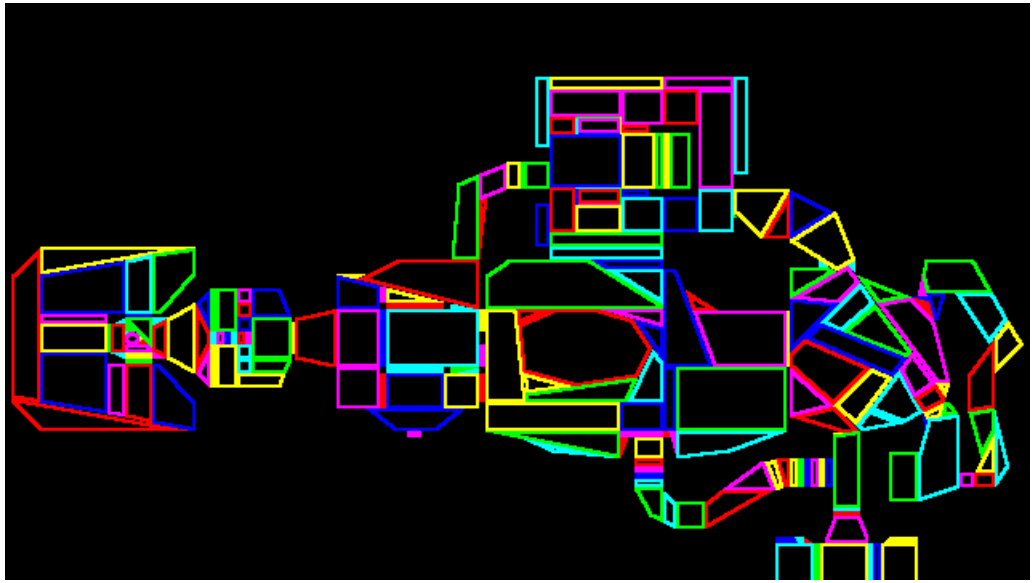
Bounding Volume Hierarchies



Some Slides/Images adapted from Marschner and Shirley and David Levin

Announcements

Assignment 4 due 9 June

Has anyone experienced trouble compiling/running A4?

Bounding Volume Hierarchy

Review of Bounding Volumes and BVHs

Constructing Object-Partitioning Hierarchies

- AABB Trees

Space-Partitioning Hierarchies

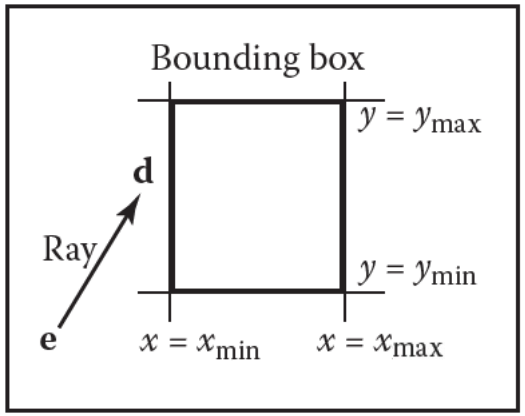
- Uniform Spatial Subdivision
- Axis-Aligned Spatial Subdivision

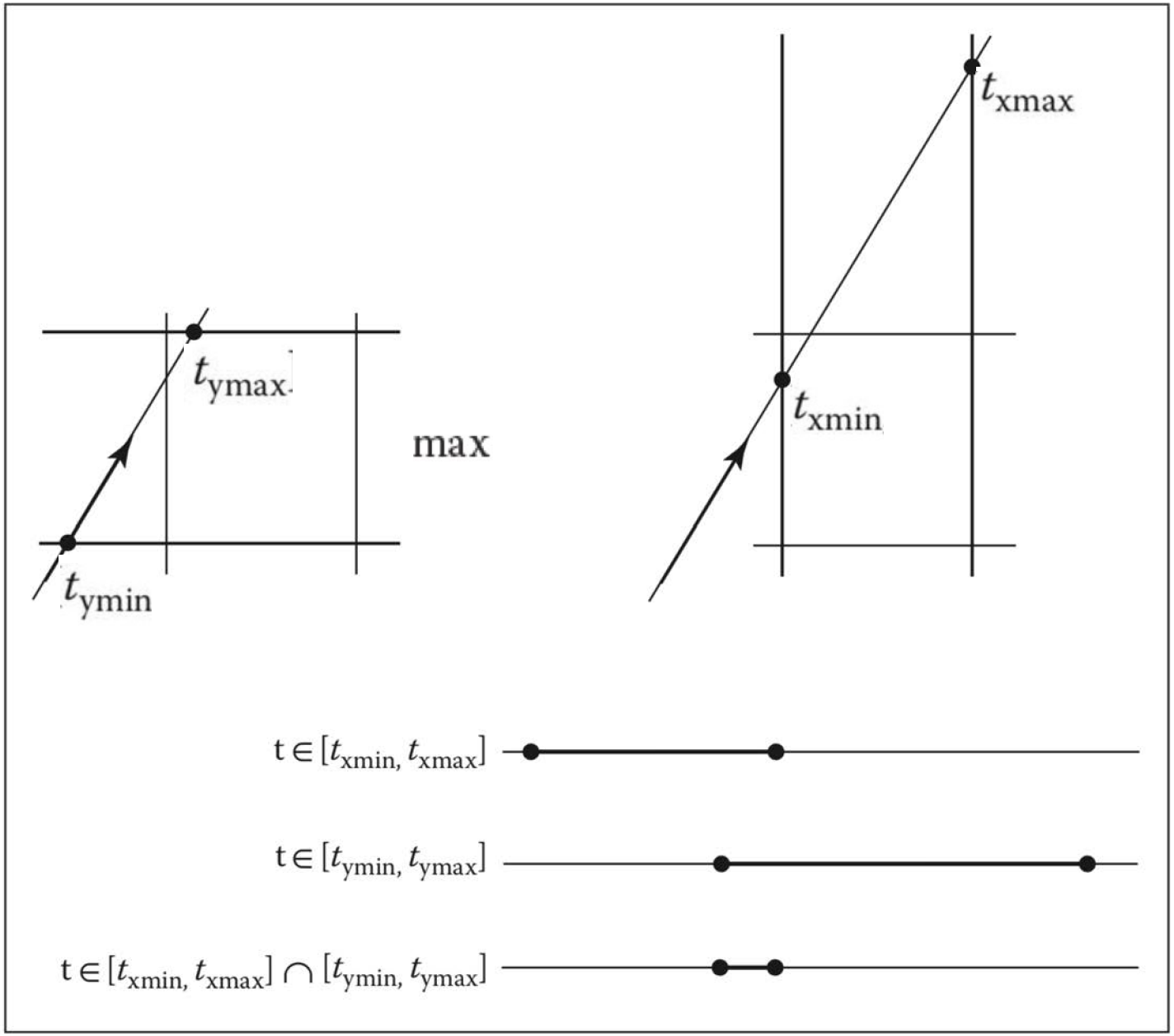
Clarifications

On ray-box intersection

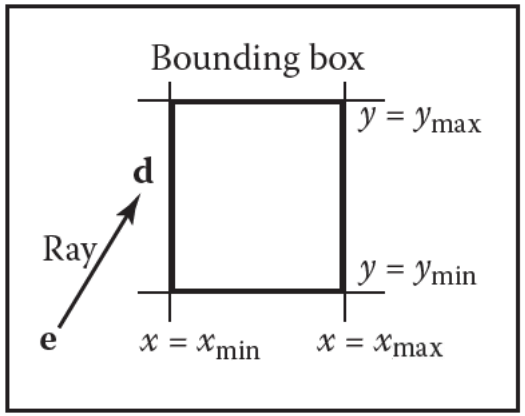
Ray-AABB Intersection

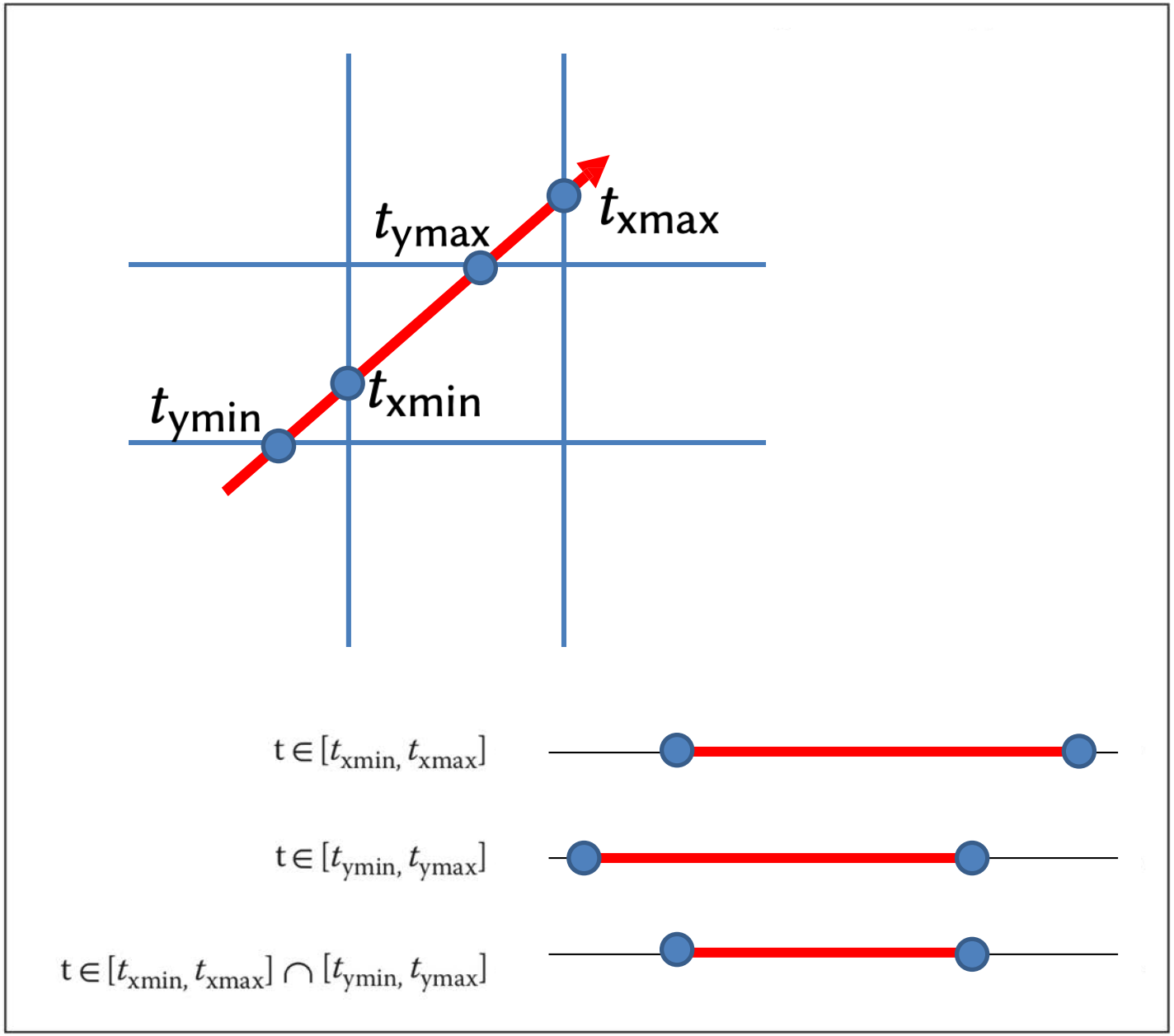
Note: I think the x pictures and y pictures are swapped



