

Collision Detection

The Basic Cases

Point - Circle

Point - Rectangle

Circle - Circle

Rectangle - Rectangle

Circle - Rectangle

The Basic Cases

Point - Circle

Point - Rectangle

Circle - Circle

Rectangle - Rectangle

Circle - Rectangle



Point - Circle

$$x^2 + y^2 \leq r^2$$

The Basic Cases

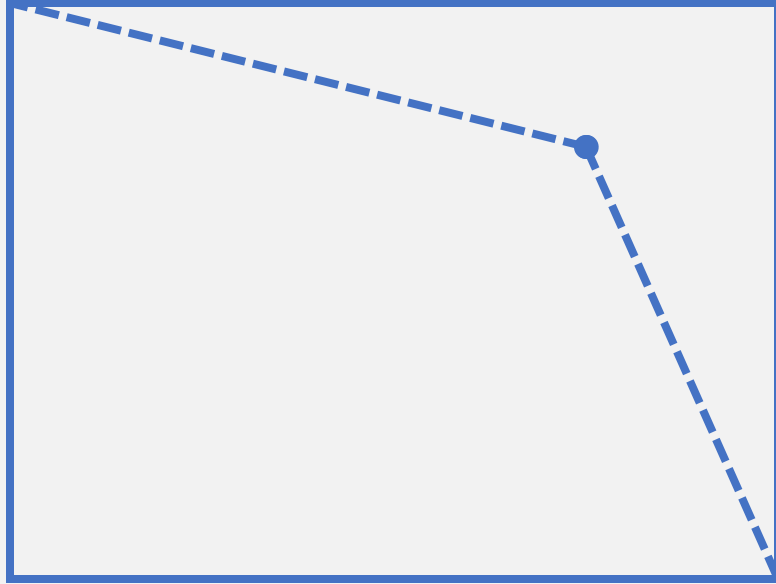
Point - Circle

Point - Rectangle

Circle - Circle

Rectangle - Rectangle

Circle - Rectangle



Point - Rectangle

$$\begin{aligned} l &\leq x \leq r \\ t &\leq y \leq b \end{aligned}$$

The Basic Cases

Point - Circle

Point - Rectangle

Circle - Circle

Rectangle - Rectangle

Circle - Rectangle



Circle - Circle

$$(x_2 - x_1)^2$$

+

$$(y_2 - y_1)^2$$

\leq

$$(r_1 + r_2)^2$$

The Basic Cases

