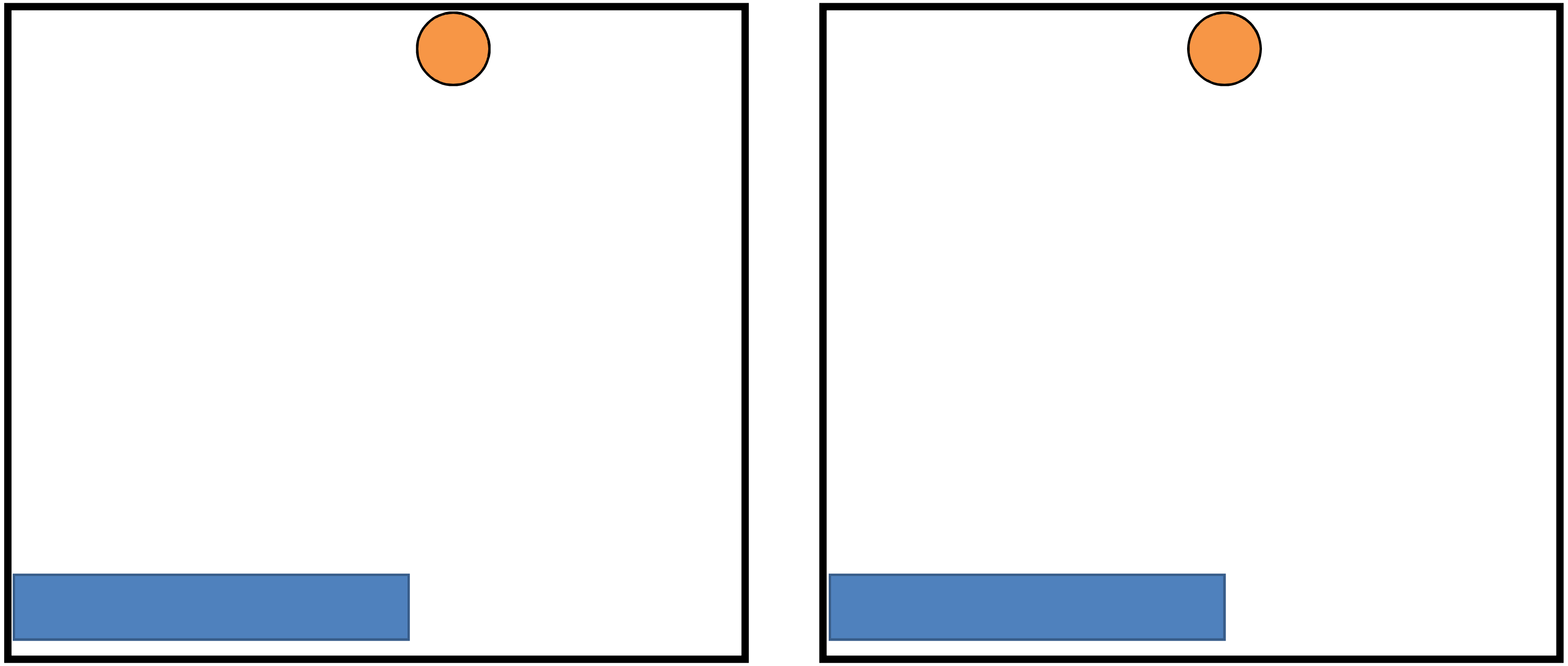
Handling Collisions

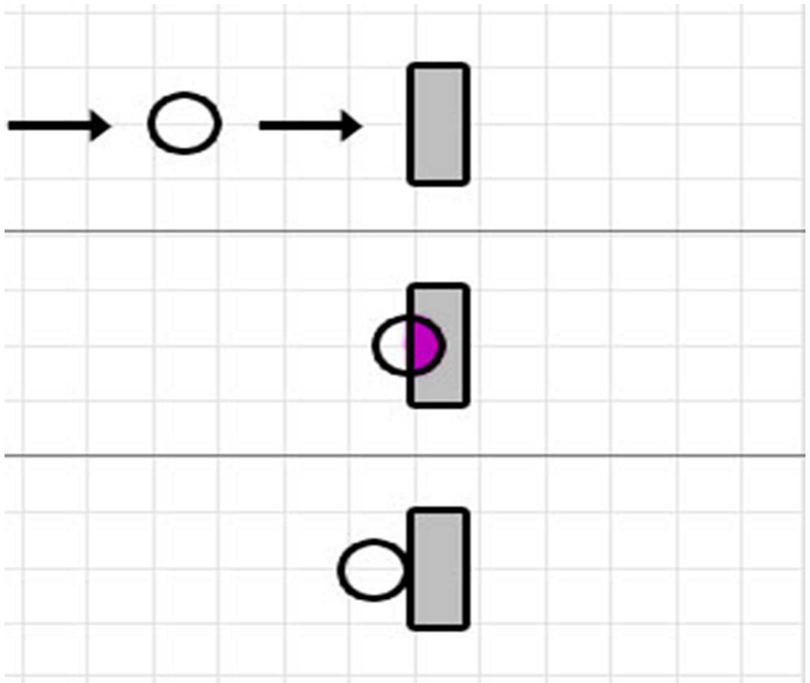
Why?

- Realisme / game play
  - Without objects pass through other objects



Three Major Parts

- Collision detection
  - Do the objects collide?
- Collision determination
  - Where do they collide?
- Collision response
  - What happens now?



Three Major Parts

- Collision detection
  - Do the objects collide?

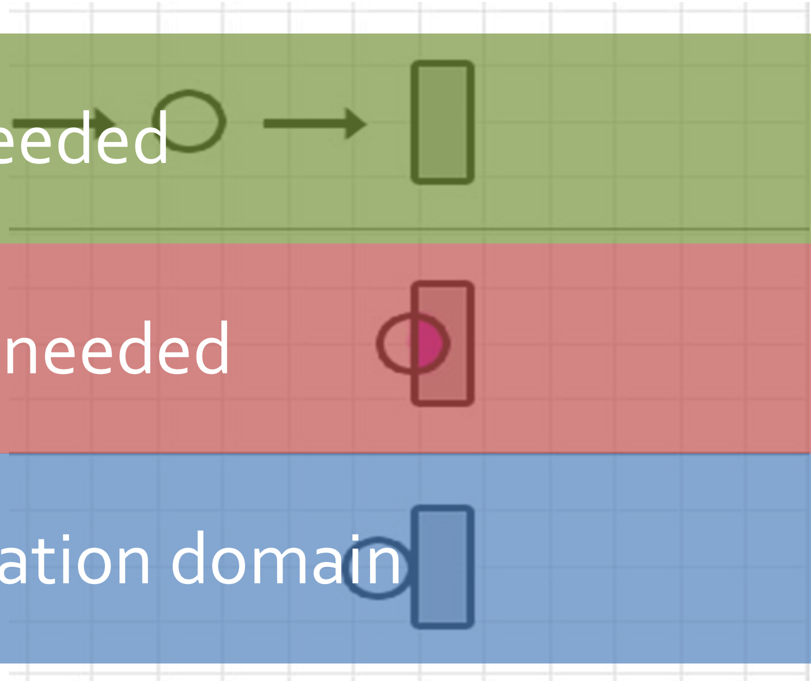
Always needed

- Collision determination
  - Where do they collide?

Not always needed

- Collision response
  - What happens now?

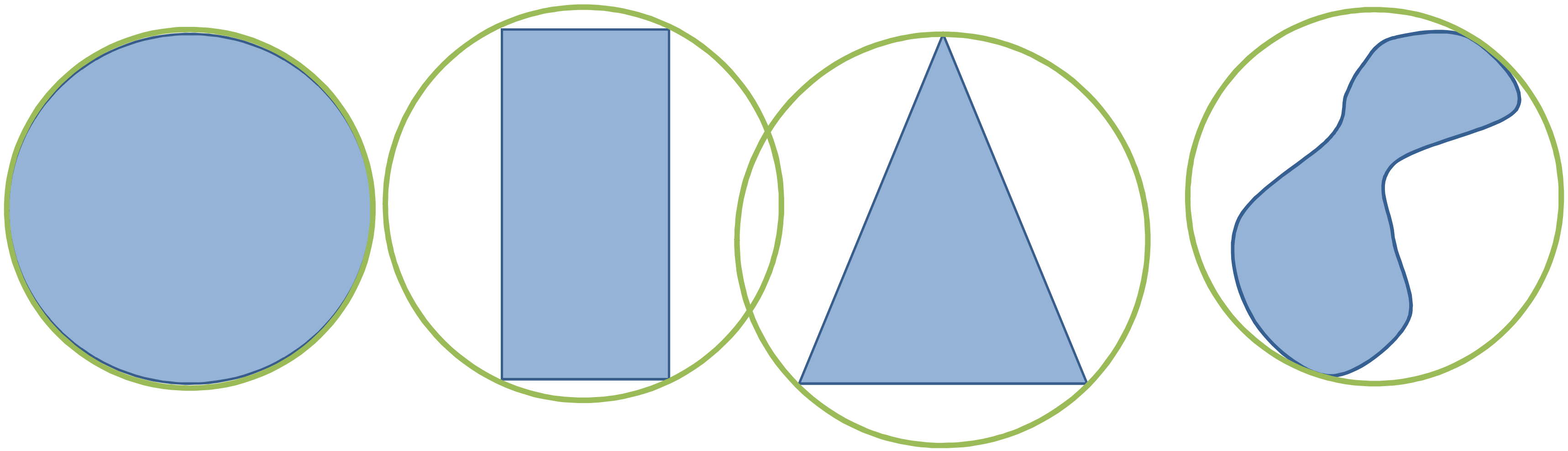
Specific to application domain



Collision Detection

Collision Detection

- Many specialized algorithms for specific geometry
  www.realtimerendering.com/intersections.html
- For complex objects a bounding geometry is used
- In games often one type of bounding geometry is used for all objects

