

Make Your Own Game Tutorial VIII: Events Part 1

Opening Thoughts

With this tutorial, I am going to be doing something slightly different than I did with all the others. We will not be opening our sample project during this tutorial, instead we will focus entirely on theory and the basic commands available to use.

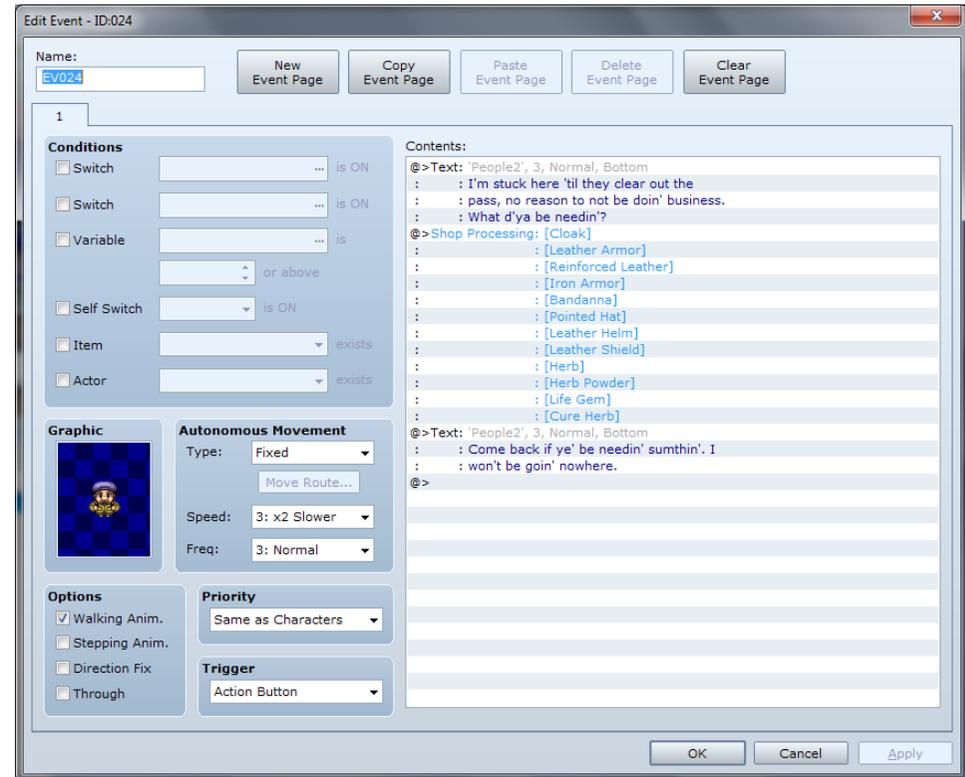
In Events Part 2, we will then apply this theory to our project, fleshing out, and leaving only polishing up the sample game between us and being finished.

Events: What do they do?

Events are used for two major purposes in RPG Maker VX Ace.

The first is to provide objects for the player to interact with. This can be an NPC for the player to talk to, a switch for them to pull, or even just something to move your characters from map to map.

We've already made several simple events that fulfill this purpose, such as the Merchant we made in the fourth tutorial shown to the right.



Simple Event made in Tutorial IV

But there is also a second, very important role for events is controlling game flow. For instance, how do you determine where a character can and can't go, how do you open up new areas for him to explore.