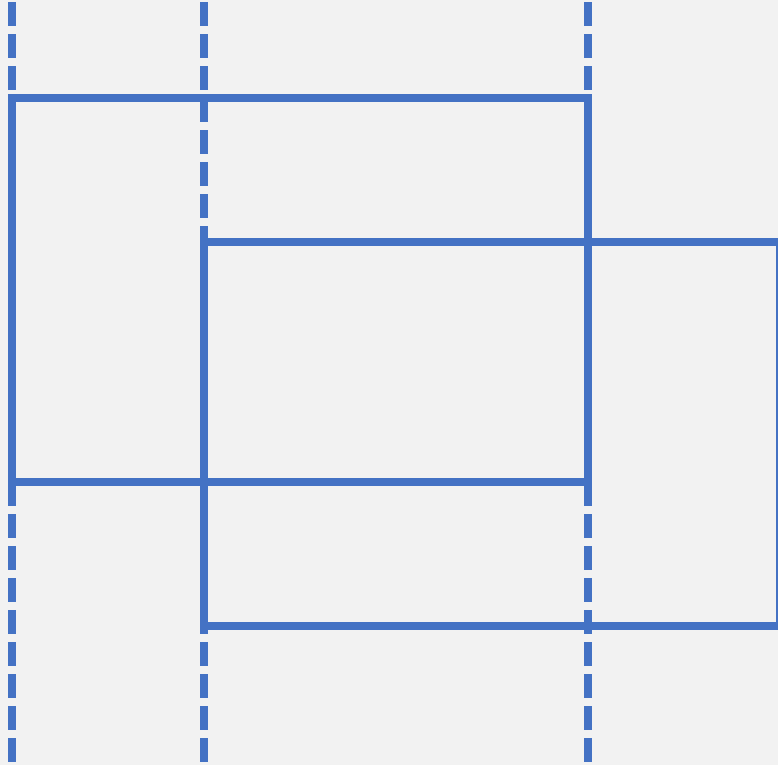
Point - Circle

Point - Rectangle

Circle - Circle

Rectangle - Rectangle

Circle - Rectangle



Rectangle - Rectangle

$l1 \leq r2$ and $r1 \leq l2$

and

$t1 \leq b2$ and $b1 \leq t2$

The Basic Cases

Point - Circle

Point - Rectangle

Circle - Circle

Rectangle - Rectangle

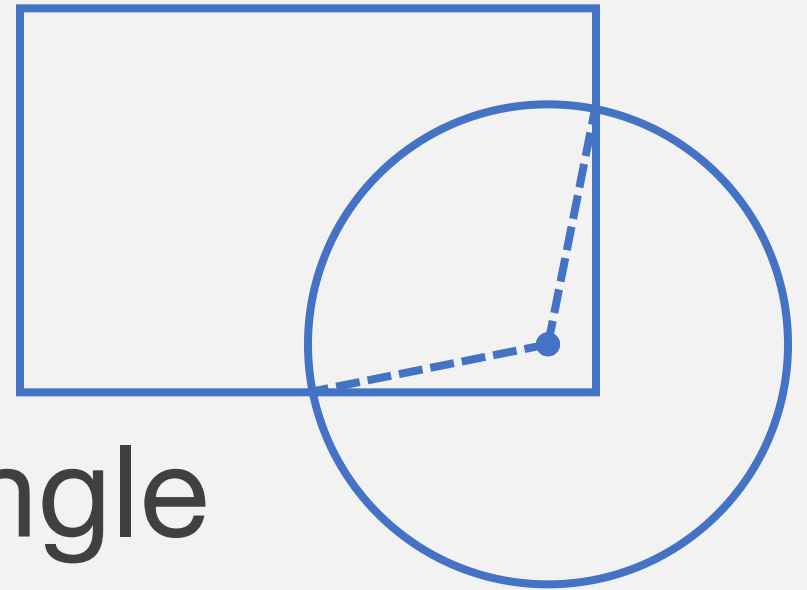
Circle - Rectangle

Circle - Rectangle

This one is tricky...

Case 1 Center of the circle is
in the rectangle

Case 2 Intersection(s)
exist between the circle
and any side of the rectangle



Efficiency?

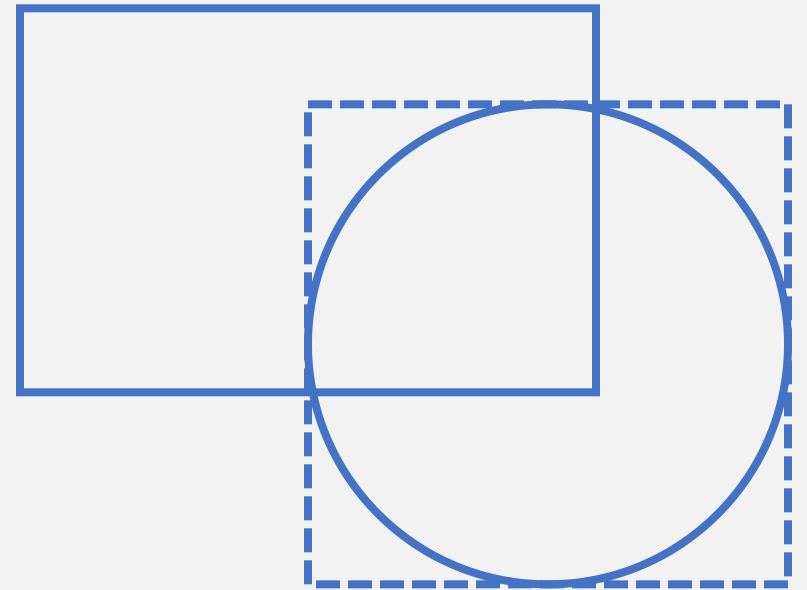
Bounding Boxes

Axis-Aligned Bounding Box (AABB)

Perform rectangle - rectangle
detection

If collides, perform the
actual detection

Otherwise, skip them!



Time Complexity?