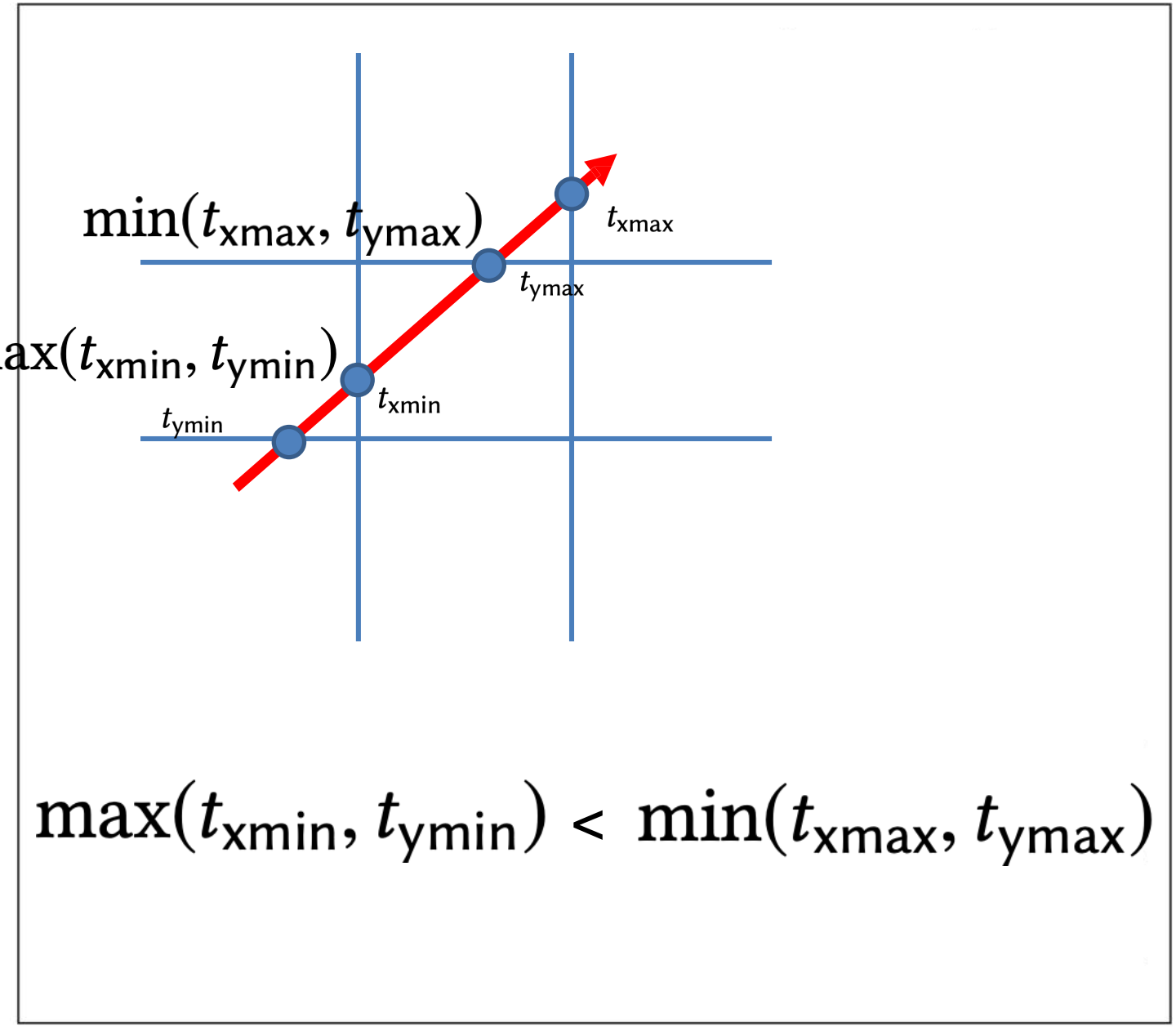


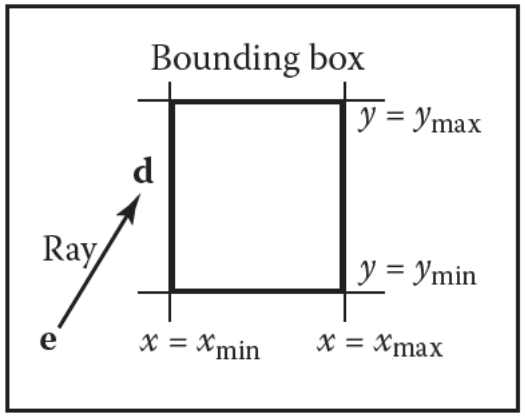
Ray-AABB Intersection



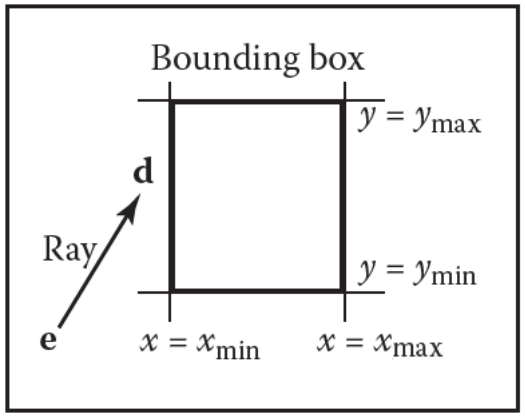


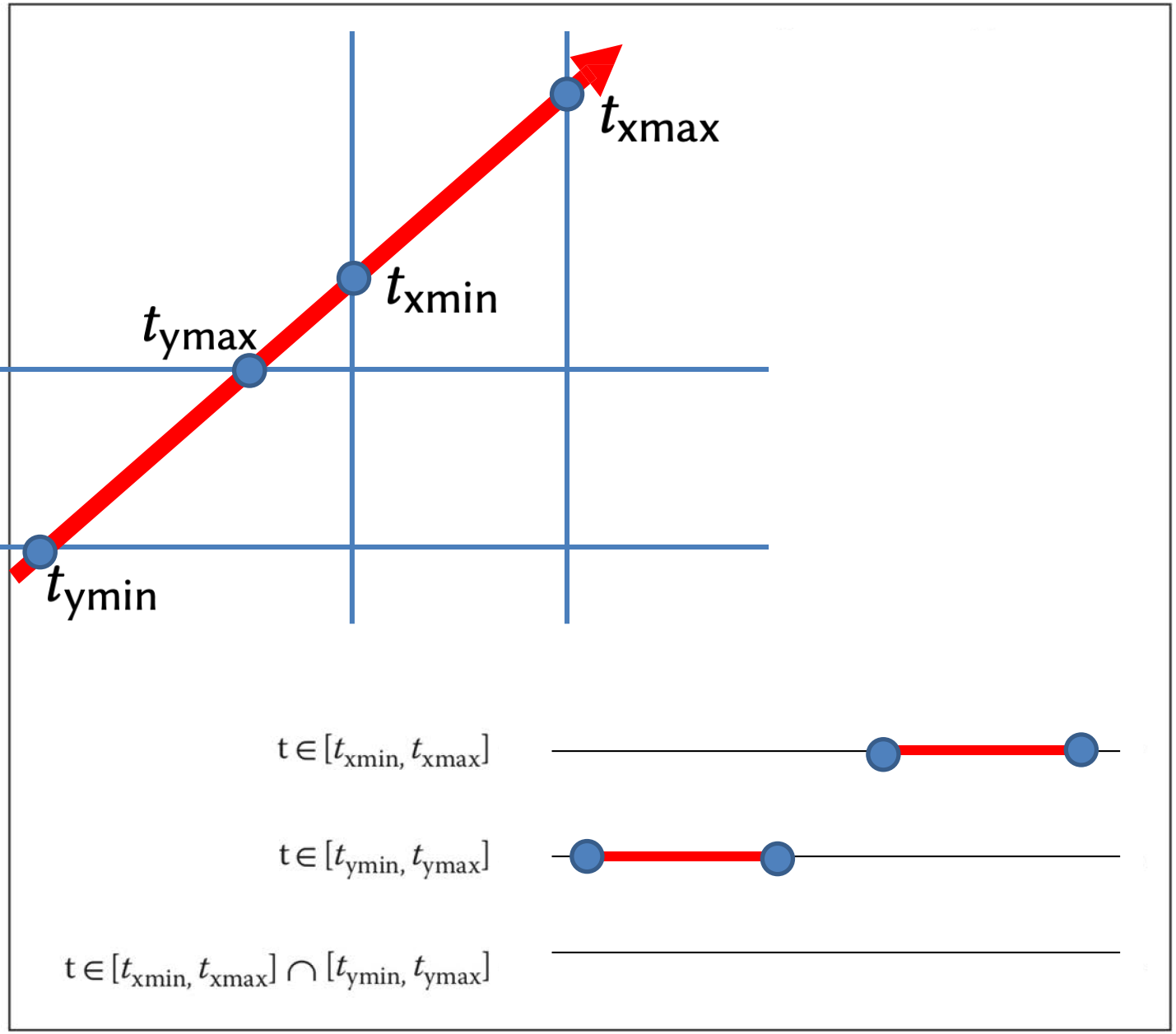
Ray-AABB Intersection



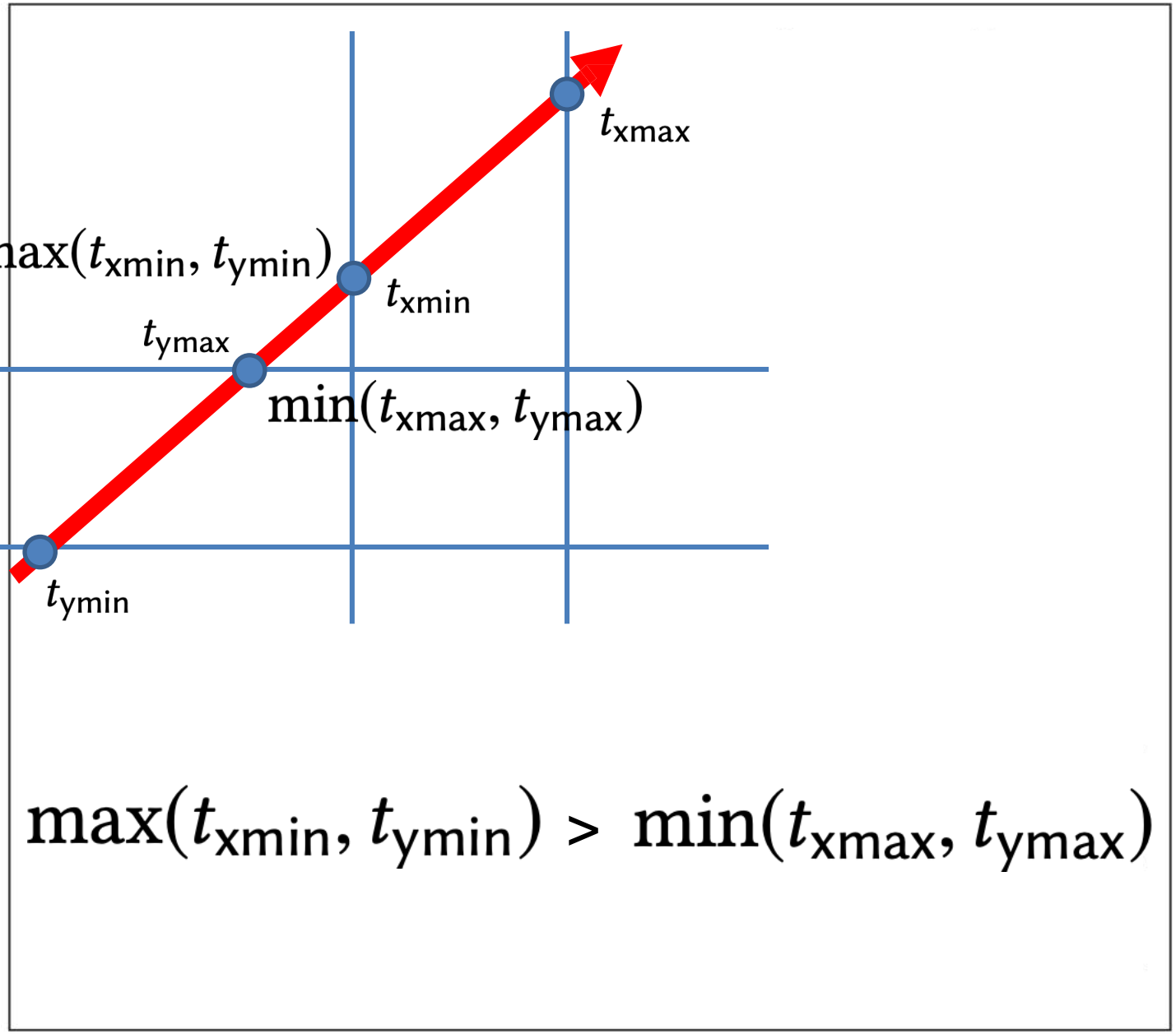


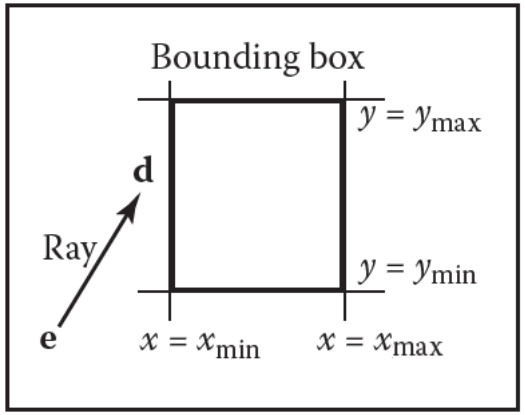
Ray-AABB Intersection





Ray-AABB Intersection





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

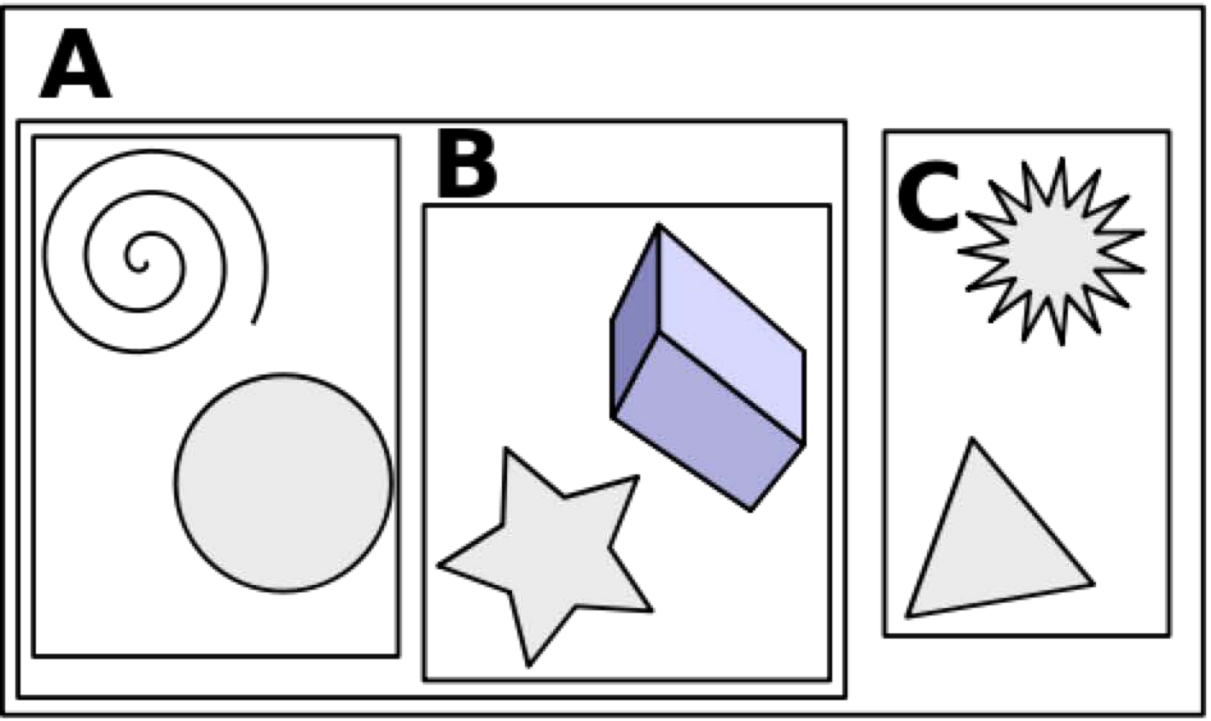
So the check would be:
The ray intersects the box if

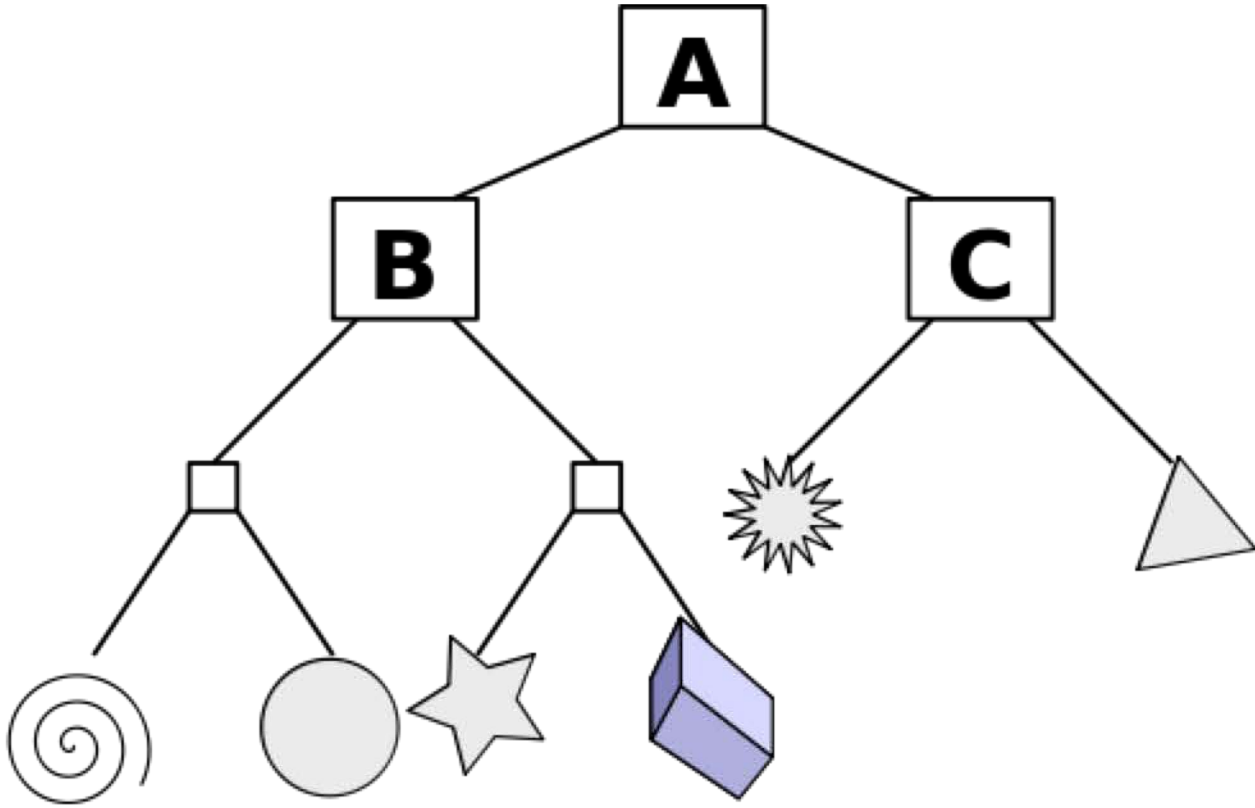
$$\max(t_{\text{xmin}}, t_{\text{ymin}}) < \min(t_{\text{xmax}}, t_{\text{ymax}})$$

Which is easily extended to 3D

Any Questions?

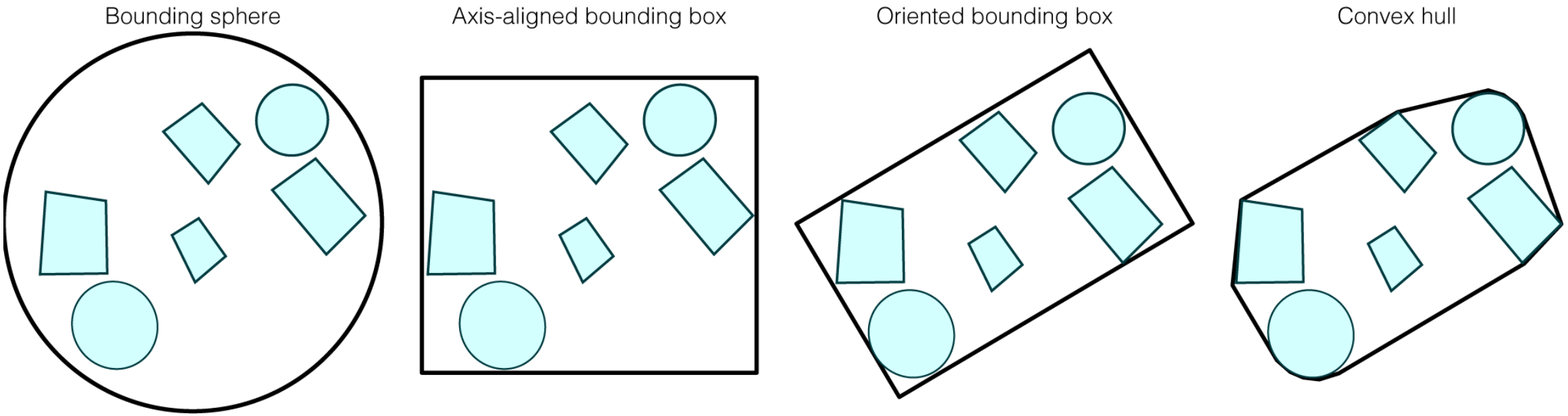
Bounding Volume Hierarchy





Bounding Volumes (BVs)

"Simple" geometry that fully encloses a **collection** of other geometry



Sphere
(a lot)

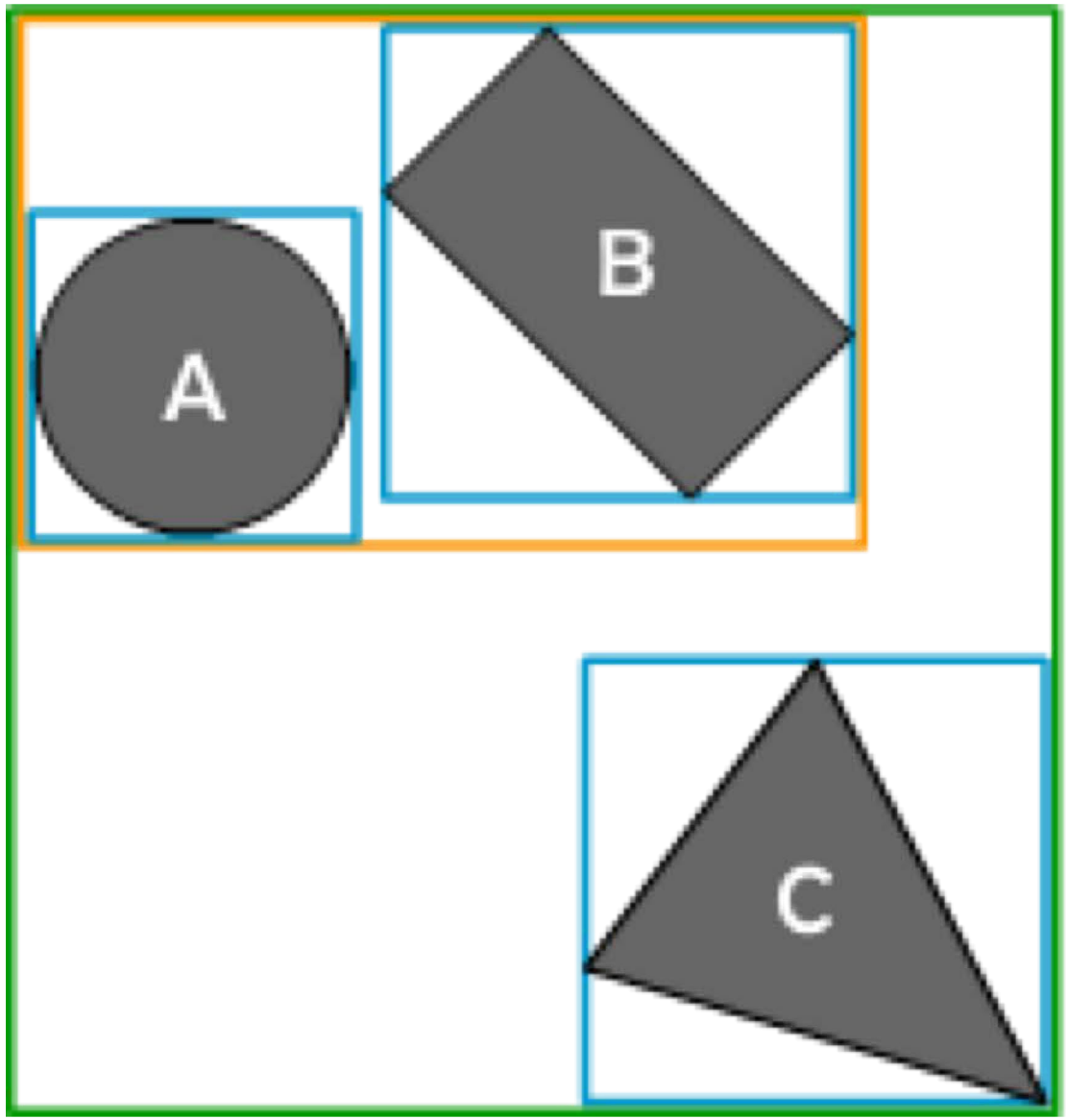
AABB
(a lot)

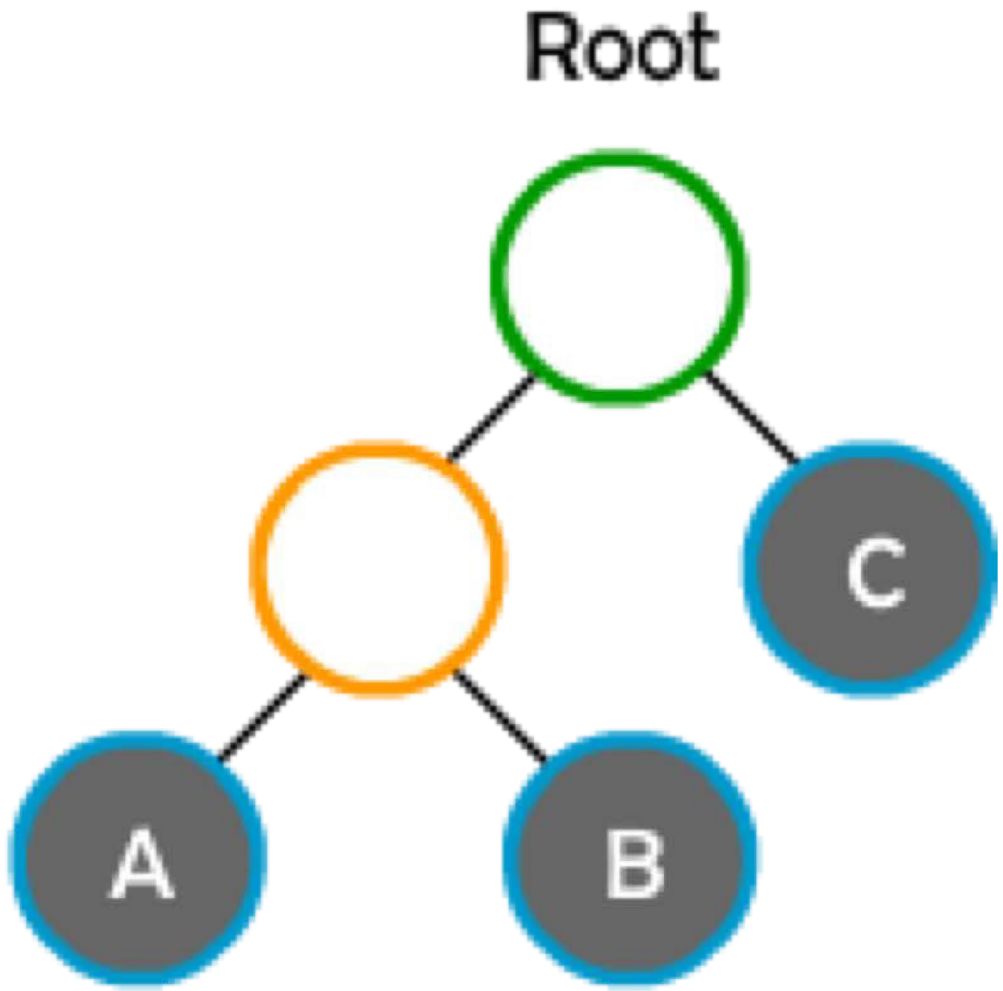
OOBB
(a little)

Convex Hull
(nope!)