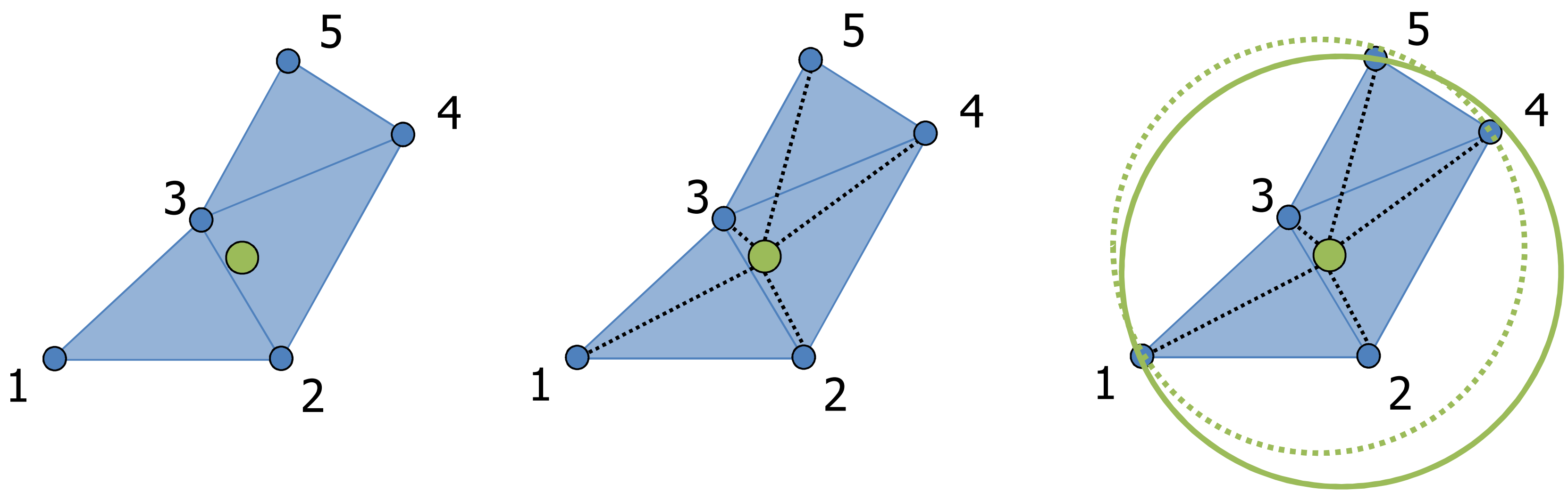


How do we find a bounding geometry?

- Calculation (again many algorithms)
- Artists defines the bounding geometry alongside the object

Bounding Sphere – Calculation

- Find the center
  - Average of all vertices
- Find radius
  - For all vertices: calculate max. distance to M
- In mathematics: minimal bounding sphere problem