Time Complexity

Suppose there are n objects

To detect collisions between one object and any one of the others: $O(n)$

To detect all collisions: $O(n^2)$

MARIO
004450

×00

WORLD
1-1

TIME
269



Time Complexity

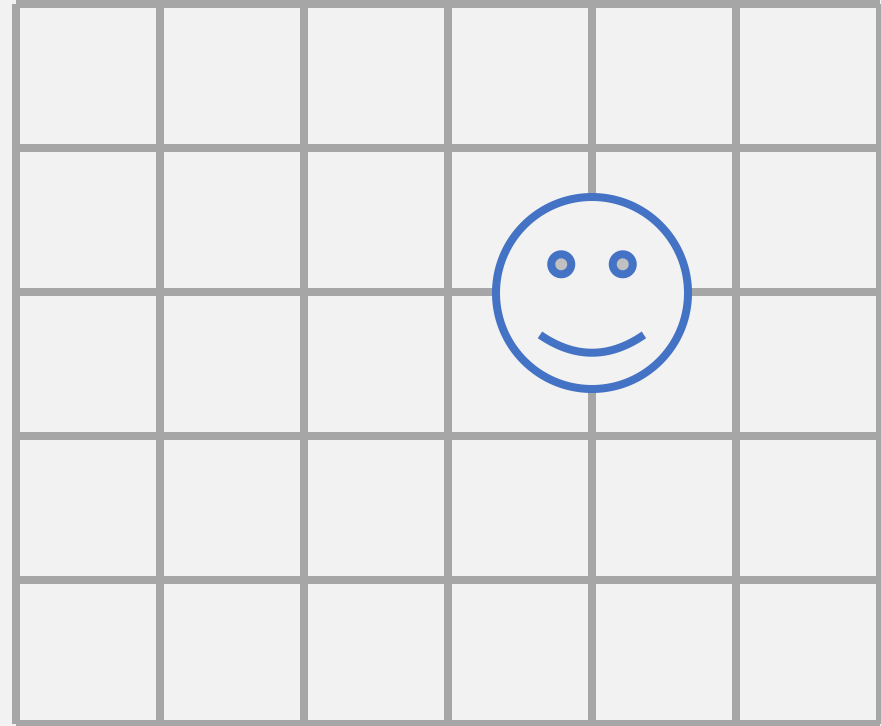
If we have 1000 objects

Collision calculation:

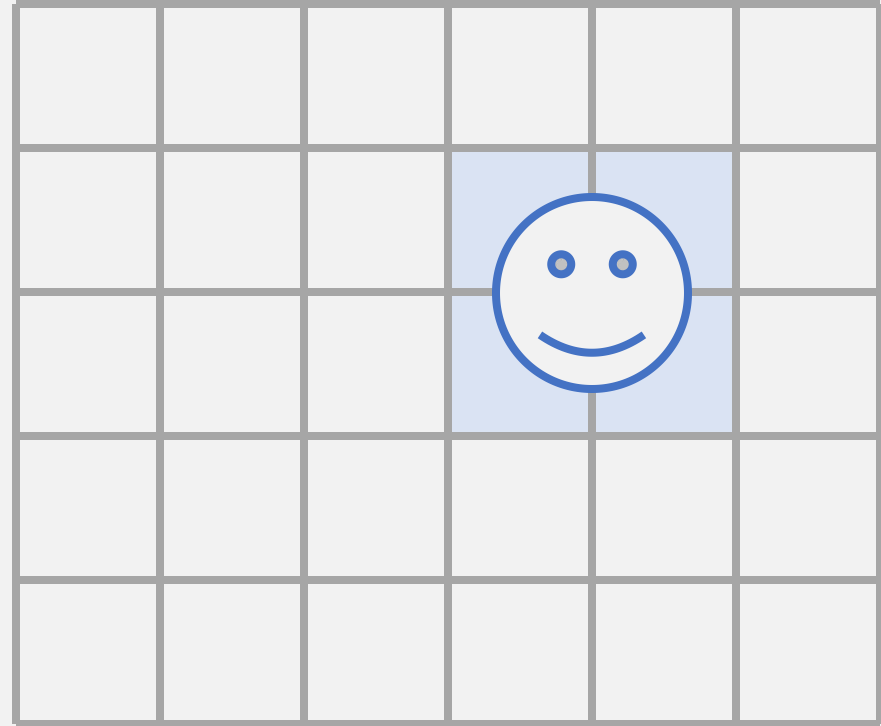
$$1000 * 999 / 2 = 499500$$

How can we optimize this?