https://en.wikipedia.org/wiki/Convex_hull

AABB Trees





http://allenchou.net/2014/02/game-physics-broadphase-dynamic-aabb-tree/

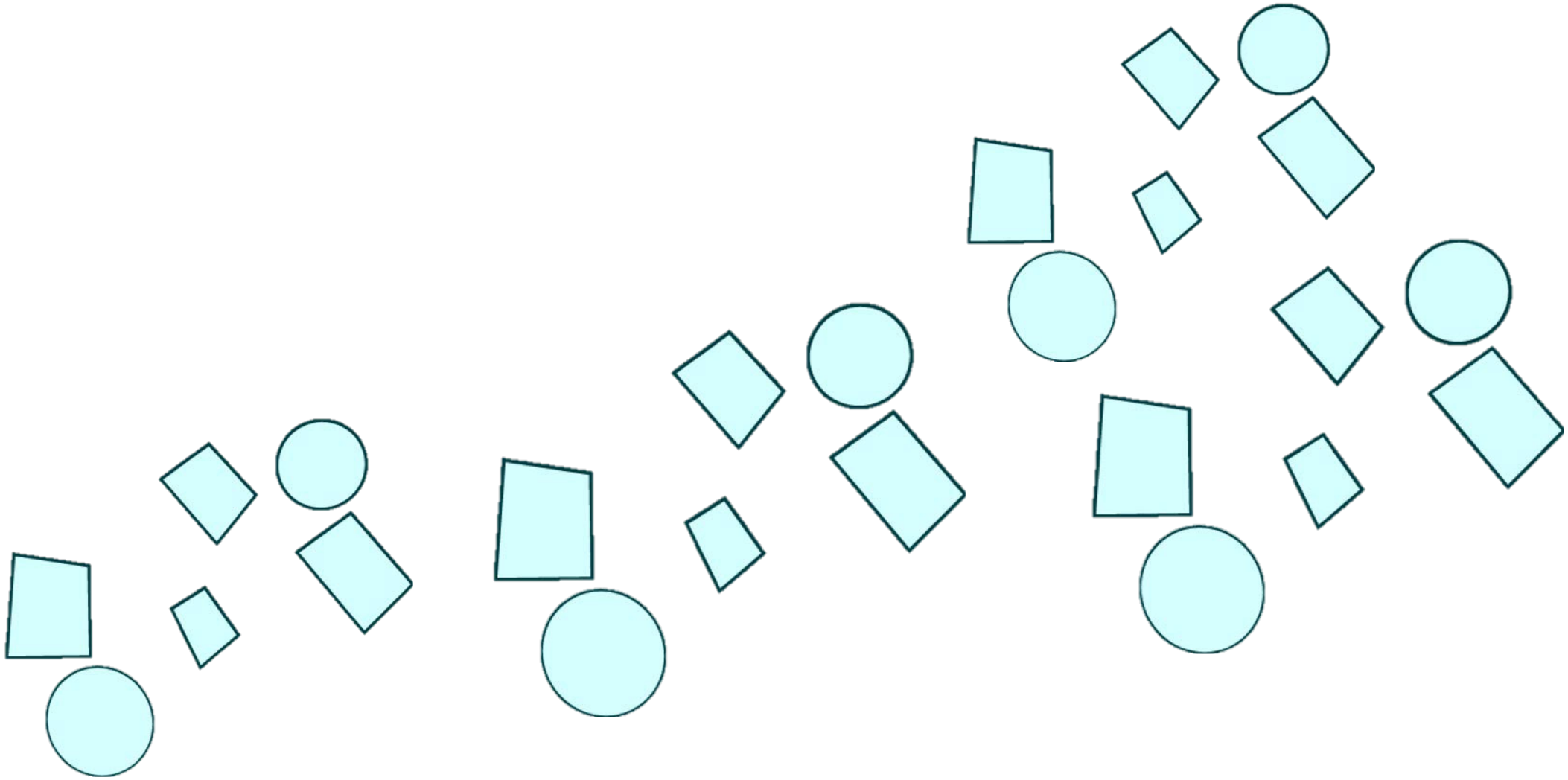
BVH Intersection Queries

```
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 BV
      if (i1 && i2) {  t=min(t1,t2); return true;  }
      if (i1) {  t=t1; return true;  }
      if (i2) {  t=t2; return true;  }
      return false;
    }
}
```

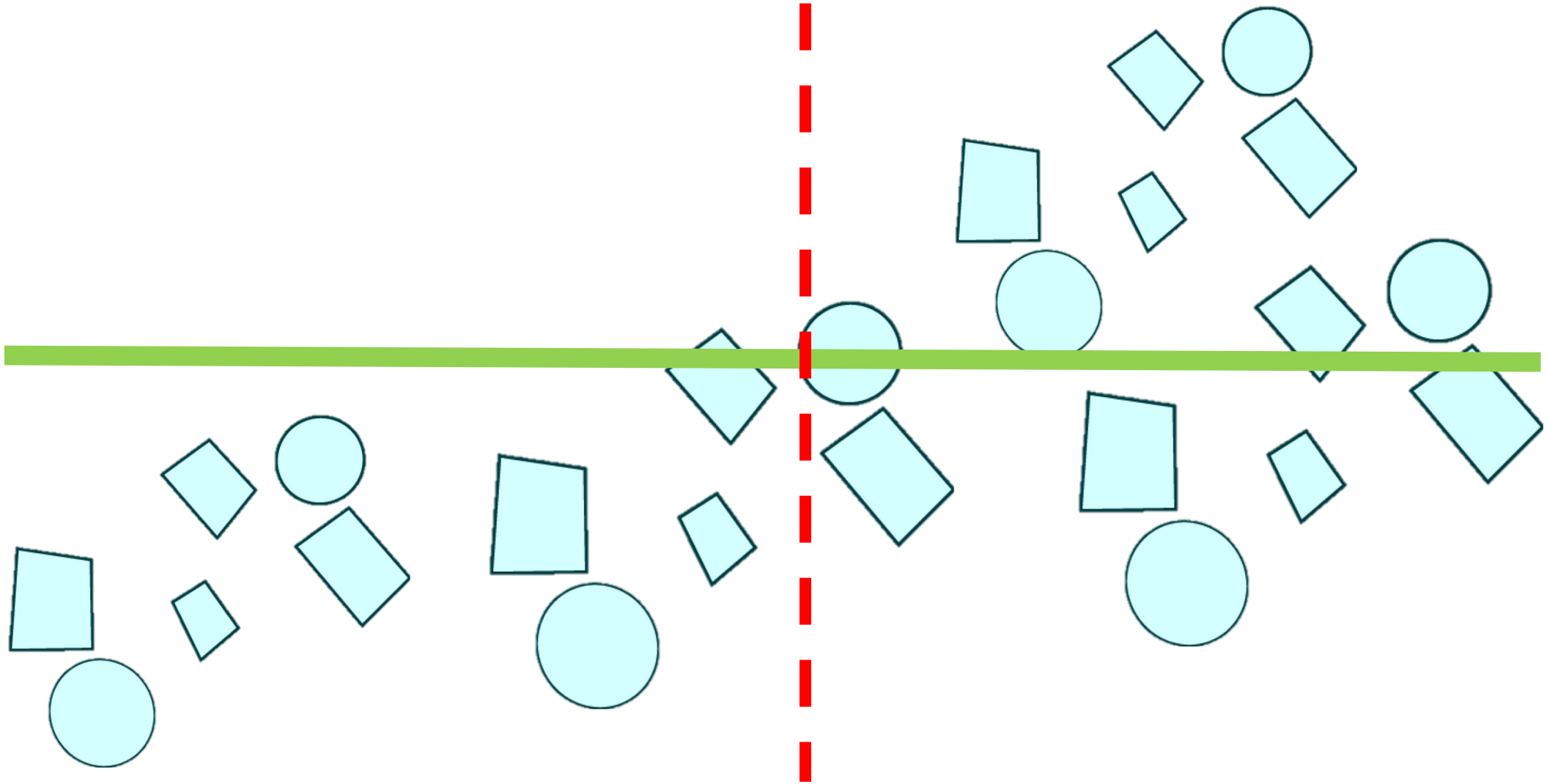
BVH Distance Queries

```
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)
}
```