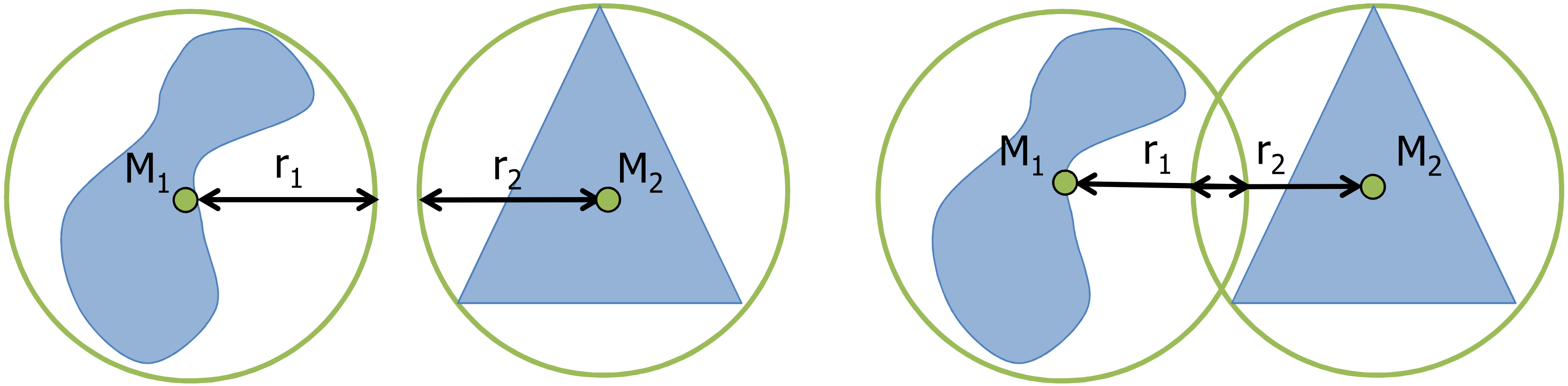


Bounding Sphere – Collision Detection

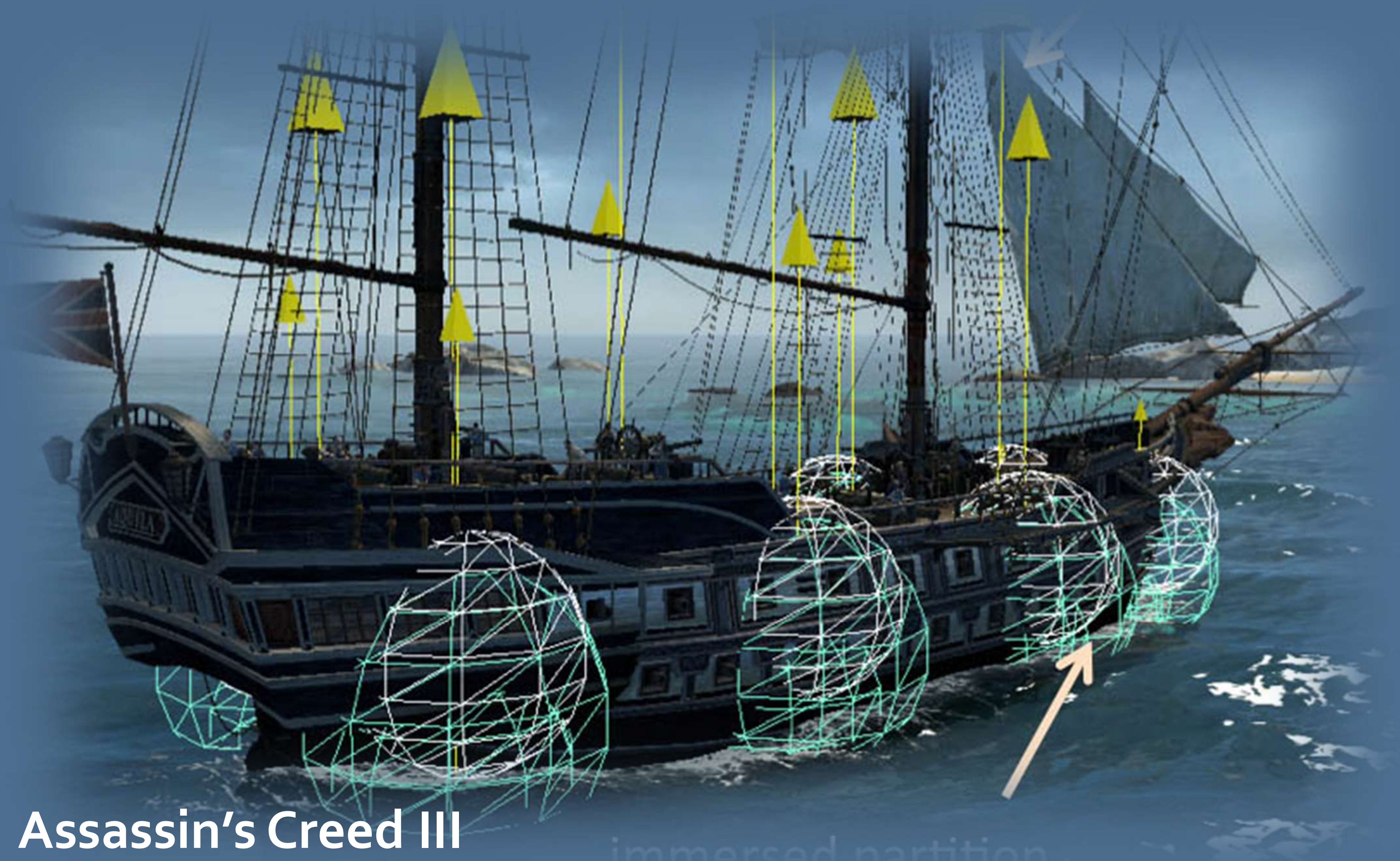
- Collision iff

$$distance(M_1, M_2) < r_1 + r_2$$

$$\Leftrightarrow distance(M_1, M_2)^2 < (r_1 + r_2)^2$$



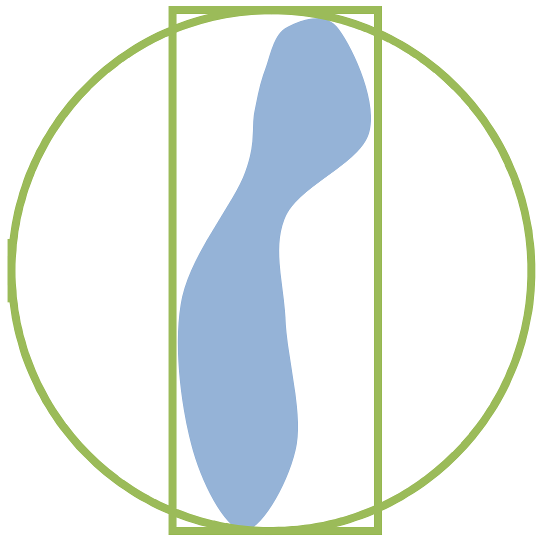
Creative Use of Bounding Spheres

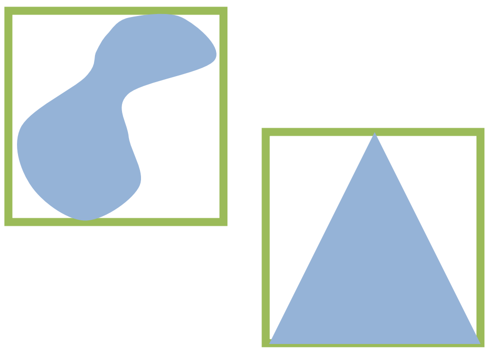


Assassin’s Creed III

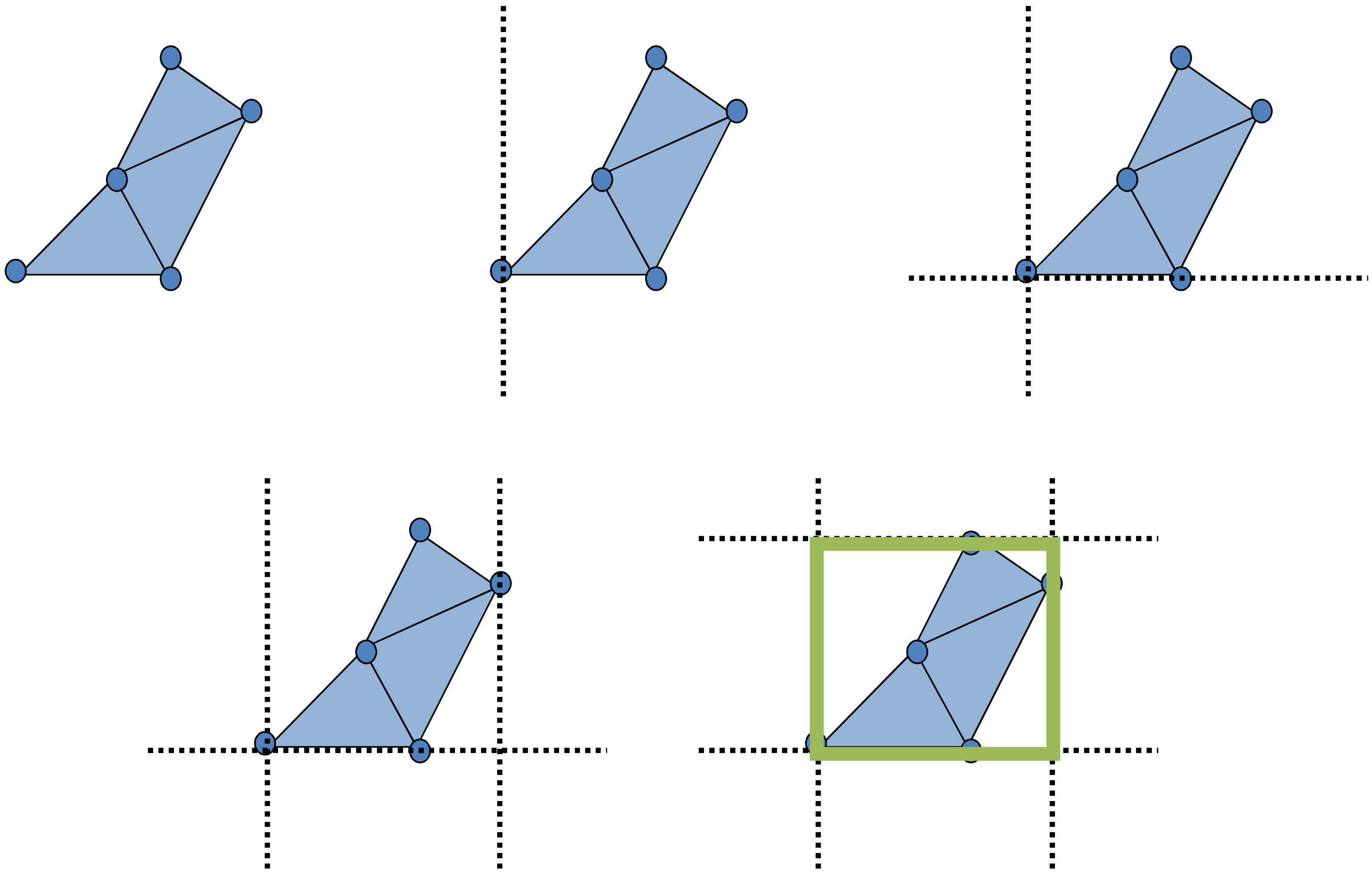
AABB-Algorithm

- Bounding-Spheres:
  - Efficient
  - Inaccurate
- Axis Aligned Bounding Boxes
  - Better fit for elongated objects
  - Only slightly more complicated



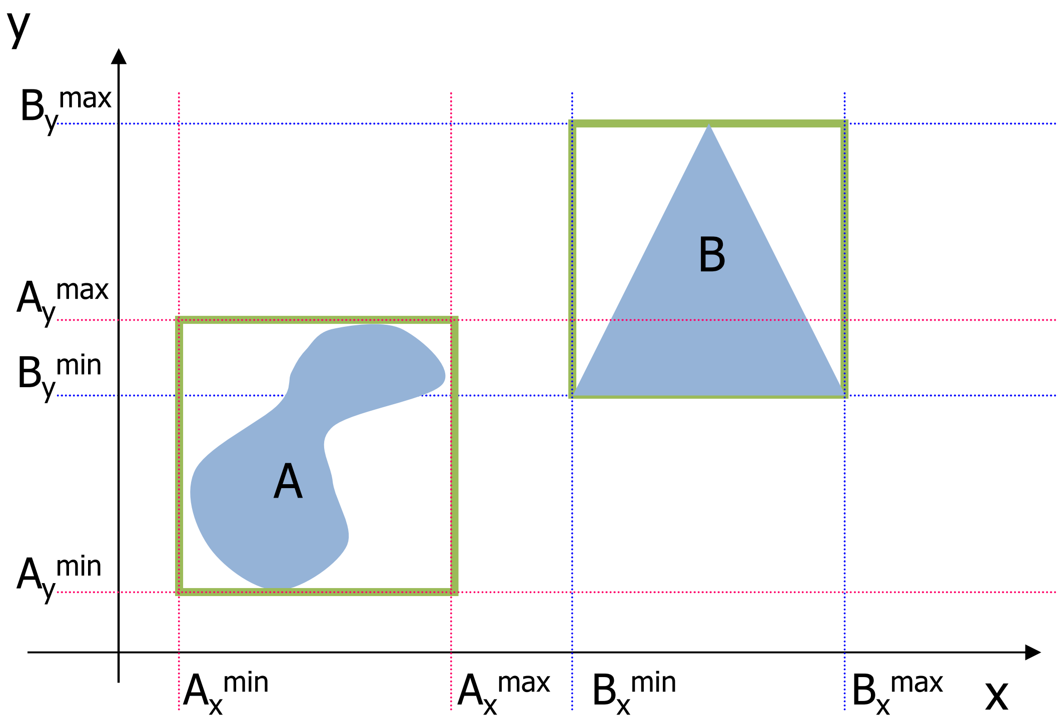


AABB – Calculation

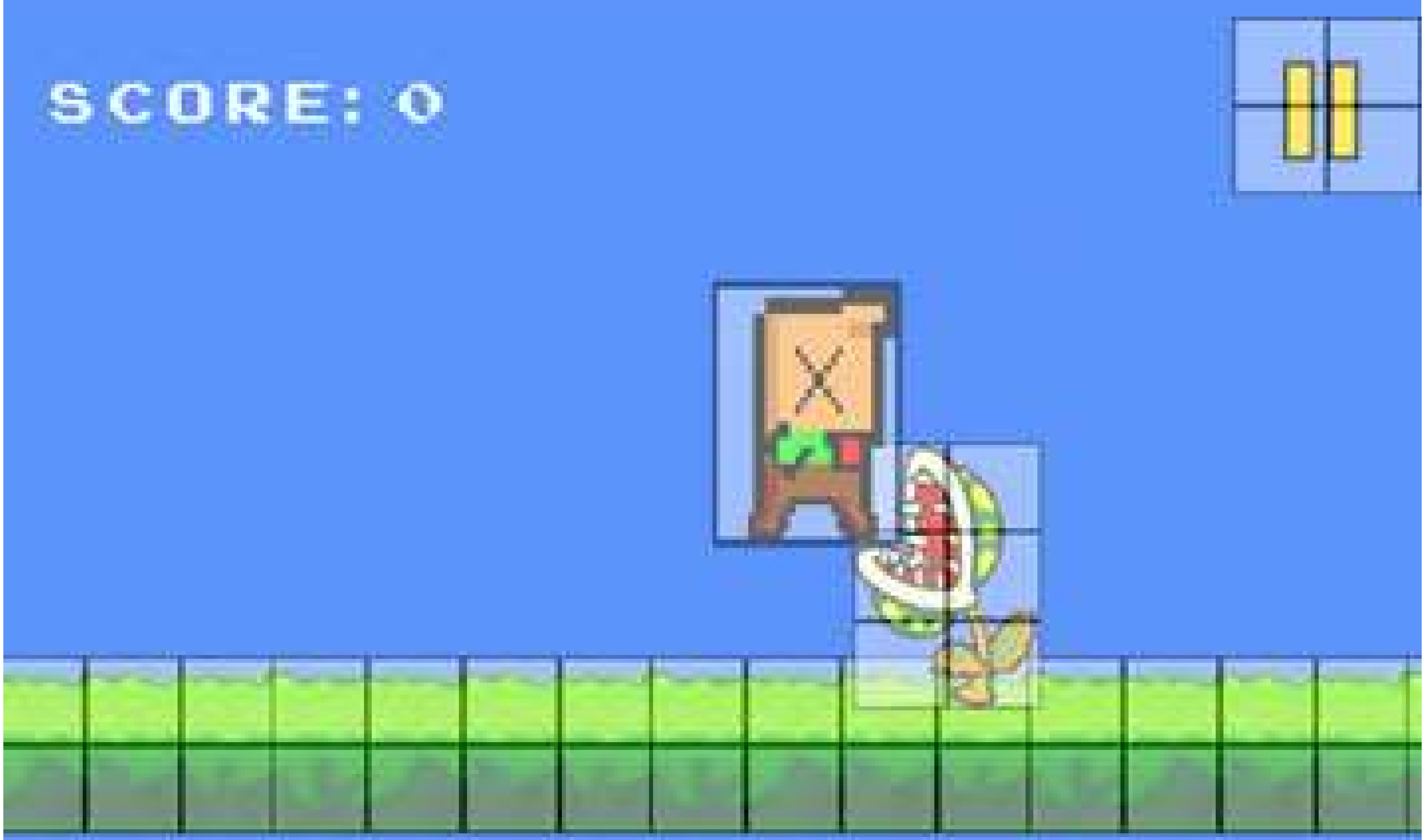


AABB-Algorithm

- No collision if
- $\exists i \in \{x, y, z\} | \left(\mathrm{A}_i^{min} > \mathrm{B}_i^{max}\right)\ or\ \left(\mathrm{B}_i^{min} > \mathrm{A}_i^{max}\right)$
  - Separating axis theorem