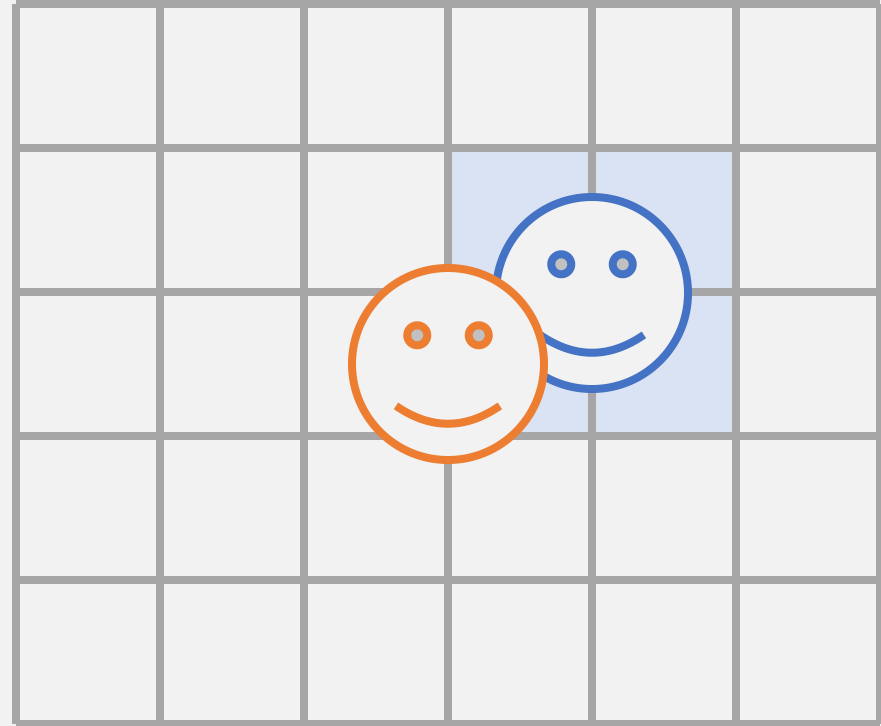
Collision Matrix



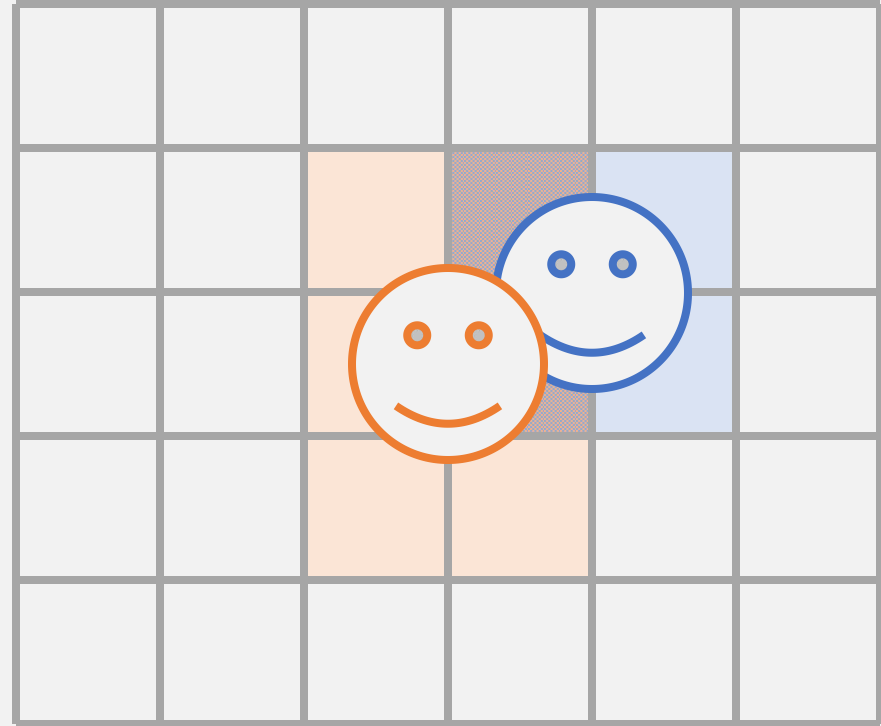
Collision Matrix



Collision Matrix



Collision Matrix



Collision Matrix

Find the slots each object occupy

Cache them in a matrix

Do collision detection only if two
objects share a slot

