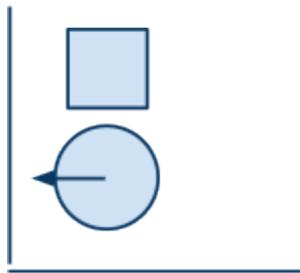
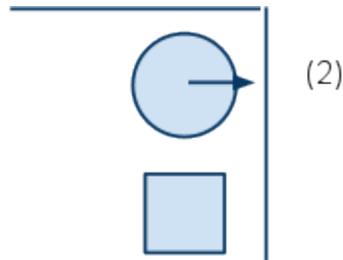


How "ASK RAYS" work



(1)

Key:

The circle is the agent facing in the arrow's direction
Square is the "player" (you)

First we found a corner like in label (1).

Then did ask rays 4

This will return 1 worldspawn [x1 y1 z1] ... 4 player [x4 y4 z4]

Note that usually the z value is 0. I bet it is just there for convention.

Suppose the ask rays 4 returned

4 player [-20 0 0]

Then we can reason that the player is at -20 in x-direction.

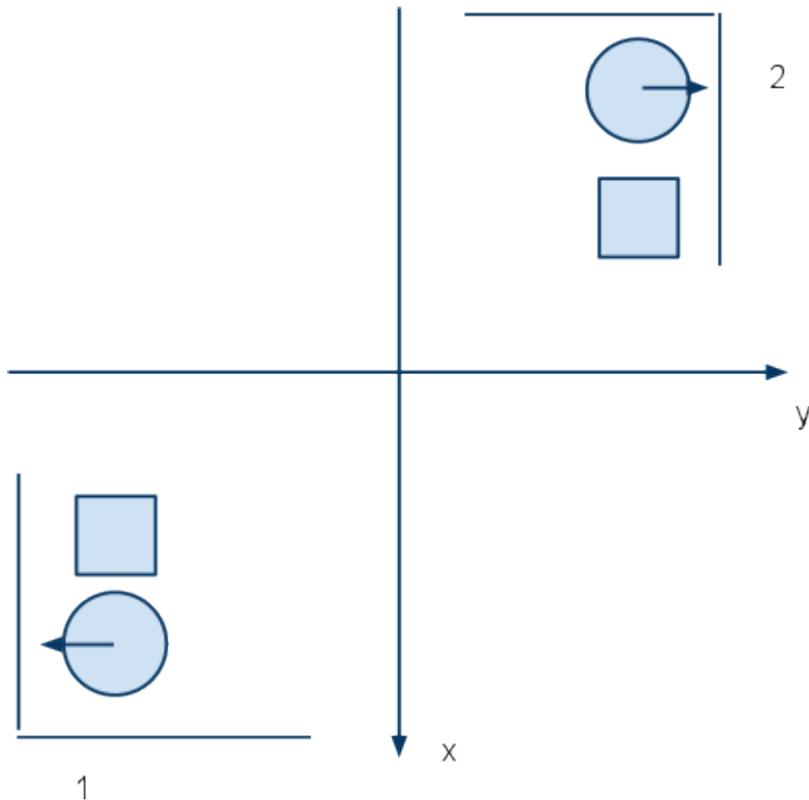
Now if I do the same "ASK RAYS 4" at label (2)

Then this will give me

4 player [20 0 0]

Then we can reason that the player is at 20 in x-direction.

But in both of the cases, the player is directly to the right of the agent, but it gives us 20 and -20. From this we can logically figure out that there is a "world coordinate system" (which is not in CB's documentation). Which looks like this:



Now here comes the more ugly part that is very difficult to figure out with out looking at the quagent SOURCE CODE.

at

~cs242/quake/scr/quagent or more precisely "/home/hoover/u1/cs242/quake2ur-20050512/src/quagent/"
we see that there is a code called "quagent_func.c"

This source has some interesting math. (on line 329)

theta = i*M_PI*2.0/num_rays + D2R(self->s.angles[YAW]);

D2R - is a function that takes degrees and changes it into radians.

M_PI is the value of PI (as you know... 3.14... stuff)

num_rays is the number of rays you specified for "ASK RAYS". (Also from experimenting the ask rays command a lot of time, we figured out that the number of rays it can shoot is at most 4.

so, num_rays will get set to 4 if too many is specified.)

Now the "evenly distributed" part in the documentation seems correct according to this equation but I only tested "ASK RAYS 4" or number greater than 4.

Now YAW (most important)

Since YAW is the direction the agent is facing (relative to the "world coordinate system"), we now see why we always have "4 player [x4 y4 z4]" when the player is directly to the right of the agent, rather than "1 player [x1 y1 z1]" or "2 player [x2 y2 z2]" or "3 player [x3 y3 z3]" when we do "ASK RAYS 4".

When we experimented this we had YAW = 120 for label (2). We did not check the YAW at label (1) but it should have been YAW = 120 - 180 = -30 or 330

Now one thing that is poorly documented:

from http://www.cs.rochester.edu/~brown/242/docs/quake_protocol.html

Command: GetWhere. Where is bot, how oriented, moving how fast? these values must be in world coordinates to be meaningful.

Response: ERR ([Echo]) [useful_description] or

OK ([Echo]) [World_State], where [World_state] is a vector of coordinates and a velocity: (world_x, world_y, world_z, roll, pitch, yaw, velocity).