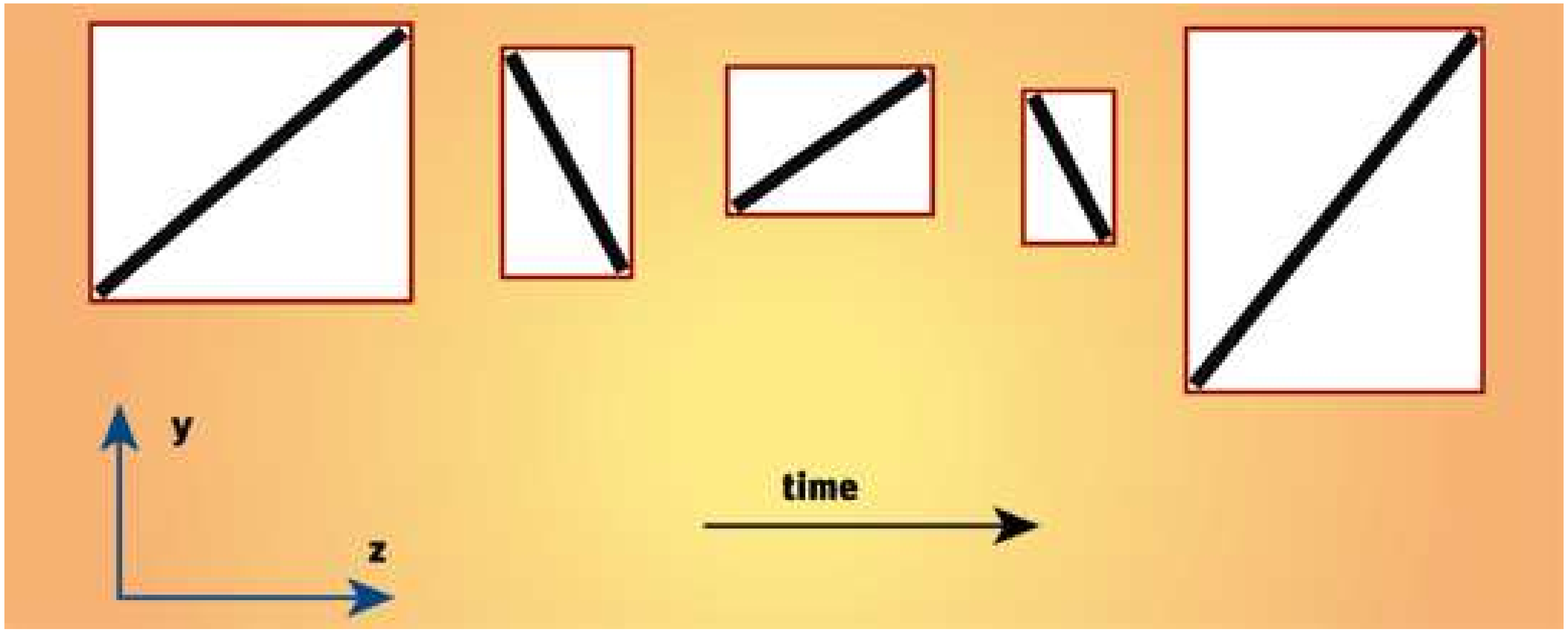


AABB Creative use



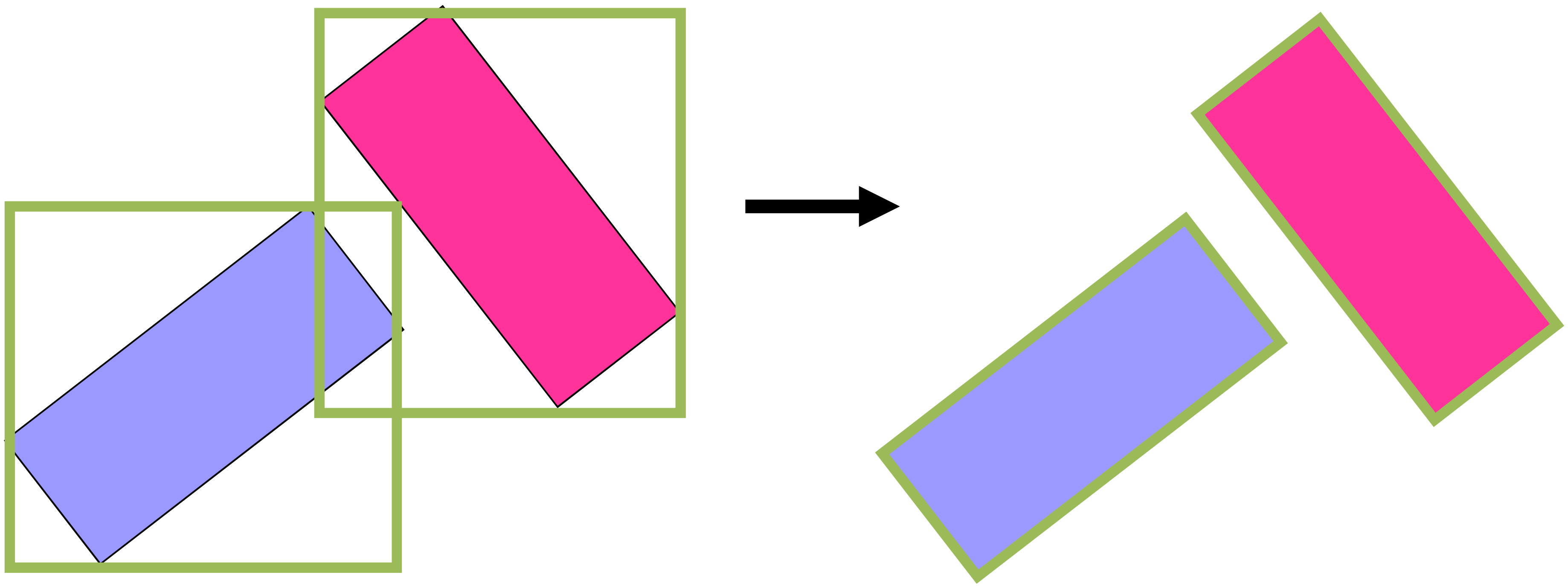
AABB - Problems

- While rotating an object, we have to recalculate the bounding box



Oriented Bounding Box

- Which problems do we have using the AABB approach?
  - *SIGGRAPH 1996, Gottschalk et al.*

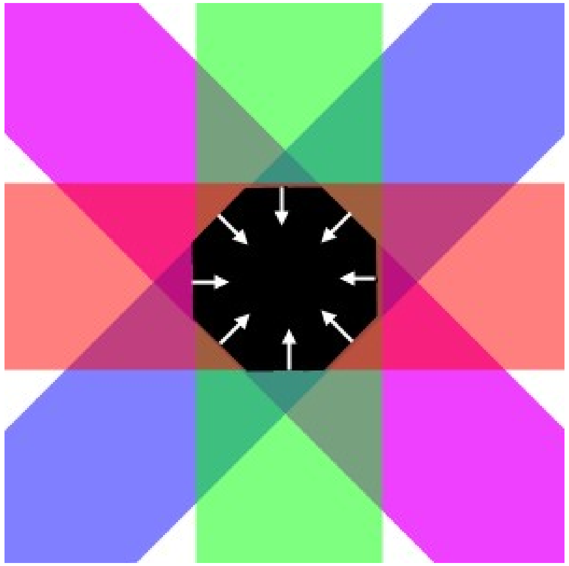


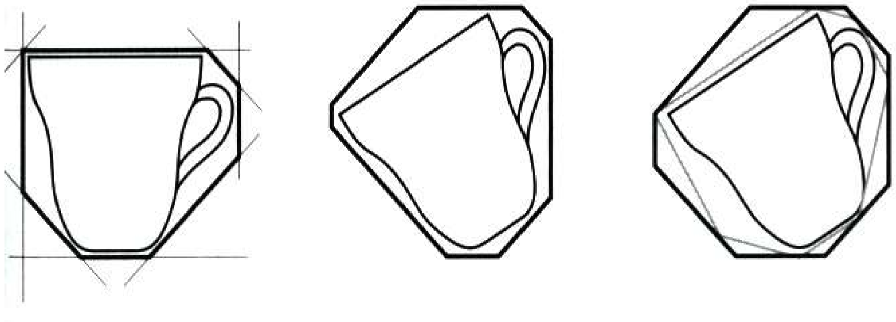
Oriented Bounding Box

- Rotation is no further a problem
- 95% of the situations are solved
- More complicated to calculate than AABB
- Separating axis theorem still works
- More math involved
- Find more information under
  - www.gamasutra.com
  - Game Programming Gems (I, II, III)

k-DOP

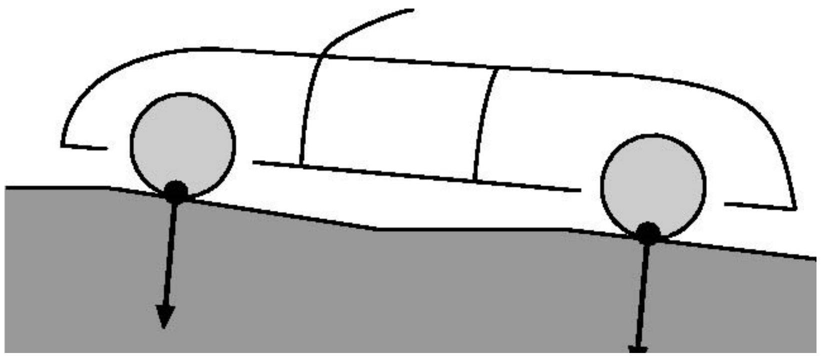
- k-Discrete Oriented Polytop
- OBB and AABB are 6-DOPs
- Optimal bounding boxes