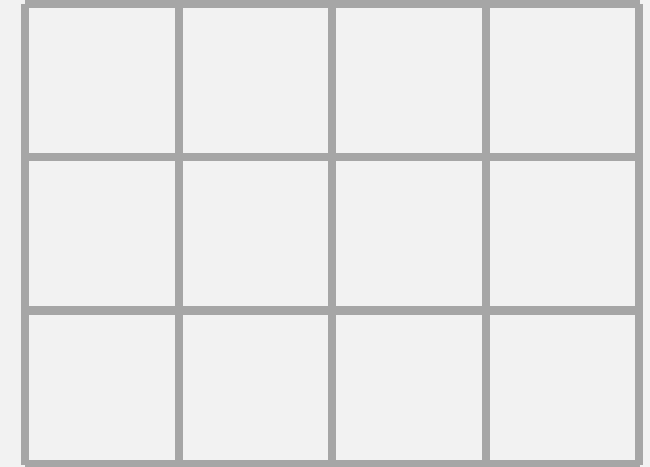
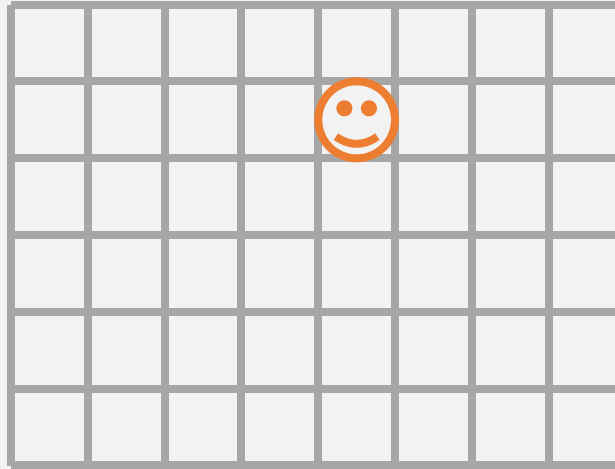
Potential Problem?

Collision Matrix

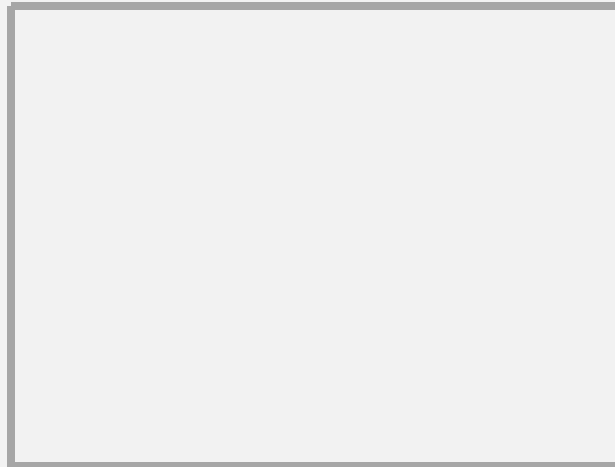
What if the object is very large?

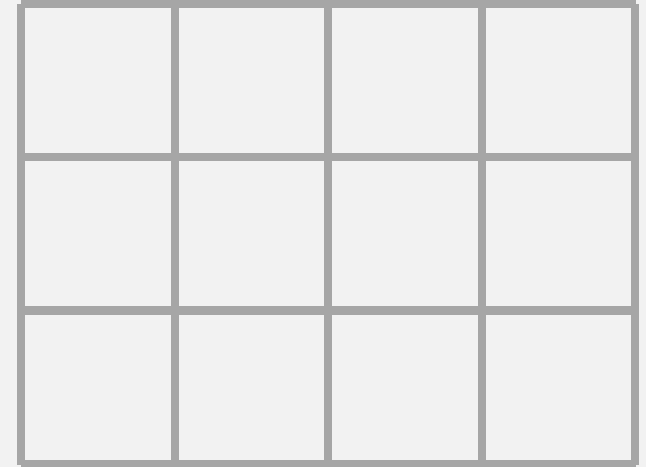
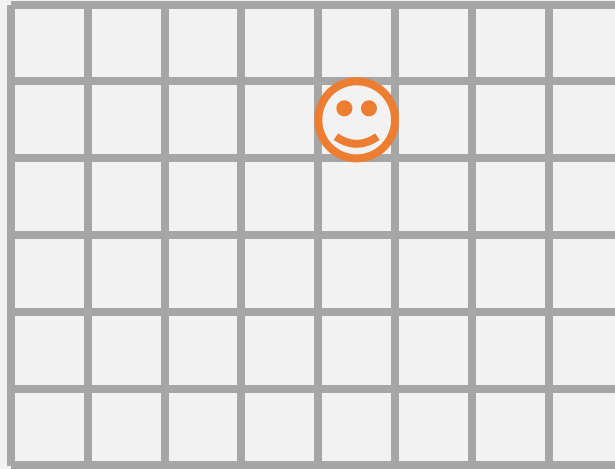
It may occupy a large number of slots

