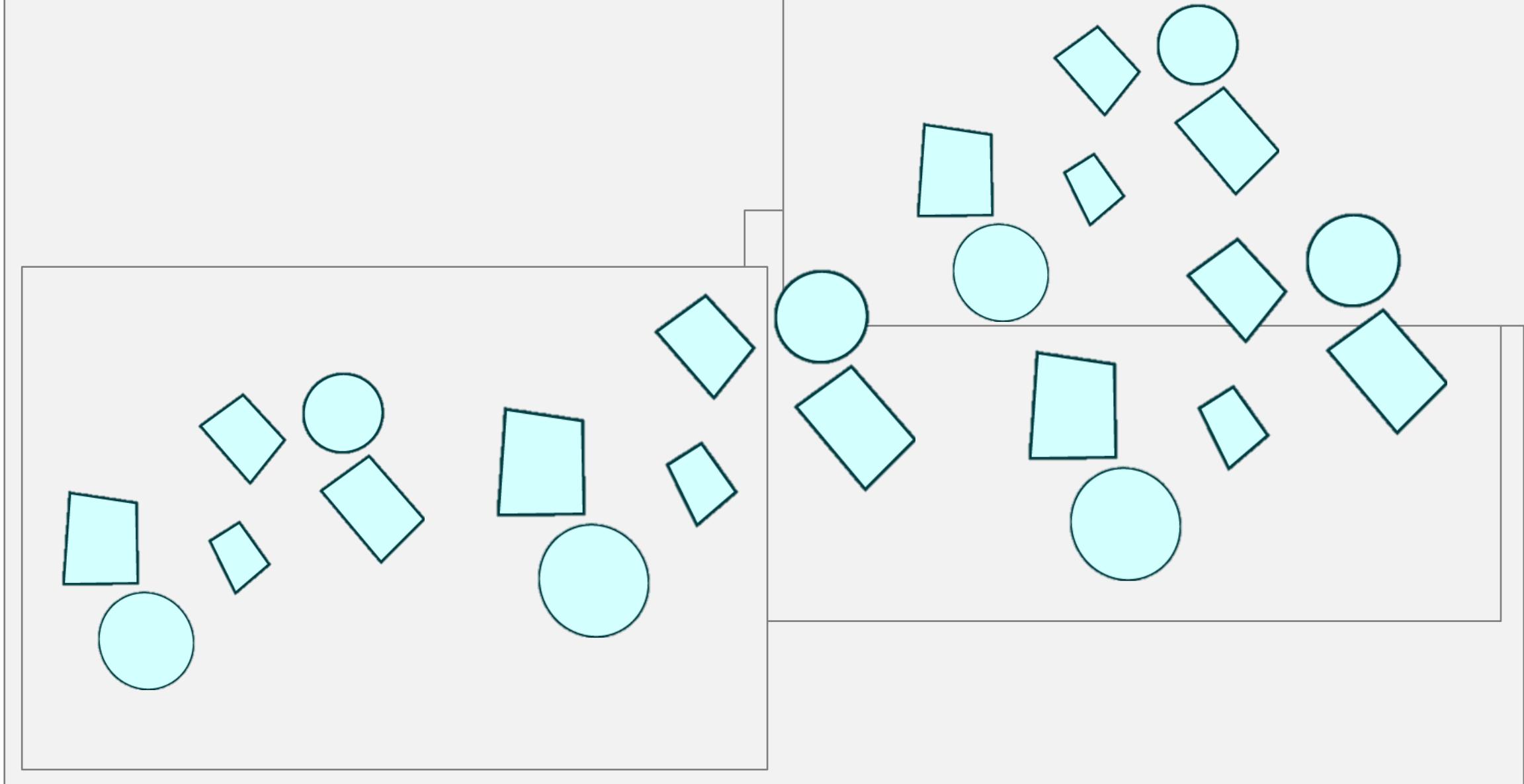
Constructing a BVH



Constructing a BVH



Constructing a BVH



Spatial Data Structures

Basic Idea – asymptotic improvement in spatial queries by subdividing

Two types of subdivisions – ***object-based*** and ***spatial***

Our object-based data structures will be boundary volume hierarchies or BVHs.

BVHs are hierarchies of BVs represented by trees

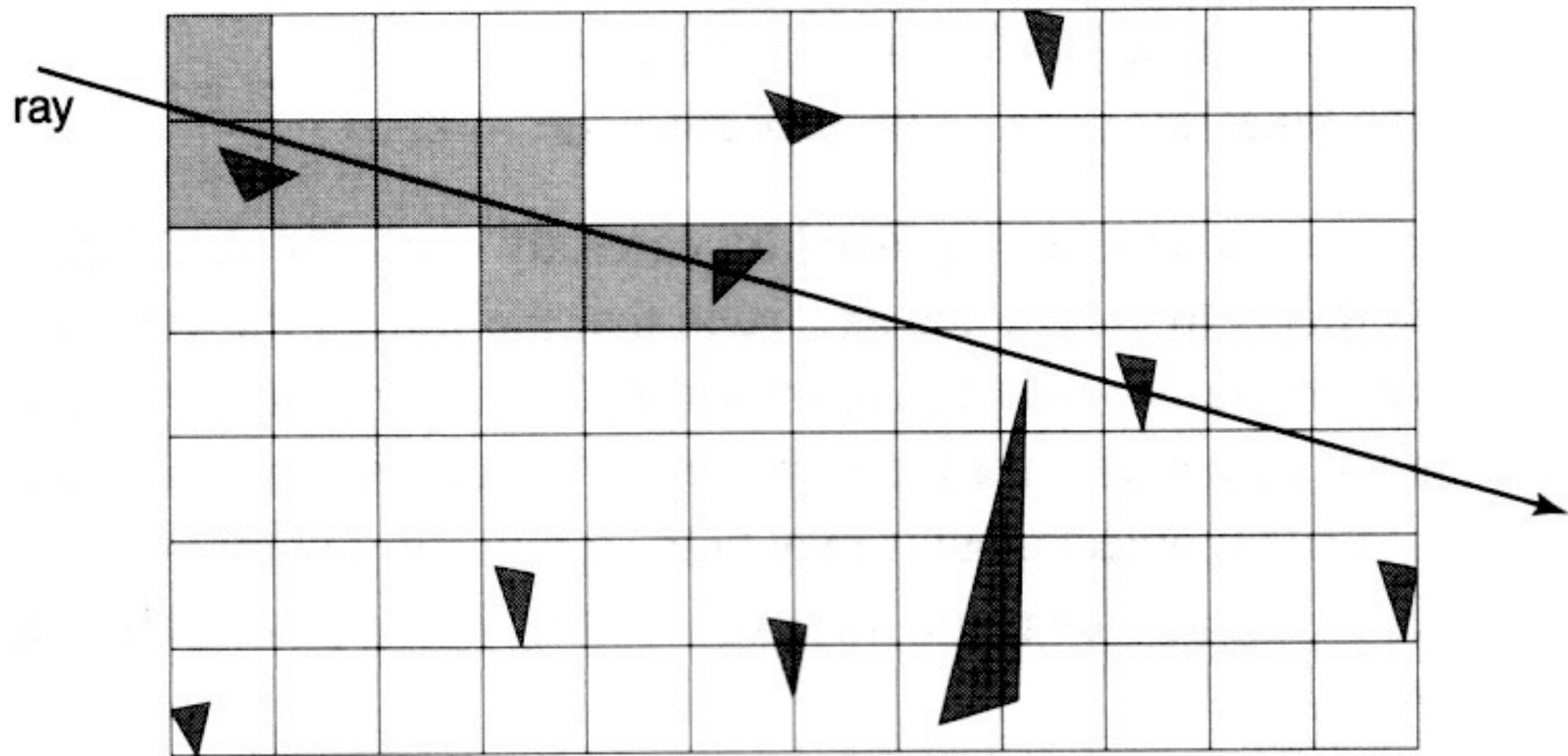
Spatial Data Structures

Basic Idea – asymptotic improvement in spatial queries by subdividing

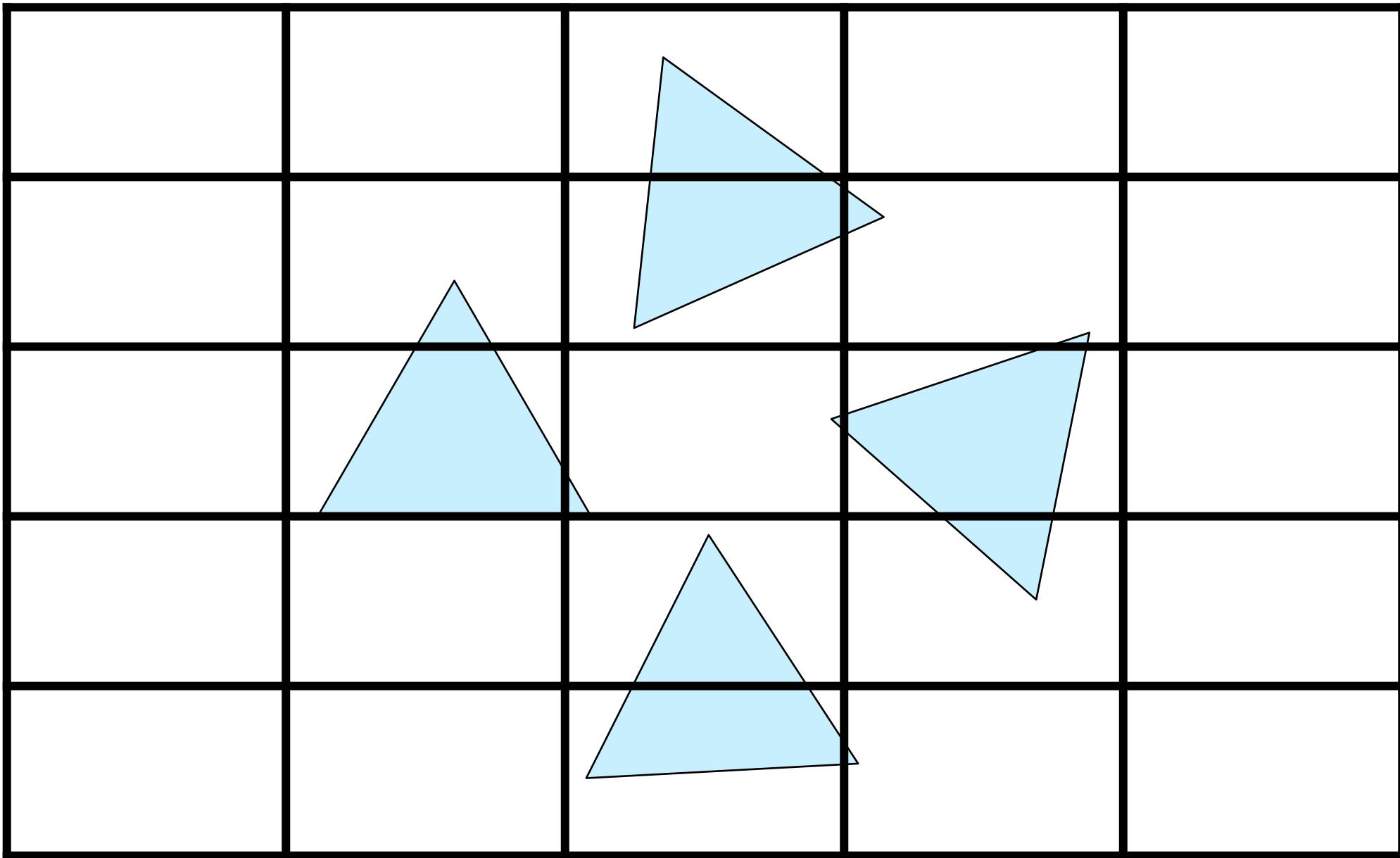
Two types of subdivisions – ***object-based*** and ***spatial***

Spatial subdivision divides space hierarchically and represents this as a tree.

