Wire Expression2:Common Errors

From GMod Wiki

This is a recreation of my thread on wiremod.com, by eurocracy, I think that by putting it here people will benefit more from the information it provides.

Contents

Common problems with E2

I'm making this thread in order to solve all those blasted problems that seem impossible to solve. 75% will be a simple problem that is a synatax error or typo etc. So get ready for the most common of them

Finding a problem area

First of all your E2 validation is your first line of defense against any annoying error, it will give you a brief of what the error might be and will give you the line that is wrong and the character number, you can then jump to that area by clicking the validation error. The E2 validation will often be errors with spelling on the code^_^ These such errors can be hard to see sometimes as we will see in the next section and such.

Quick problem solve

The most common problems arise in spelling, some are extremely hard to see, for example take a look at this piece of code and try telling me the error in it:

```
if(Range<1000|Bearing==0) {Fire=1)
```

Now in bigger if statements this would be harder to see, but as you can clearly see here, there is a simple error. If you didn't see it then lets take a look again:

```
if(Range<1000|Bearing==0) {Fire=1)
```

As you can see in the bold there is a small problem. Instead of a } to end the if statement there is a). This means that E2 cannot read the expression and as such will not work at ALL. To see a finished result:

```
if(Range<1000|Bearing==0) {Fire=1}
```

Now the error is solved we can validate again, in order to check for more errors Another common problem for beginners is the difference between variables and functions, for instance some new E2 coders may put this:

```
ApplyForce(vector)
```

This is a VERY WRONG example with 2 instances in it, lets take a look

```
ApplyForce(vector)
```

As you can see it has defined the force application as a variable and the vector as a function, because of some misplaced capital letters, a fixed result would be:

```
applyForce(Vector)
```

Now the capital letters are replaced and solved, this should come with no error now, fixing the problem

Matte has submitted this annoying problem, I think you can see what is wrong:

```
if(Button == 2 & Length = 6){Door = 1}
```

See it? Well as you can see it clearly says Length = 6, this is NOT comparing, use 2 = signs to compare, use 1 to set a variable. Fixed version:

```
if(Button == 2 & Length == 6) {Door = 1}
```

Common function errors

Now just to settle a few slight problems, these problems are common mis-beliefs of the functions, here is an example:

```
shootPos()
```

Now shootPos is often confused with aimPos, shootPos is where you view is, NOT WHERE YOU ARE AIMING! aimPos on the other hand gives the XYZ vector of where you are aiming.

One blasted little confusion here:

```
lastSpoke()
```

This little blighter was a living nightmare in one thread, everybody was confused and bewildered what it was until Matte came in and solved it. Now for those of you who don't know, this returns the entity of the person who spoke last, for instance you could use like this:

```
@name Test
@outputs Name:string
runOnChat(1)
Name=lastSpoke():name()
```

Now in that very same thread someone thought lastSpoke() was WHEN someone put a chat message last, that is not the case, you can still find out when though via this command:

```
lastSpokeWhen()
```

Using lastSpokeWhen() you can get the time when someone or anyone last said something, allowing you to store times in a database or something P:

Void error

Now you may have gotten a void error once or twice, for an example of a void error lets take a look at an example:

```
Sphere=entity():isWeldedTo()
applyForce(Sphere:forward())*(Sphere:mass()*2)
```

You should have spotted the error, if not then let me explain. The * sign is not in a bracket that contains the vector for applyForce(), so therefore its not connected to that applyForce() and thus is in the void. To solve simply put on another bracket:

```
Sphere=entity():isWeldedTo()
applyForce((Sphere:forward())*(Sphere:mass()*2))
```

Now that it is fixed you should be able to use it as normal

Things that DO NOT WORK

Now these are the dumb errors that just fail, fail and FAIL AGAIN. These are the more annoying breed and will take ages to solve without help.

Firstly armor on NPC's, NPC's do not use armor, they use health only! So doing this:

```
Screen=TargetNPC:armor()
```

Is doomed to fail, just stick to:

```
Screen=TargetNPC:health()
```

Another great fail is trying to use this:

```
owner():setAlpha(0)
```

It does NOT WORK, if you wanna be invisible then use this:

```
MyString="models/effects/vol_light001"
owner():setMaterial(MyString)
```

This will set you to the volume light material, this is as close as the can get to becoming invisible

If you don't see the problem in this code you will kick yourself in the face:

```
@name test
```

```
interval(1)
LS=owner():lastSaid()
@persist LS:String
```

That code has **2** code problems in it, first of all @persist is AT THE END and secondly the LS:String has a capital on the "string" which is variable defining NOT type defining.

Another problem to look out for is the E:health() problem on a prop, the health is designed to get a players health amount NOT to get a prop's health, so it will MOST LIKELY fail. An exception to this failing is if it is used on a breakable prop, in which you will have a working expression

Apply Force

Alright, applyForce does not apply force towards a vector, it applies force on a directional vector, I'm just putting this here because I have seen this problem TOO MANY times.

To be updated <(^-^)> --Eurocracy 17:20, 6 September 2009 (UTC)

Retrieved from "http://wiki.garrysmod.com/?title=Wire_Expression2:Common_Errors"

- This page was last modified on 7 September 2009, at 19:05.
- This page has been accessed 4,710 times.

- Privacy policy
- About GMod Wiki
- Disclaimers