Solution: multi-layer space partitioning

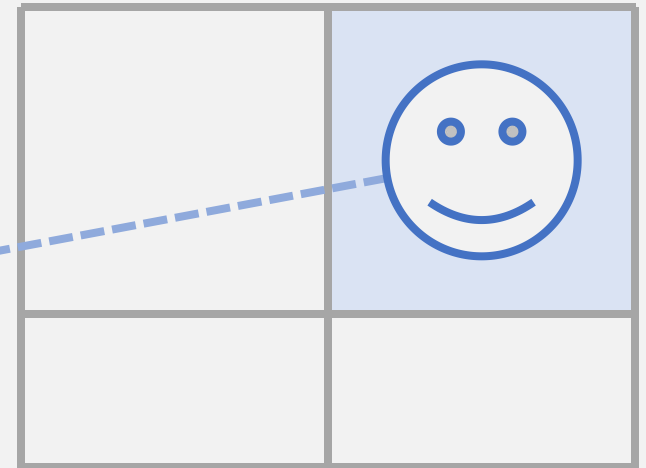
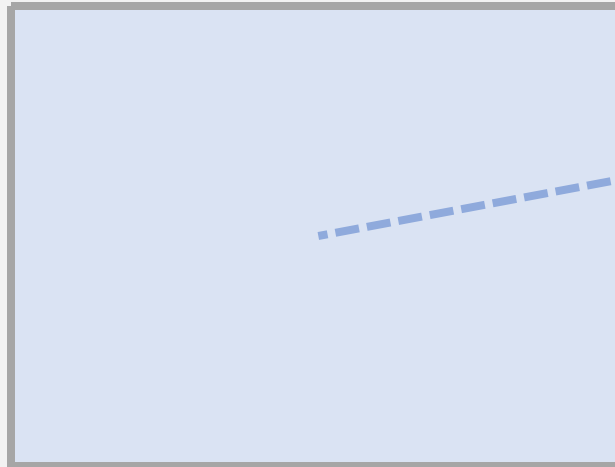