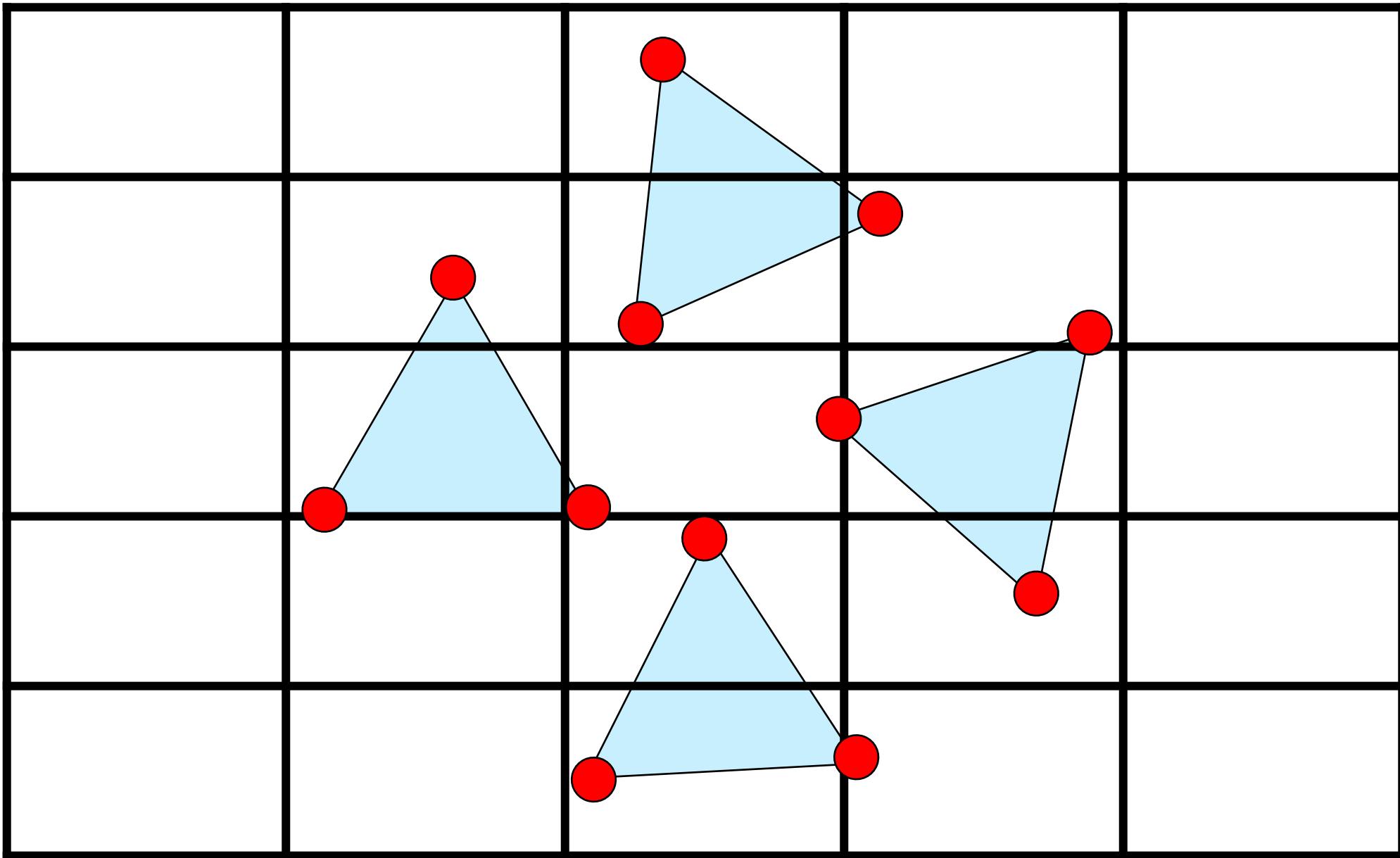
Axis-Aligned Spatial Subdivision (Uniform)



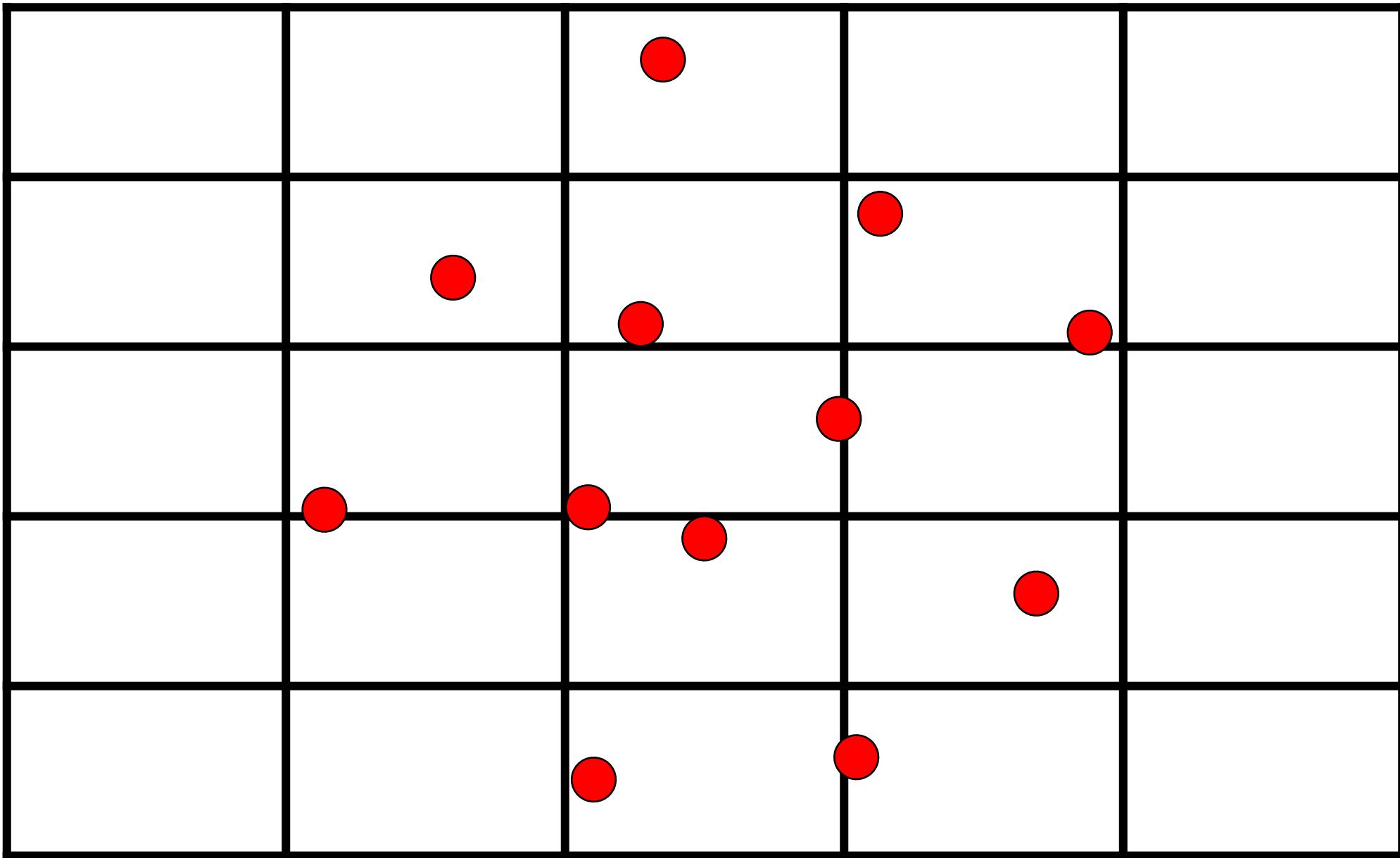
Construction



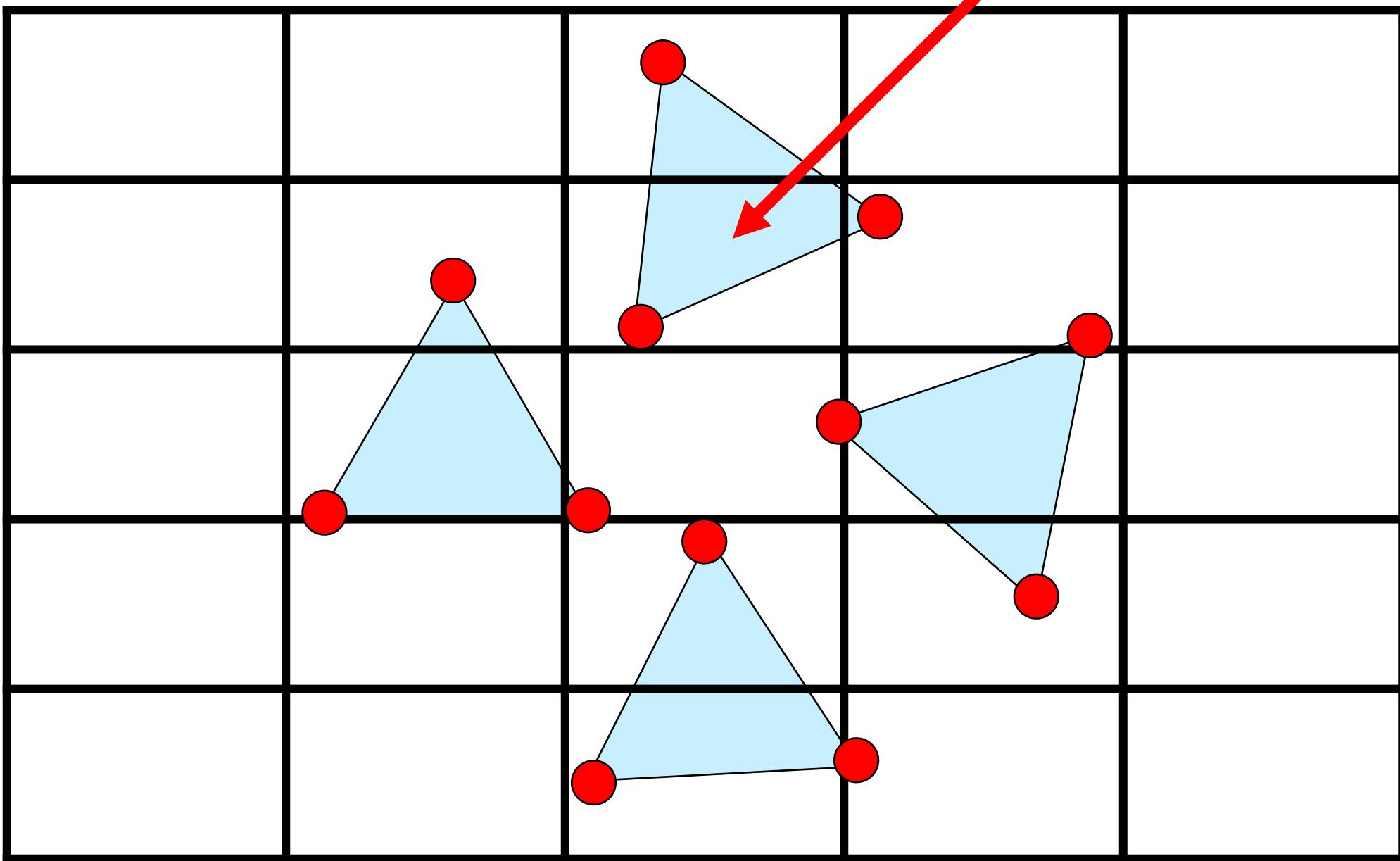
Construction



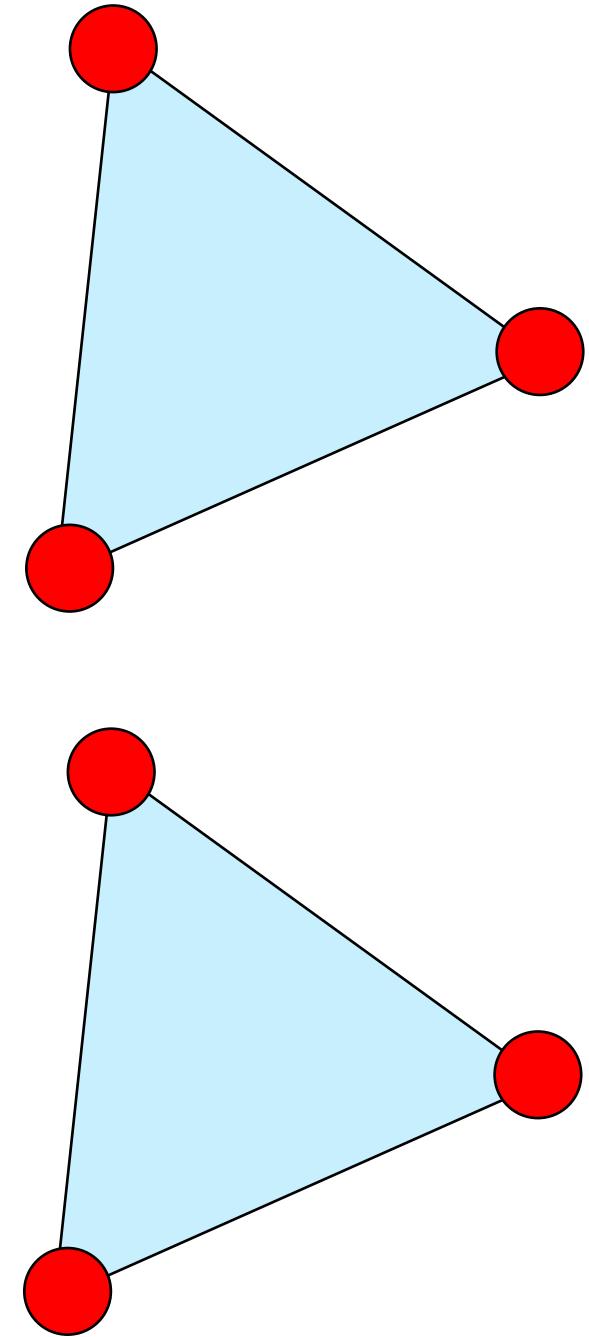
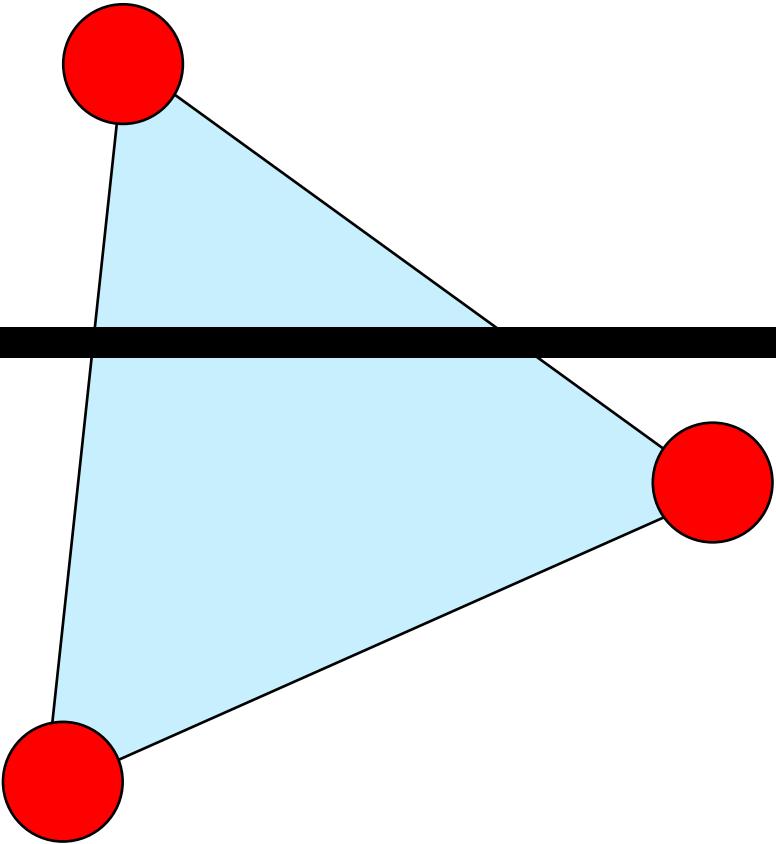
Construction



