Revan's Expression Gate 2 Guide - Intermediate Level - Commonly Used Functionality

Things to Notice as you Read the Guide:

There are several symbols/colors I may use throughout this guide to help you find things, so here's a list of them all:
Term - A definition you should know
Other things I may make reference to:
The Expression Gate 2 Wiki - Every now and then I will link to a page on the wiki for you to reference. It's a good idea to bookmark this page, because even experienced E2 programmers such as myself will pull up the wiki mid game to find a function. To make opening it in game easier, I recommend that you either use the steam in-game browser or run Garrysmod in windowed mode (not full screen), a setting which can be altered from options.

Introduction

Welcome to part two of Revan's Guide to the E2 (Expression Gate 2). In reading this, it is my hope that you will better understand how and why the E2 works the way it does. I will make an attempt to keep my data here as accurate and current as possible, but considering that I no longer really play Garrysmod, I cannot guarantee that everything you see on this page will still function in the future, nor can I claim that the ways I show you how to do certain things here are the best ways to do them. The reasons for that include features being added later, my being unaware of the existence of a feature, or even just to teach you a concept, I may use a less appropriate way of achieving it for the sake of the example. Understand and respect those faults as you read.

Before continuing with reading, I advise that make certain that you understand all of the basics of E2 as I described them in my previous guide. If you have not read it already and you are familiar with the E2, I still advise that you look at my previous guide, if only just to skim it. I will assume in this guide that you understand each of the concepts I presented, and that you have made use of them in one form or another in Garrysmod. I assume that you did not just read the text and expect to be a master, but rather that you read, then tried to make your own code based on my explanations and then read some more. I expect that you've made some mistakes, and that you have been able to rectify them, either on your own, or by asking the community at http://www.wiremod.com. This being the case, you should be capable of understanding and using the concepts that I present to you in this guide, however, if you realize midway through that you are not, do not fear, reread my last guide, and focus on the basics for a while longer before you move on; I split the guide for a reason, you don't want to cover too much ground at once.

Chat

The chat extension is an interesting set of functions that allow the E2 to find out what users are saying in chat. By typing 'lastSaid()' you can get a string which contains the last message in chat, and then using the string functions, you can dissect it and take different actions. I won't spend a lot of time on this, it's really basic, I just didn't want to focus on extensions in the basic guide. This is the list of functions for this extension.

Built-In Ranger

One of the unique things about the E2 is how expansive it is. The E2 can replicate the functionality of most other wire components, rangers included. To get ranger data, represented on the E2 wiki by 'RD', you can type a number of different things, but to detect data to 100 units off the top of the E2, type:

```
X = ranger(100)
```

In doing so, we just turned 'X' into a ranger data variable, which we can now find information from like so:

```
X:distance()
```

Finds the distance to the first entity in the path of the ranger or if it finds no entity, '100', the max distance. You can find out other information by using several other functions which can be seen here.

Sound Playback

Sound playback is another one of the extensions that allows you to mimic another wire component, in this case, the sound emitter. To create a sound, simply type:

```
soundPlay(1, 1, "path/file.ext")
```

The first number allows you to refer to the sound after it's been created, remember this value if you plan on changing the sound. The second, is the duration of the sound. The last one, is the local path to the file from your garrysmod/sound directory, the path must include the extension of the file, I know that the WAV format works and I'm fairly certain MP3s do as well. After a sound is created you can adjust it's volume and pitch or cause it to cease. To do so, run one of the functions listed here, using the aforementioned index to refer to the specific sound.