Make Your Own Game Tutorial IX: Events Part 2

Introduction

In Tutorial VIII, we covered a lot of theory on how to organize and structure events.

There are 2 major things that we learned in the last tutorial that are the most important:

1. The concept of storying information as switches (yes/no information) or variables (storing a number value).

2. The concept of flow control, the ability to have events do different things based on stored information by branching and looping.

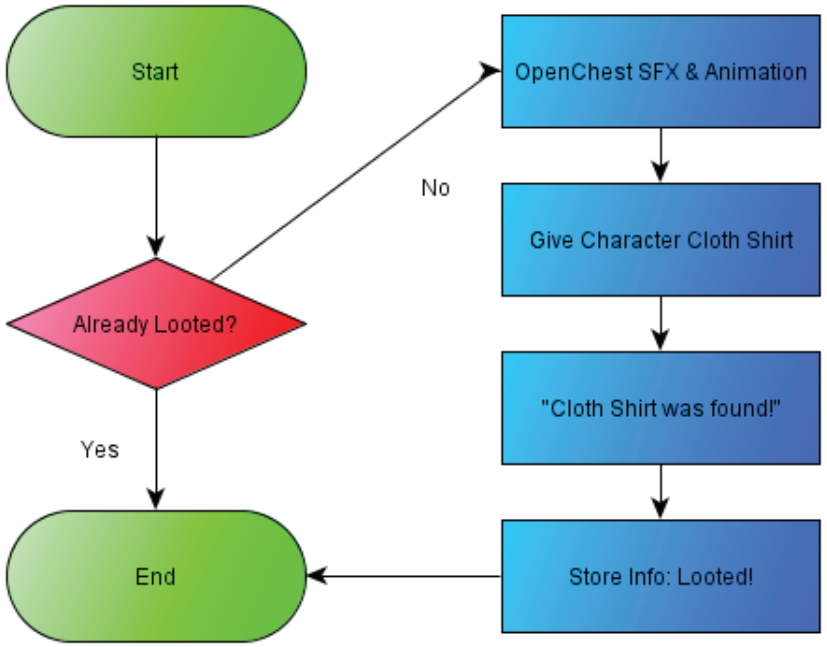
Clever applications of these to things will allow you to make almost any event you can think of. We understand them in theory, but how do we create these inside the editor?

In this tutorial, we will translate these two things into their counterparts in the event editor .

We will go over two types of Information Storage:

1. Switches

2. Variables



Flow Control with Switches from Tutorial VIII

And four types of Flow Control:

1. Conditional Branch

2. Show Choices

3. Loops

4. Label and Jump to Label

Make Your Own Game Tutorial IX: Events Part 2

Information Storage: Switches

Ok, so remember that Treasure Chest we worked on last tutorial? It uses what is called a Switch.

A switch stores boolean data. Boolean is just a fancy word meaning that it only has two states. These two states in the editor are referred to as ON or OFF. You can also think of it as YES/NO or TRUE/FALSE.

To change a switch we use the "Control Switch" Command, it is on the first page of the event commands under "Game Progression" (shown below).

Control Switches is incredibly simple, giving you an option to select the switch, and turn it off or on. (You can also select a group of switches using the Batch option).

Standard Switches are **global**. What this means is that there is one set for the entire game, and those switches can be accessed or changed by an other event in the game.

Switches are numbered, but you an also label them. To add a label, when selecting which switch to use during a process, type a name into the "Name" box.

There is also a second type of switch. This is called a Self Switch. Self Switches are **local**. This means that instead of being able to be accessed or changed by any event, they can only be accessed or changed in a local area. The local area in this case is within that single event.

To change a self switch, you use the "Control Self Switch" command in the "Game Progression" set, also shown in the image on the left of this page.

Each event has its own set of self switches, labeled A, B, C, and D. Remember that what you do to a self switch ONLY applies to that event.