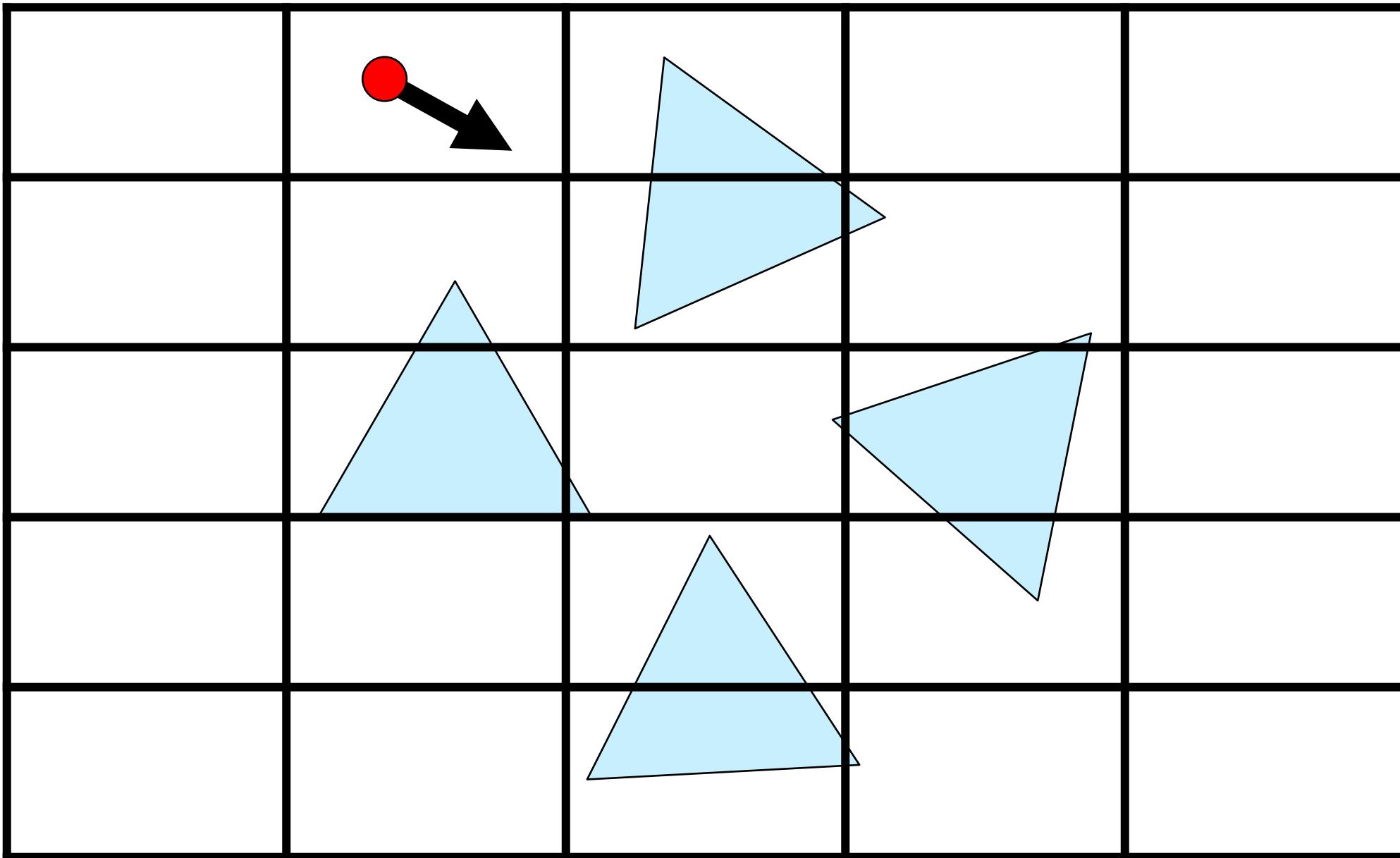
Construction



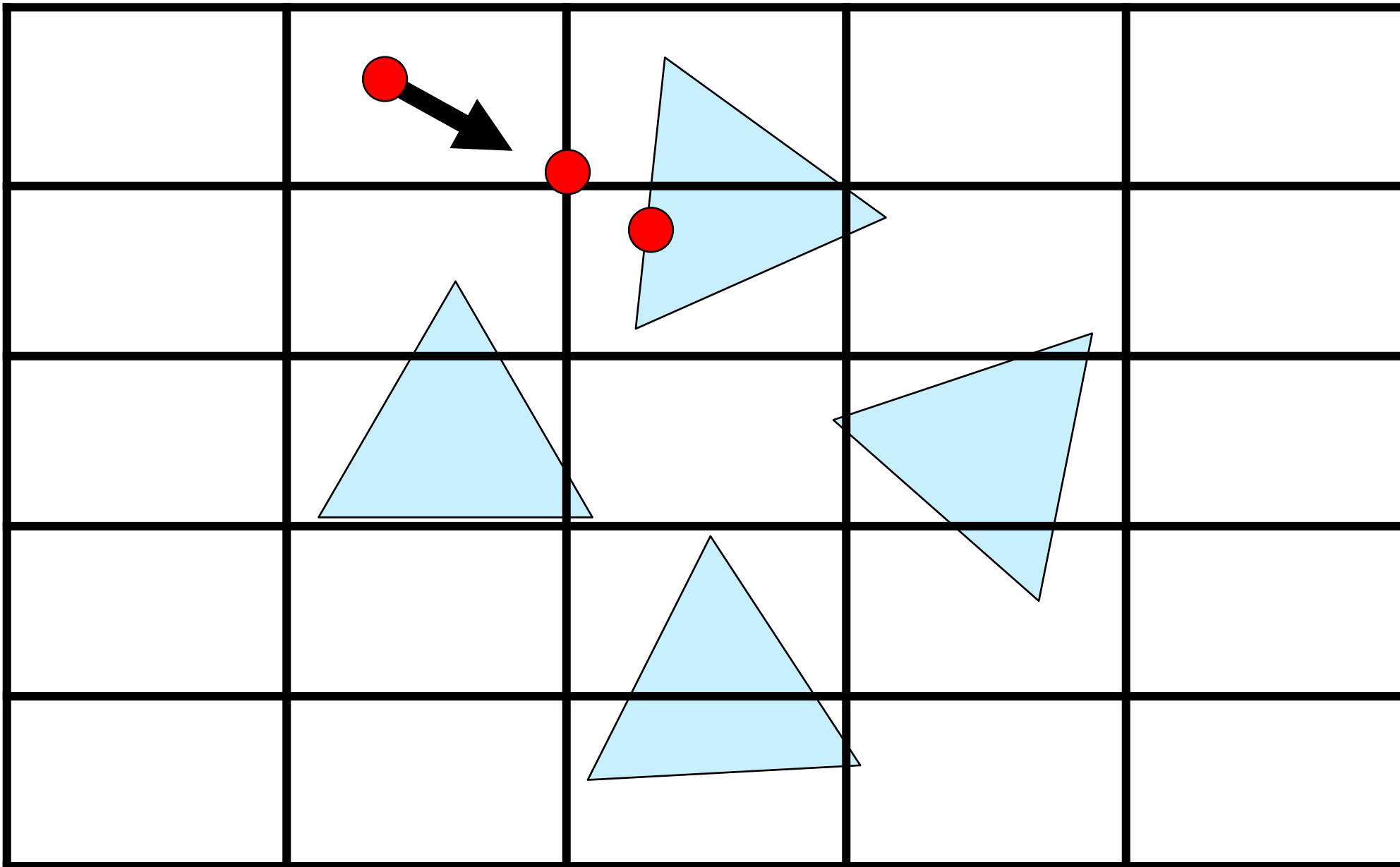
Duplicate Triangle



Intersection Tests

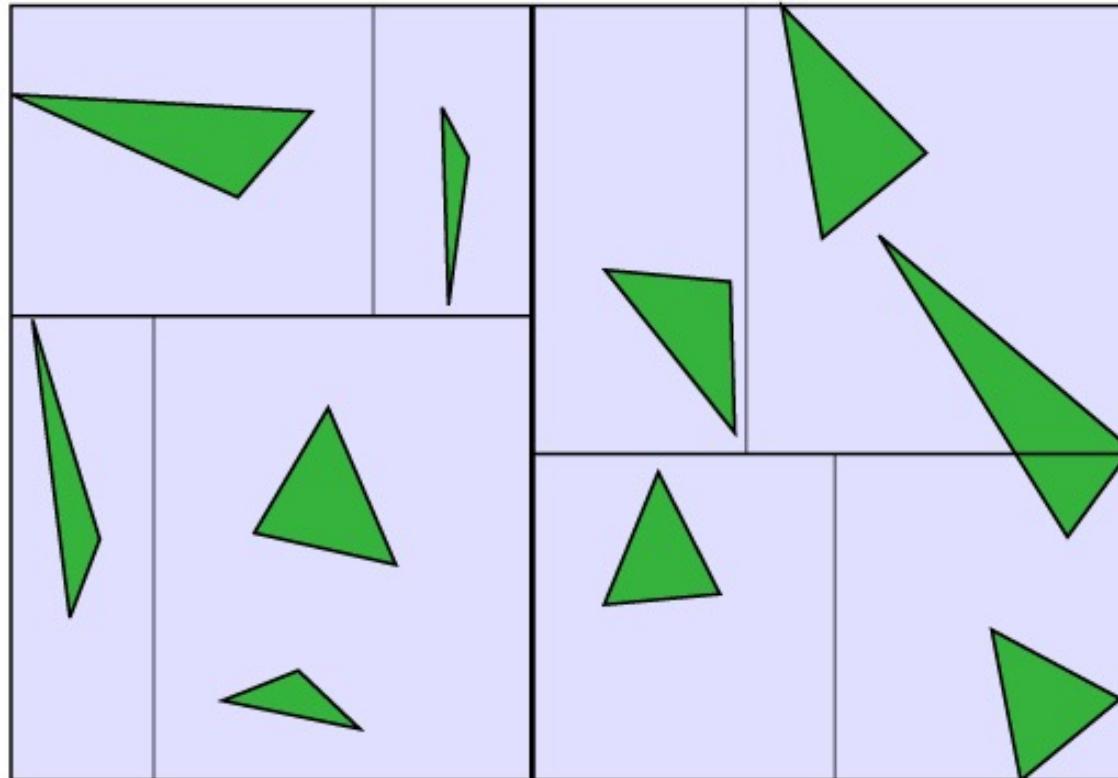


Intersection Tests

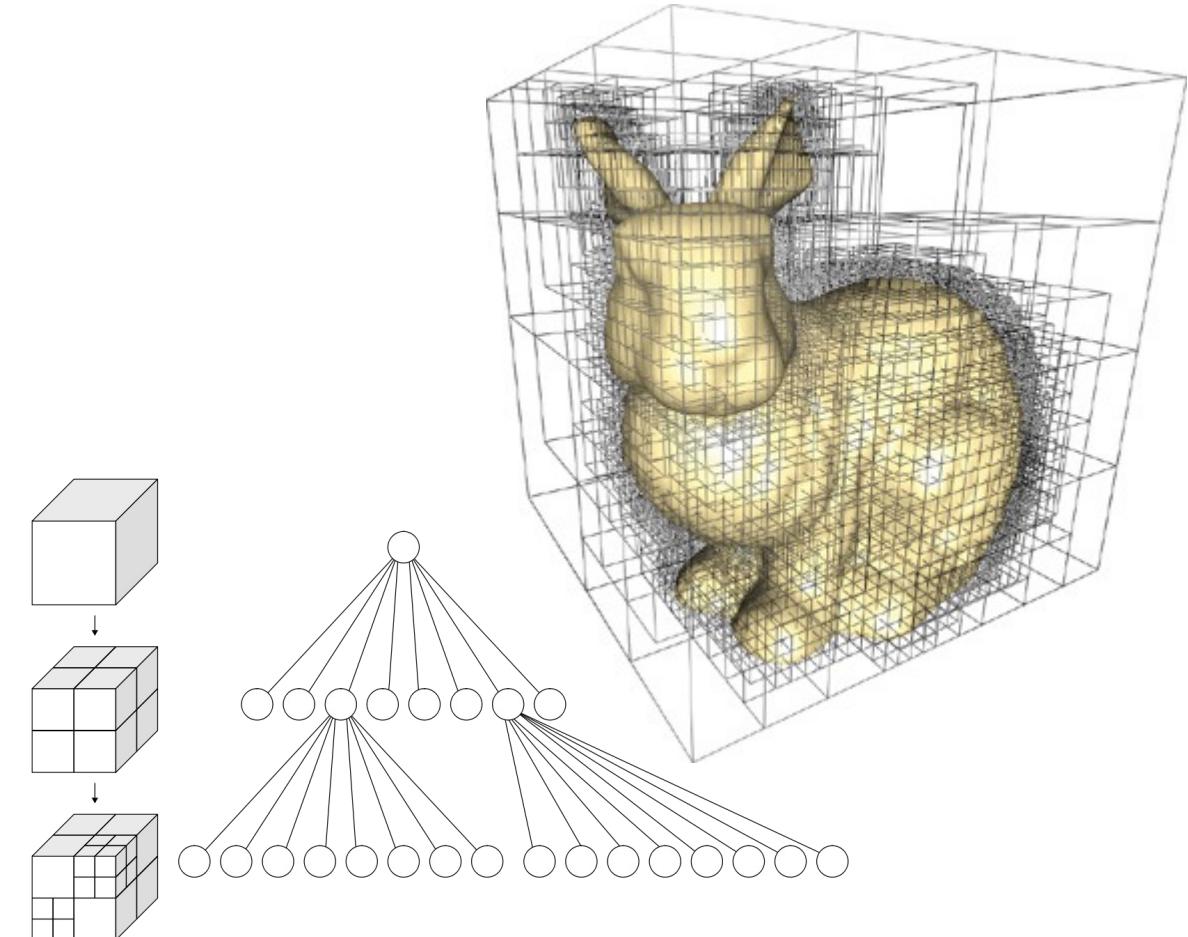


Axis-Aligned Spatial Subdivision (Non-Uniform)