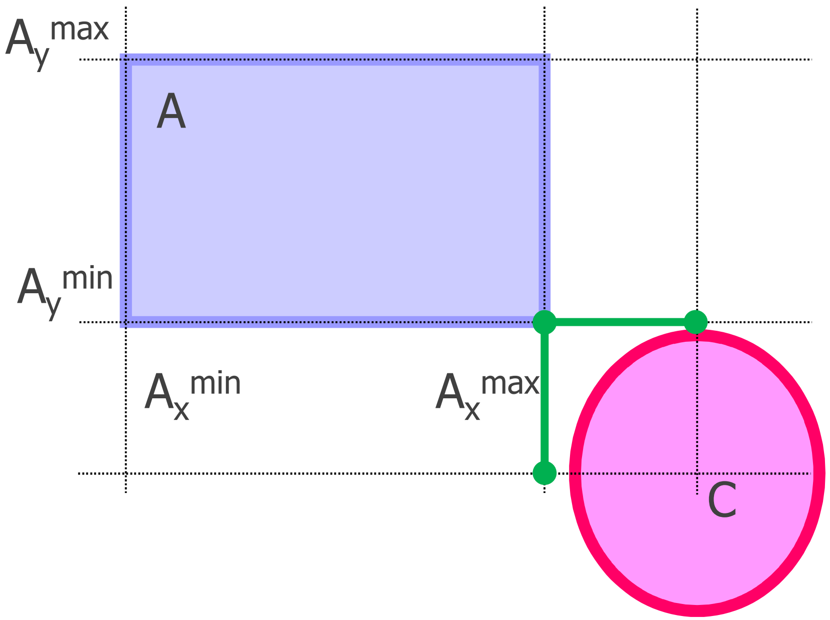
Collision Detection with Rays

- E.x.: car on road, player on terrain
- Test all triangles of all wheels against road geometry
- Often approximation good enough
- Idea: approximate complex object with set of rays



Sphere-Box Intersection

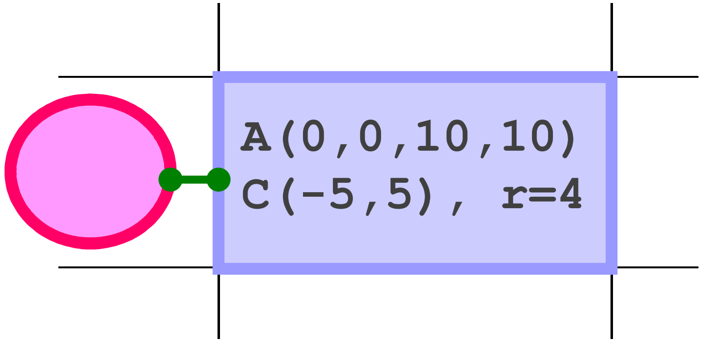
- Idea: Coordinate-wise Euclidean distance



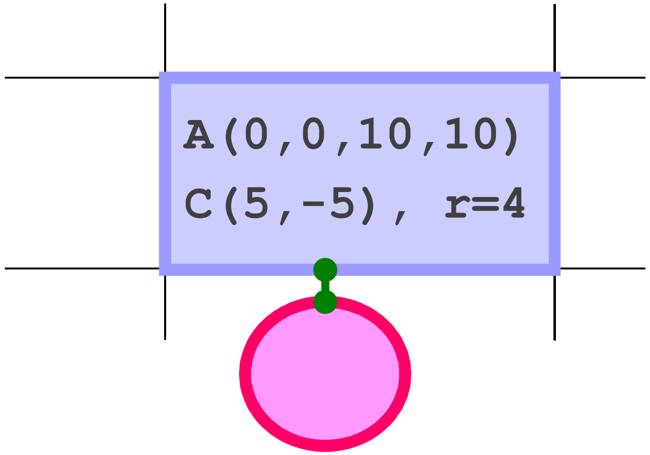

```
d = 0
for each i ∈ {x, y, z}
{
  if (C_i < A_i^min)
    d = d + (C_i − A_i^min)^2
  else if (C_i > A_i^max)
    d = d + (C_i − A_i^max)^2
}
if (d > r^2)
  return DISJOINT
else
  return OVERLAP
```

Sphere-Box Intersection

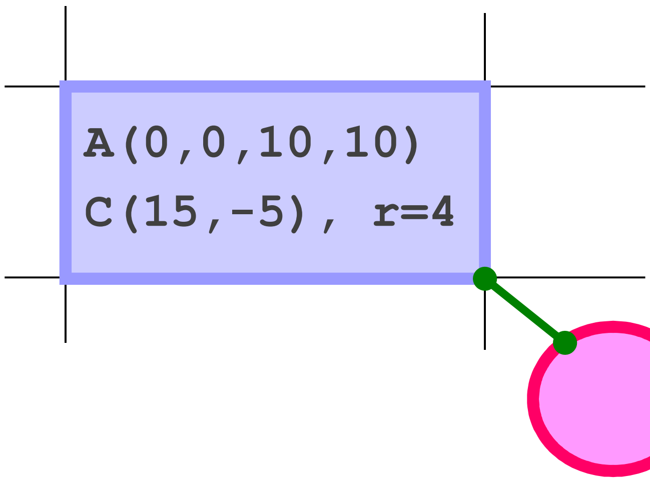
(1)



(2)



(3)

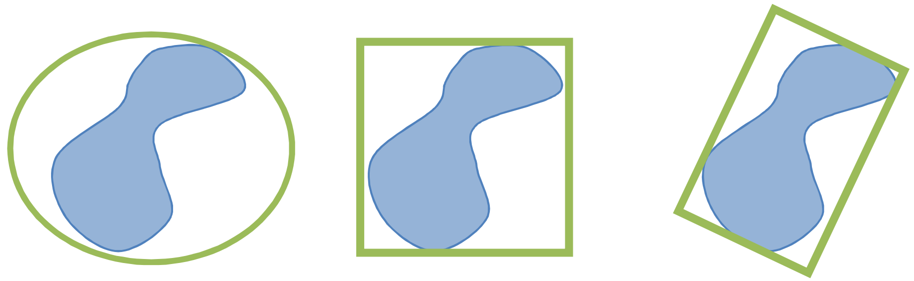


(4)

3D?

Phases

- Broad Phase (use spacial data structure for speed)
  - Grids
  - Spatial subdivisions hierarchies
  - Sweep and prune
- Narrow Phase (real object is intersected)
  - Bounding objects
  - Point-Line
  - Point-Triangle
  - Triangle-Triangle
  - ...

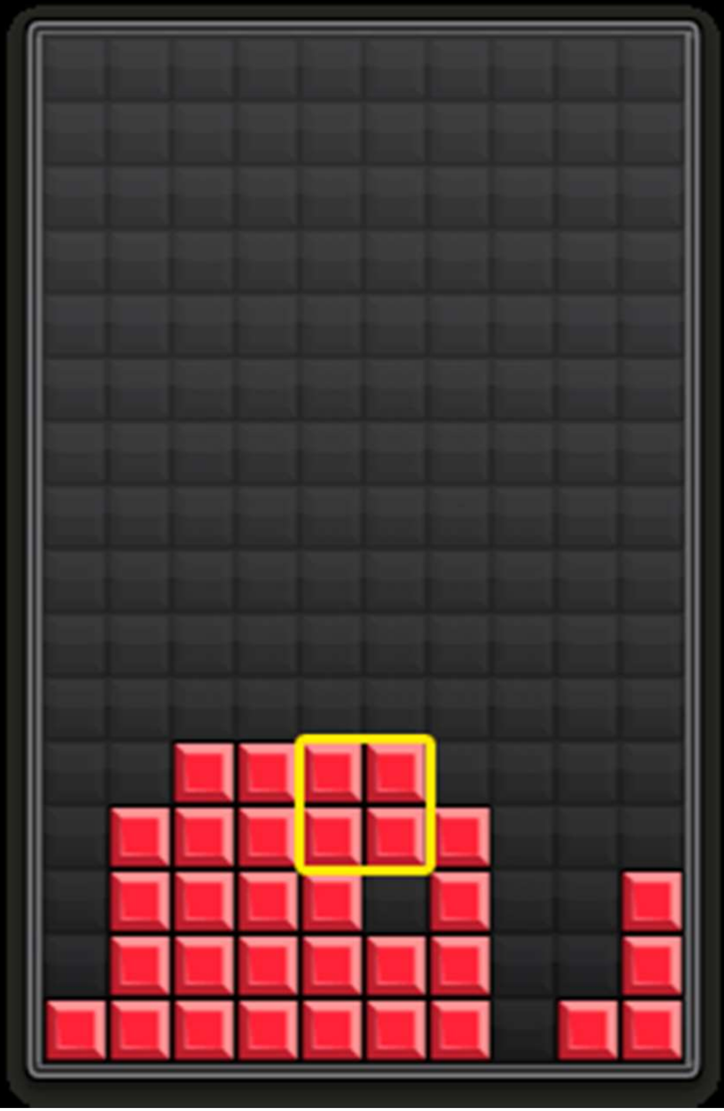


Broad Phase

Handling High Numbers of Objects

- Have to check each object with every other
  - $N \cdot (N1) \approx N^2$
- Hierarchical iregular subdivision
- Hierarchical Regular suvdivision
- Regular subdivion