Make Your Own Game Tutorial IX: Events Part 2

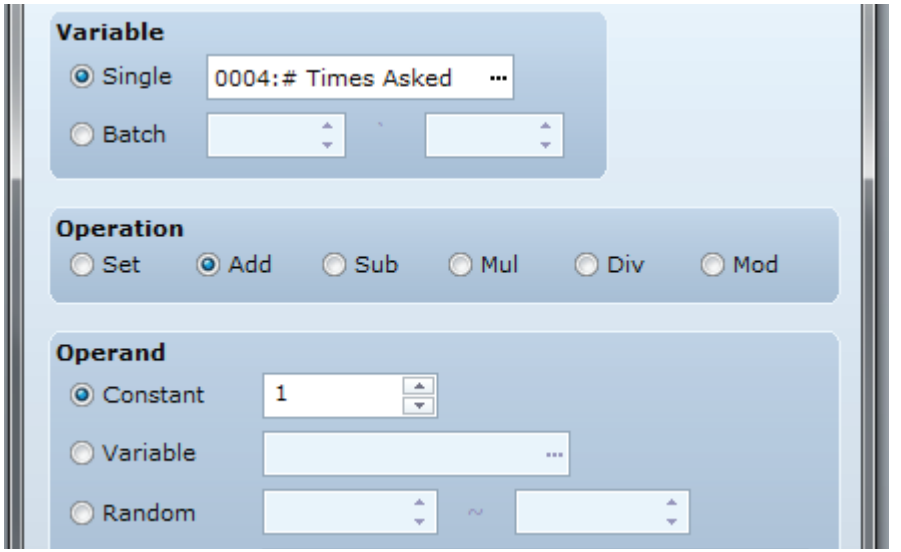
Information Storage: Variables

Variables are the other type of information we can store in RPG Maker VX Ace. For an example from our theoretical exercises in Tutorial VIII, you can look at Quiz Problem 4, where we stored the number of times we asked the Old Man his age.

Variables, rather than storing "Yes/No" information can store a whole integer, negatve or positive. You can change a variable using the "Control Variable" command, located between the "Control Switch" and "Control Self Switch" Commands in the "Game Progression" set of the event commands.

Because Variables are numbers rather than just yes/no information, we can do more than just set them. We can do math with them.

When doing a Control Variables command, we have 3 parts. The Variable (what variable you want to change), the Operation (what process you are using to change it), and the Operand (the number used in the Operation to affect the Variable).



Shown above is Variable 0004 (labeled as # Times Asked) using the Operation "Add" and the the Operand of "Constant: 1".

What this does is take Variable 0004, and then add (the operation) 1 to it (the operand). You can add, subtract, multiply, divide, divide and save the remainder, or just set the variable to the Operand. You can use all kinds of things as your Operand, from random numbers, to the numbers stored in another variable to things from the Game Data.

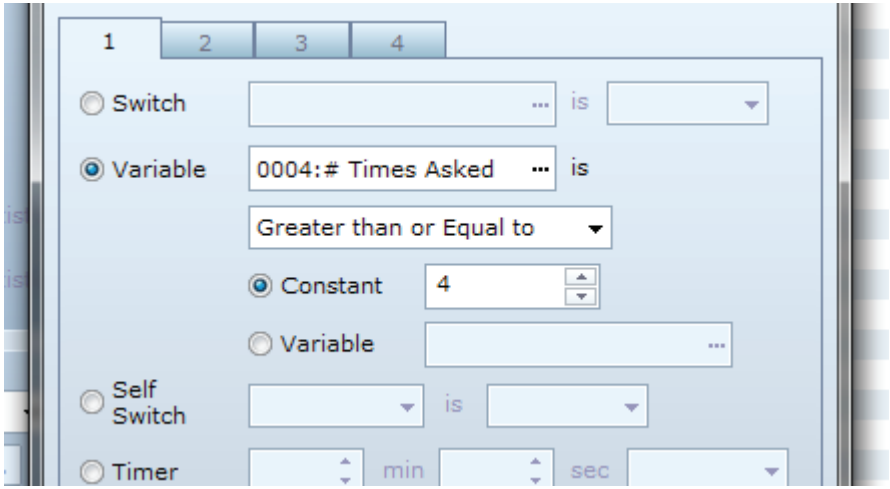
Unlike Switches, all variables are **global**. There are no local switches.