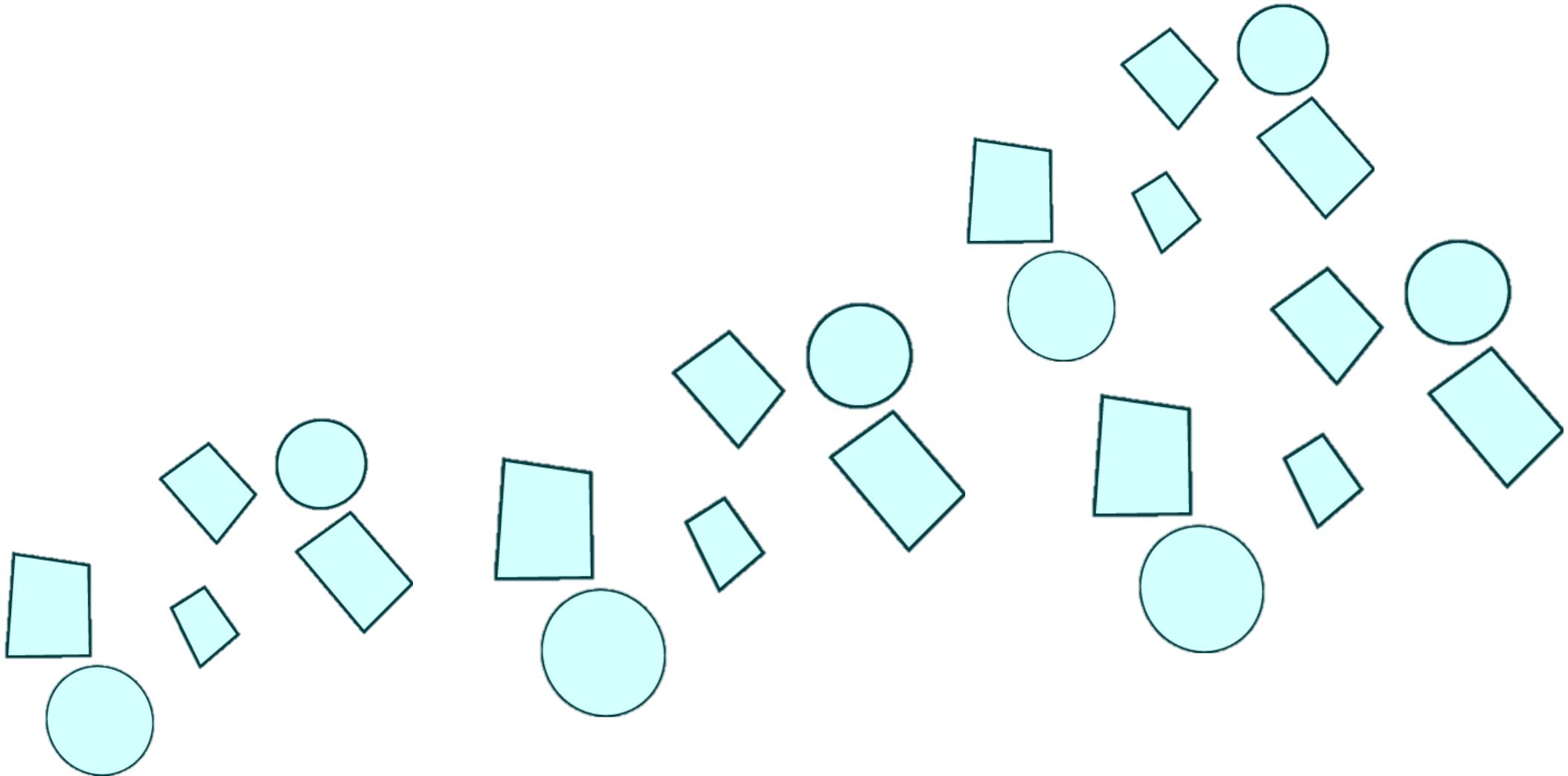
BSP Tree



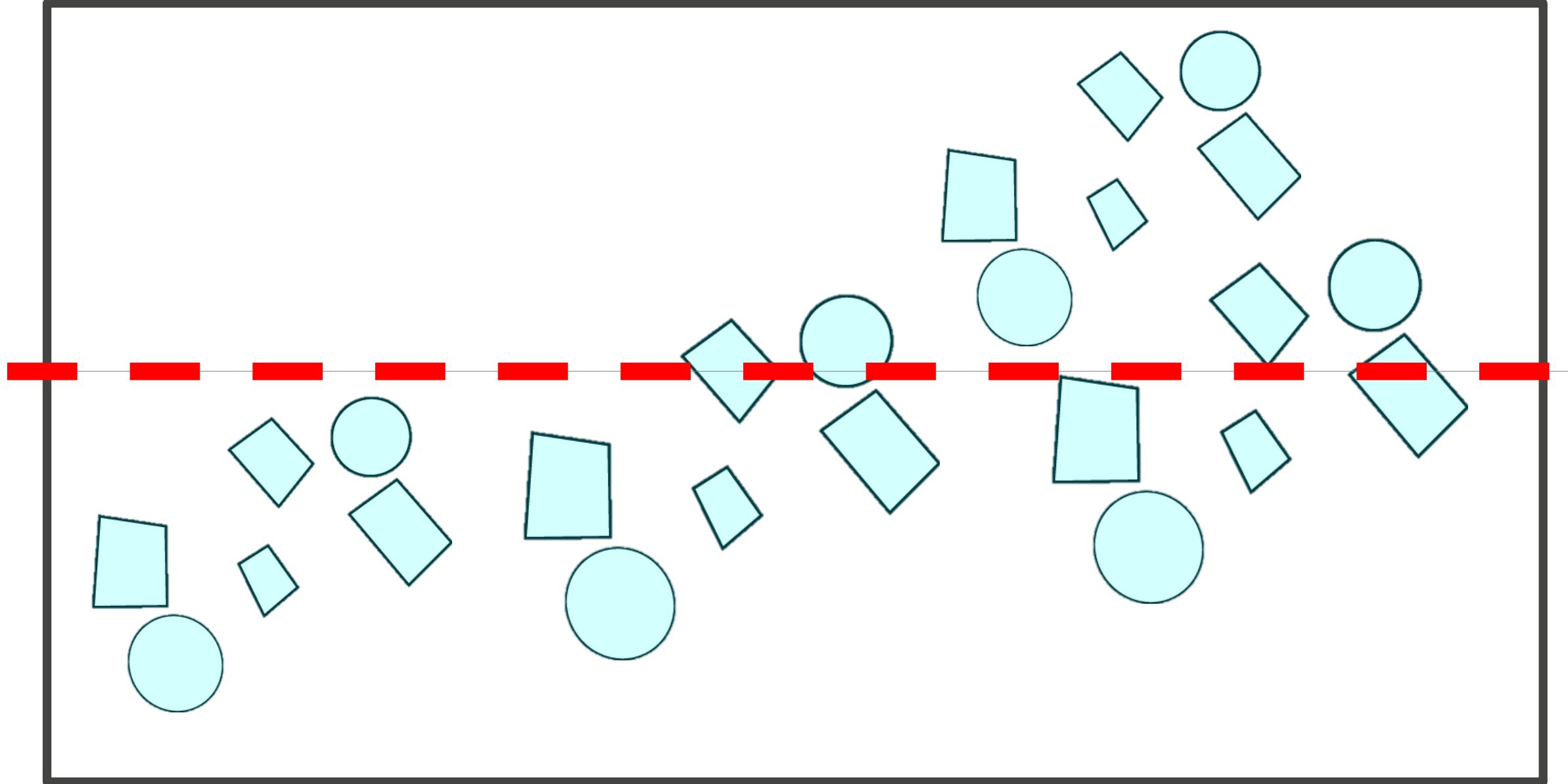
Octree



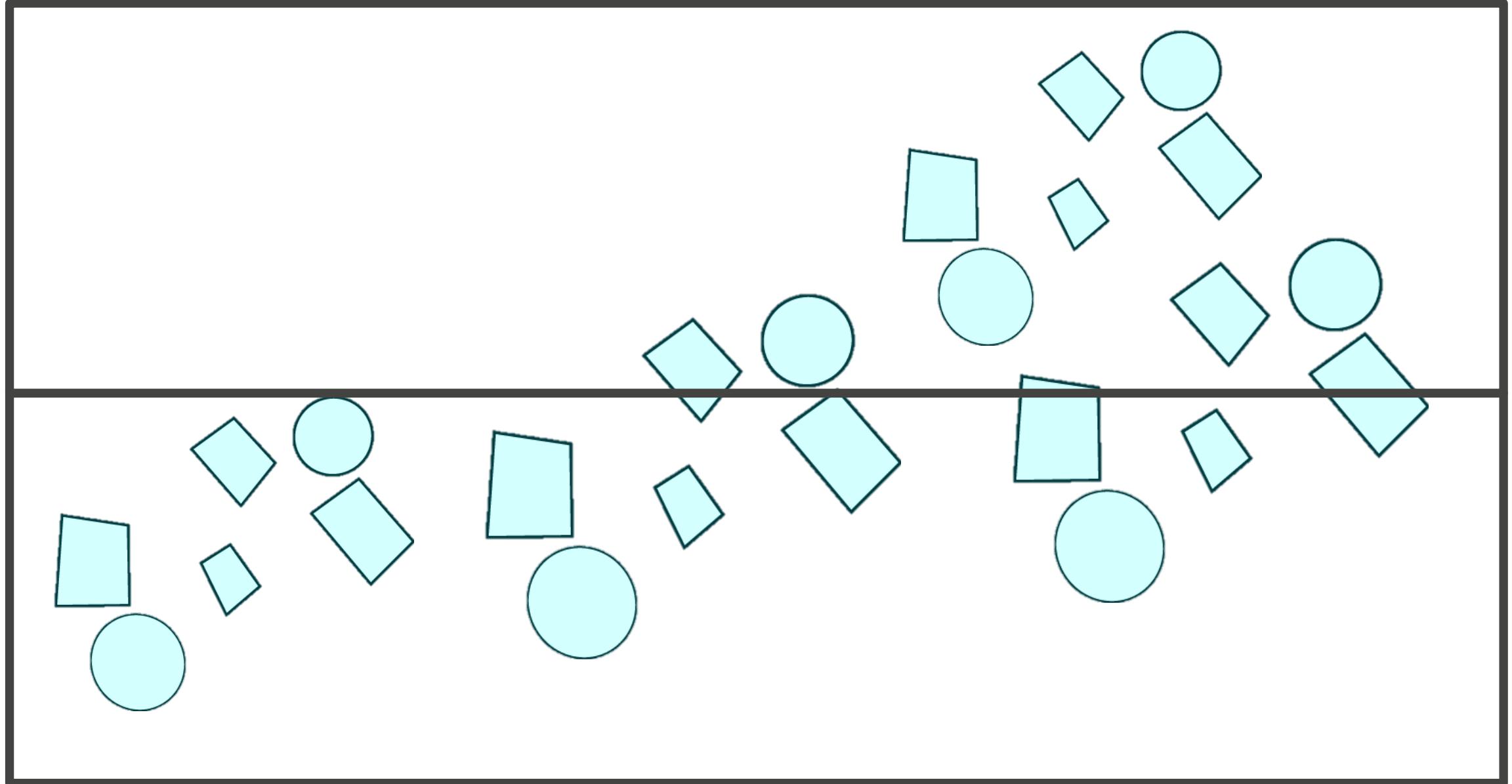
Constructing a k-d Tree



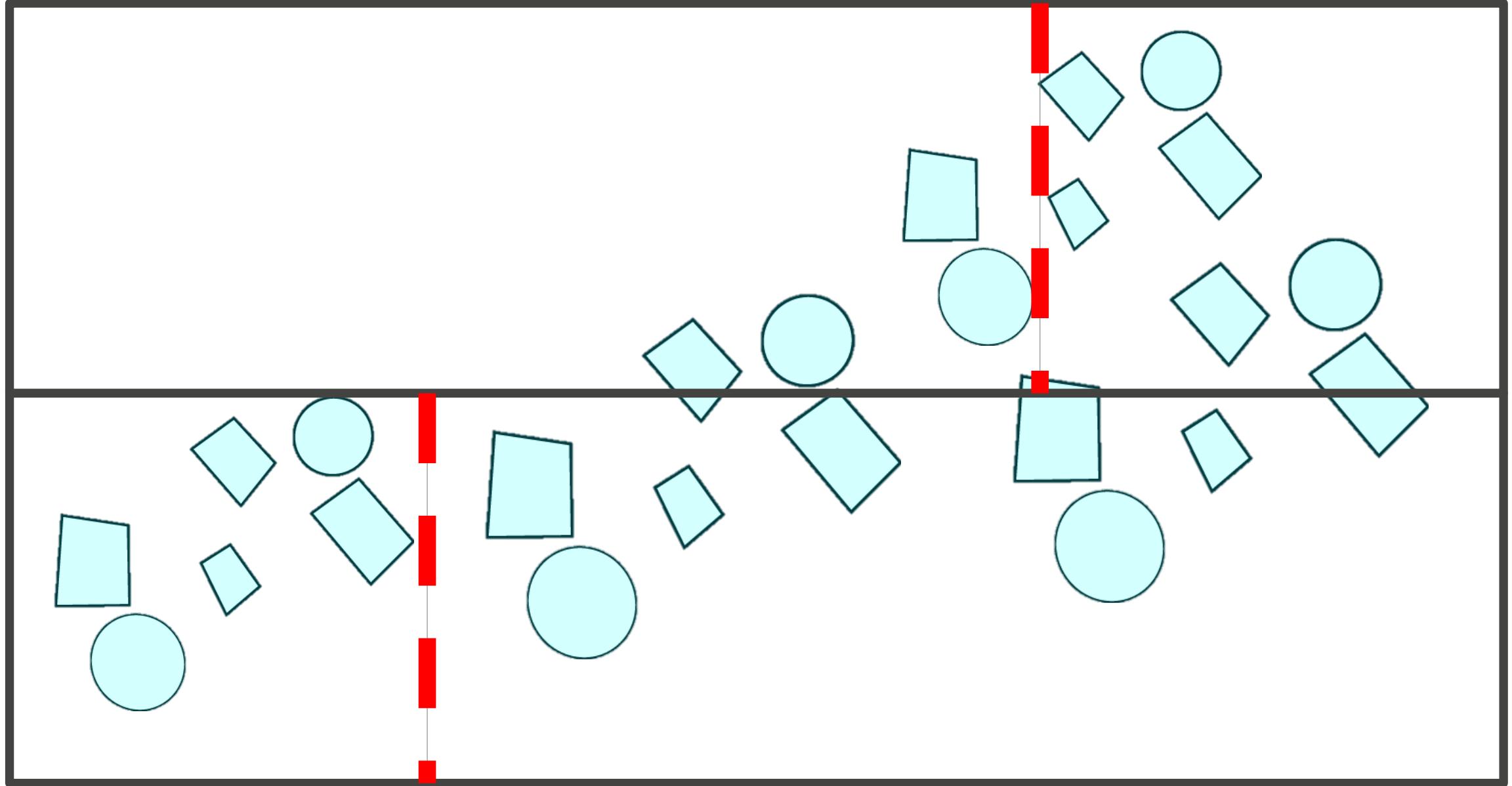
Constructing a k-d Tree



