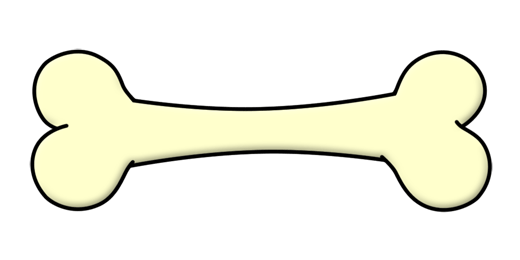
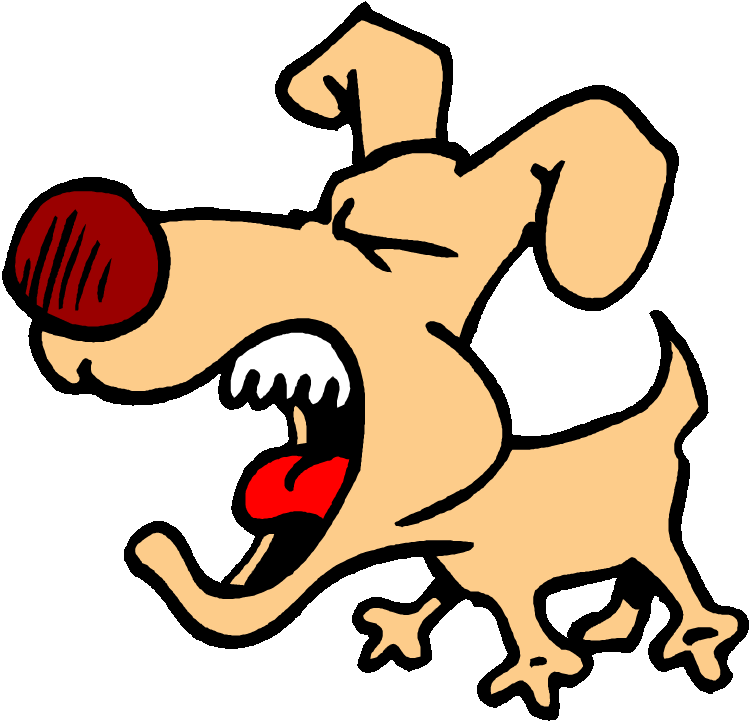
**Hit Boxes**

Let's suppose you have a game where a dog has to find bones and put them in a dog house to gain points. Trust me, this game will be a hit and make you millions.



The question is, how does the computer know when the dog has "caught" the bone? The answer is, the dog needs to be at or around the same location as the bone. That is, the X and Y coordinates of the dog need to be the same as the X and Y coordinates of the bone. Basically the code is:

if dogX == boneX and dogY == boneY:

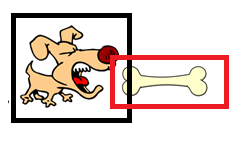
# dog has caught the bone

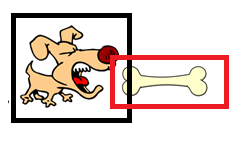
Well, sort of.

The real answer is that the coordinates don't have to exactly match. Instead, a region around the dog needs to overlap with a region around the bone. In gaming we call these regions "hit boxes".

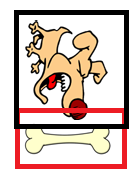
Suppose the dog in our example can approach the bone from four directions:

From the left:

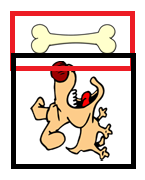


From the right: 

From the top:



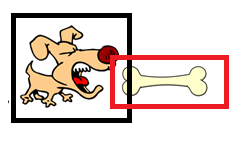
From the bottom:



The annoying black and red boxes are simply for our reference - so we can see when a "collision" occurs between the dog and bone. A collision occurs when the boxes overlap.

We will look at only the case where the dog approaches the bone from the left, and hopefully you can figure out the rest from there.

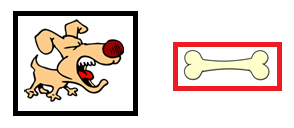
Case 1: The Dog Approaches from the Left



In this case, a collision occurs when the ***right*** side of the dog passes (is to the right of) the ***left*** side of the bone.

Let's assume the dog and the bone are both 50 pixels wide. Let's put them at two random spots:

(40,?)



50

(120,?)

In the above picture, the x coordinates are shown. The y coordinates are not important here, so they have been left as question marks. In Python we have these variables:

dogX = 40

dogWidth = 50

boneX = 120

As we can see, the dog hasn't "caught" the bone yet. The left side of the bone is at 120. The right side of the dog is at 40+50=90. An x coordinate of 120 is "to the right of" an x-coordinate of 90. Another way of saying "to the right of" is "with X coordinate that is greater".

So we rewrite this:

"If the right side of the dog passes the left side of the bone"

as this:

"If the X value of the right side of the dog IS MORE THAN the X value of the left side of the bone"

The right side of the dog is just the left side plus the width (40 + 50), so we rewrite this again:

"If the X value of the left side of the dog plus the width of the dog IS MORE THAN the X value of the left side of the bone"

Finally we can translate this in Python:

if dogX + dogWidth > boneX: # if the dog's right side crosses over the bone's left side

# do stuff

**Adding the other three directions**

Your work is 90% done. You have the first direction taken care of. Now you just have to apply the same logic to the other three directions and **combine them in one long if statement**. The general look of the if statement will be:

If case1 AND case2 AND case 3 and case 4:

# the dog has the bone

This is just pseudo-code of course. I will leave you to figure out the details.