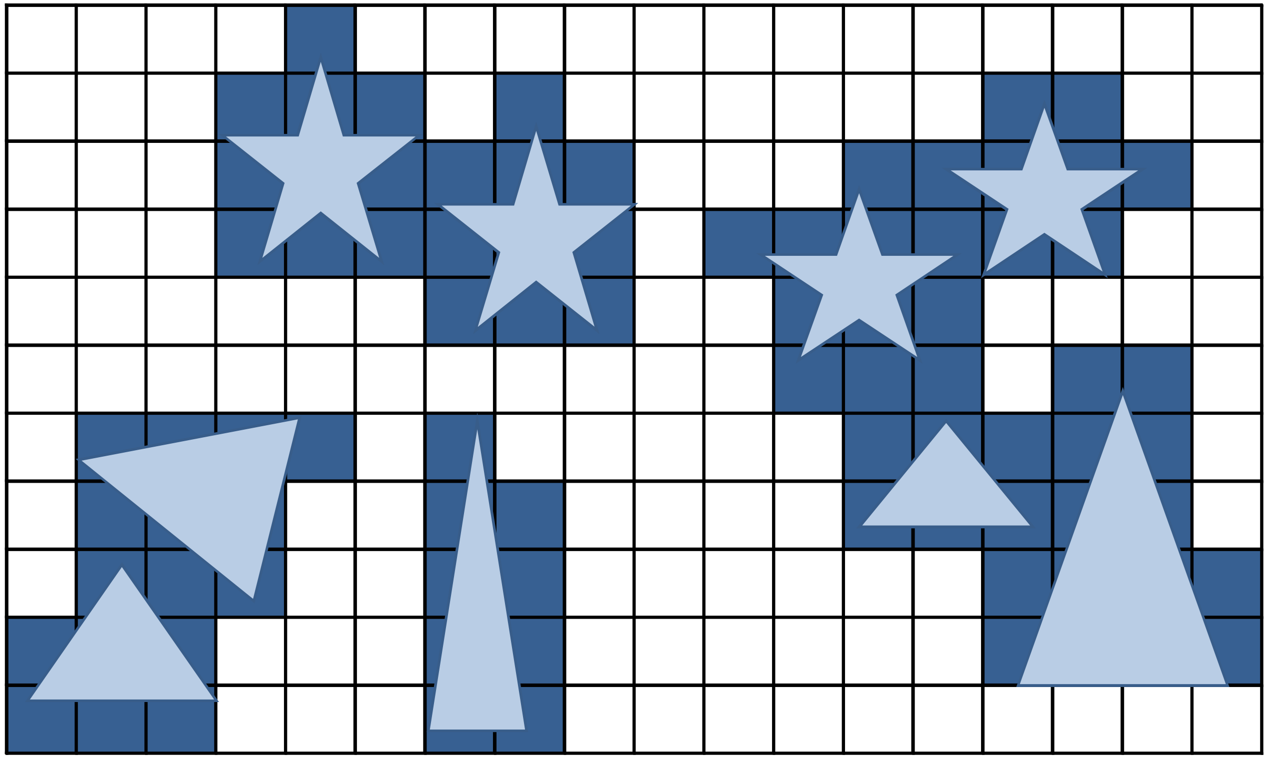
Regular Subdivision



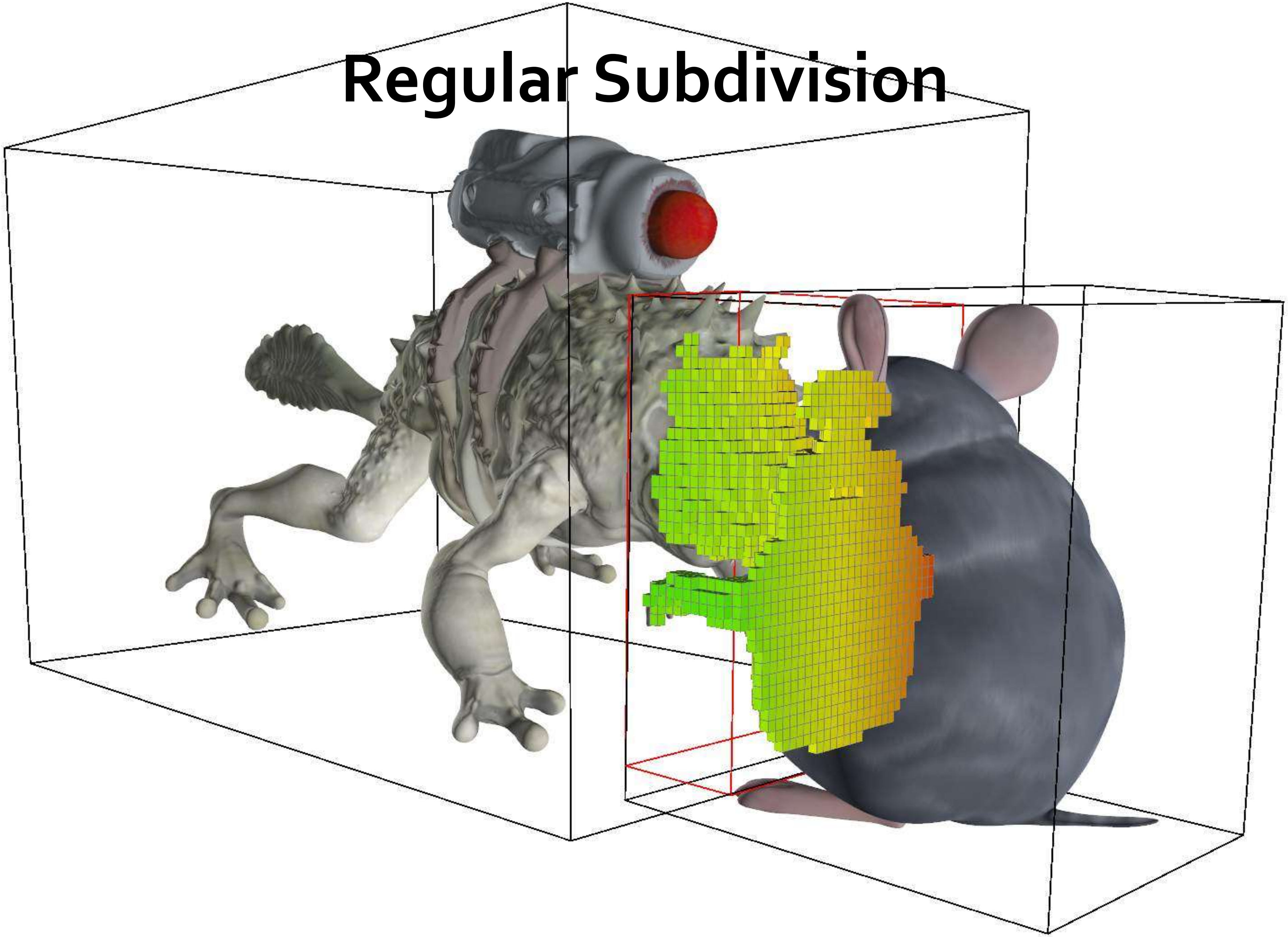
```
[[0,0,0,0,0,0,0,0,0,0],
 [0,0,0,0,0,0,0,0,0,0],
 [0,0,0,0,0,0,0,0,0,0],
 [0,0,0,0,0,0,0,0,0,0],
 [0,0,0,0,0,0,0,0,0,0],
 [0,0,0,0,0,0,0,0,0,0],
 [0,0,0,0,0,0,0,0,0,0],
 [0,0,0,0,0,0,0,0,0,0],
 [0,0,0,0,0,0,0,0,0,0],
 [0,0,0,0,0,0,0,0,0,0],
 [0,0,0,0,0,0,0,0,0,0],
 [0,0,1,1,0,0,0,0,0,0],
 [0,1,1,1,0,0,1,0,0,0],
 [0,1,1,1,1,0,1,0,0,1],
 [0,1,1,1,1,1,1,0,0,1],
 [1,1,1,1,1,1,1,0,1,1]]
```