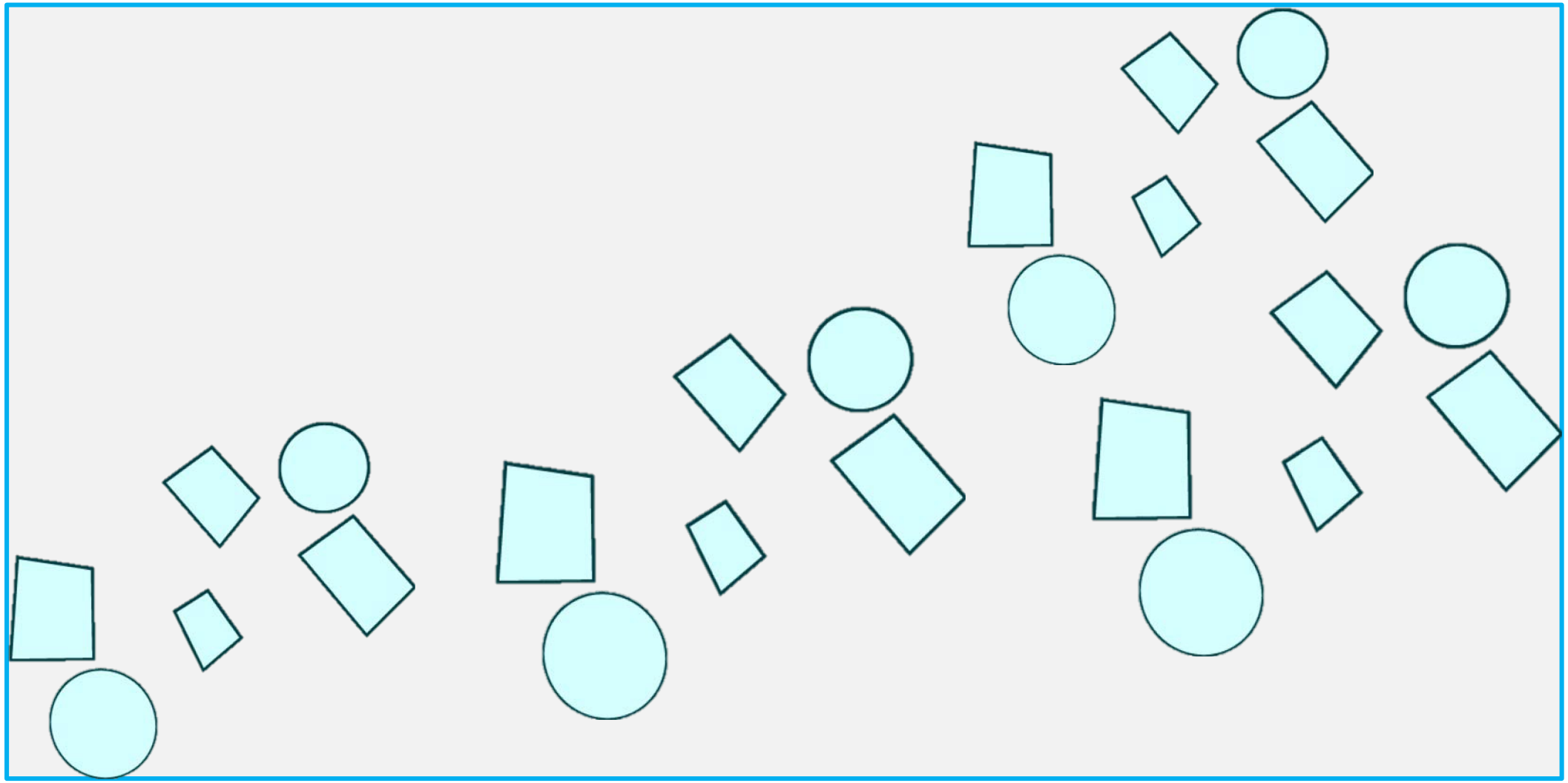
Constructing a BVH



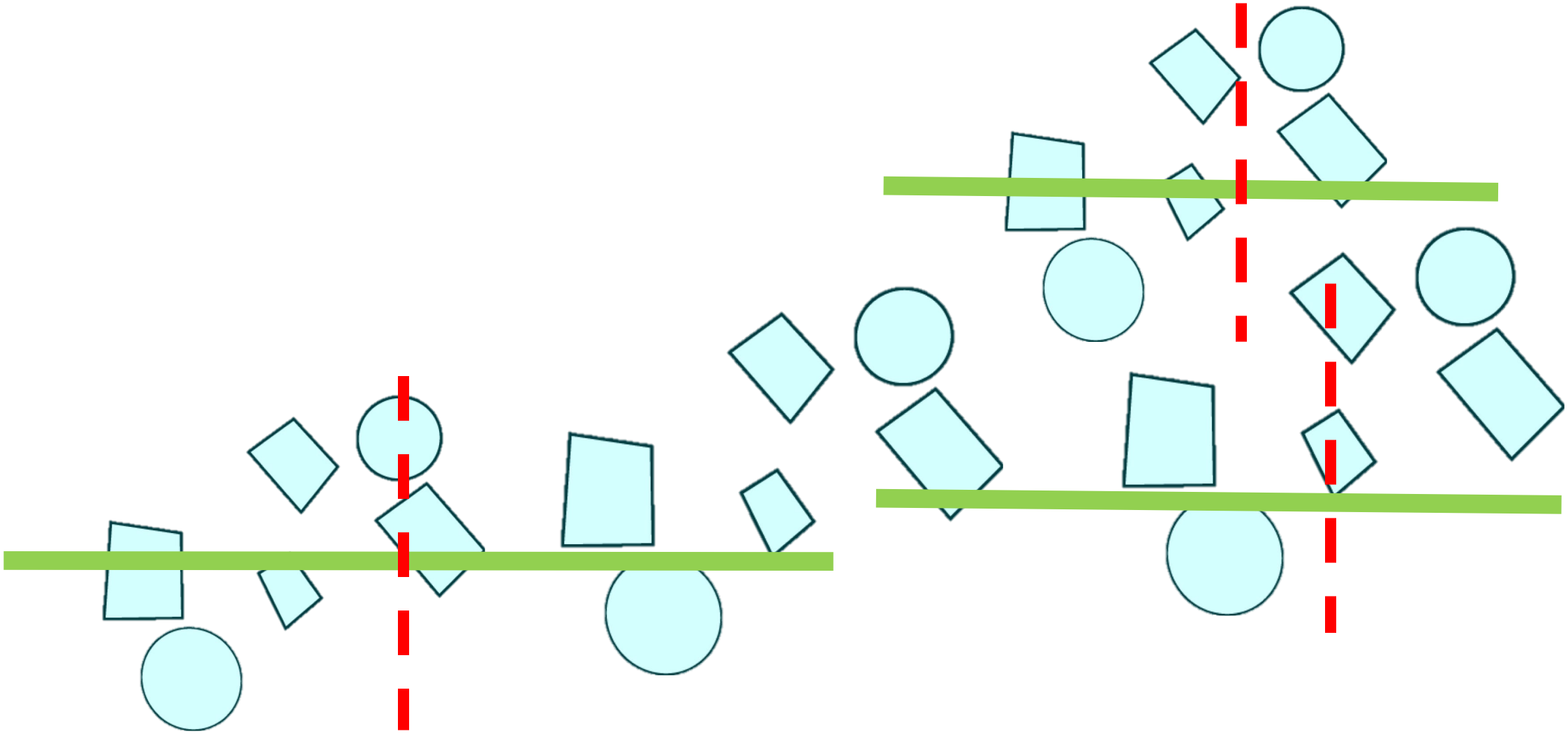
Constructing a BVH



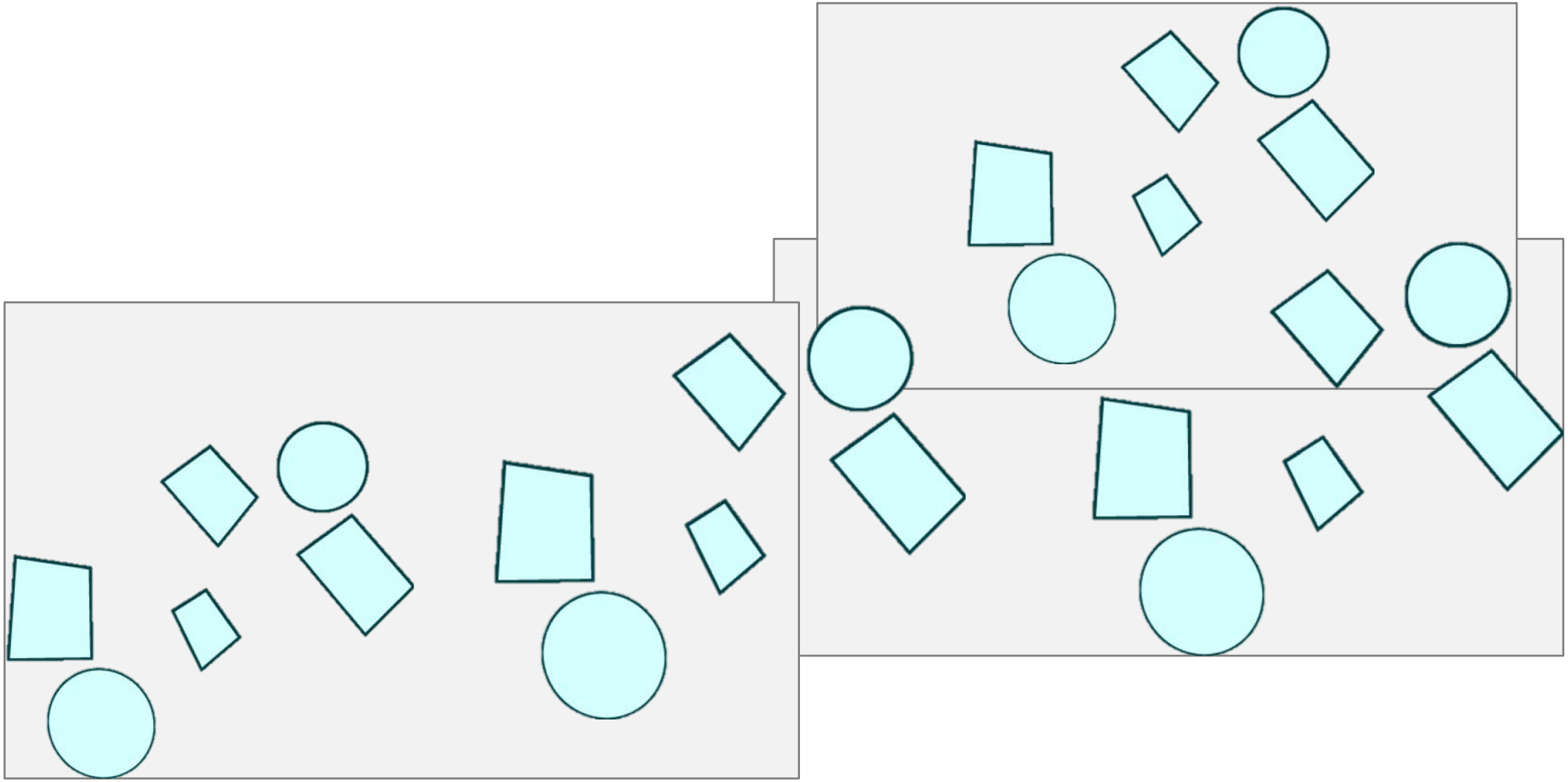
Constructing a BVH



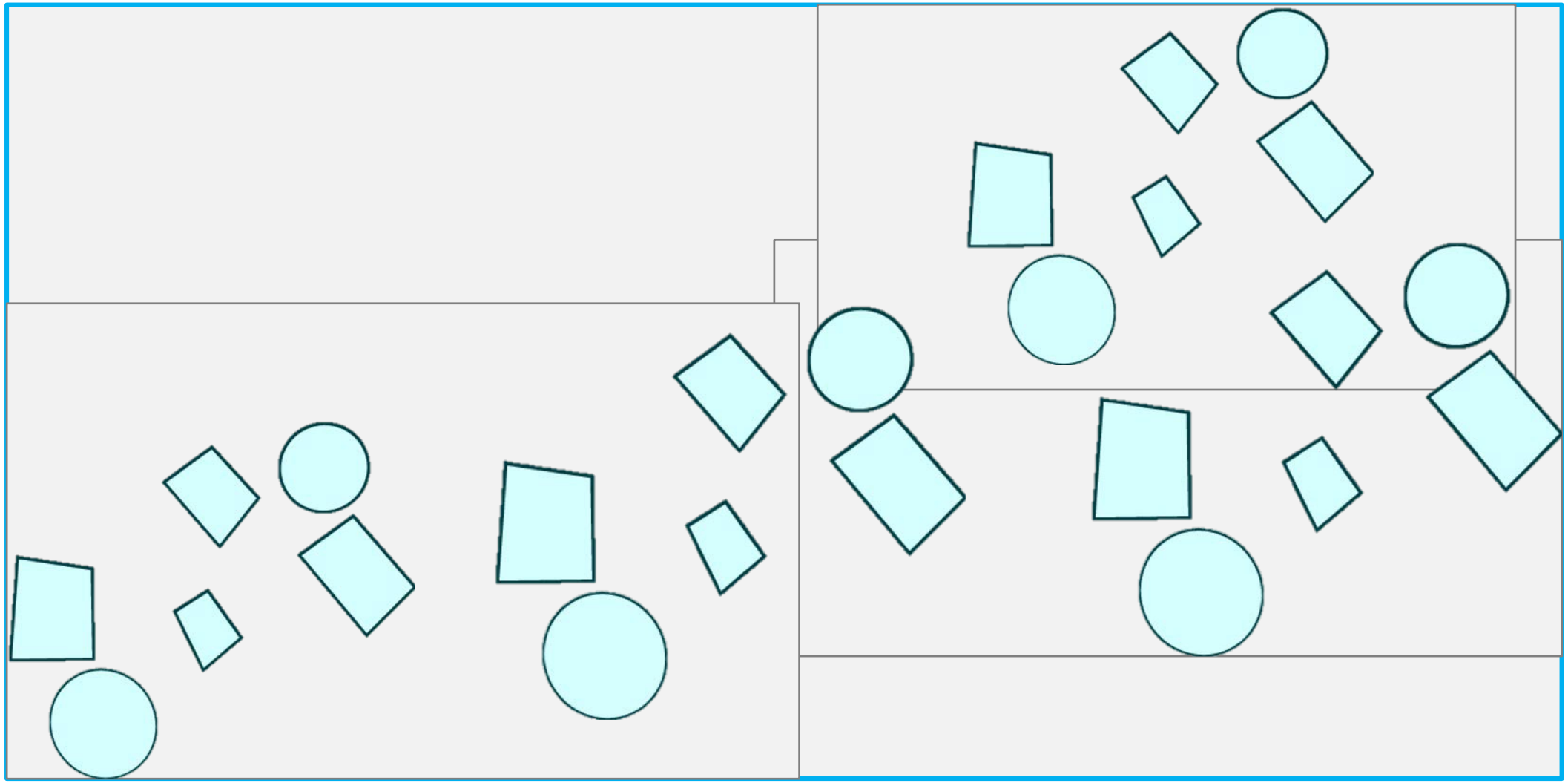
Constructing a BVH



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

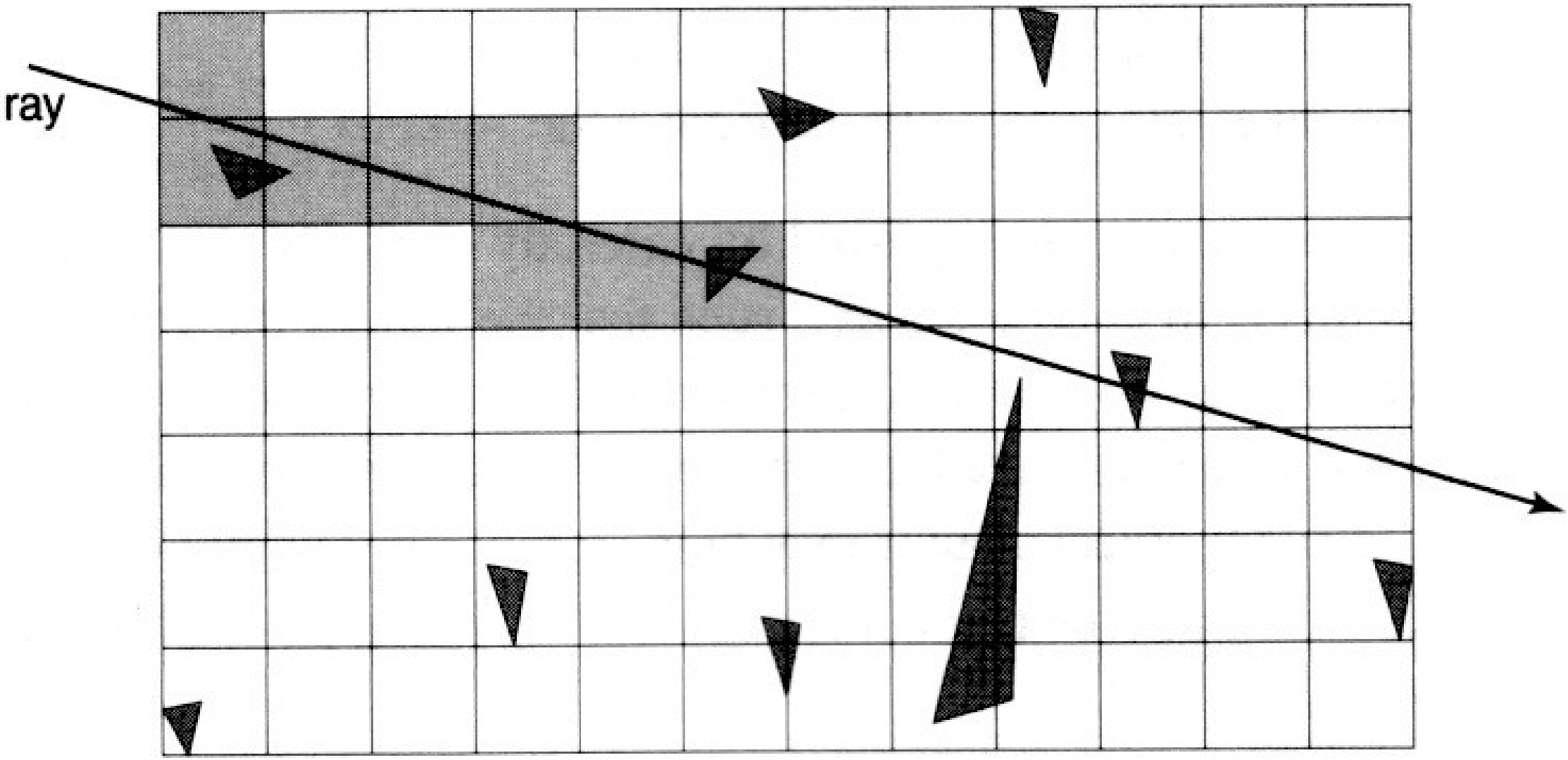
Spatial Data Structures

Basic Idea – asymptotic improvement in spatial queries by subdividing

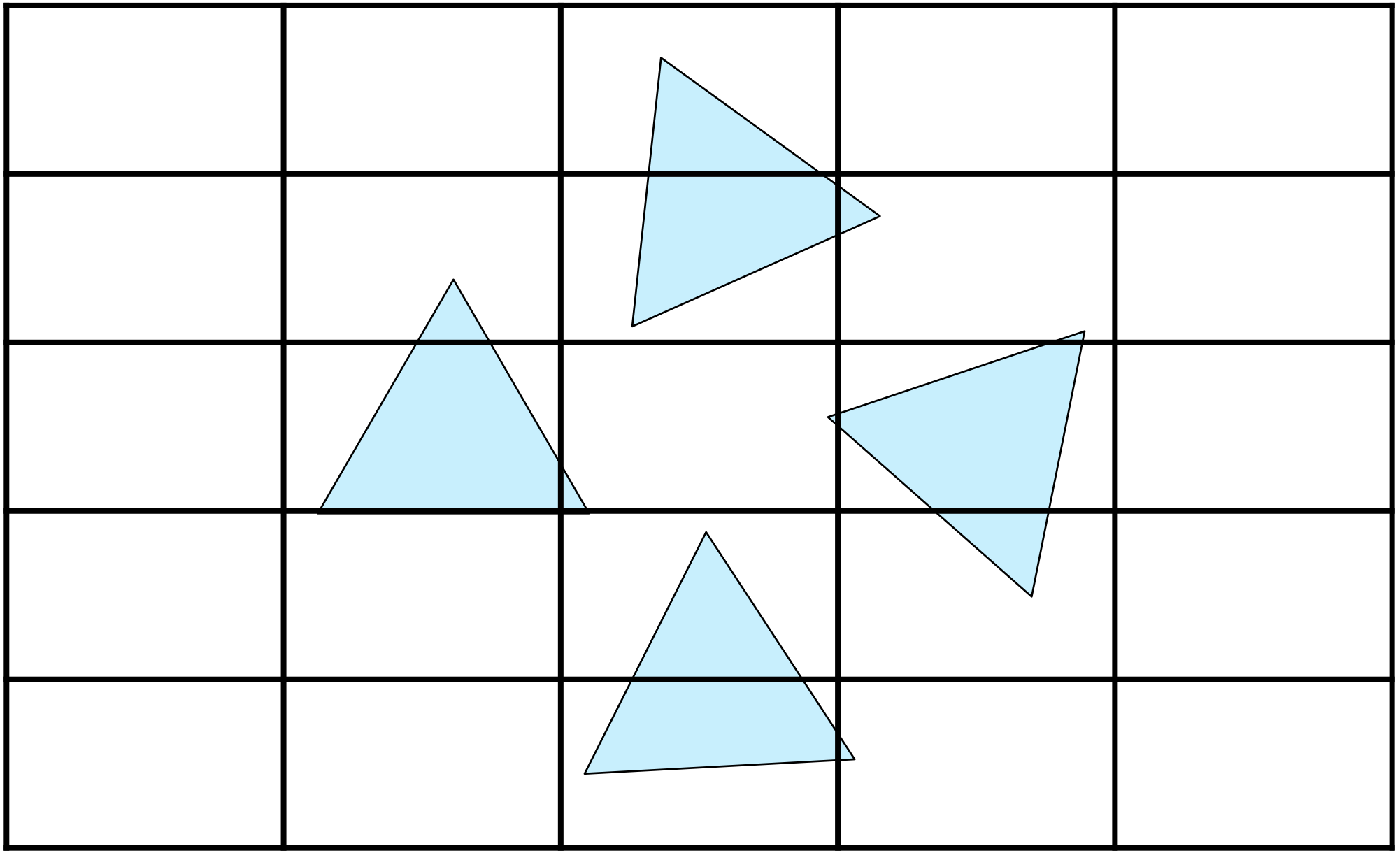
Two types of subdivisions – object-based and ***spatial***

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

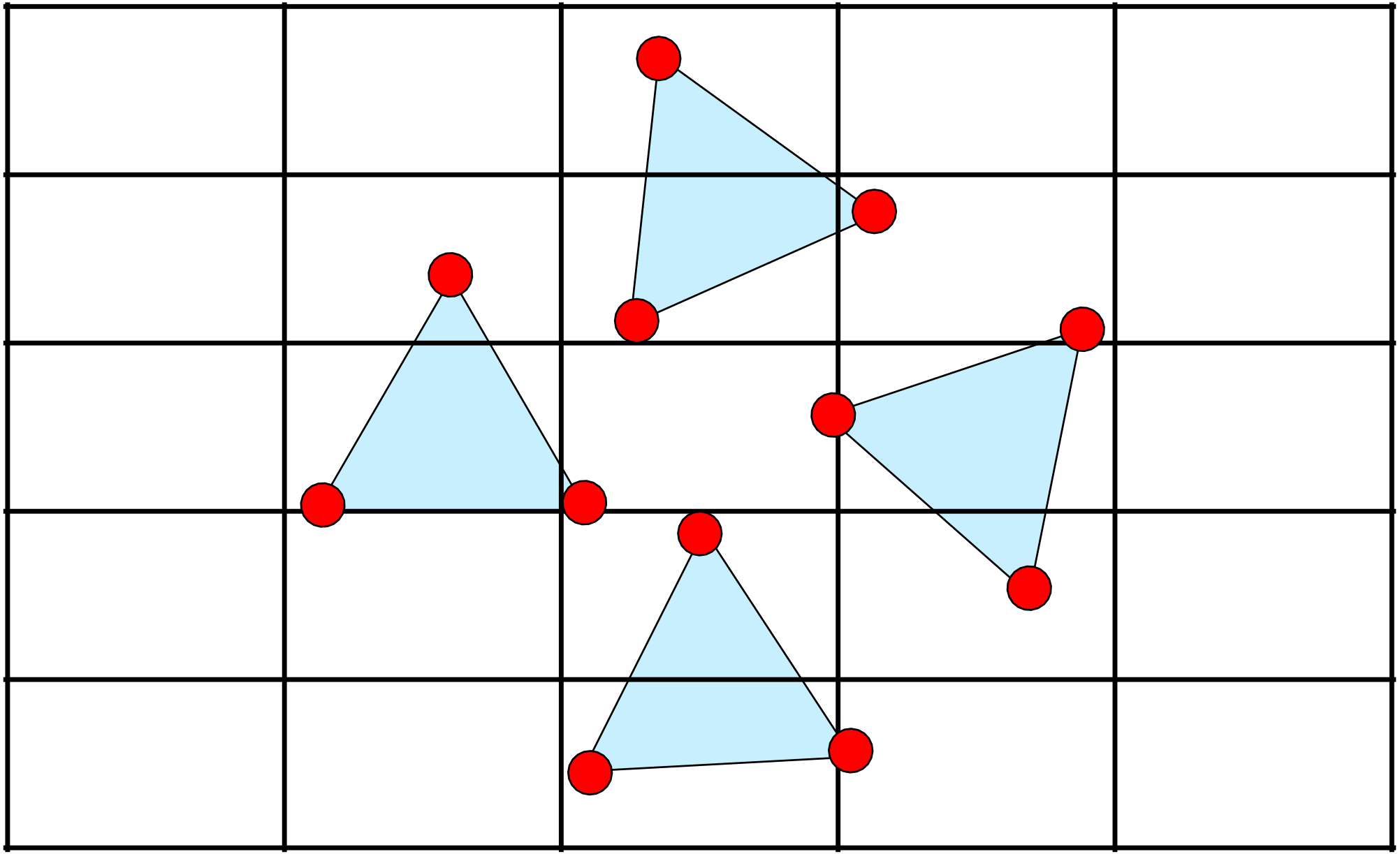
Axis-Aligned Spatial Subdivision (Uniform)