Space Partitioning

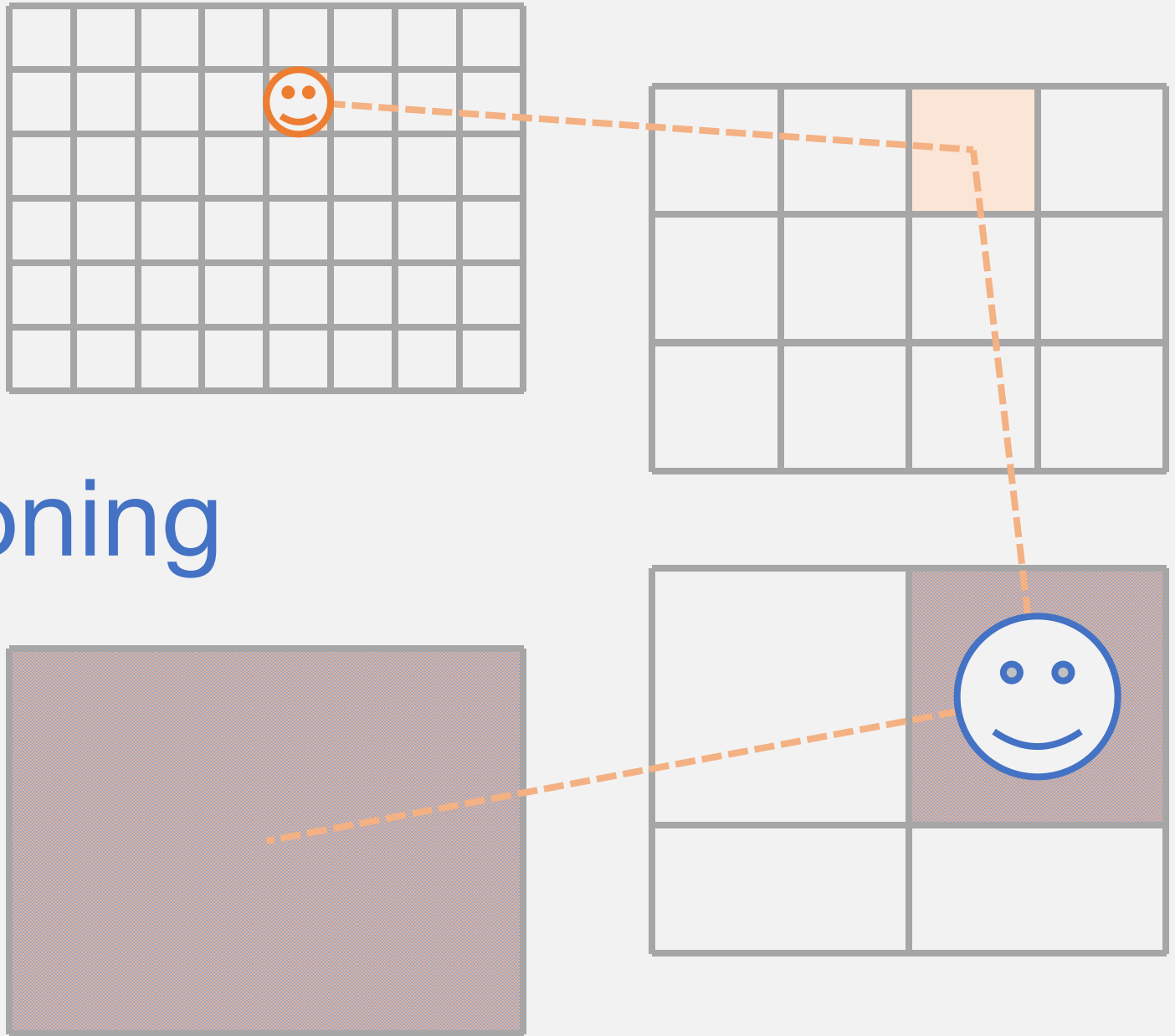




Space Partitioning



Space Partitioning



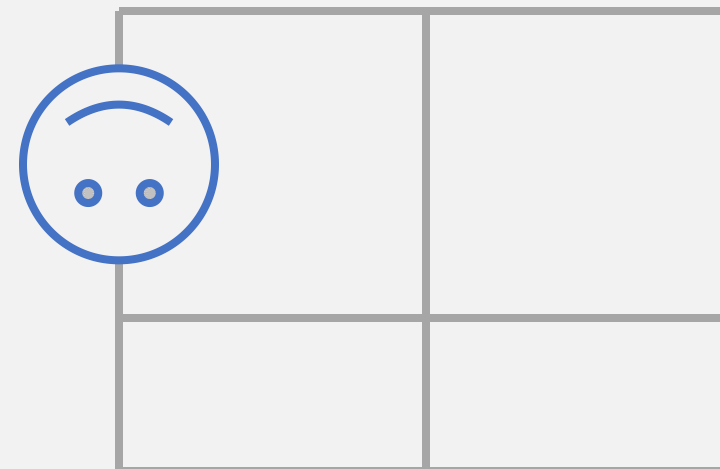
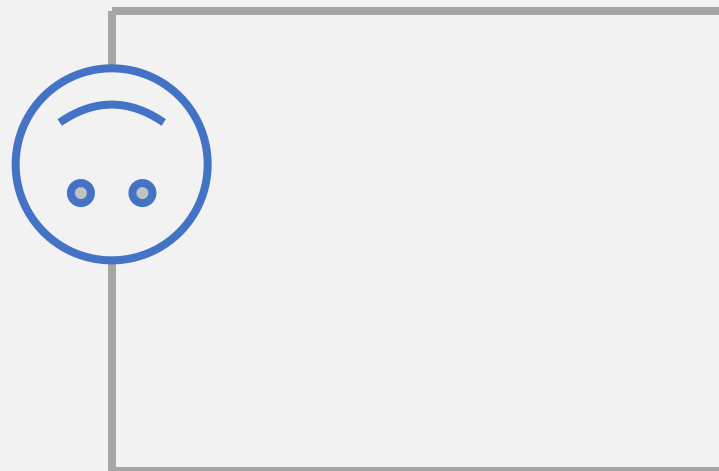
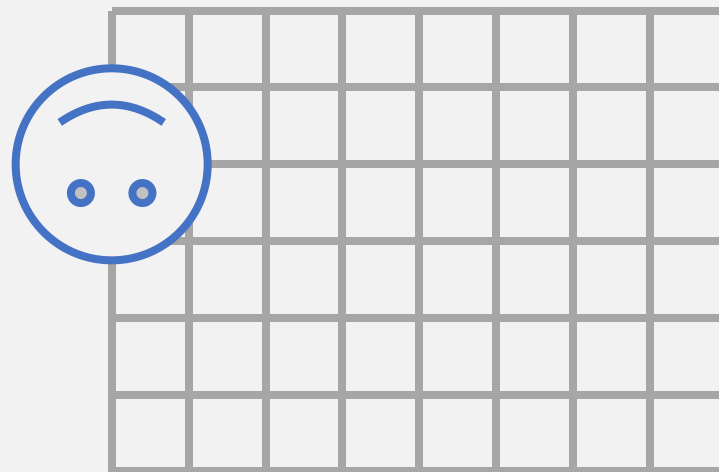
Space Partitioning (Quadtree / Octree)