Make Your Own Game Tutorial IX: Events Part 2

Flow Control: Conditional Branch

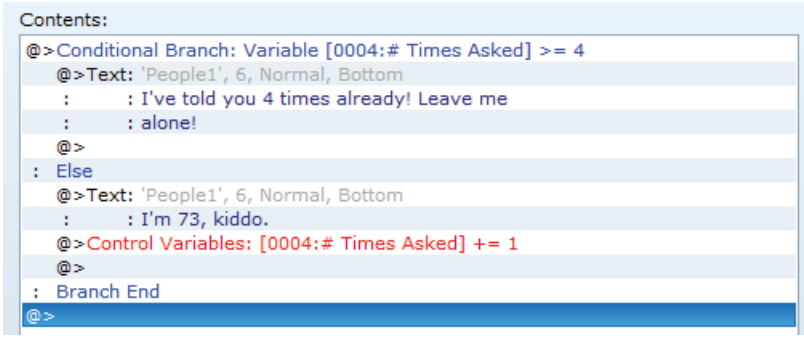
The Conditional Branch is one of the simplest forms of Flow Control in RPG Maker VX Ace. A Conditional Branch takes a piece of information, either from the Game Data, Switches, or Variables, and determines whether something about it is True or False and makes the event do different things in each case.

The Conditinal Branch command is found in the first tab under the "Flow Control" set. Inside you will find all kinds of possible "questions" you can ask the conditional branch to check. Below, I have it set to check if we have asked the Old Man four or more times how old he is.



Once you have selected the conditions, you are taken back to the event. There you will see indented lines, one after the conditional branch command, and one after a line saying "Else". You will also see a "Branch End" line.

Anything you want to happen if the conditional branch is **true**, put directly after the conditional branch command. Anything you want to happen if it is **false**, you put directly after the "Else" line. The Branch End line closes the conditional branch, and anything after that will happen regardless of whether it is true or false.



Above, if the variable is 4 or above, he will say the top line, if the variable is less than 3, he will say the bottom line and it will add one to the variable. If you wanted something to happen afterwords either way, you would add it on the highlighted line.

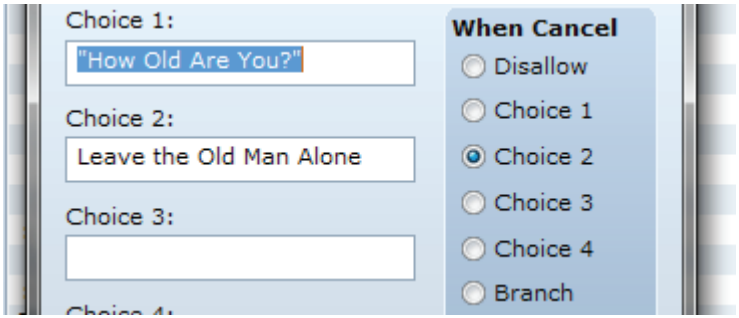
Make Your Own Game Tutorial IX: Events Part 2

Flow Control: Show Choices

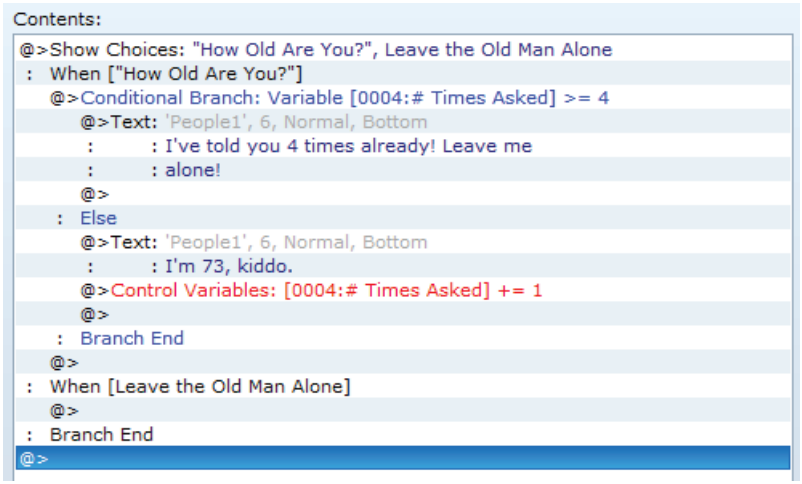
This one actually isn't listed in the Flow Control set in the event commands, but its still very much used to control the flow of an event, so we will include it here.