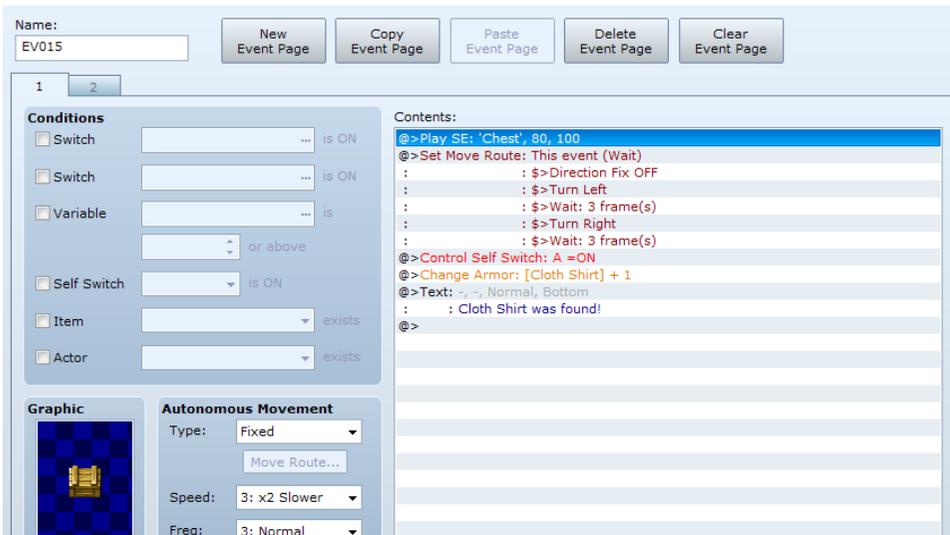
All of this is done with events.

Make Your Own Game Tutorial VIII: Events Part 1

Approaching Event Design

Event design is all logic. There are lots of tricks and tips you can pick up over time, but the best tool you will ever have is your own brain. What I hope to teach in this tutorial isn't how to create a specific type of event, its how to walk yourself through ANY event.

To start with, we are going to be designing the treasure chest we created in Tutorial IV, except this time, we are going to do it from scratch, and we are going to go step through step of the logic it takes to create it.



Creating the Treasure Chest

The first thing to do when planning an event is to think, in plain speak: What does this event do?

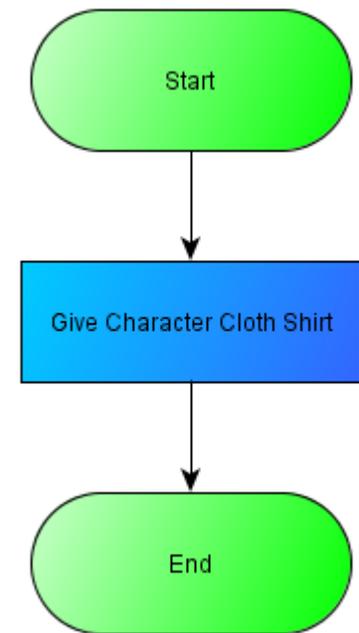
With a treasure chest, this is a pretty simple thing to describe: Player opens it, gets an item. The item given with this one is a cloth shirt.

To illustrate event flow I'm going to be drawing each step as a flowchart.

I encourage everyone to use flowcharts in their event design, as it will give you an excellent idea of how to organize your events.

To the right is the simple plain text explanation of what our event does. The event starts, it gives a cloth shirt to the player, and then it ends.

Do you see the problem with this flowchart?

