
Subject: Re: Renegade Coordinates

Posted by [BlueThen](#) on Sat, 07 Aug 2010 15:43:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

The one in your quote looks a little broken, TechnoBulldog.

From what I can tell, `Commands->Get_Facing(obj)` gets the angle of which an object is facing in degrees. (`Commands->Get_Position(obj)` is self explanatory).

In the equations

```
Position.X += static_cast<float>(1.5*cos(Facing*(PI/180)));
```

```
Position.Y += static_cast<float>(1.5*sin(Facing*(PI/180)));
```

we're pretty much taking Facing, and converting it to Radians (multiplying by $\text{PI}/180$). We convert it to Radians because cosine and sine are both built to handle values between 0 and 2PI , or $-\text{PI}$ and PI , whichever you prefer. Radians just take the range of values from 0-360 and maps it to 0- 2PI .

Cosine and Sine both return values between -1 and 1, which makes it simple for us to change the distance at which our object is plotted. Multiplying 1.5 to these 2 values puts our object 1.5 units from the center. Then, after finding out the coordinates where the position should be from the origin, we translate it by `Position.X` and `Position.Y` by adding this value to them.

So all what this does is take our coordinate `Position` and moves it 1.5 units towards an angle (`Facing`).

I said that the formula in your message looks broken because you have the $\ast (\text{PI} / 180)$ outside of the trig functions, when they should be inside.

After getting this position, you can spawn your object and face it towards/away/perpendicular to the initial `Position`, all in `a000clown`'s post.

Edit: I'm also on Nite-Serv. You can ask me for help on the math part, but I don't really know Renegade's API as much as Mike does.