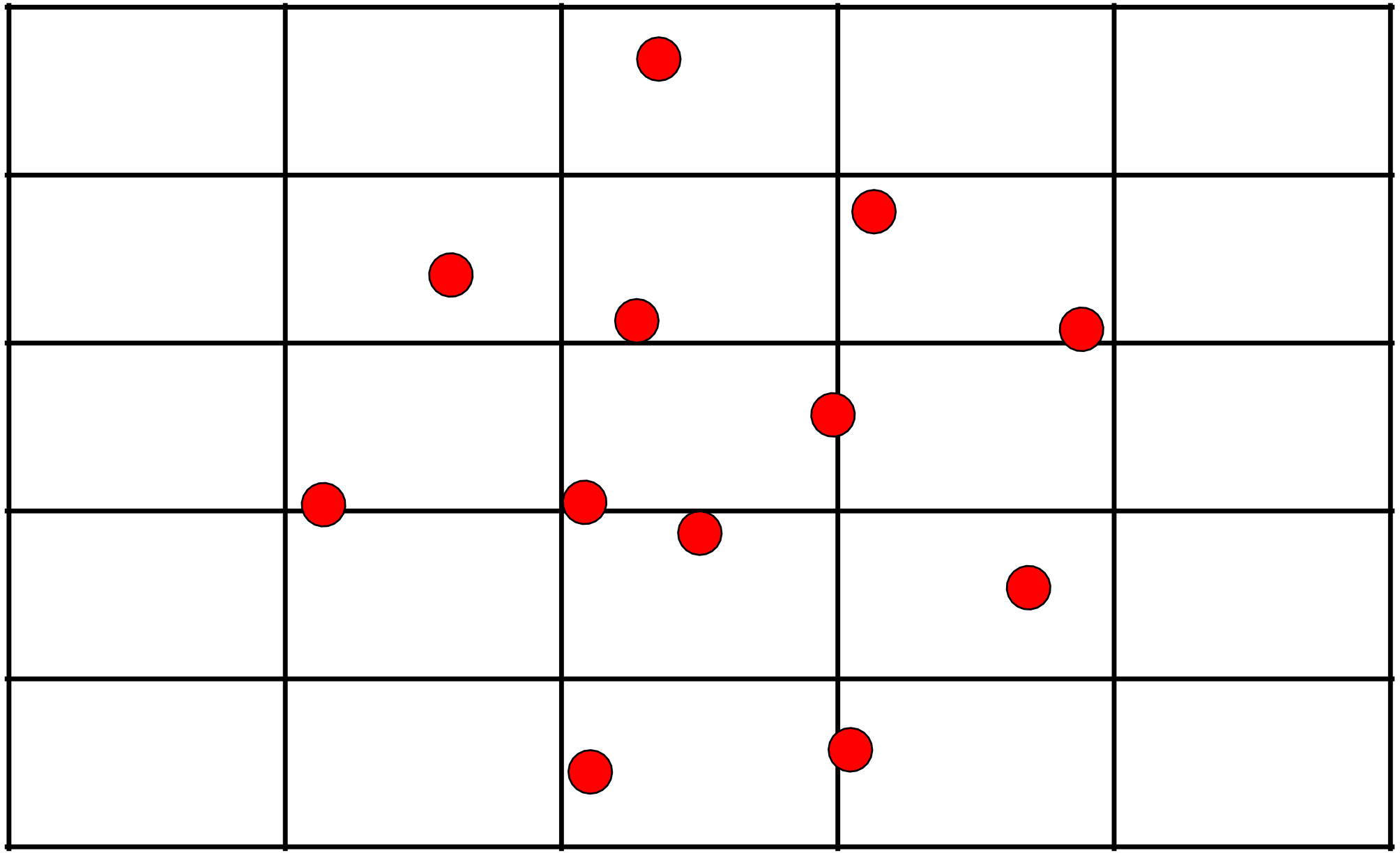
Construction



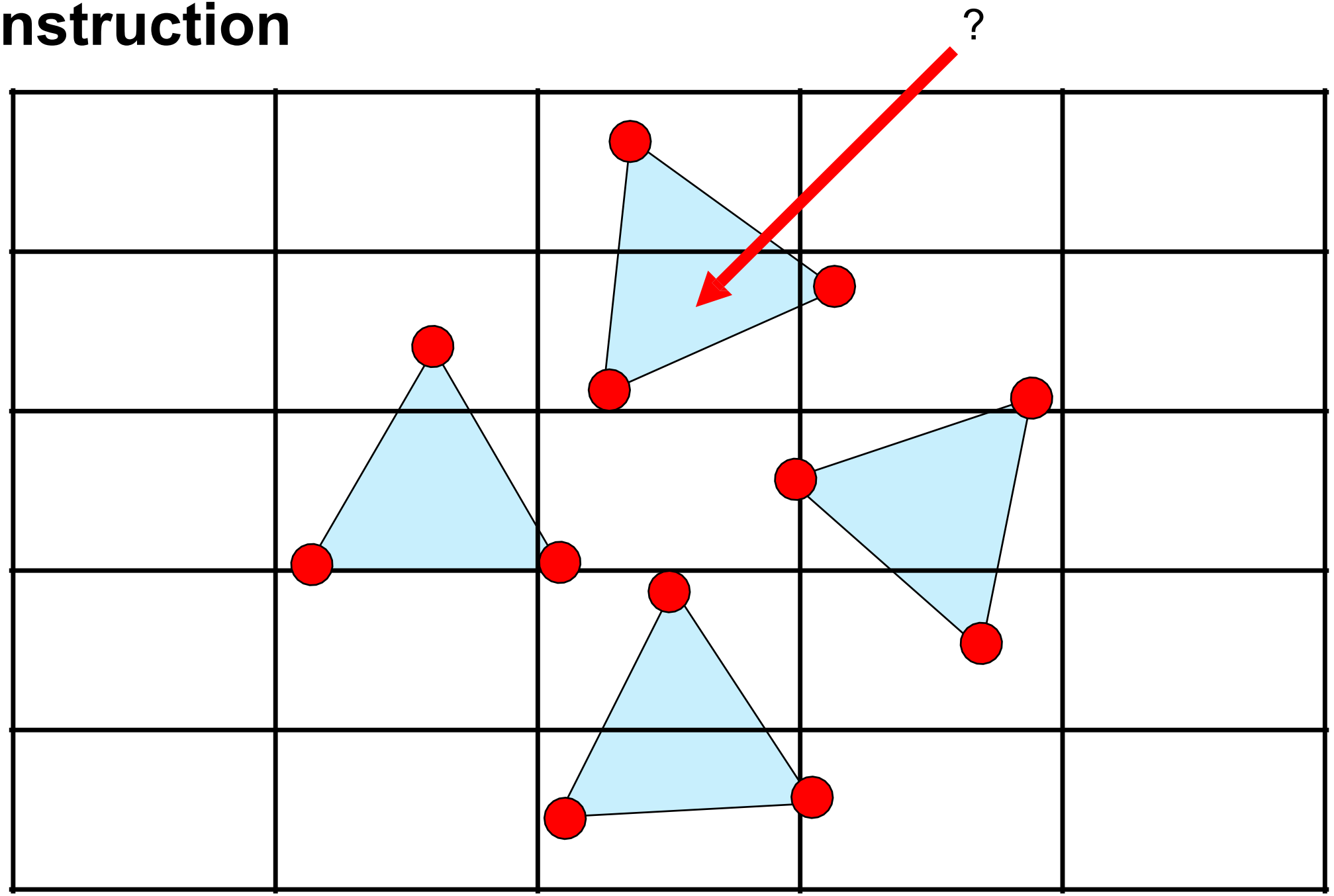
Construction



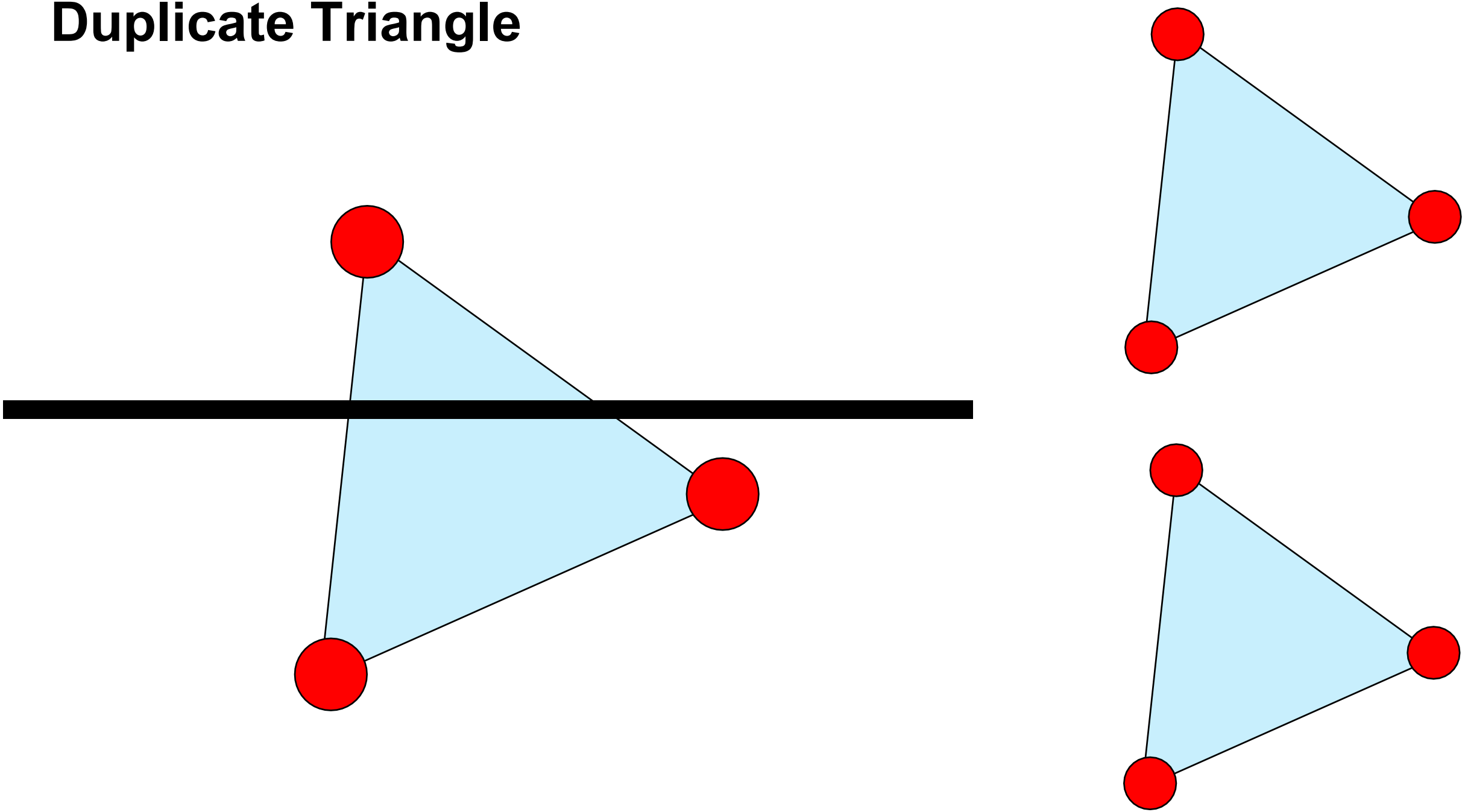
Construction



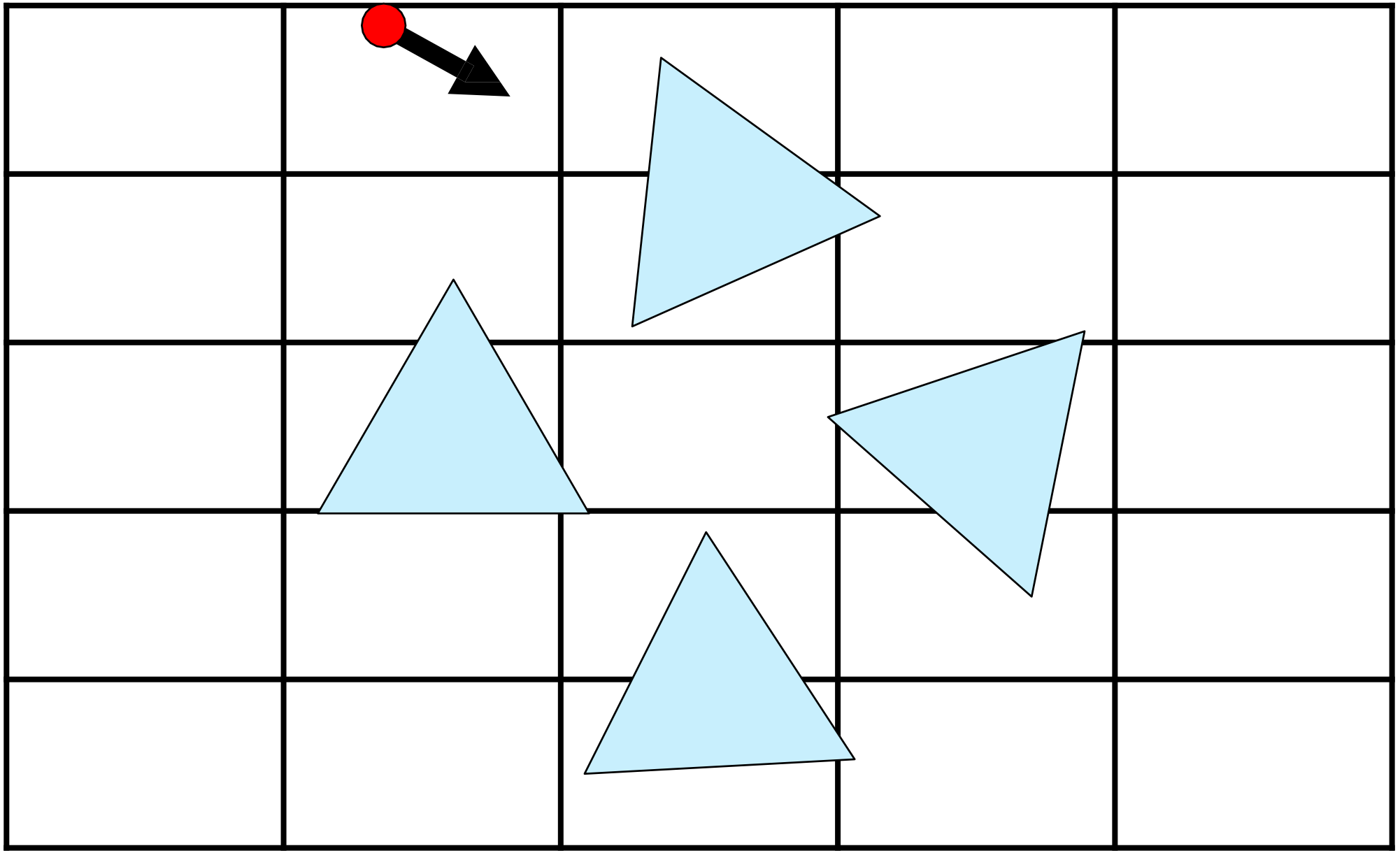
Construction



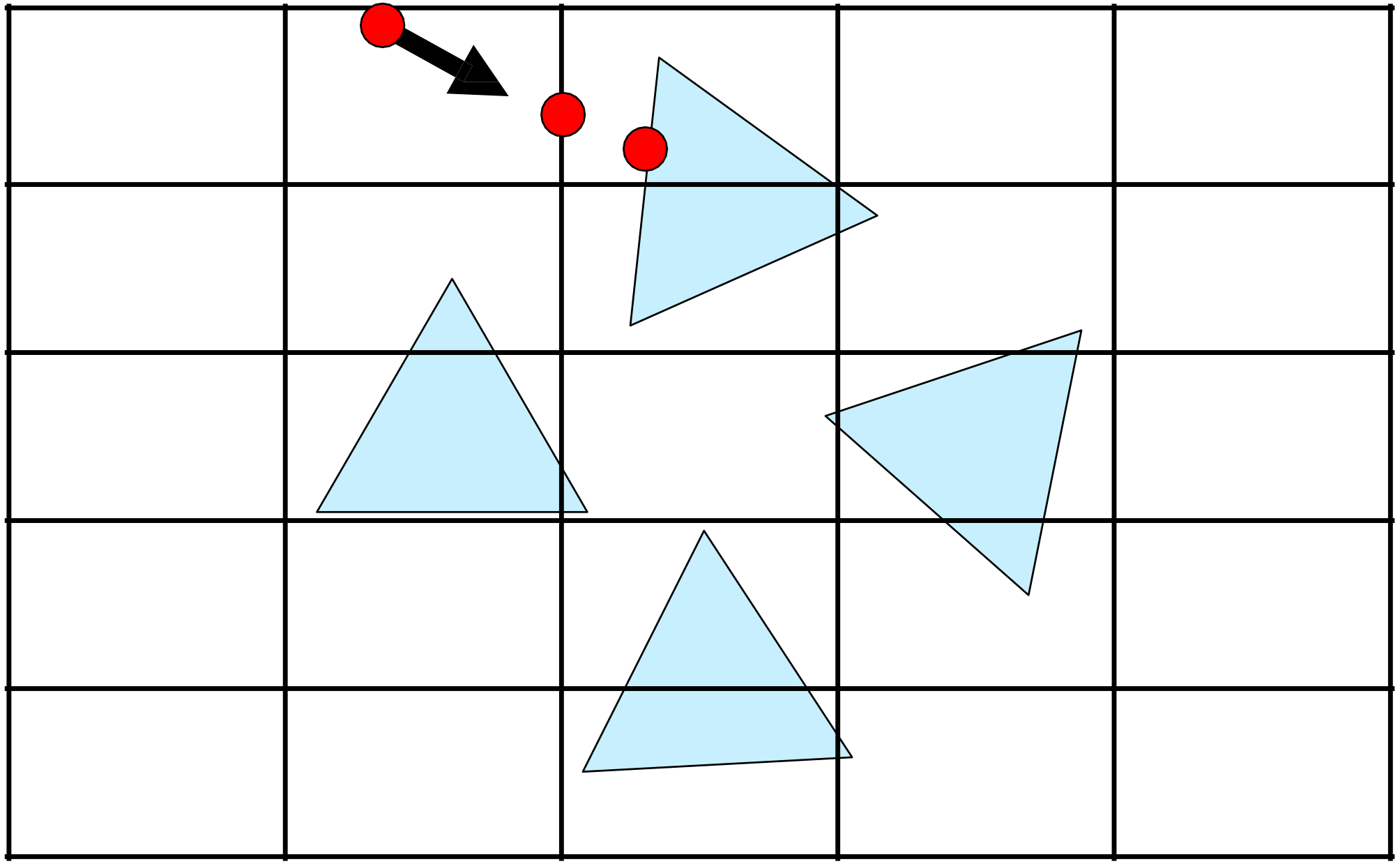
Duplicate Triangle



Intersection Tests

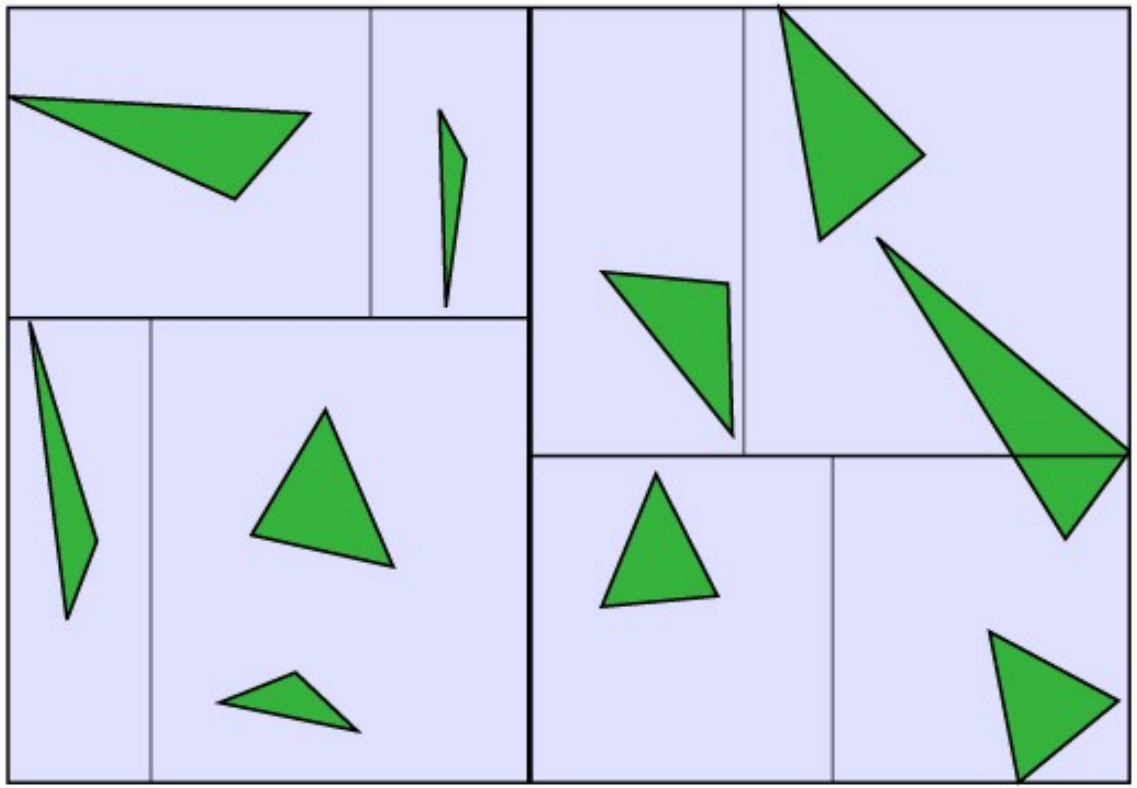


Intersection Tests

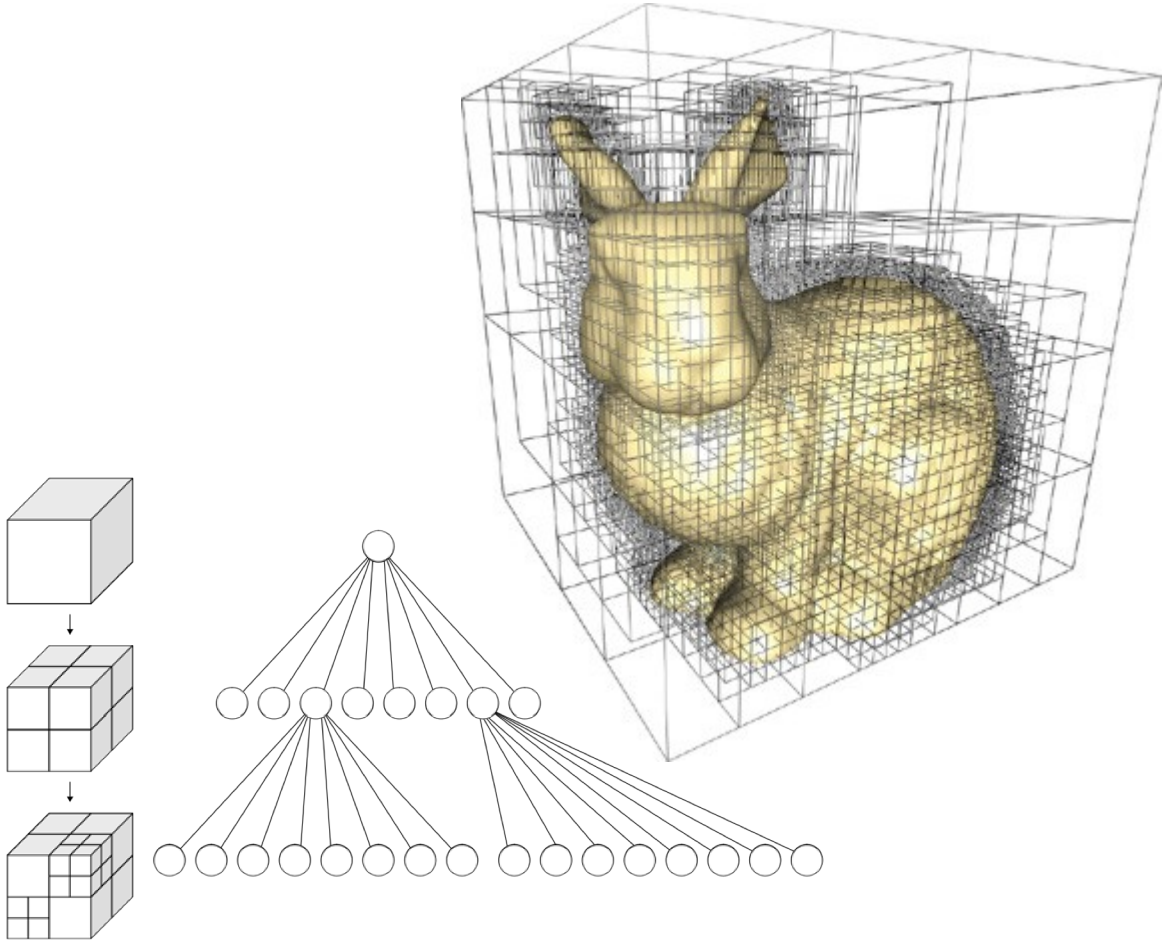


Axis-Aligned Spatial Subdivision (Non-Uniform)