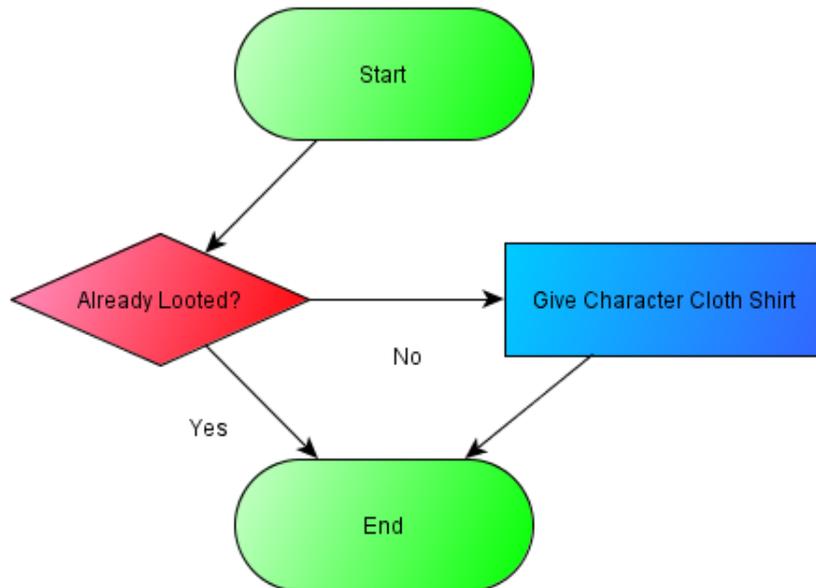


Make Your Own Game Tutorial VIII: Events Part 1

Creating the Treasure Chest (Cont.)

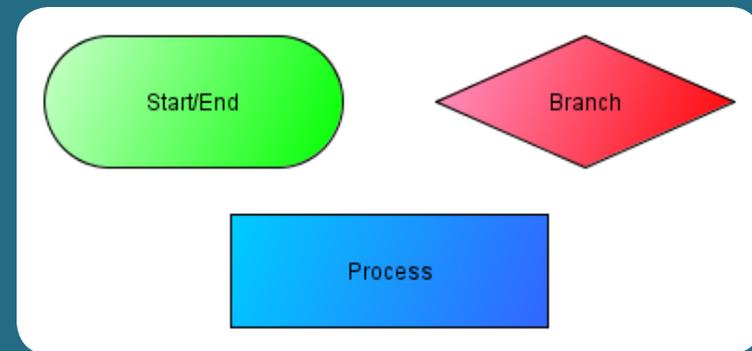
That's right! Every time the player activates this event it would give him another shirt. Infinite shirts would be useful, but would break the game. The player could sell them for infinite money!

So we need to add a branch near the beginning of the event that checks if the chest has already been looted. Let's not worry about how this is one in the editor quite yet, instead, in plain language, let's put the branch into our flowchart.



Flow Charts!

Flowcharts are an excellent resource to use to actualize events. A lot of the ways I am using them are the same as how programmers are taught to begin thinking of programming. There are many flowchart shapes, but I am only using the three listed below combined with arrows.



You can find info on many more shapes if you are interested by searching the internet.

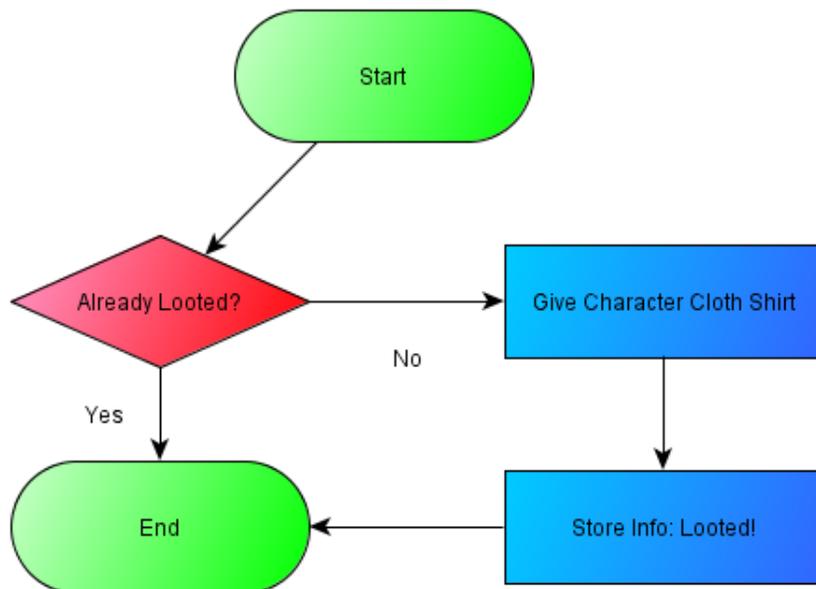
Also, all of the flowcharts made in this tutorial were made with the freeware program yEd.