Show Choices lets you give the player up to four choices to select from, and then has the event behave based on the selection chosen.

It can be found in the "Message" set at the top of the fist tab in the Event Commands.



On the left, you are given the option of putting in the choices, while on the right, you can check which option to select if the playre hits the cancel button. If you leave an option name blank, it will not show up to select in game.



Like with the Conditional Branch, you place the things you want to happen within the indented lines after the choices lines. In the above, the conditional branch starts if you ask him his age, and the event does nothing and just carries to the end if you choose to leave him alone.

As with the Conditional Branch, you can also put things after the branch end line if you want it to happen after the rest is processed no matter the choice selected.

Notice that you can nest flow control options inside each other. Always keep an eye on your indentions and the number of Branch Ends you have to make sure you are inserting commands on the right line!

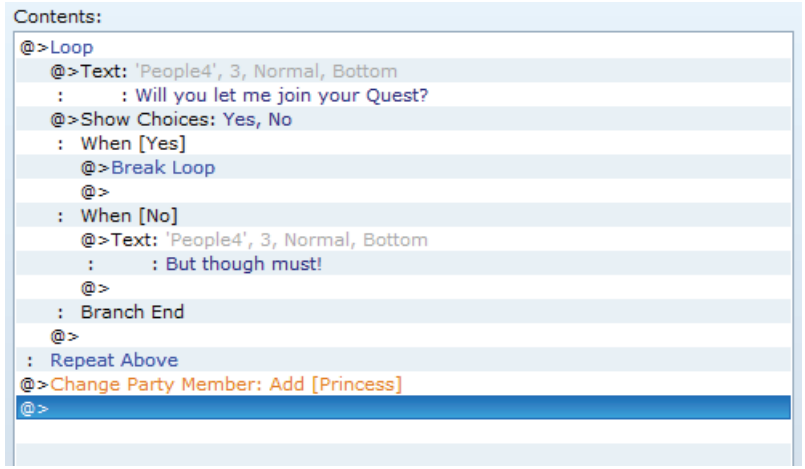
Make Your Own Game Tutorial IX: Events Part 2

Flow Control: Loop

Loops are a different kind of flow control. Rather than do something based on a condition, they just repeat a set of commands inside the loop until something breaks the loop.

The Loop command is in the "Flow Control" Set. There are no options, once you select it you will get a line labelled "Loop" and a line labelled "Repeat Above".

Everything you put between these two commands will loop. You can break the loop (usually using some other kind of flow control such as a conditional branch or show choices) using the Break Loop command.



The Break Loop command is also found in the "Flow Control" set. Placing a Break Loop command will cause the event to skip the rest of the commands in that loop and jump directly to the next command after the "Repeat Above" line.

As you can see on the right, the Princess will continue to ask to join your quest as long as you select no. When you select yes, the loop breaks and it will skip to the line directly after Repeat Above. In this case, it skips to the line that adds the Princess to the party.

"But I dun' wanna!"