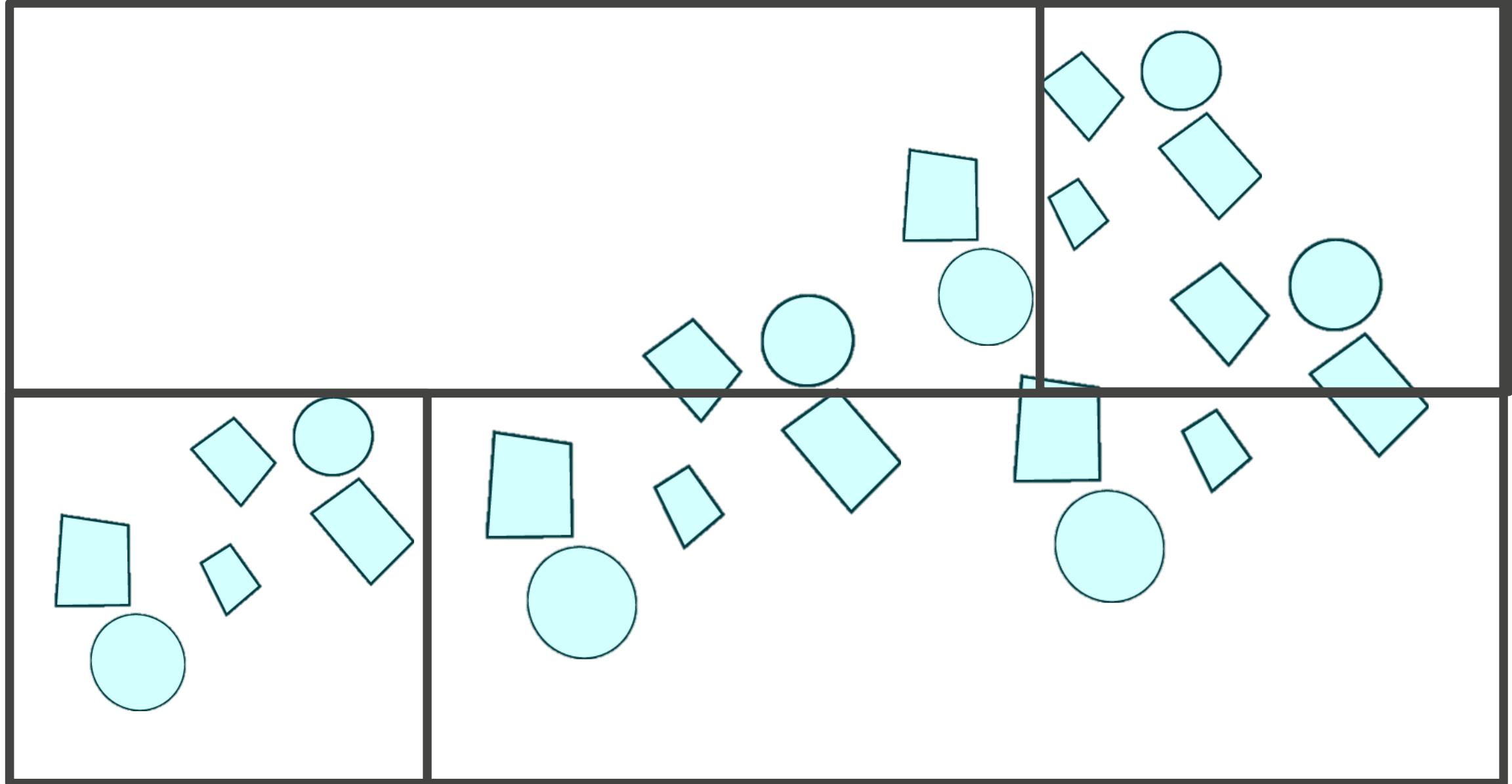
Constructing a k-d Tree



Constructing a k-d Tree



Constructing a k-d Tree



Ray Intersection Tests

Depth First Search Again

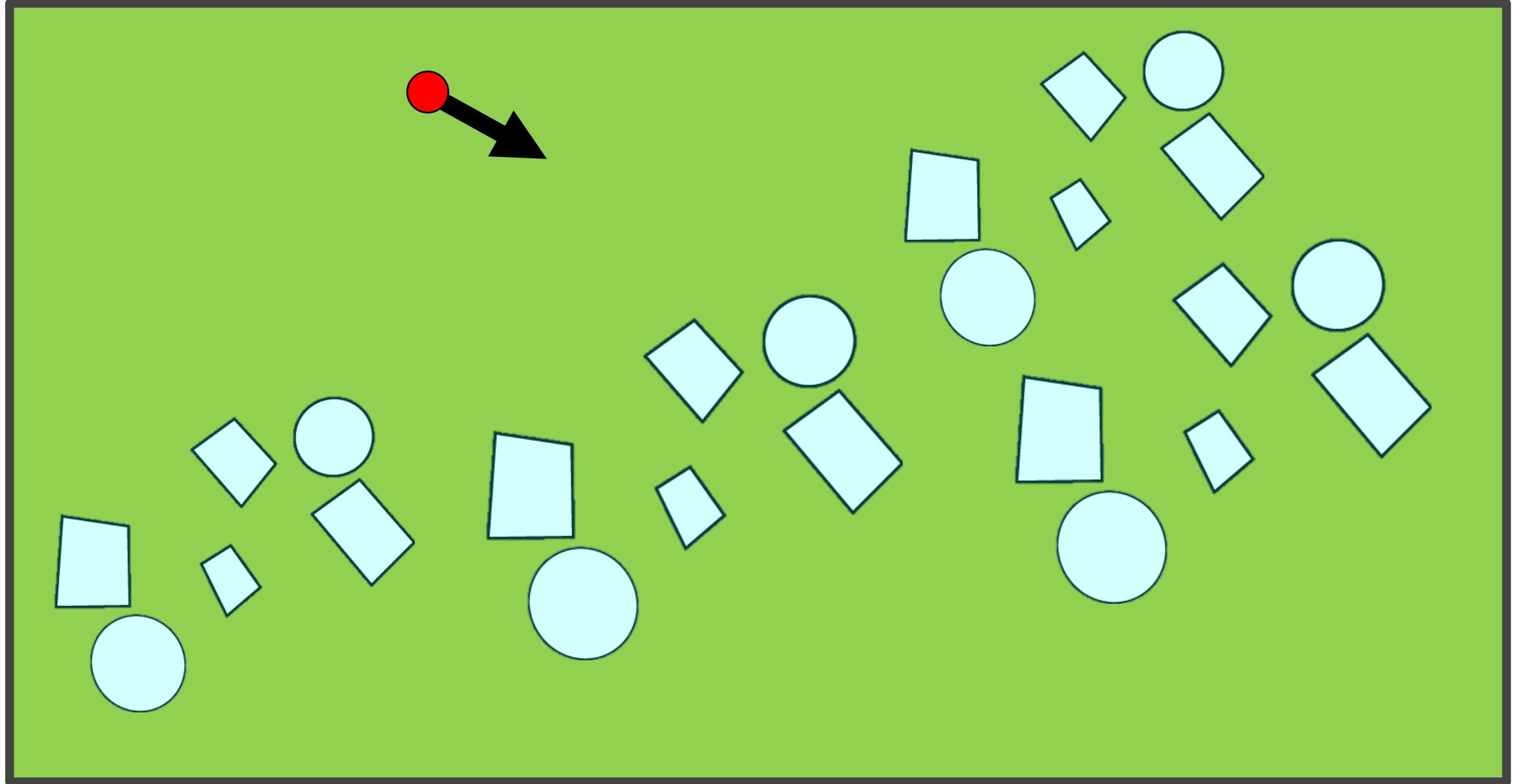
If ray interacts with child node then recurse