Make Your Own Game Tutorial VIII: Events Part 1

Creating the Treasure Chest (Cont.)

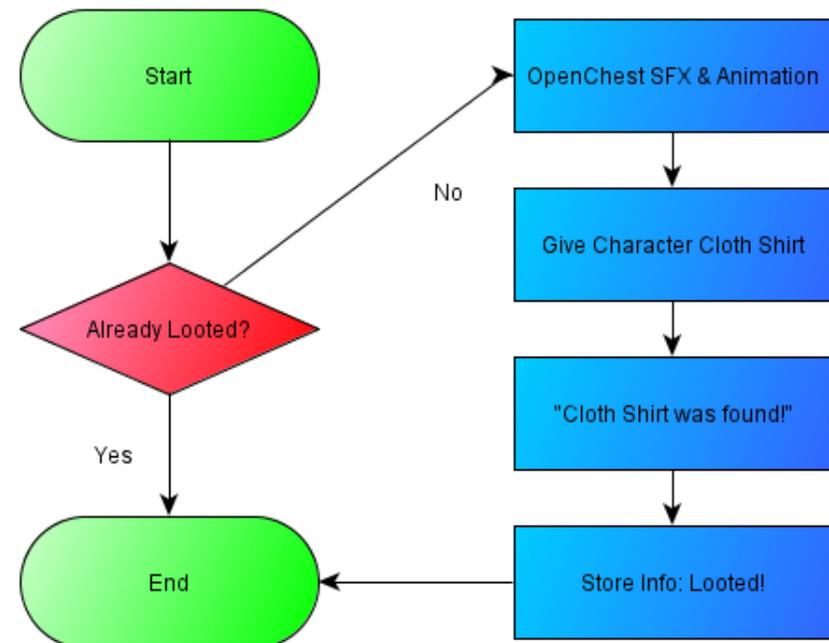
This flowchart works perfectly for us, because we remember that it was looted already. But a computer doesn't know to remember this unless we tell it to remember it.

Once again, we won't worry about how this is done in the editor. All we need to know right now is that it has to be told to store the information about whether it was looted. Its easiest just to fill this in as the last thing the branch on the "No" side does before ending the event. You can see this version of the flowchart below.



We are nearing the end of our plain text version of our event. Right now, it will perform the game mechanics portion of what we want it to do, but it doesn't actually visually do anything on the screen.

What we need to do is add in the sound effects, treasure chest opening animations, and a message to the player that he received something. All of this, of course, needs to be stuck in the "No" branch.



Make Your Own Game Tutorial VIII: Events Part 1

Quiz!

As you can see, plain description and flowcharts can put you in a good position to start plotting out your events in the editor.

Here, because the main purpose of this tutorial is to teach **process** rather than how to do a specific thing, I'm going to do something a little different and give a little quiz. Don't worry, its not going to be graded!

On this page, I will describe several event scenarios and ask you to draw your own flowchart (either by hand or on a computer) to show how that would be accomplished.

Once again, knowing the exact commands is not important. What is important is understanding the logic of how the program runs through events. Use plain English, and don't get hung up on whether I write things slightly different. The main point is to see how well your flowcharts match mine, and if they don't, if it would still accomplish the same thing.

The following pages will show the answers to these scenario.

1. Recreate the Dragon Quest I "But Though Must" scene: The Princess asks to join you, if you say no, she says "But Thou Must" then asks you again. This will happen as long as you continue to say no. If you say Yes, she joins the party. (Remember that the Princess shouldn't be able to join twice!) Answer Page 6

2. Using the treasure chest flow chart we created as a base, create a flow chart for a treasure chest that will only open at the cost of 3 Lightning Gems from your inventory, that will give the character a Lightning Blade. Answer Page 6