Make Your Own Game Tutorial IX: Events Part 2

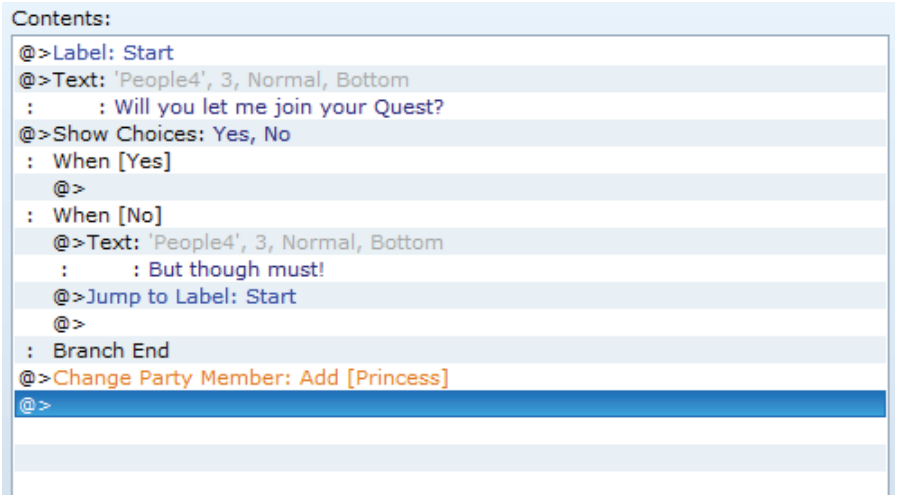
Flow Control: Label and Jump to Label

Like Loops, Labels and Jump to Label commands do not by themselves check any information stored.

Labels by themselves do nothing. You place a label in your event and it just sits there with a name that you gave it. Pretty boring huh?

But that is where Jump to Label comes in. Whenever a Jump to Label command comes up, the event jumps and starts processing back wherever the indicated Label was. At the most basic level this can be used to create Loops, but it can also be used in combination with conditional branches to skip parts of an event, to jump back to a specific point in the event, etc.

The uses of this might not seem as obvious as some of the others we covered, but at some point you will be making an event and suddenly this little feaure will jump back to you. Technically, everything CAN be done with just loops, conditional branches and show choices, but Label and Jump to Label commands can create huge shortcuts in your events.



For the example, we will recreate the exact same circumstances that our Loop example created, but this time, we will do it with Labels and Jump to Label commands.

Instead of starting with a Loop, I placed a Label named start at the beginning.

When the player selects Yes, it skips over everything in the “No” condition, so it goes directly to the end of the Branch.

But instead, if they select No, it jumps back up to the Start Label!

Make Your Own Game Tutorial IX: Events Part 2

Let's Make an Event

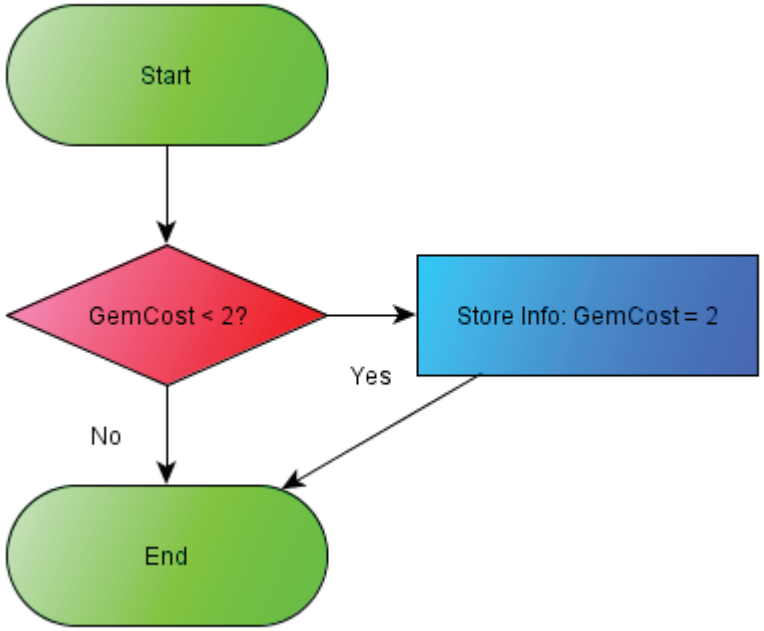
So now that we know all the pieces, let's go all the way through creating an event for our game.

