

Detecting Whether Two Boxes Overlap

Posted on [September 8, 2011](#) by [Fletcher](#)

Questions asked on job interviews also make for interesting blog post topics. (And very rarely, they are also practical!) This post discusses one of my favorite interview questions: “How can you tell if two axially-aligned 2D boxes intersect?” The adjective “axially-aligned” means that the sides of the box are parallel to the x - and y -axes; the box is not arbitrarily oriented. The acronym *AABB* is often used for axially-aligned bounding box.

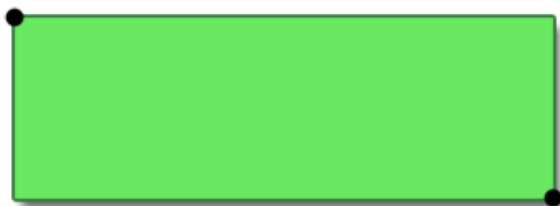
I like this question because it is both a test of experience and problem solving ability. Entry-level applicants aren’t necessarily expected to know the answer, but they should be able to suggest an approach (with very high frequency they suggest the incorrect solution discussed below), work through some examples, and arrive at the correct one. Most experienced web/GUI programmers have worked with 2D boxes and have encountered the problem, and should know the proper solution. However, they often do not immediately perceive the principle that extends the idea beyond *AABB*’s to arbitrarily-oriented boxes. In summary, it’s a problem with a deceptively simple starting point and many branching points depending on the skill of the applicant, which is precisely why it’s a great interview question.

Let’s say that we have basic 2D vector and bounding box classes such as:

```
// Simple 2D vector class
struct Vec2D
{
    float x,y;
};

// 2D axially-aligned bounding box.
struct Box2D
{
    Vec2D min, max;
};

(min.x, min.y)
```



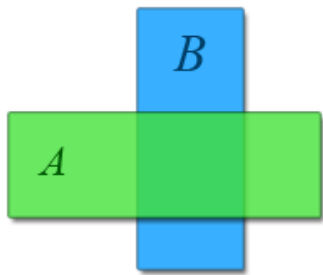
(max.x, max.y)

So the goal of the question is a function with a prototype such as

```
bool BoxesIntersect(const Box2D &a, const Box2D &b);
```

When faced with this problem, inexperienced programmers produce a particular solution that

doesn't work with such regularity, that it is worth mentioning. They suggest to check the four corners of box *A*, to see if any are containing within box *B*, and also check *B*'s corners to see if they are contained within *A*. But this approach fails in the following example.



The correct approach is to work by process of elimination. What are some situations when it is clear that the two AABB's do *not* intersect? We can immediately think of an obvious case: if box *A* is completely to the left of *B*, then the boxes cannot intersect. Likewise if *A* is completely to the right of *B*, or above or below it. Are there any other cases to consider? No. If *A* is completely to the left or right of *B*, then *the vertical positions of the boxes do not matter*. If two boxes do not intersect, they will fit into at least one of the four cases just mentioned. (If you are not convinced of this, a useful exercise is to try to categorize all the possible possible cases.) So the only remaining task is to express “*A* is completely to the left of *B*,” and so forth, in C.

This is the correct answer:

```
bool BoxesIntersect(const Box2D &a, const Box2D &b)
{
    if (a.max.x < b.min.x) return false; // a is left of b
    if (a.min.x > b.max.x) return false; // a is right of b
    if (a.max.y < b.min.y) return false; // a is above b
    if (a.min.y > b.max.y) return false; // a is below b
    return true; // boxes overlap
}
```

(By the way, whether “<” and “>” should be “<=” and “>=” depends on exactly what you mean by boxes “intersecting.” If you are using floating point coordinates and this distinction matters, you are probably doing something wrong.)

Most everybody understands the solution to this problem, and it extends in a straightforward way into 3D. So that's pretty much all there is to say about AABB's.

But what about boxes that are arbitrarily oriented? How about convex polygons (or polyhedra in 3D) with an arbitrary number of sides? To solve these problems requires that we distill from our ad-hoc solution a more fundamental principle, known as the [separating axis theorem](#). It can be used to [detect whether two convex polygons overlap](#).

■ [Post to Facebook](#)

1

1

Like

1

Recommend this on Google

This entry was posted in [Uncategorized](#) and tagged [2D](#), [Geometric Tests](#), [Interview Questions](#). Bookmark the [permalink](#).