Cut the space recursively

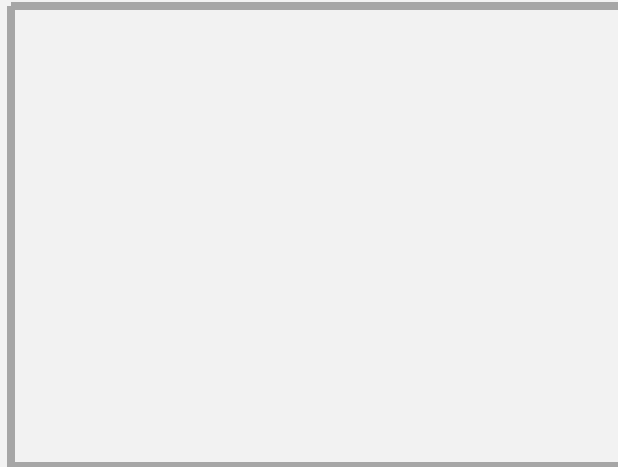
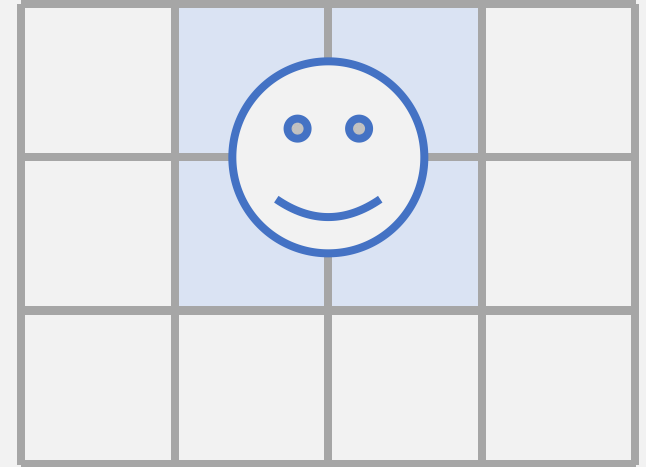
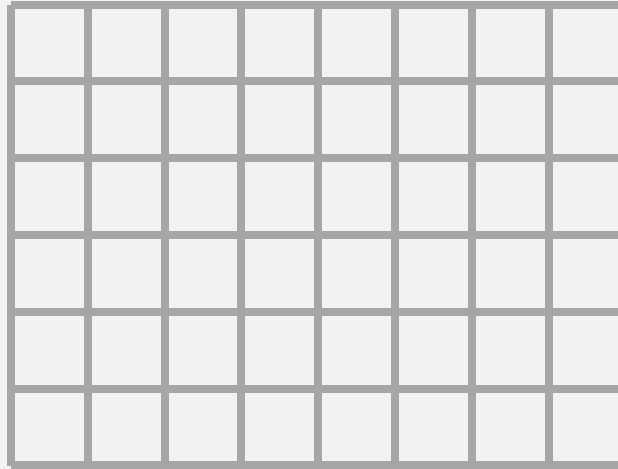
Find the smallest slot the object fits in

For collision detection, search upward
until reaching the root

What if...



Solution:
Allow 2x2 occupancy



Problem Example

Hitting the Targets

From ICPC NAQ 2012

<https://open.kattis.com/problems/hittingtargets>

Problem Example

Simple Polygon

From CTU Open 2011

<https://open.kattis.com/problems/polygon>