So I've decided instead of using treasure chests that unlock with gems, I'm going to have the special elemental weapons in our game provided by a fairie who turns elemental gems into magic weapons.

Instead of having a flat cost of 3, I'm going to have the first one you buy cost 2, the second one cost 3, and the fourth one cost 4.

Now, we need to start working on an event that will do that. Let's start by doing what we did in the last tutorial and write it out using a flowchart. Because we are more aware of the options though, we can better structure our flowchart using conditional branches, loops and show choices.

The first thing we need is a variable that will control the cost! And it needs to start at 2. So let's put a routine at the beginning that sets it to 2 if the variable "GemCost" is equal to less than 2. Because it rises from there, the only time this will be the case is the first time you talk to her.



The next thing we need to do, is give her some kind of introduction spiel. This will tell the player how the trading system will work and introduce the character.

I'm going to have it play through every time we talk to her, so we can place it right after the GemCost<2 conditional branch.

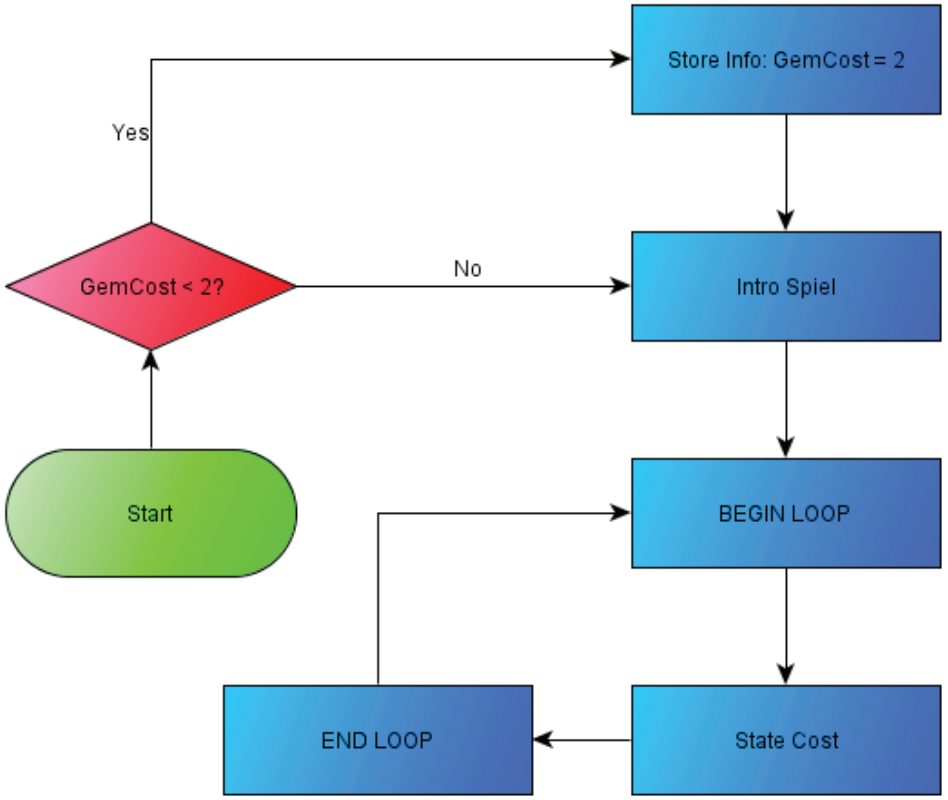
From there, we can start the Loop that will be the meat of the event.

Make Your Own Game Tutorial IX: Events Part 2

The Loop

Now, let's place a Loop. Inside the Loop, let's have her give the current price in gems of the next weapon.