Regular Subdivision

- Test with regular grid

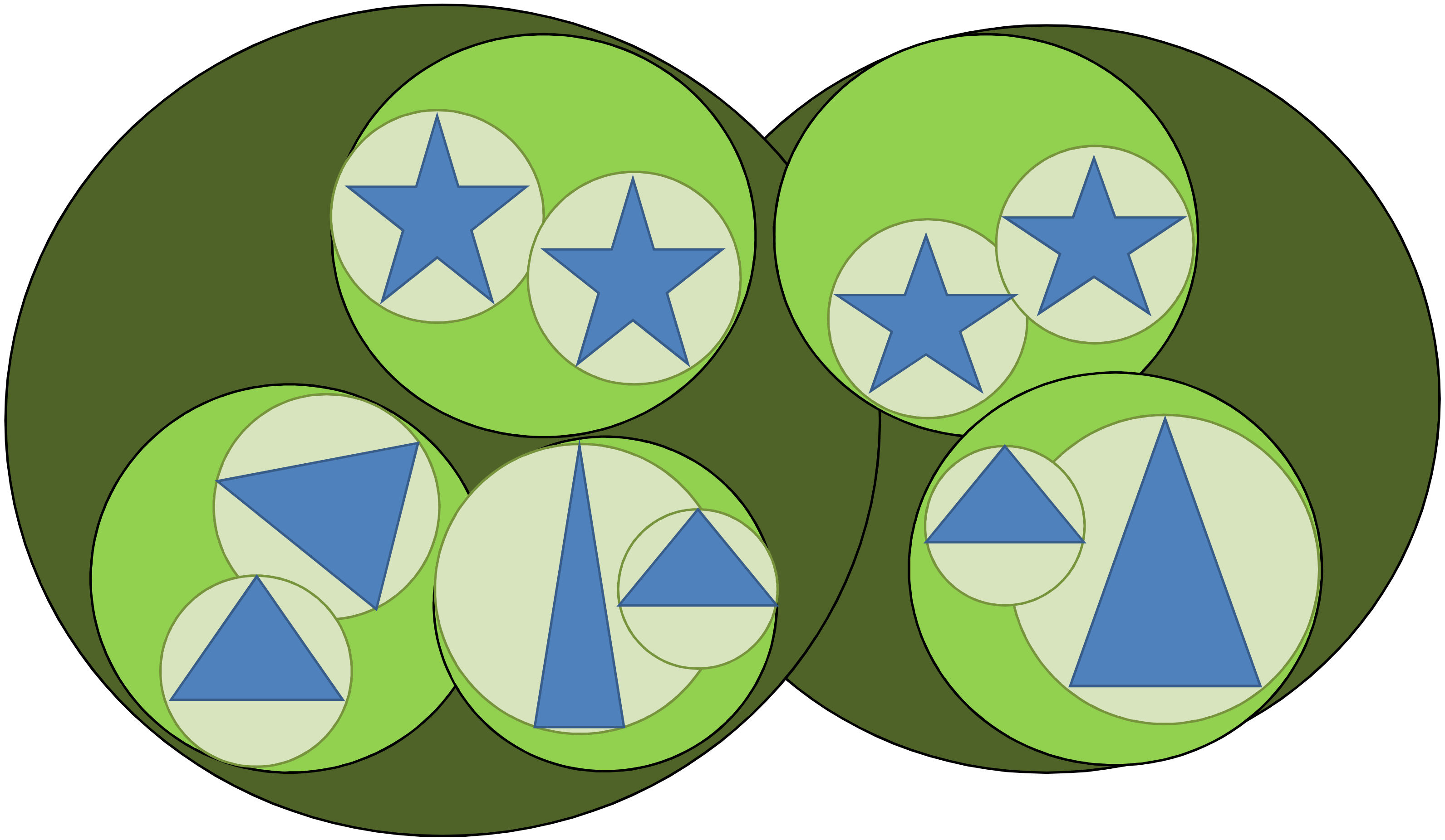


Regular Subdivision

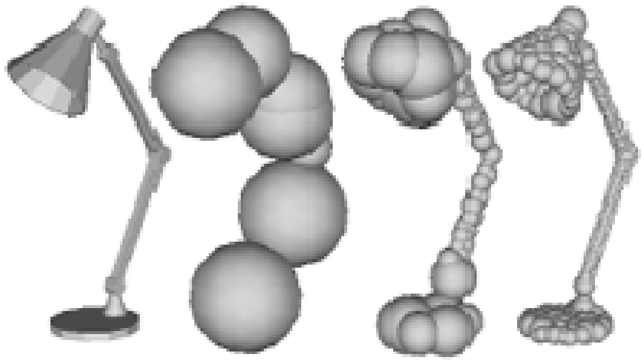


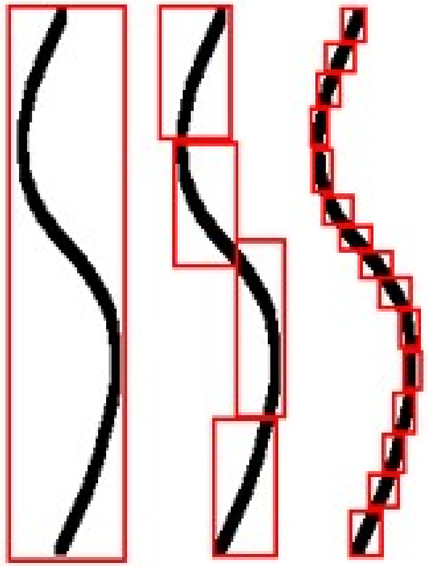
Hierarchy Trees

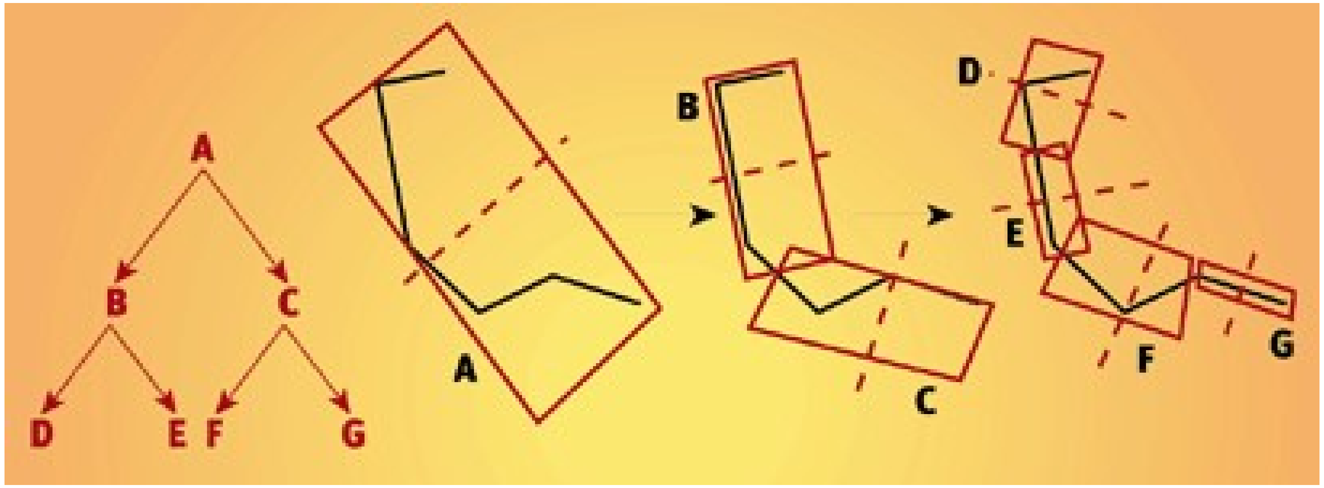
- Bounding Volume Hierarchy = BVH



Hierarchy Trees

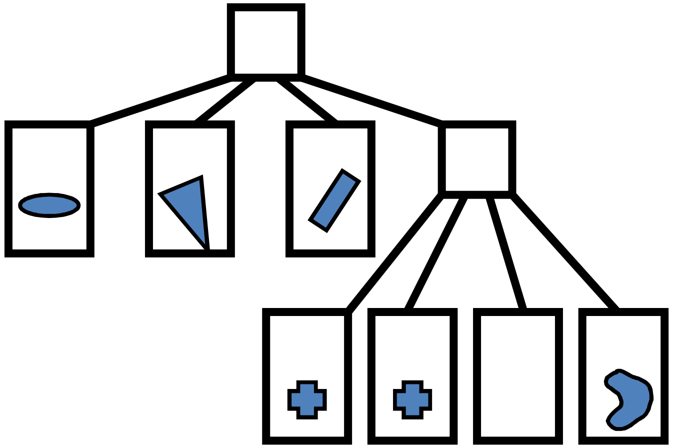


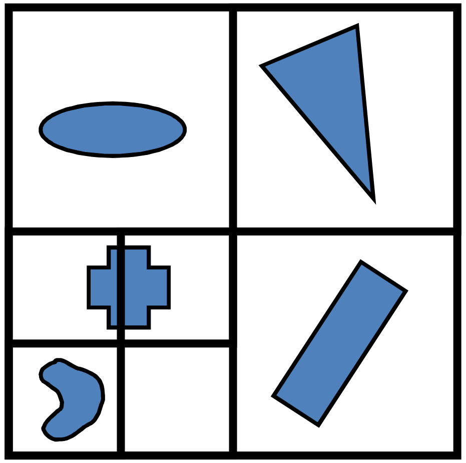




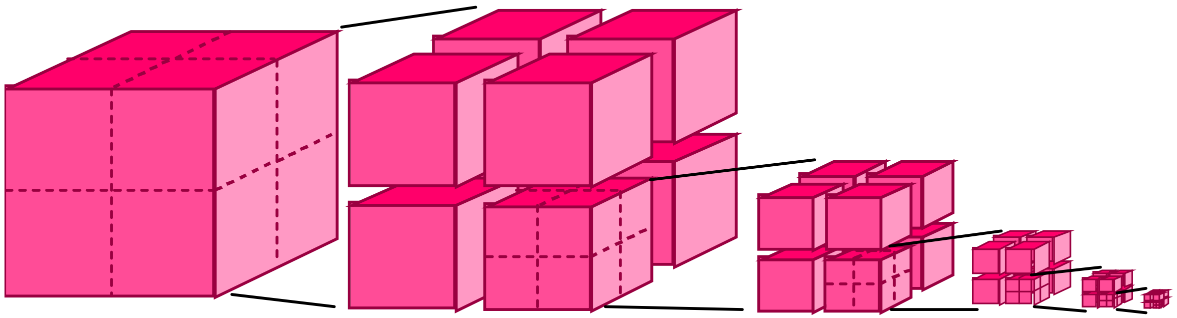
Quad/Octrees

- Quadtree (2D)



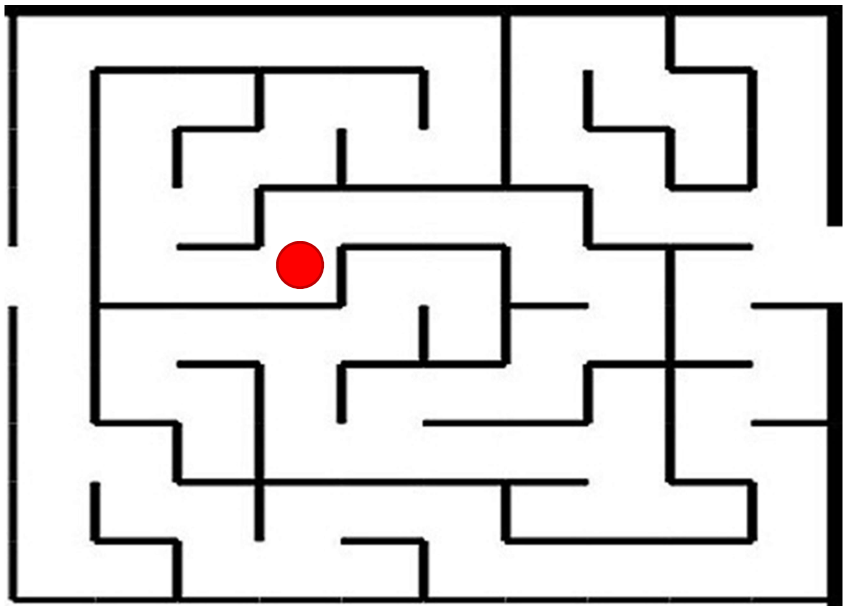


- Octree (3D)