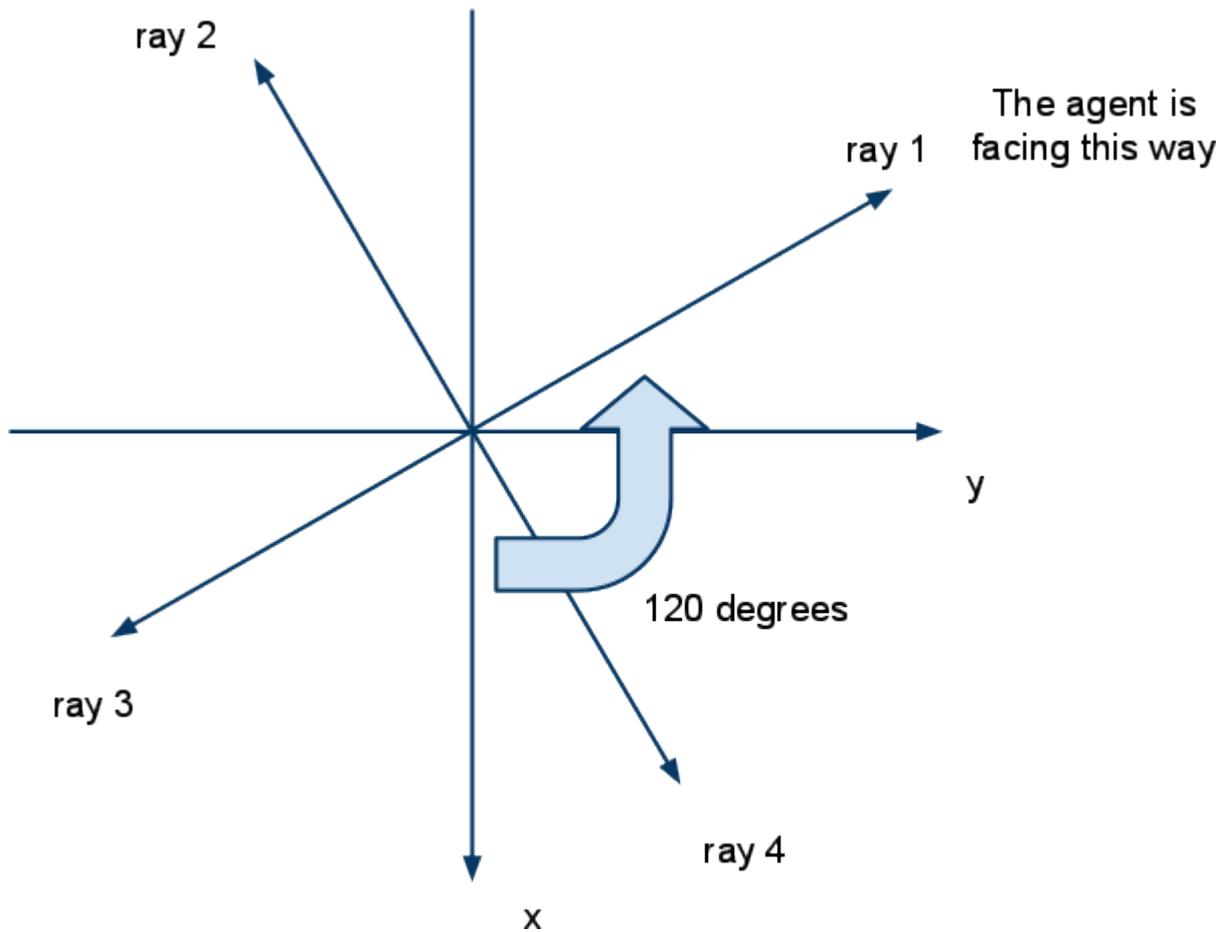
It is not (world_x, world_y, world_z, **roll, pitch, yaw**, velocity) but (world_x, world_y, world_z, **X, YAW, Y**, velocity) where X or Y could either be roll or pitch, and it does not matter because it does not make sense for the agent to be tilted or facing up (or down).

We found this by doing

```
do getwhere
OK (do getwhere) -2040.29 1817.7 -23.875 0 120 0 1
do turnby 90
OK (do turnby 90)
do getwhere
OK (do getwhere) -2040.29 1817.7 -23.875 0 210 0 1
```

which means that YAW is the 3rd to last value and not 2nd to last.

Now going back to the equation (sorry for jumping back and forth), YAW is the angle off from the X axis of the "world axis" which means if YAW is 120 then if I do "ASK RAYS 4" the rays will be shot like this.



This is a more thorough description of ASK RAYS command than the one on the course web page. We are guessing that this coordinate system will be the same for ASK RADIUS command.

If you want to replicate our observation then

1. Stand in front of a wall (but give enough space for the agent to spawn).
2. use telnet to connect to "telnet localhost 33333" to spawn an agent.
3. do "do getwhere" and find the YAW value (x).
4. do "do turnby -x" (x is from procedure 3)
5. do "ask rays 4" to check the coordinate of the closest wall (world spawn).
6. do "do turnby 90"
7. do "ask rays 4"

Now you can see that the wall's coordinate did not change.

This implies that there exists a "world axis" which is not (necessarily) a coordinate from the quagent's perspective.