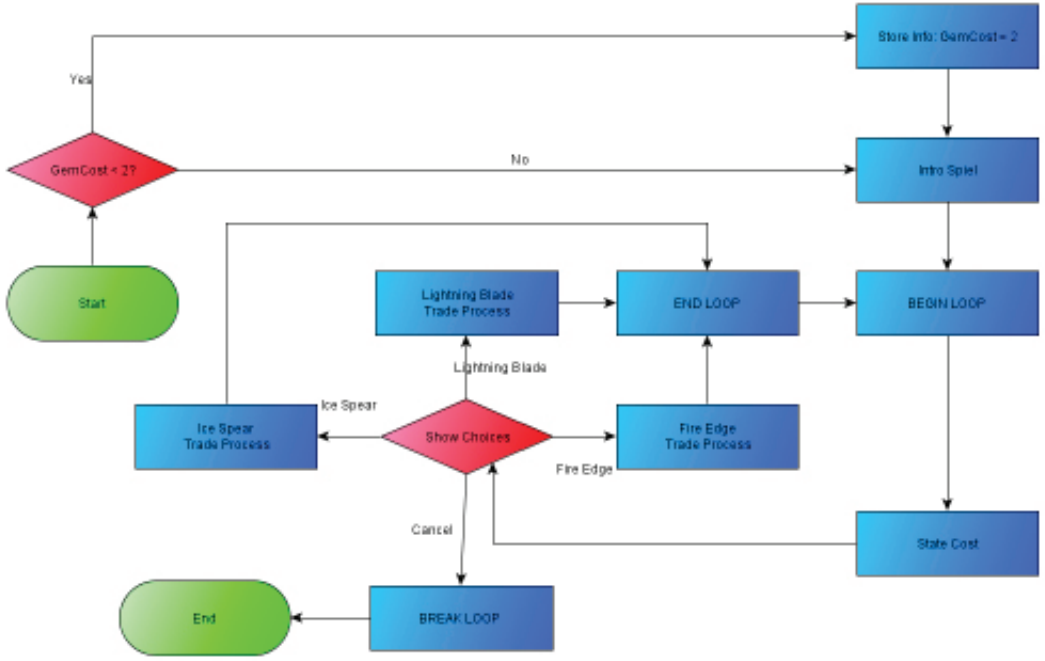
Everything else we add will be add on the inside of this loop.



Show Choices

Obviously, show choices is the next step. Add it in to allow the character to pick which weapon they want. Add in a Cancel option to break the loop. Let's add in an abstracted trade process to the others that we will break down in the next step.

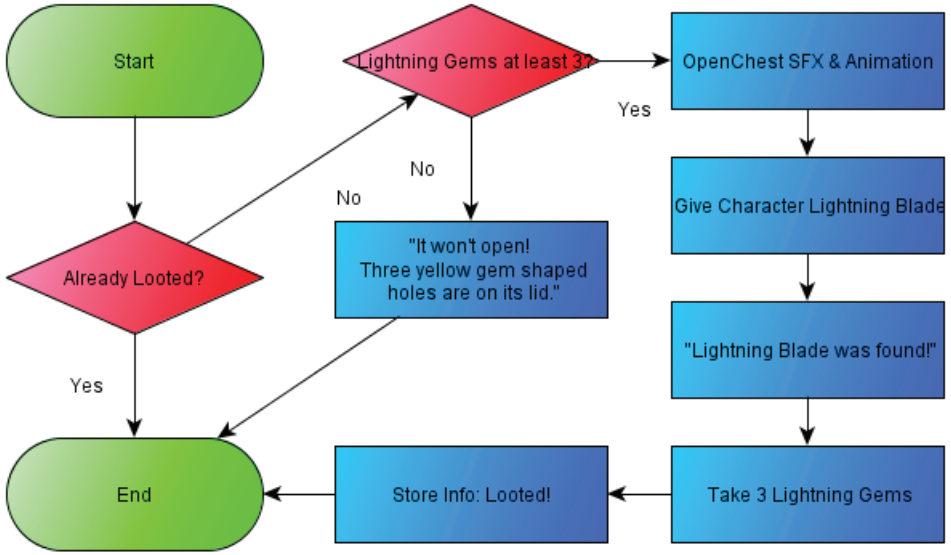
Instead of continuing to use the full chart, we will map the trade process on a separate chart.