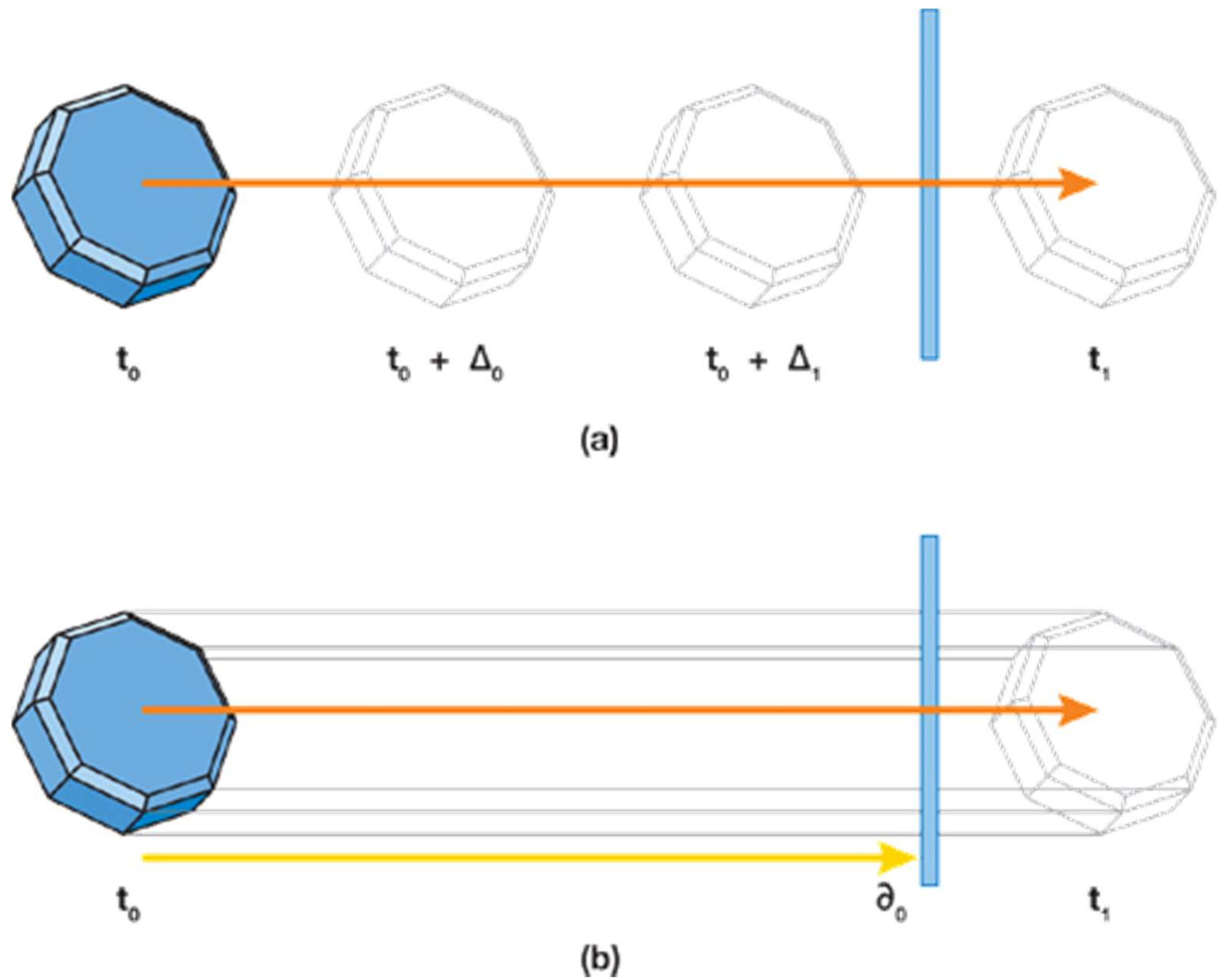
Another Simplification

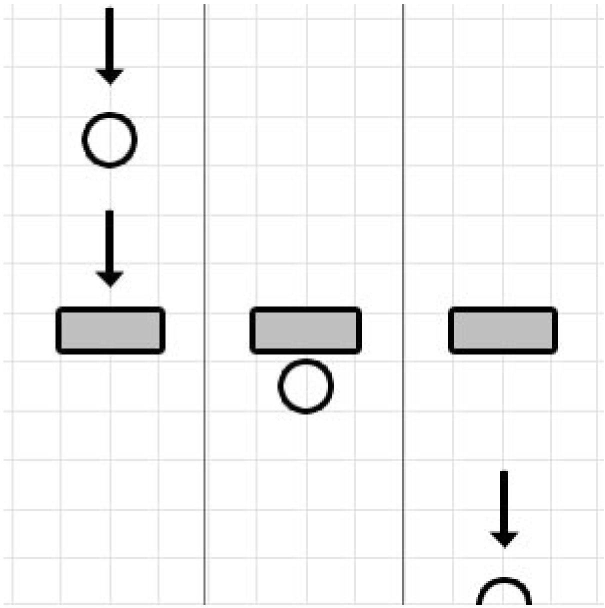
- Sometimes 3D can be turned into 2D operations
- Example: maze
- Approximate player by circle
- Test circle against lines of maze

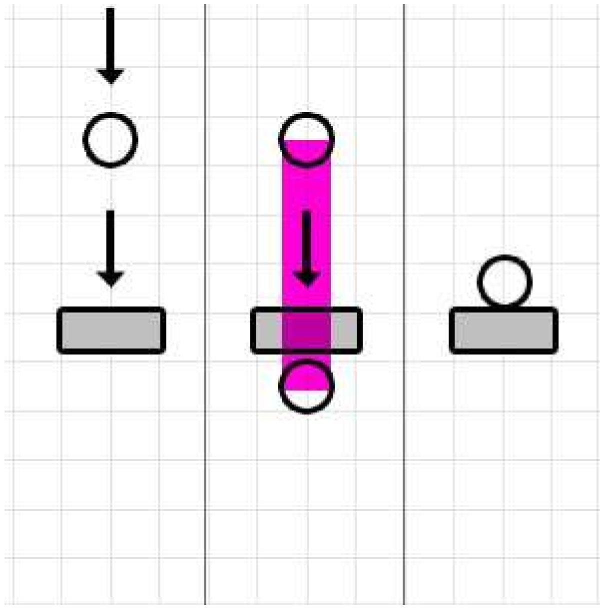


Animated Objects

Trouble with Animated Objects

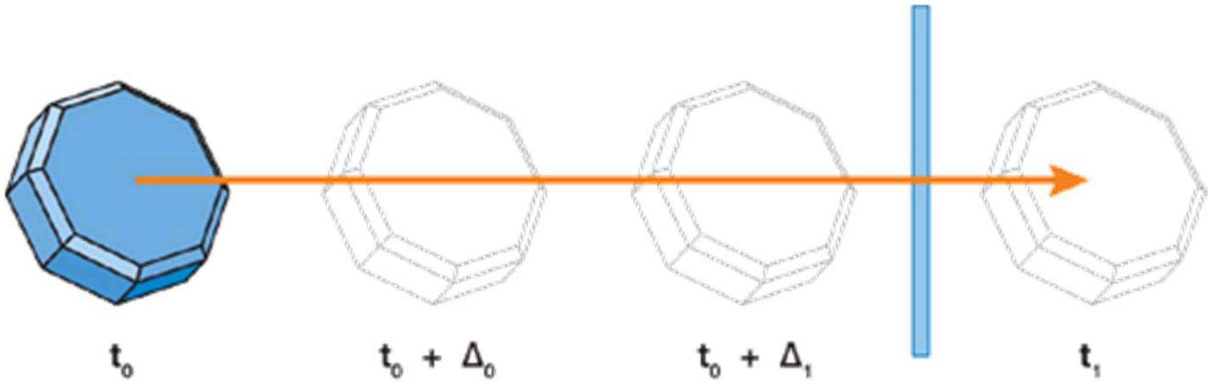






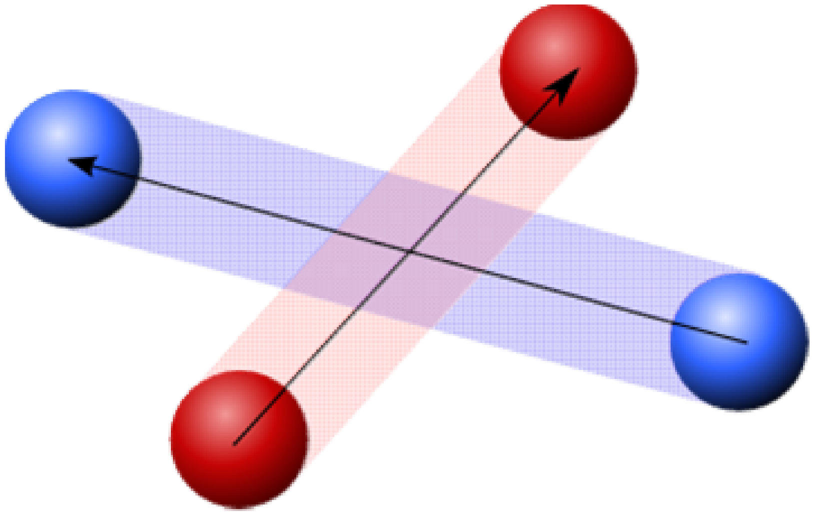
A posteriori (Discrete)

- Advance physics by time step then check for collision
- Simple
  - List of objects → return list of intersections
  - No time variable in calculations
  - Miss actual time of collision
- Need to "fix"



(a)

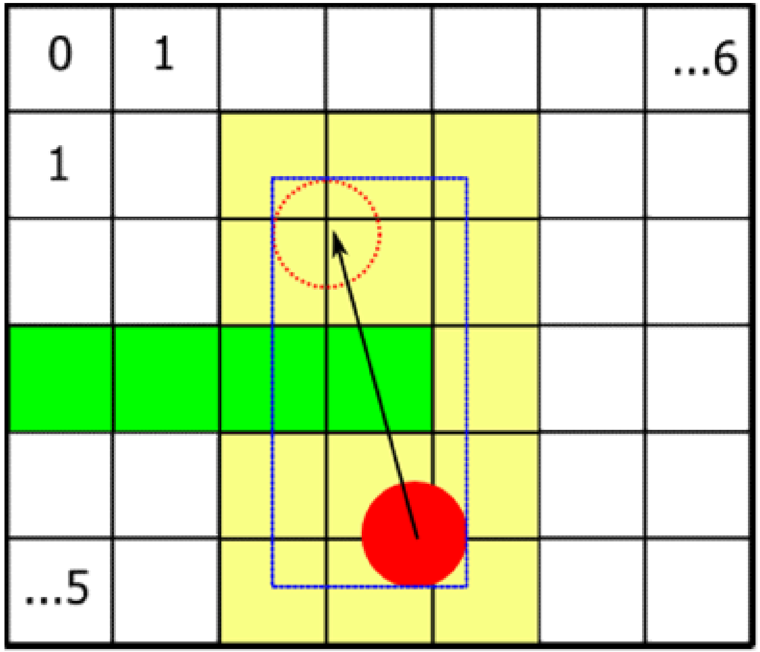
A priori (Continuous)



- A priori (continuous)
  - Predict future movement
    - Trajectories
  - Can be more precise
  - Can be more stable
  - More complex
    - Dimension of time
    - Often no closed form solution (numerical approach)
    - Aware of how objects move
      - Elastic objects (deforming)

Animated Objects - Practical Solutions

- Use extruded geometry
- Use overesized geometry
- ...
- Cast ray(s)
- Evaluate often enough
  - Restrict speed
- Extensive testing
- Some cases will be missed



Independent Render and Game Loop

- Do update in predefined intervals
- Independent from rendering loop
  - Slow rendering does not impact update cycle