Interactions are

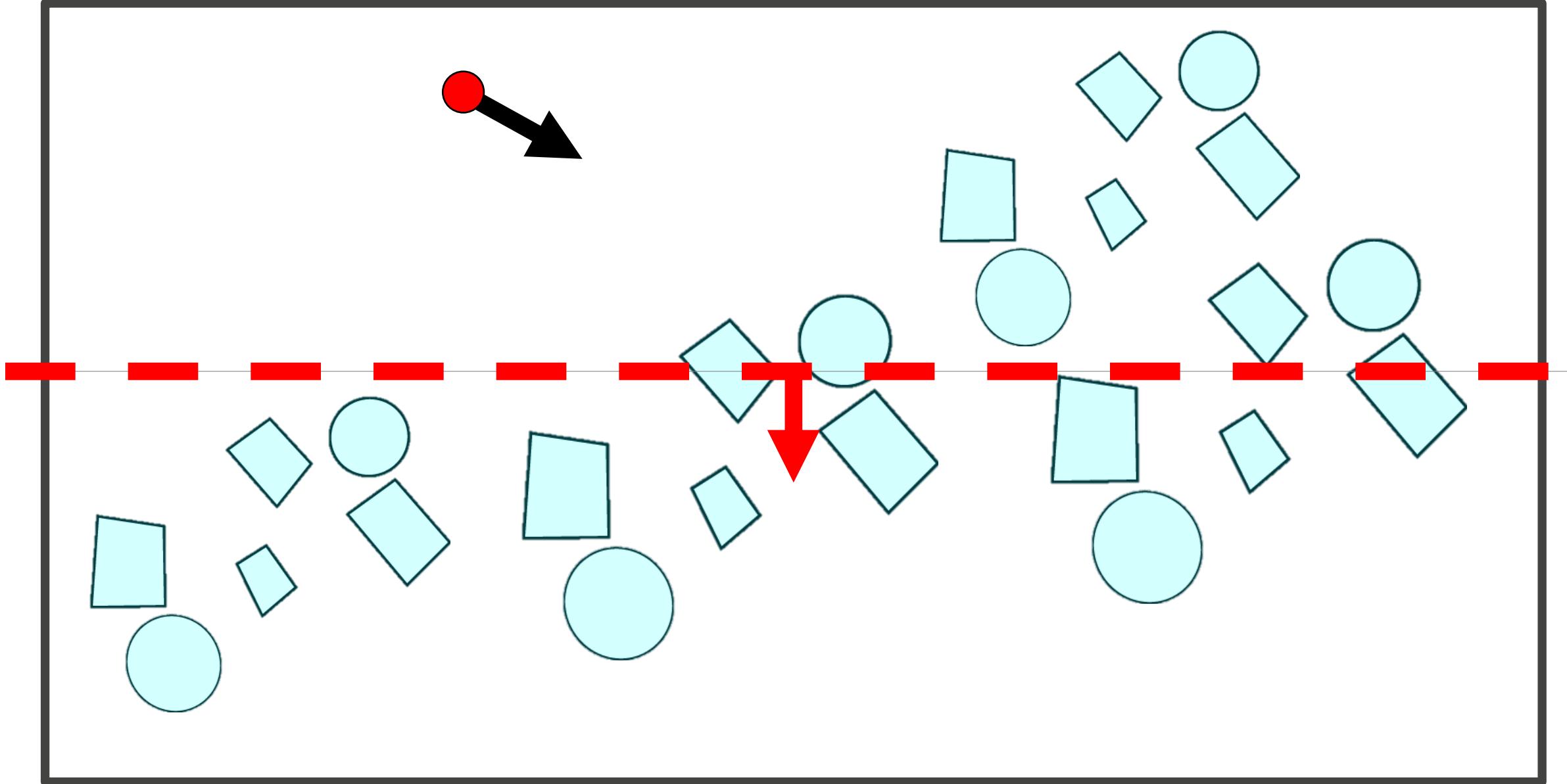
Child contains ray origin point

Ray crossed into child node

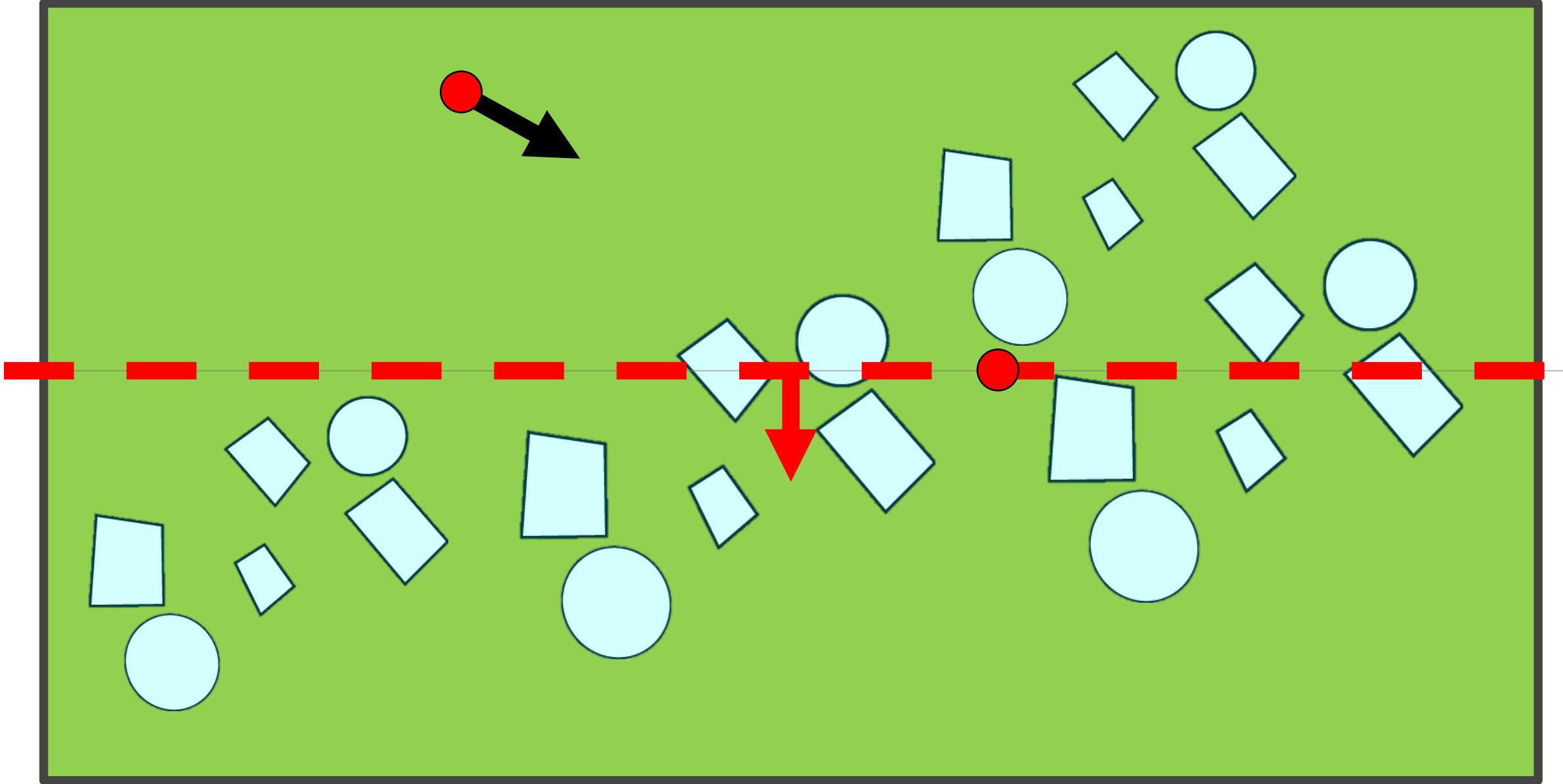
Ray Intersection Tests



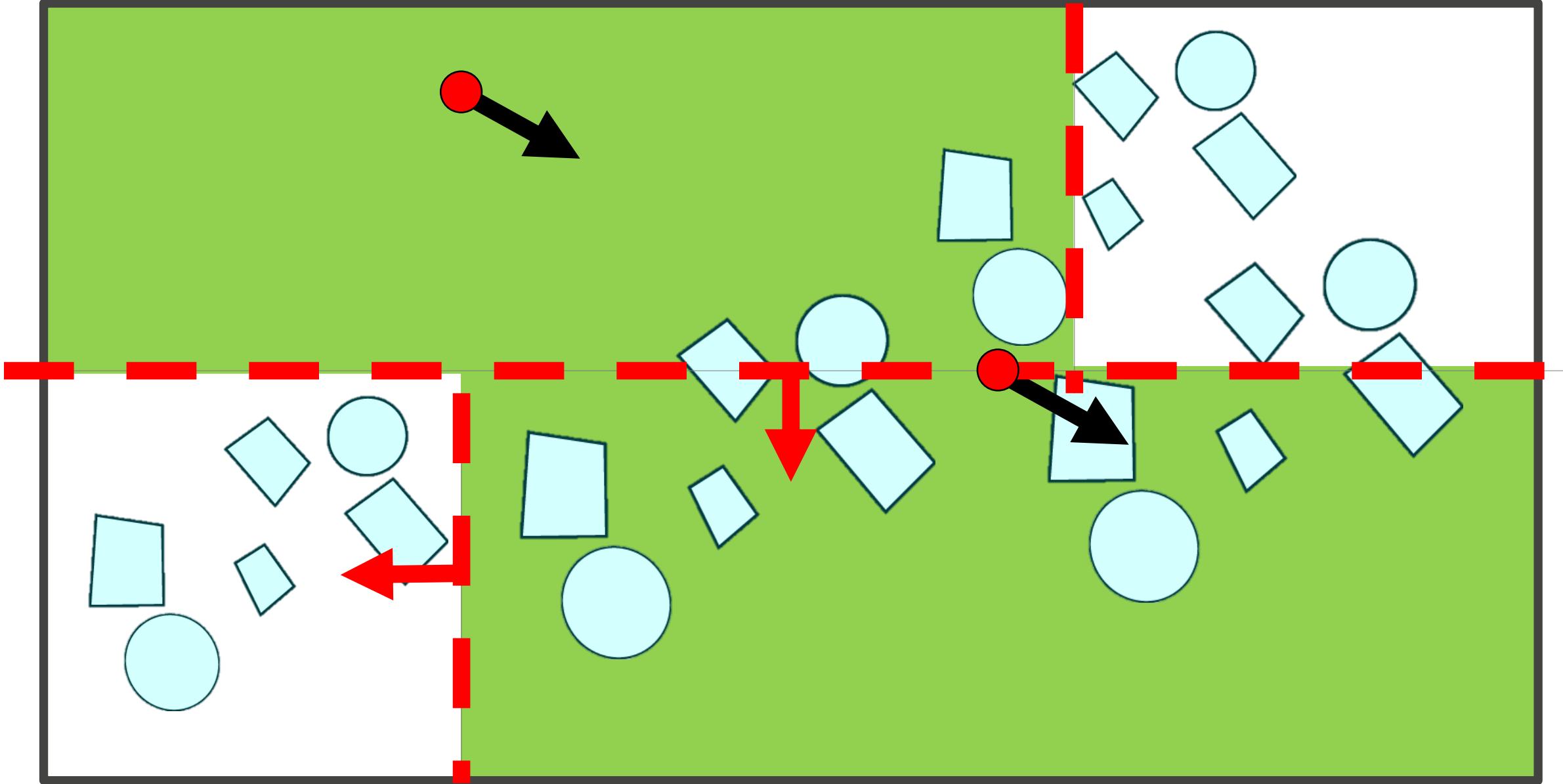
Ray Intersection Tests



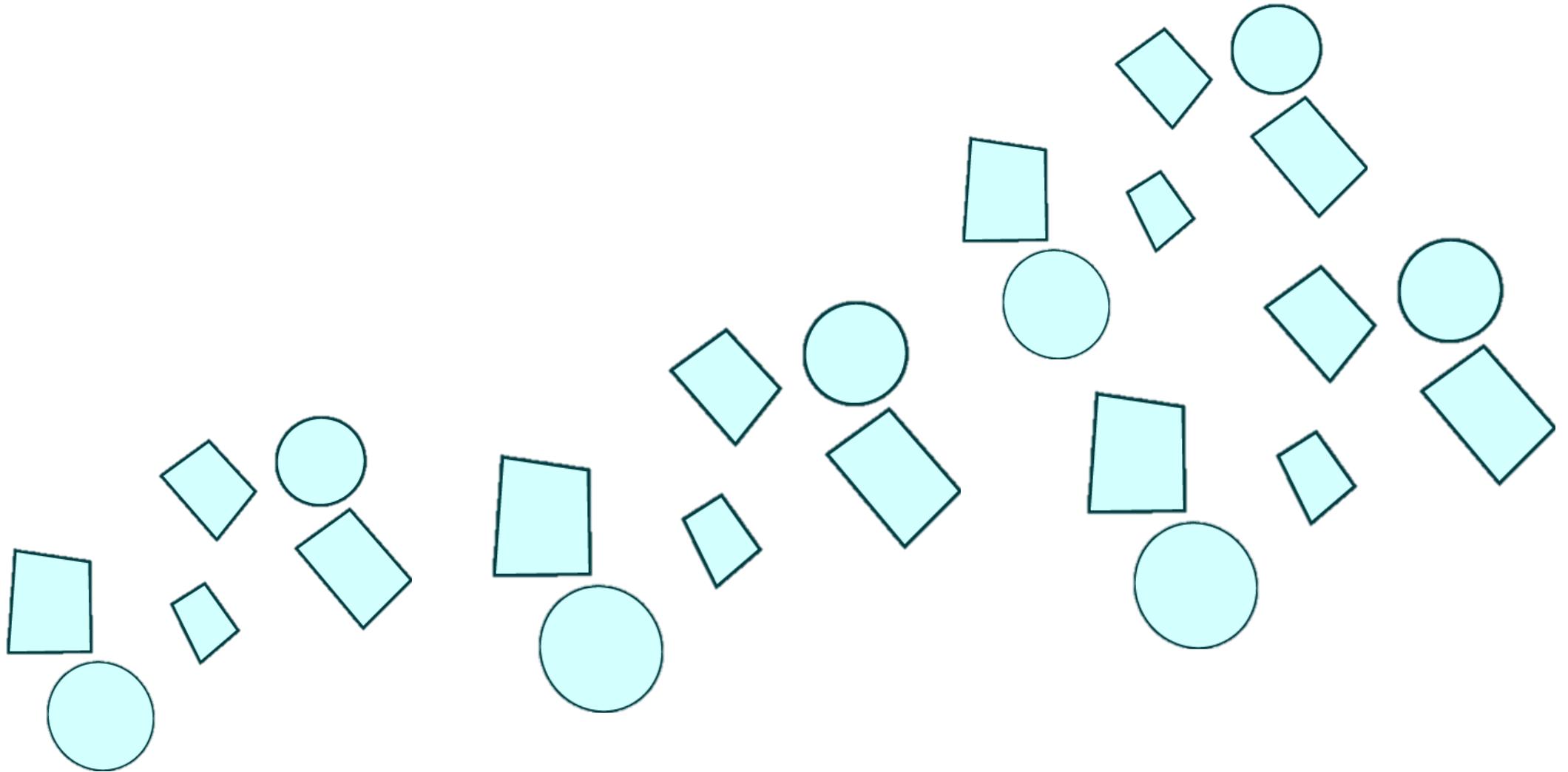