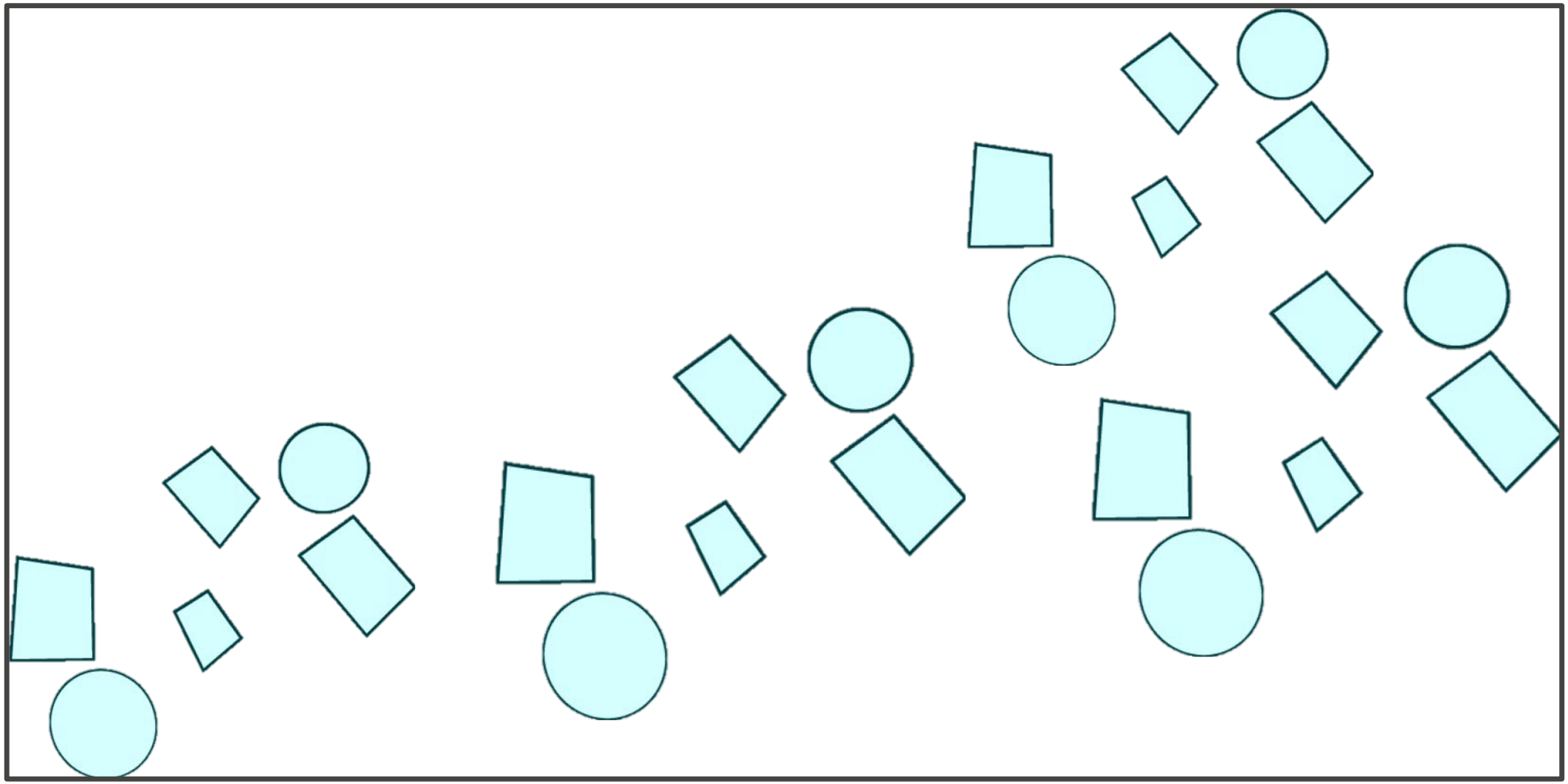
BSP Tree



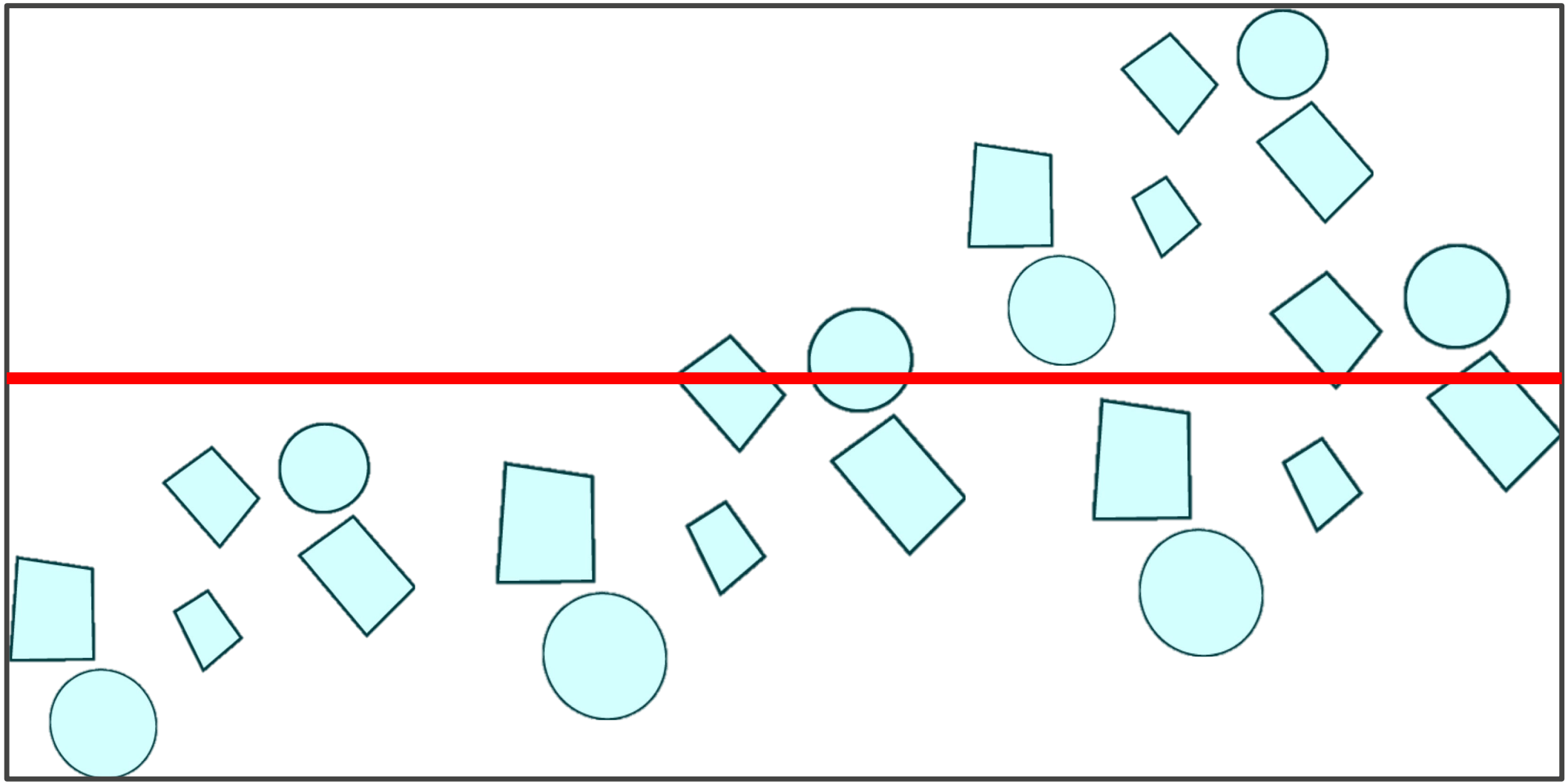
Octree



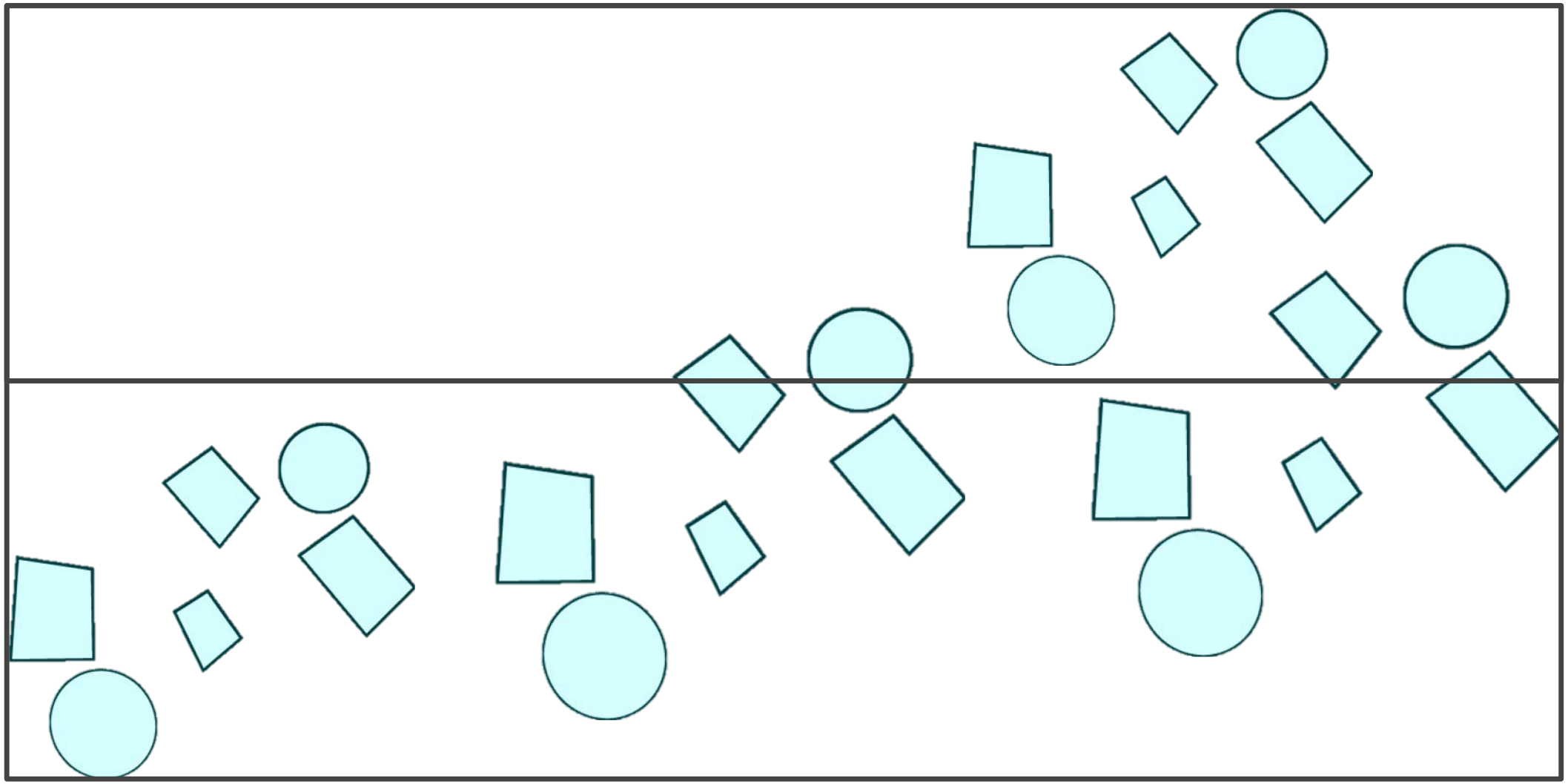
Constructing a k-d Tree



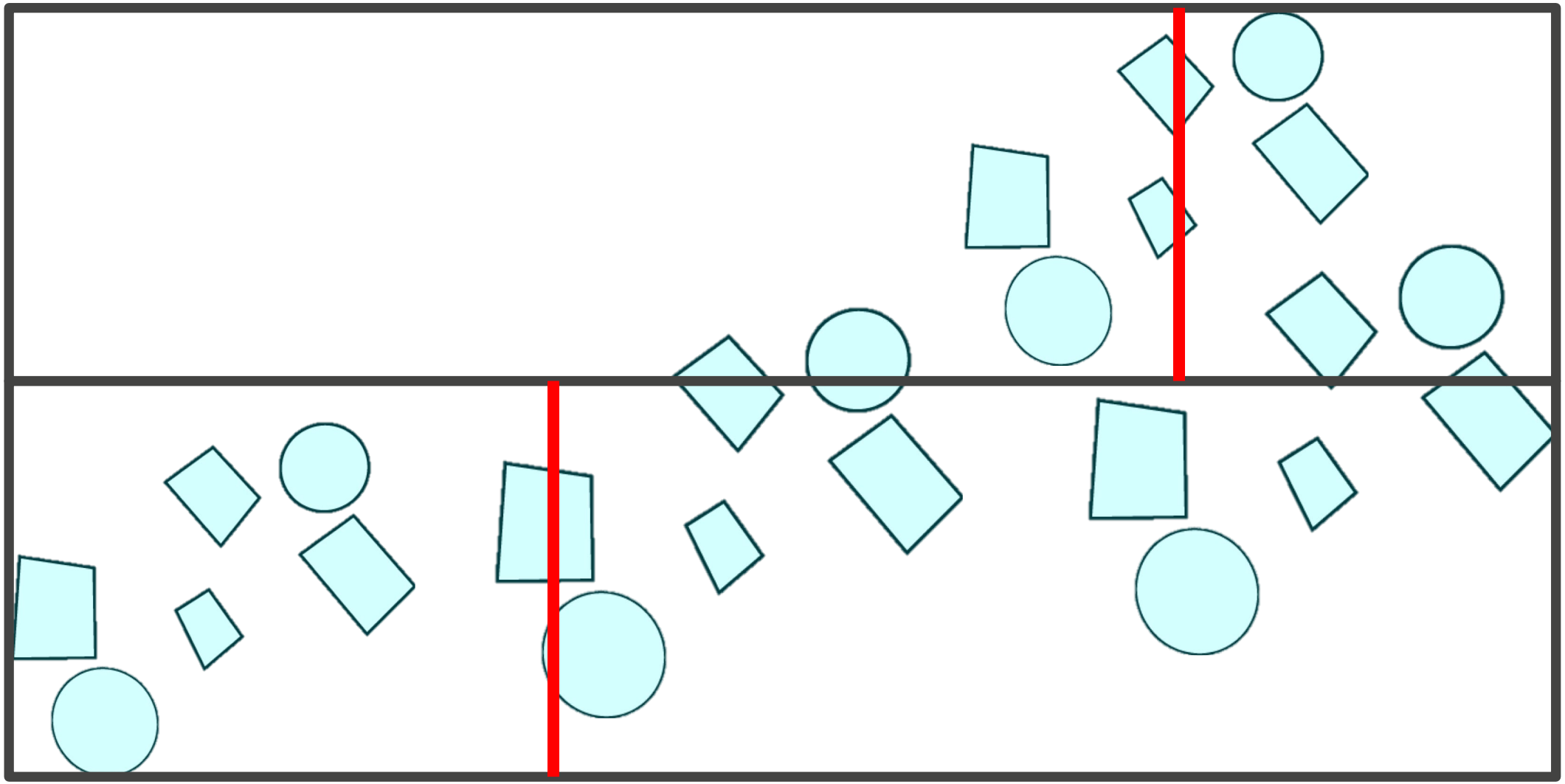
Constructing a k-d Tree



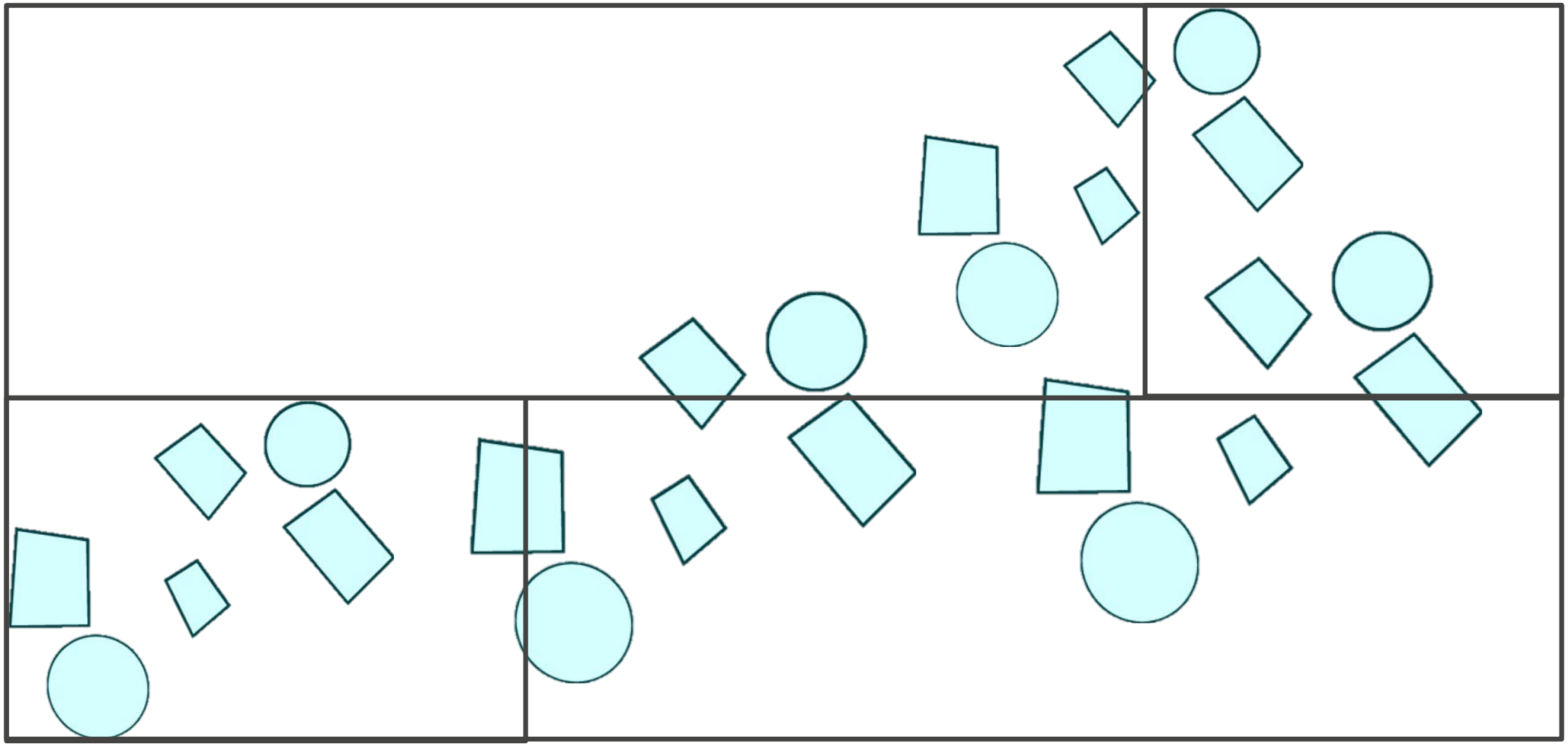
Constructing a k-d Tree



Constructing a k-d Tree



Constructing a k-d Tree



Ray Intersection Tests

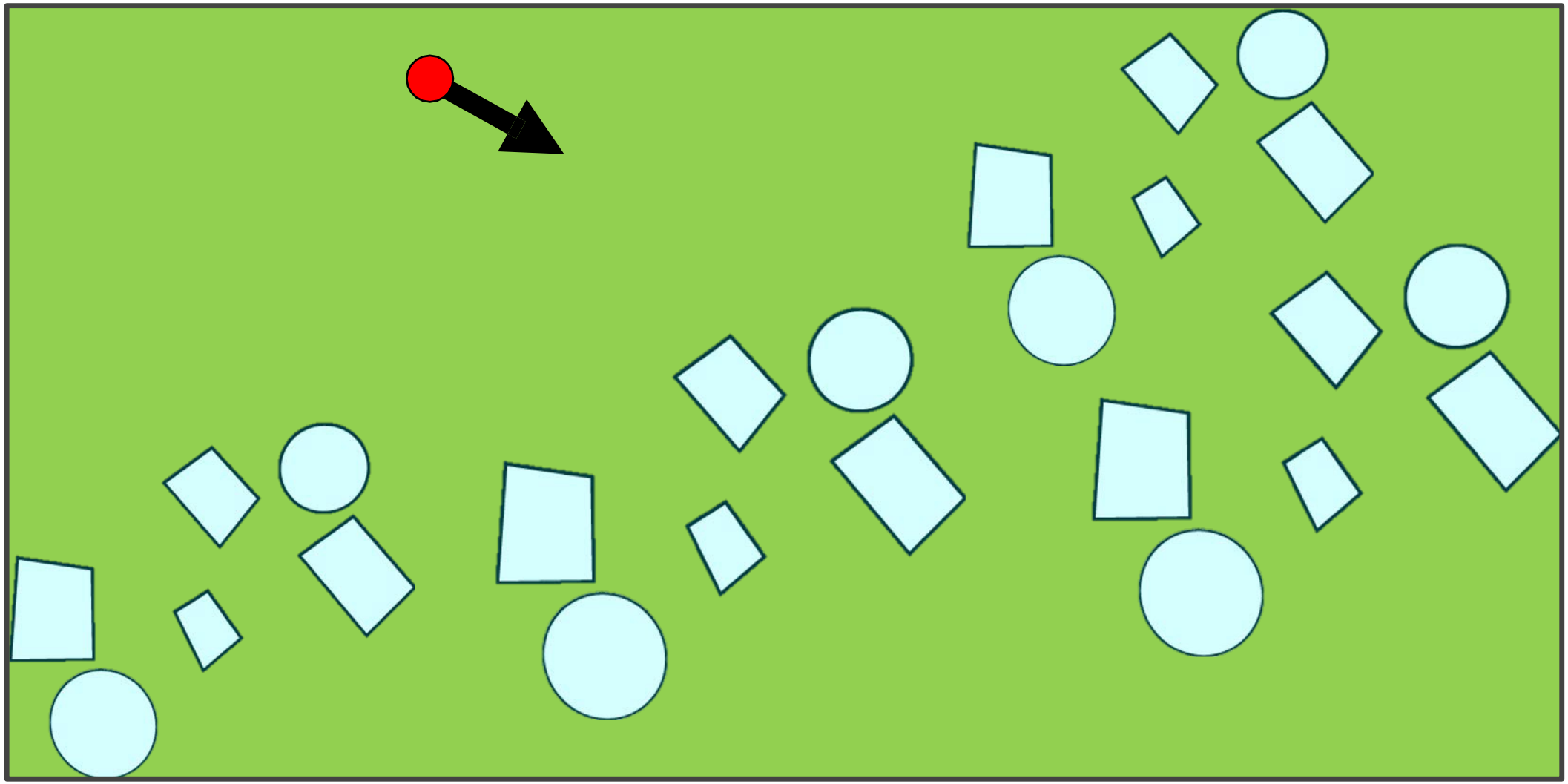
