LESSON 1: Traditional Logic Gates in Forge

INSERT INTRODUCTION TO TOPIC HERE
INSERT REFERENCE TO SCRIPTING SYNTAX HERE

AND gate

A 2+ Input, 1 Output, circuit where all Inputs must be triggered to activate the Output.

Traditional Digital Logic

In traditional electronics/logic, it is described as requiring all Inputs to be “high”, or powered (1 rather than 0), for the Output to go from “low‟ to “high” (from 0 to 1). A period (.) is used to designate the AND gate in the output in traditional shorthand.

Some Examples of 2 Input AND Gates in Traditional Digital Logic

A=0
B=0
A.B=0

A=1
B=0
A.B=0

A=0
B=1
A.B=0

A=1
B=1
A.B=1

Using AND Gates in Forge

In Forge, this is a bit more flexible, as the Message/Power: Multi Condition allows us to use Messages on Message Channels, Power States (On/Off), and Power State Toggles (On/Off/Toggle), to define what “high” actually is. (Also, yes, you’re hilarious, very original joke you’re thinking up there)

Some Examples of 2 Input AND Gates in Forge Logic

```
Script Brain - AND Gate
1| Multi [2] (Message{1}, Message{2})
 | <Message{3}>
```

This would require the Inputs, Messages on {1} and {2} ,to occur at the same time and would trigger the Output, a Message on {3}.

```
Script Brain - AND Gate
1| Multi [2] (Power{1}=ON, Power{2}=ON)
 | <Message{3}>
```

This is much more akin to the Traditional Digital Logic gate in the examples above, where ON represents the input being “high”. Keep in mind, OFF can also be used to represent “high”. Because of this, it’s much easier to think of “high” as another way to say True

```
Script Brain - AND Gate
1| Multi [2] (Power{1}=OFF, Power{2}=ON)
 | <Message{3}>
```

```
Script Brain - AND Gate
1| Multi [2] (Power{1}=OFF, Power{2}=TOGGLE)
 | <Message{3}>
```

```
Script Brain - AND Gate
1| Multi [2] (Power{1}=ON, Message{2})
 | <Message{3}>
```

These are all legitimate AND gates made using the Multi Condition. You could also use more than 2 Input Conditions for an AND gate, but you MUST require them all to be True for it to be considered as such.

OR gate

Traditional Digital Logic

TEXT

Using OR Gates in Forgev

TEXT

NOT gate

Traditional Digital Logic

TEXT

Using NOT Gates in Forge

TEXT

NAND gate

Traditional Digital Logic

TEXT

Using NAND Gates in Forge

TEXT

NOR gate

Traditional Digital Logic

TEXT

Using NOR Gates in Forge

TEXT

EXOR gate

Traditional Digital Logic

TEXT

Using EXOR Gates in Forge

TEXT

EXNOR gate

Traditional Digital Logic

TEXT

Using EXNOR Gates in Forge

TEXT

INSERT WELL-WISHING AND GENERAL GOOD CHEER HERE

LESSON 2: Advanced Scripting

TEXT

LESSON 3: Advanced Scripting Mechanisms

TEXT

LESSON 4: Creating Scripted Objectives

TEXT

LESSON 5: Creating Scripted Gametypes

TEXT