Make Your Own Game Tutorial IX: Events Part 2

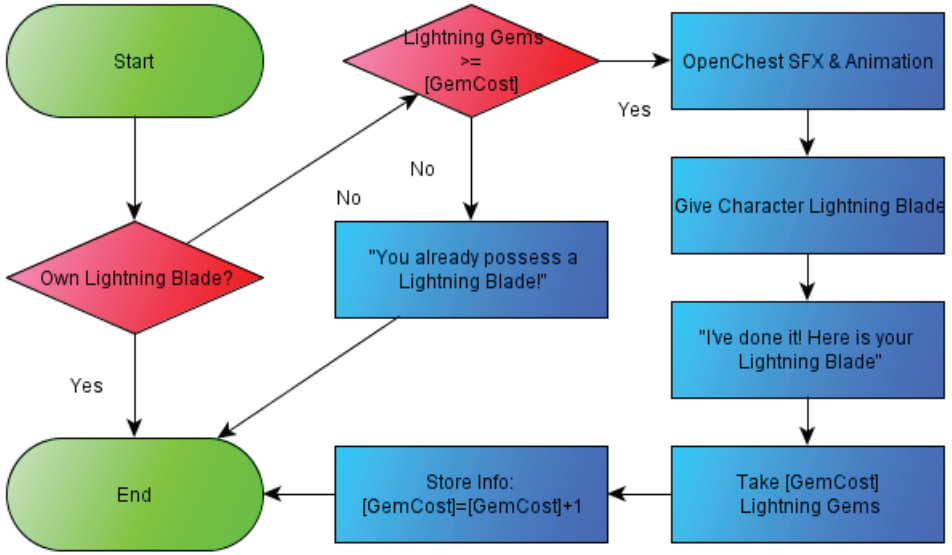
Trade Process

Instead of doing this step by step, let's go back to the one we made in the last tutorial.



The only changes that need to be made is changing the message, changing the 3 to GemCost, and having it raise the GemCost by 1 after purchase.

I am also going to remove the store info for whether it is looted, as I'm instead going to track it based on whether the character owns the weapon or not.



We now have the trading process for the Lightning Blade, which can be directly plugged in to the Show Choices flow chart above in the "Lightning Blade Trade Process" box*.

The Ice Spear and Fire Edge trade processes are identical, just change the Weapon and Gems used.

*Note: Its important sometimes to break up huge flow charts into multiple chunks, it lets you think in smaller bits, rather than trying to manhandle the whole thing.

Make Your Own Game Tutorial IX: Events Part 2

Mapping to Event Commands

So we know what each of the flow control commands are, but let's start mapping some other parts.

The intro spiel is obviously just a show text command, something we have used in a previous event in an early tutorial. As a note: When you show the cost, you can put \V[n] with the variable number in place of the n to have it replace it with the value of the variable.

To change the number of items in a players inventory you use Change Item for the gems (in the Party set of the first tab), and Change Weapon for the weapons (also in the Party set).

One thing we have to work around is the conditional branch can't be set to more than variable items in inventory. Instead we have to add in an intermediary step where we save the number of that item to a variable.

Attached in the zip file is a demo that has this event, as well as some crystal events to give you the necessary gems. Read through the event, play around with it a bit, and remember, keep practicing.

Tutorial Wrapup

In this tutorial, we covered two types of Information Storage. Switches, which hold YES/NO style information and Variables, which hold whole integers.

We also covered Flow Control such as conditional branches that branch off events based on game data and show choices, which branches based on player choice.

We also covered Loops and Labels/Jump to Labels, which give you more power to change how events play out.

Finally, we did a single complex event that used most of these commands to trade gems for weapons in our sample game.

Next Tutorial Preview

In the next tutorial, we will go one step further, and learn about event pages, event triggers, move routes, and many other methods to make more complex and powerful events.

Until next time!