

Scripting Variables

- By Gamma927

Alright. I want to make this nice and short, so that I can spend the rest of my summer playing Call of Duty 4, instead of typing up a tutorial. You probably don't want to read a three page guide on variables as well, so it's a win for me, and a win for you. Onto the good stuff.

Chapter 1: Variables

There are three basic kinds of variables.

- Short: A number. Holds a value up to 32 thousand something something. If you think that your number is going to exceed 32 thousand, or be less than negative 32 thousand, use a long.
- Long: A number. Holds up a value to around 2 million. Yes, that's over 9000. If your variable is going to exceed even that, I suggest you rethink whatever you were planning to do.
- Booleans: True / False statement. Set it to either true or false.

To initialize a variable, put it into something like this:

```
(global [VARIABLE TYPE] [NAME] [VALUE])
```

Replace [VARIABLE TYPE] with either a short, long, or boolean. Then, replace [NAME] with whatever you want to call the variable later on. [VALUE] is the initial value. This can change later on. Example:

```
(global short x 1)
```

This creates a variable called 'x', with the initial value of 1. Simple enough? I sure hope so, because I have no idea what to say if you don't get it. It's basically "fill in the blanks", except the blanks have been replaced with words surrounded by brackets.

Note: Do NOT put brackets around the words. Leave it like the example above.

Chapter 2: Changing Variables

So you now have a variable, and you've initialized it to something. Now what? The point of a variable is so that it changes. If you don't want it to change, then you shouldn't have used a variable in the first place, and instead used a nice solid number. Now, the command to set a variable, is:

```
(set [VARIABLE NAME] [VALUE])
```

Should be simple enough. It's like the command to initialize variables, but instead it just modifies an existing variable.

Now, what if you wanted to increase a variable by one? Well, first of all, we need to find out the value of the variable + 1. Then we could use the set command. So, to increase a variable, you use the addition command.

```
(+ [VARIABLE NAME] [VALUE])
```

Looks almost like the set command, except set is replaced with a +. Guess what the subtraction command is! Just replace set with a -, and you're set. For multiplication, it's *, and division is /.

Looking back at the example to set the variable to the variable + 1, you would do something like this, assuming your variable is named 'x':

```
(set x (+ x 1))
```

Look at it, and you'll see the basic idea. For the set command, set the [VALUE] to the + command.

Chapter 3: Getters

Sometimes, you don't know what value you want to set the variable to. You just know that you want it to the amount of health a unit has, or the amount of AI still alive. It could be 0, 1, or as many AI as you put. To check the amount of AI still alive in an encounter, you use the "ai_living_count" command. To set the variable to that amount, you use the set command from the last chapter. So, it would look something like this:

```
(set x (ai_living_count AI))
```

Assuming, of course, your variable is called 'x', and the AI encounter is called 'AI'. It's basically replacing the [VALUE] with the getter.

Well, it was only 2 pages. That's less than 3. I hope you learned something. If you didn't, feel free to throw angry words at me. My xfire is: Gamma927. Have a good life.