Ray Intersection Tests



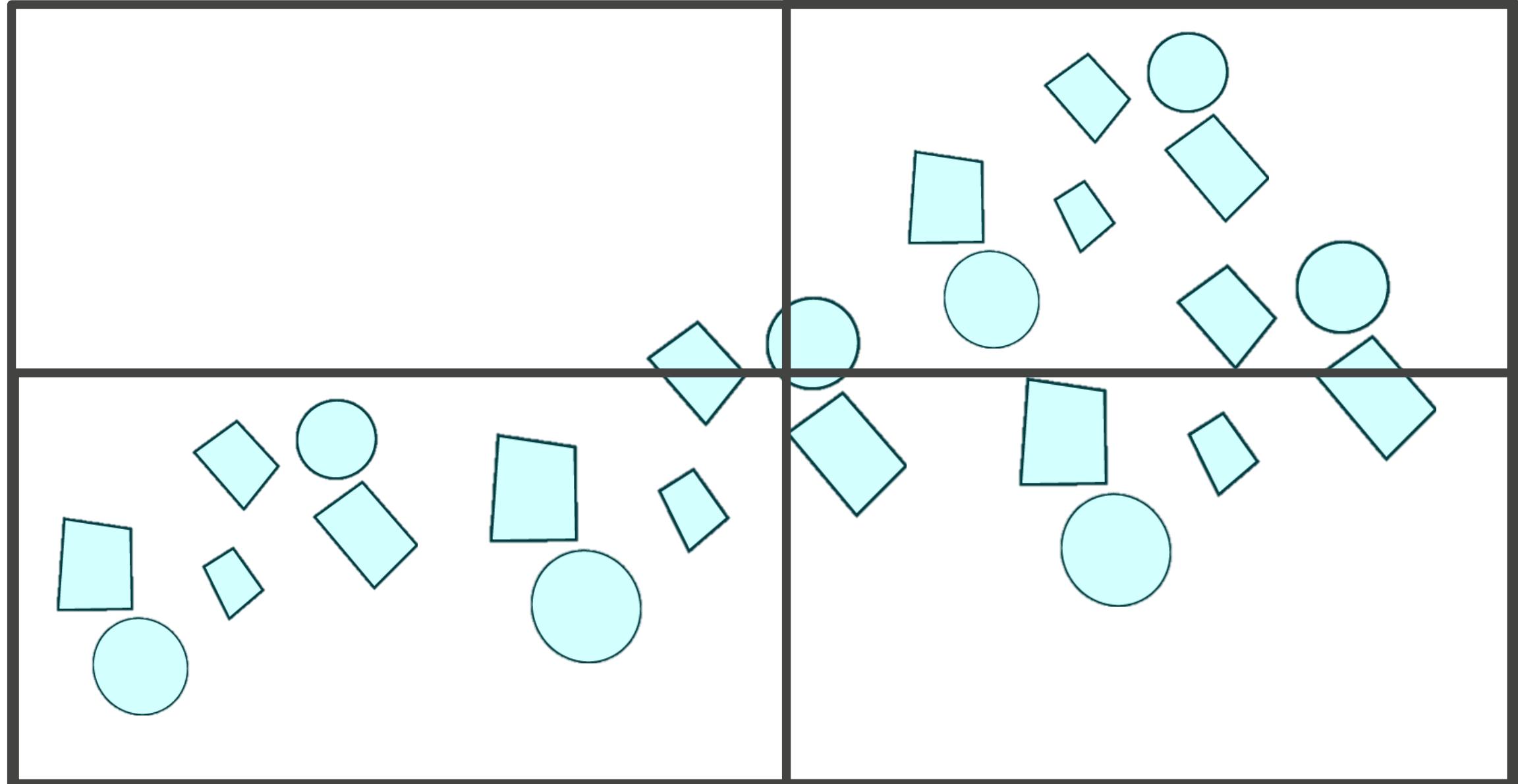
Ray Intersection Tests



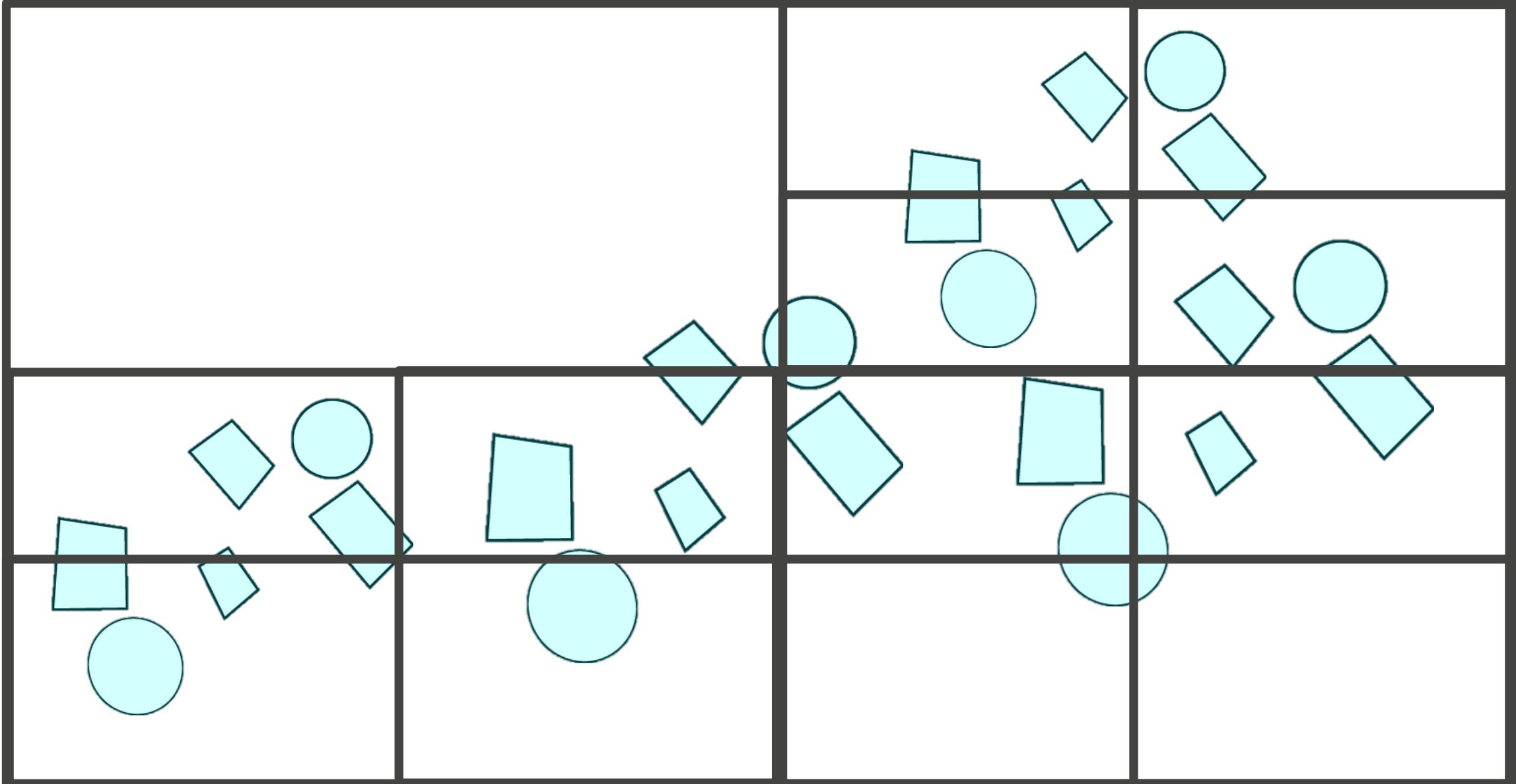
Constructing an Quadtree Tree



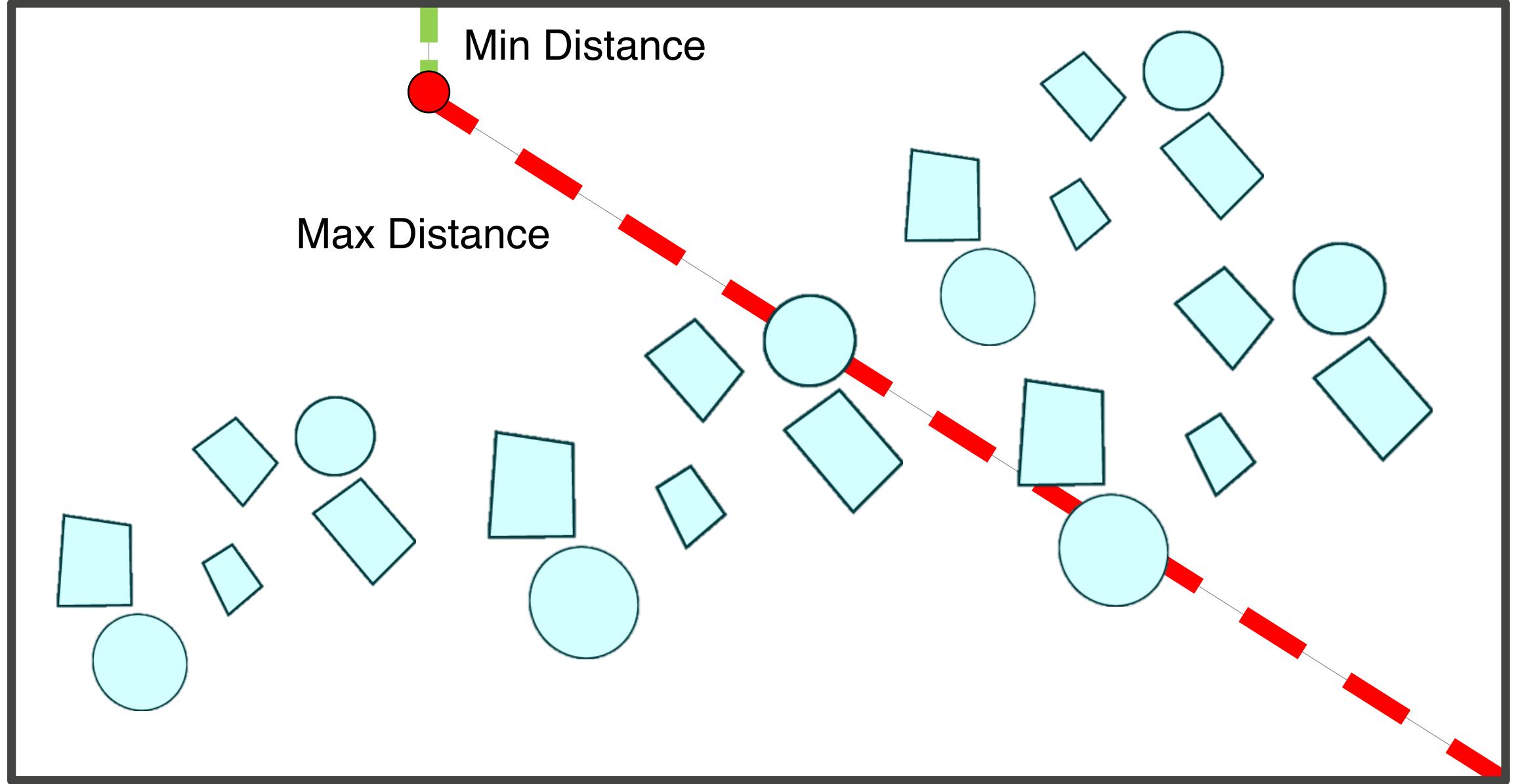
Constructing an Quadtree Tree



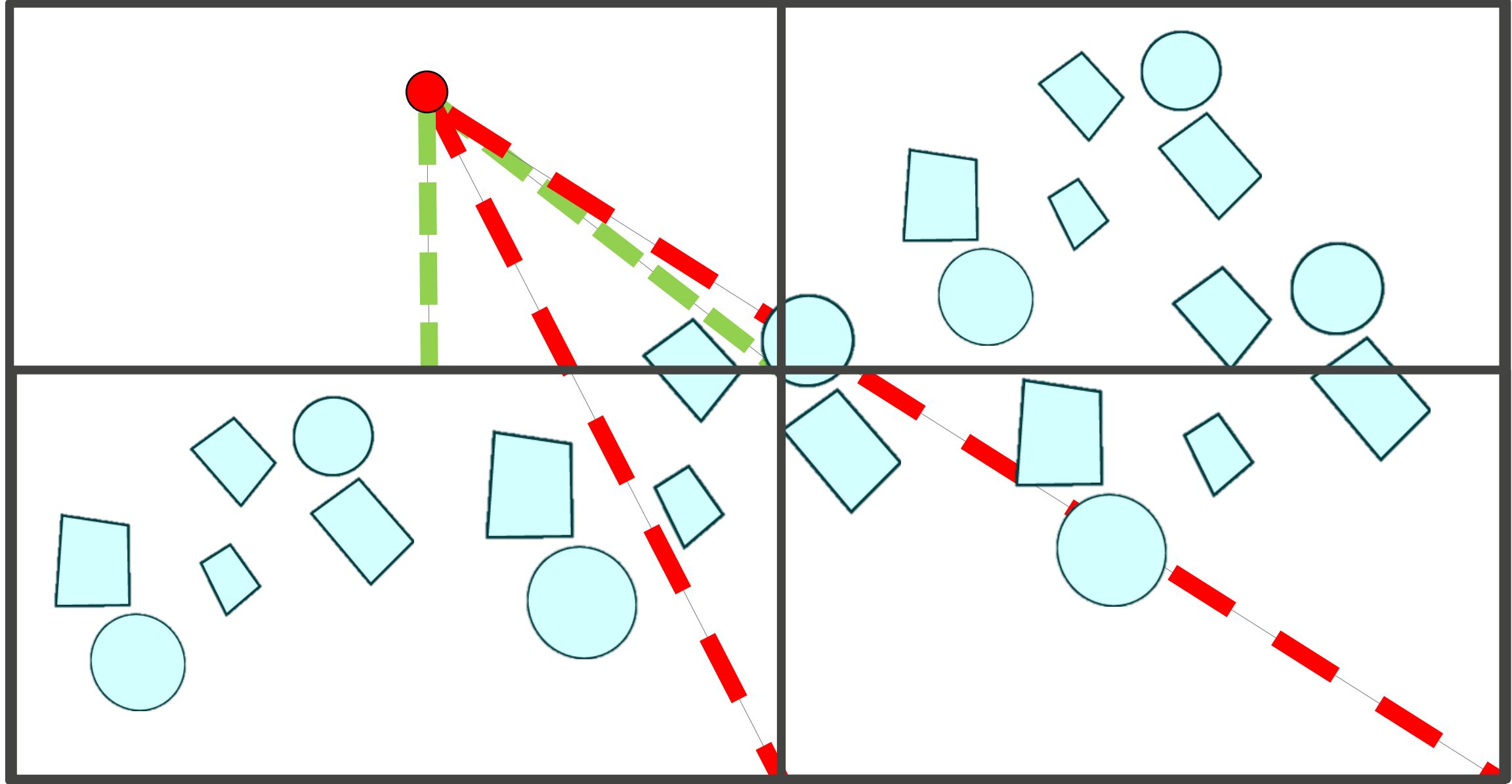
Constructing an Quadtree Tree