Depth First Search Again

If ray interacts with child node then recurse

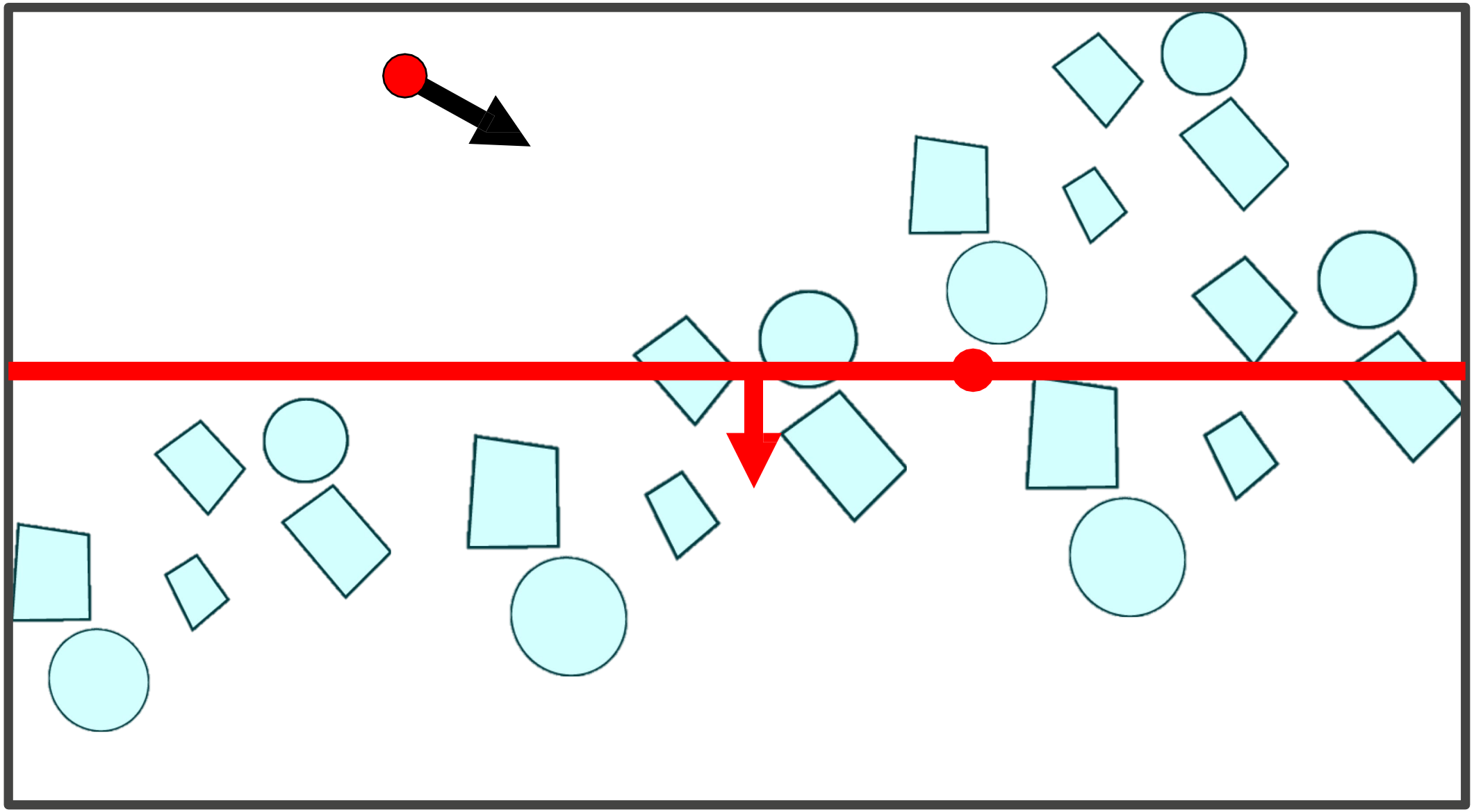
Interactions are

- Child contains ray origin point
- Ray crossed into child node

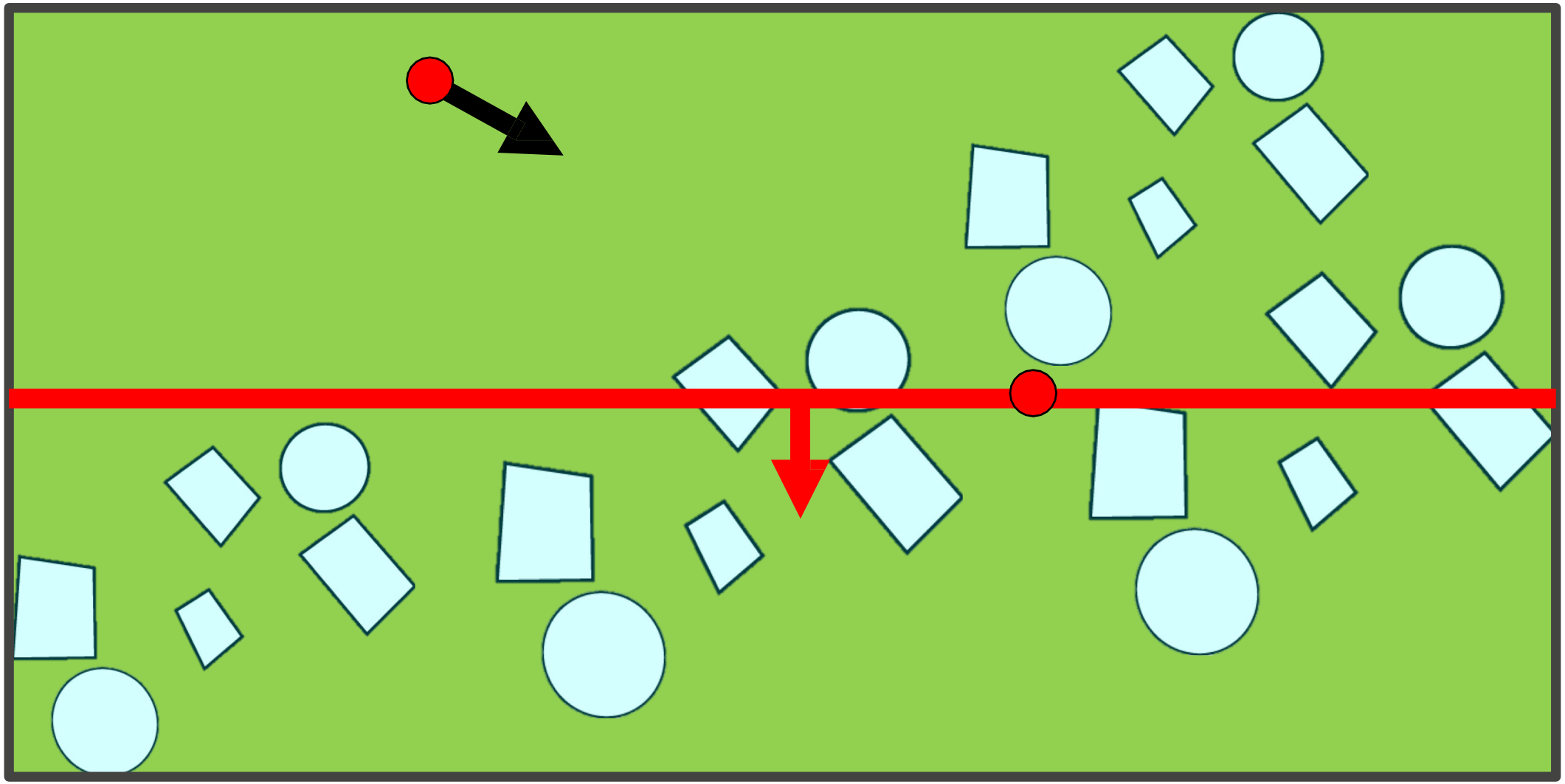
Ray Intersection Tests



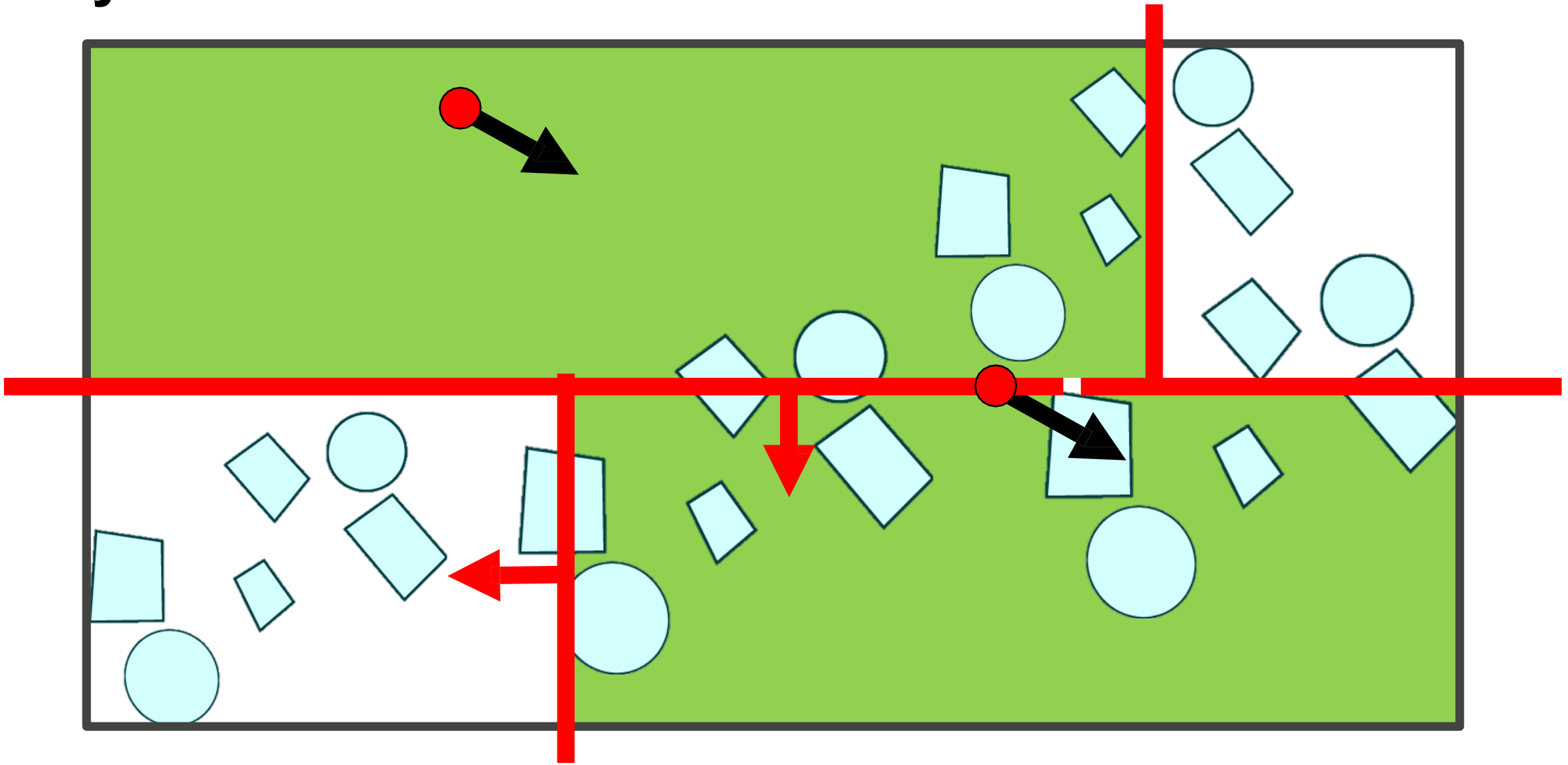
Ray Intersection Tests



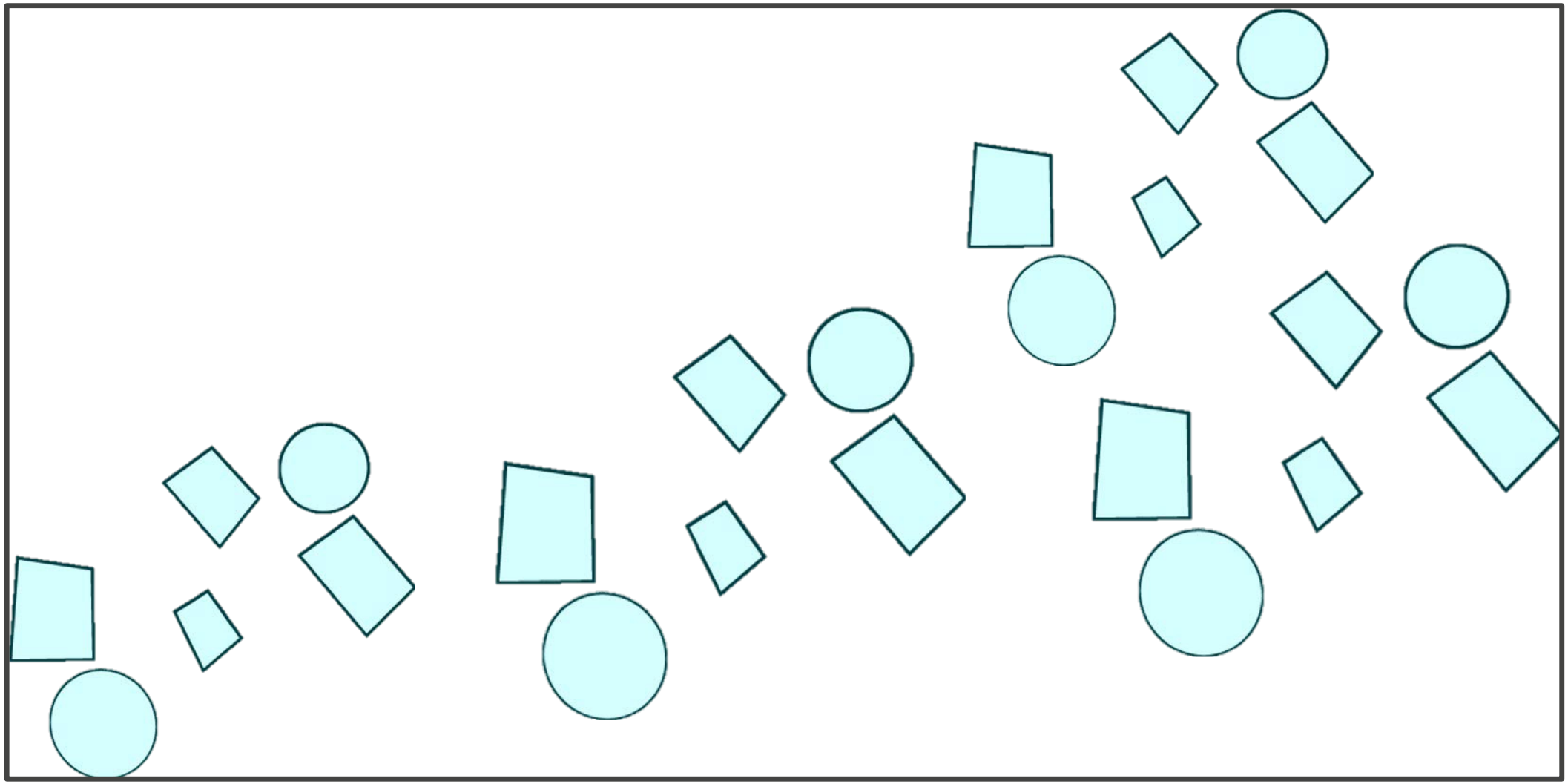
Ray Intersection Tests



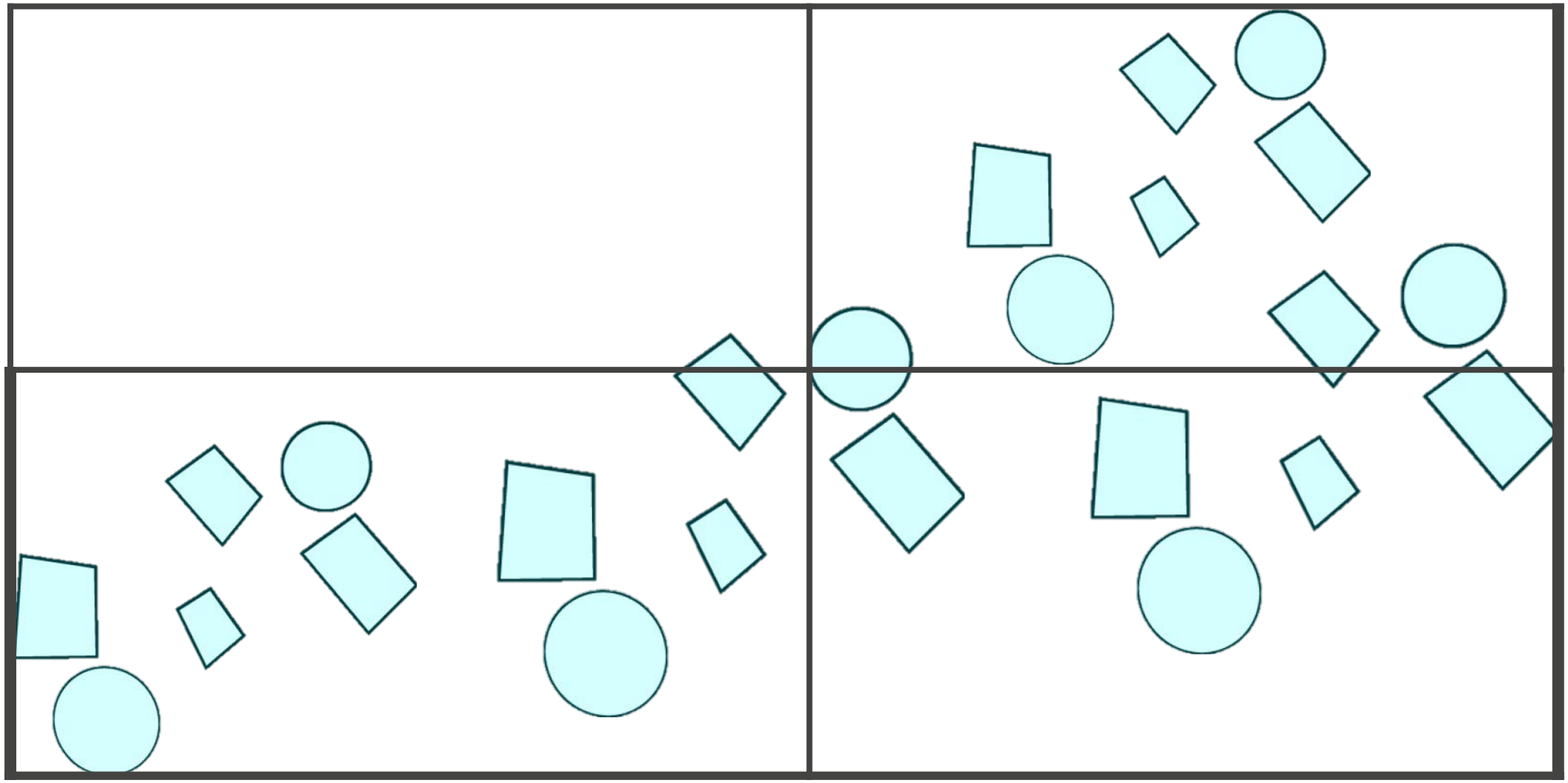
Ray Intersection Tests



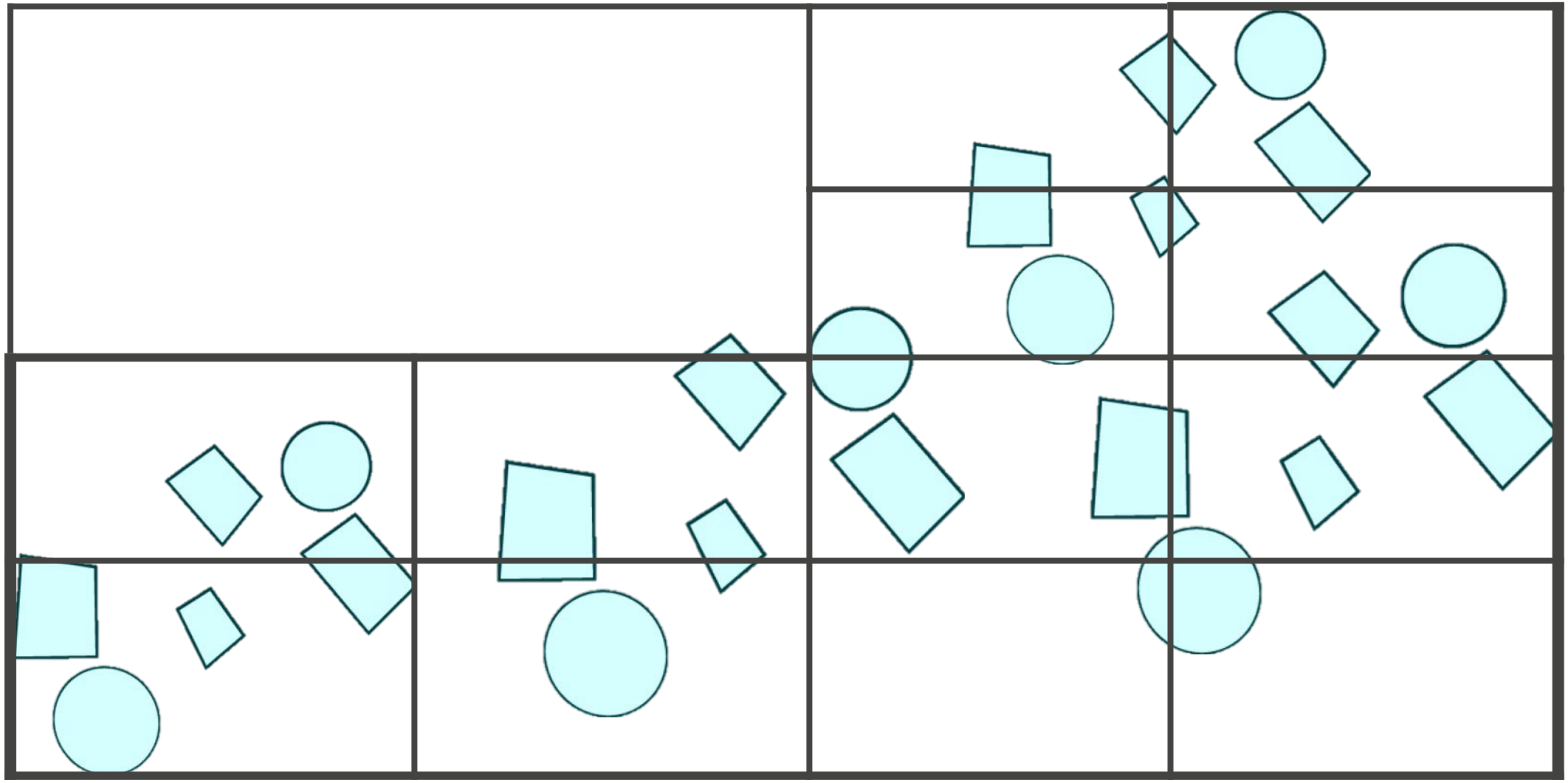
Constructing a Quadtree



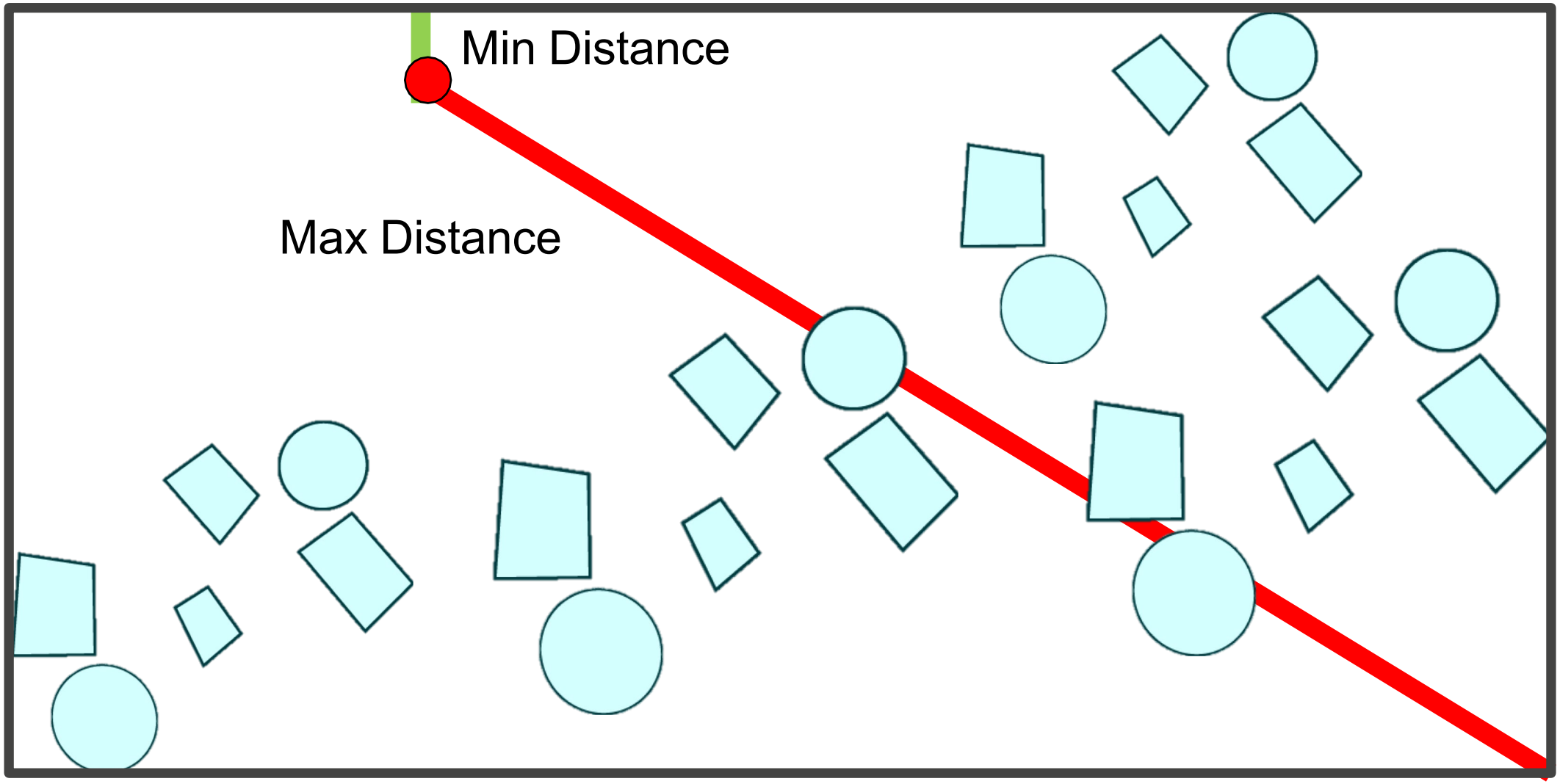
Constructing a Quadtree



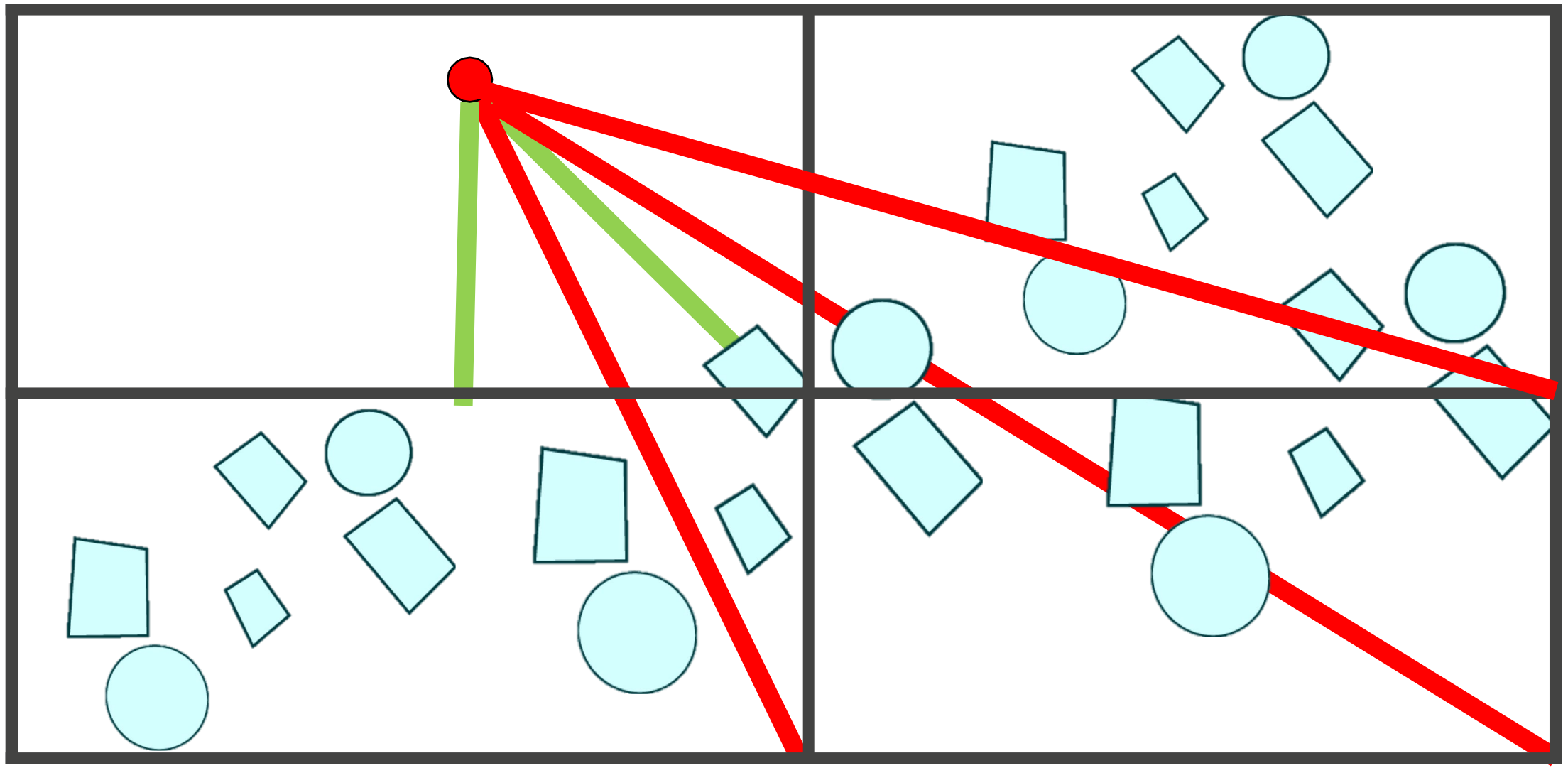
Constructing a Quadtree



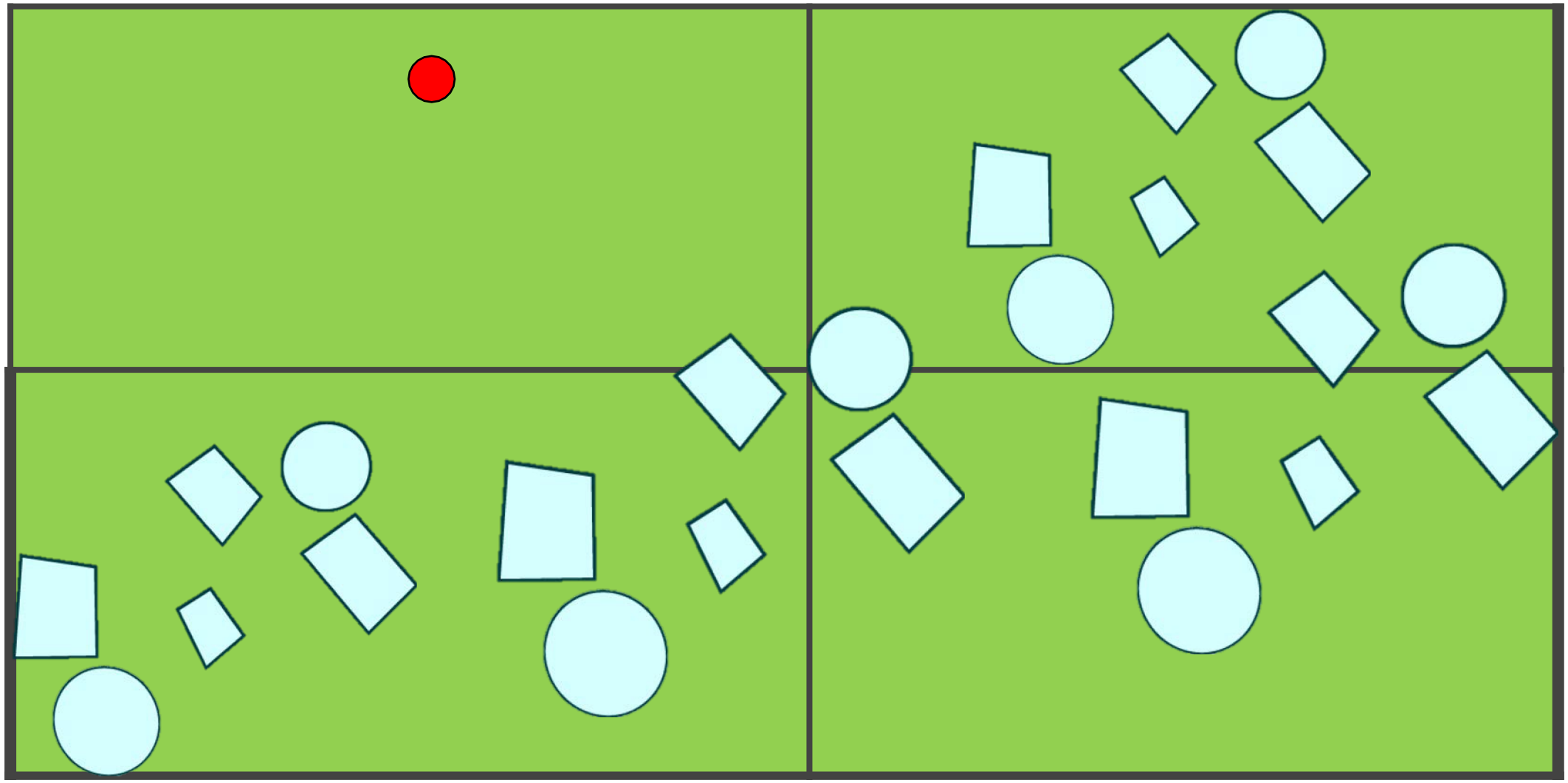
Distance Query



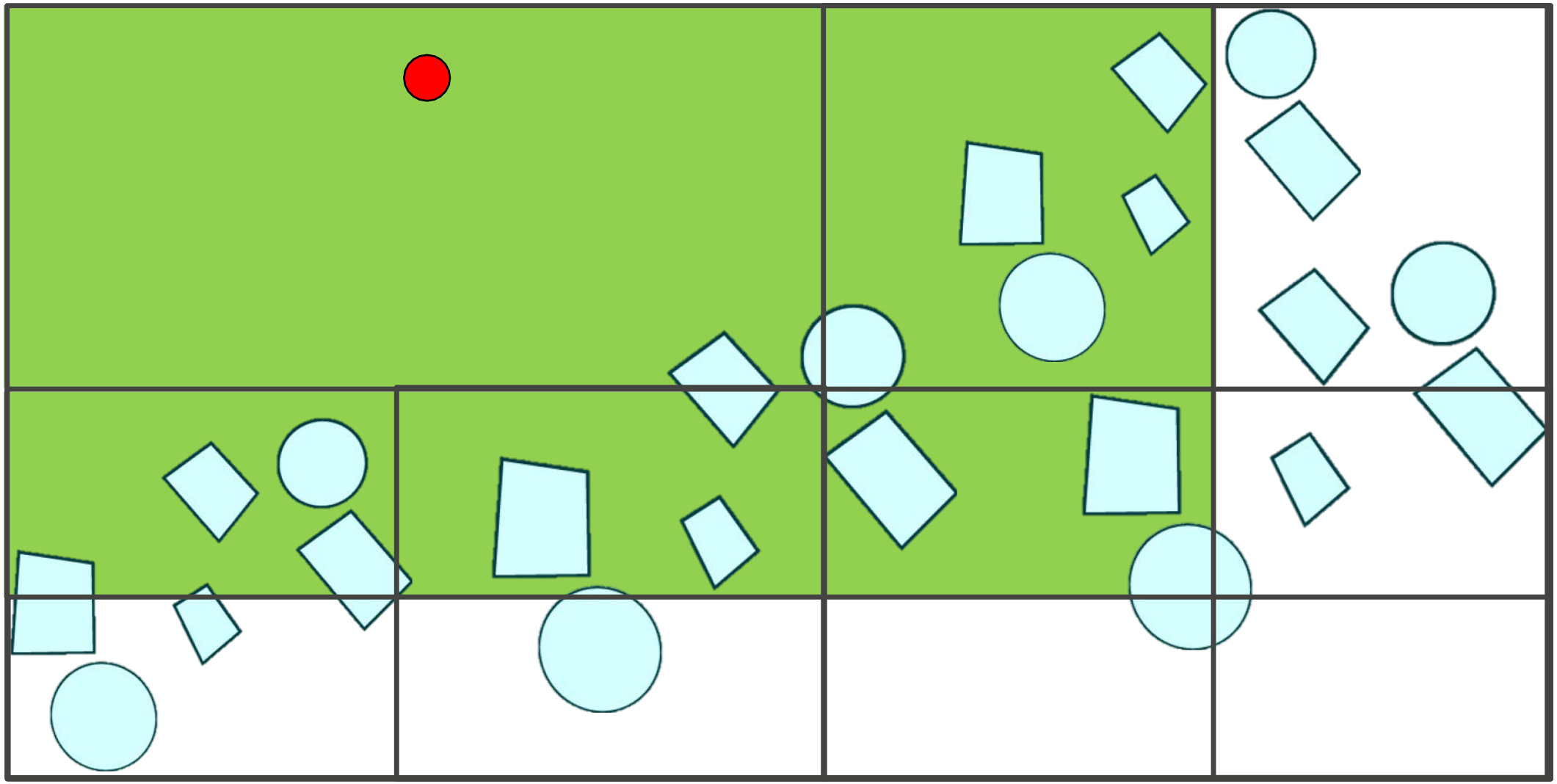
Distance Query



Distance Query



Distance Query



Done