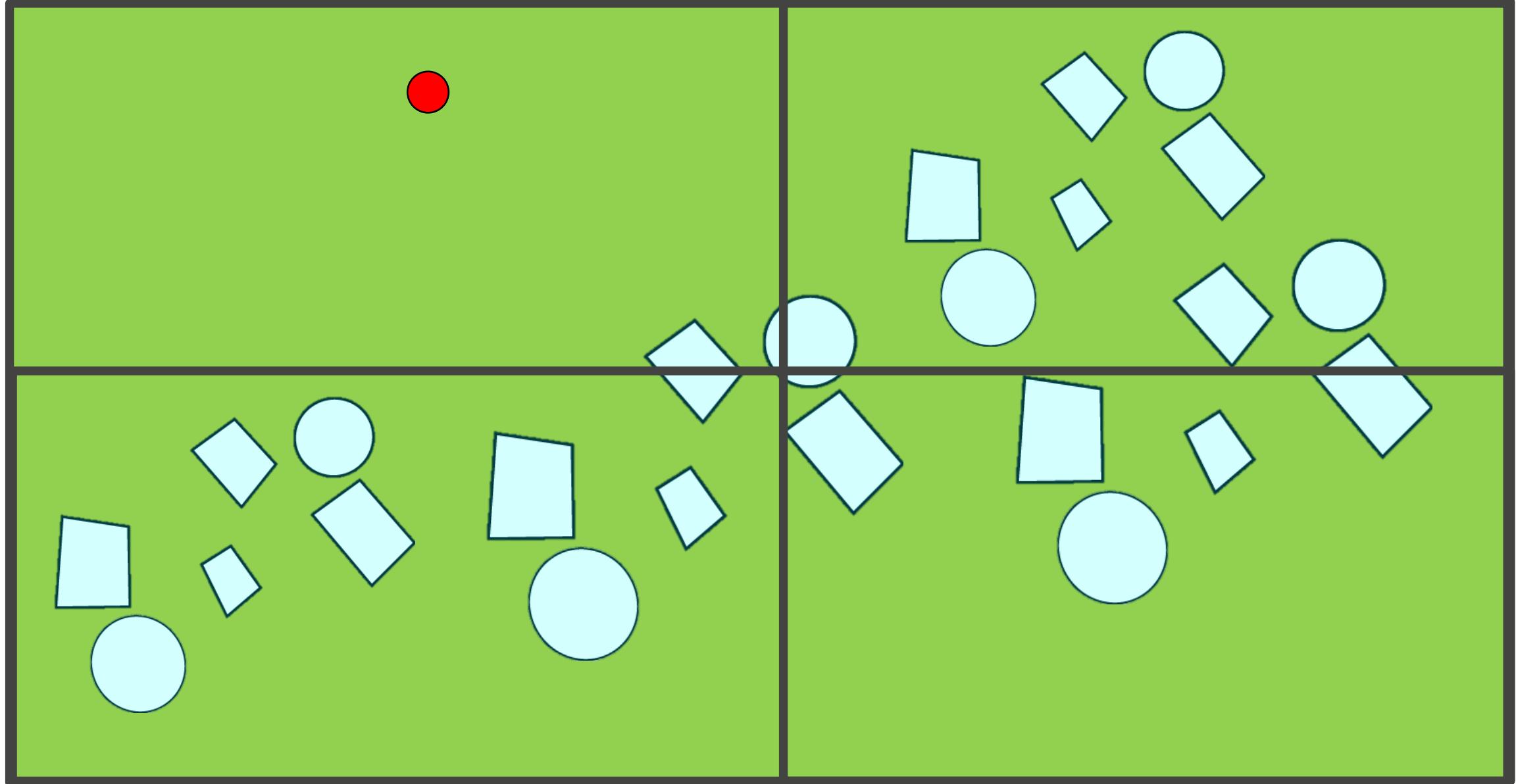
Distance Query



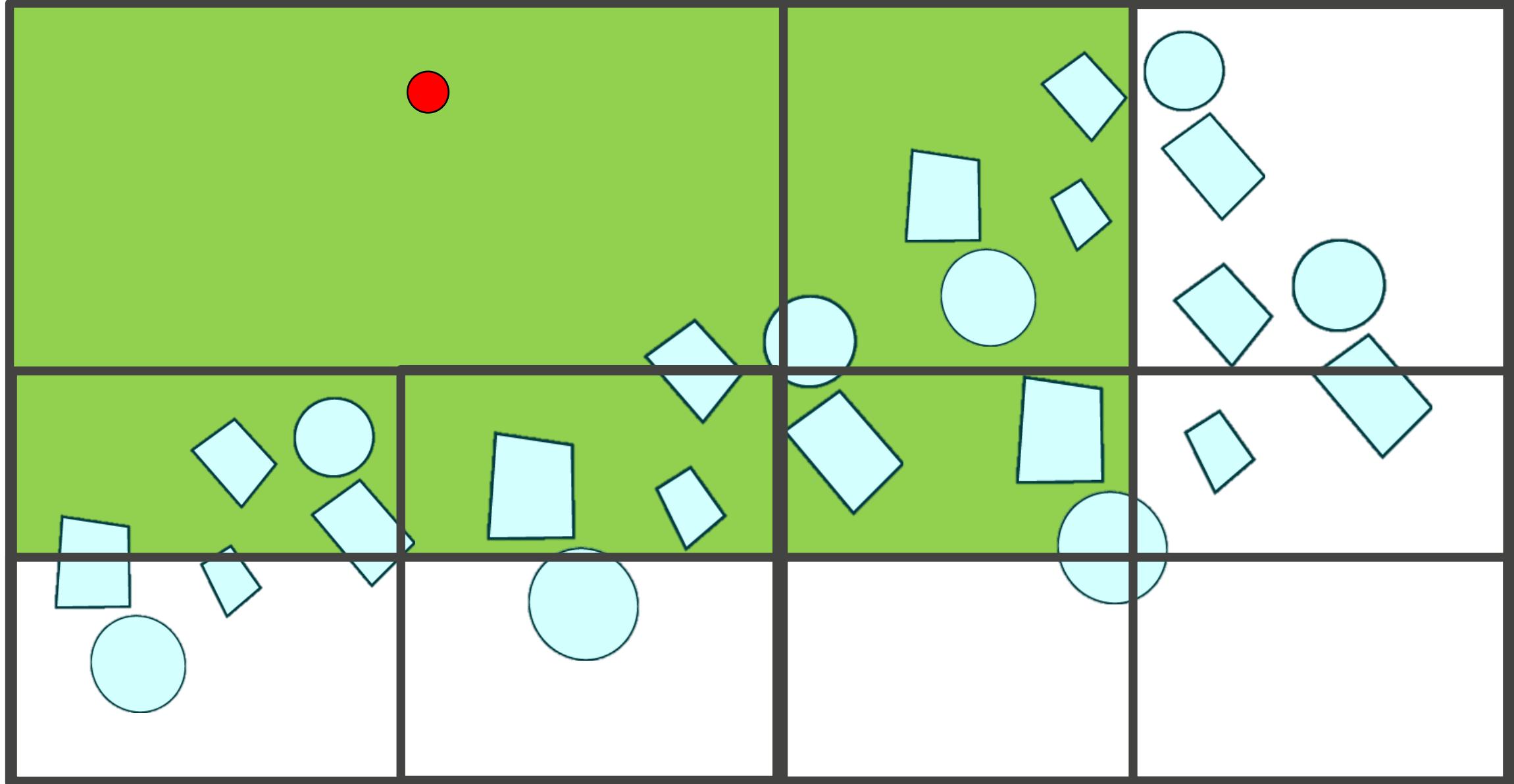
Distance Query



Distance Query



Distance Query



Done

Assignment 1 grades out (please request remark in 1 week)

Assignment 4 is out today (due October 13th)

Office hours Friday 1-2pm