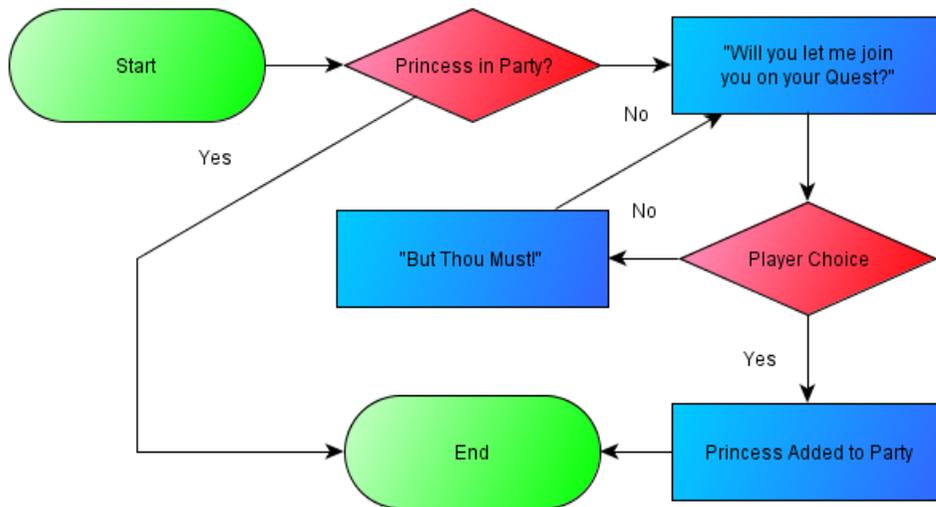
3. Create an event that allows you to do one of the following two trades: 2 Steel Bars for a Katana or 1 Yarn for a Magic Sock. Make sure to have it give "not enough" messages and the option to loop around to the beginning after you are done. (Note: These items are not used in our game, they are entirely hypothetical) Answer Page 7

4. Create an event where you can ask an old man a question. He answers 4 times, then refuses to answer the same question again, telling you to leave him alone. Answer Page 8

Make Your Own Game Tutorial VIII: Events Part 1

Quiz! Answers

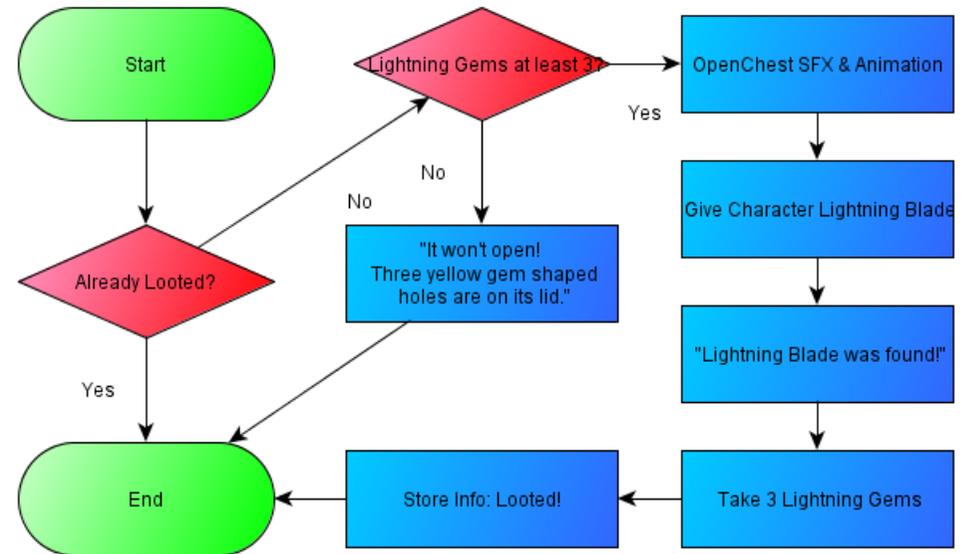
1.



The important thing here is to make sure that we loop back to the initial question every single time the player says no.

You have two options on preventing it from happening again. The first is what I used, just check if the Princess is in the party. This doesn't work if she is removed later though! So another option would have been to Store the info that the event didn't need to happen again.

2.



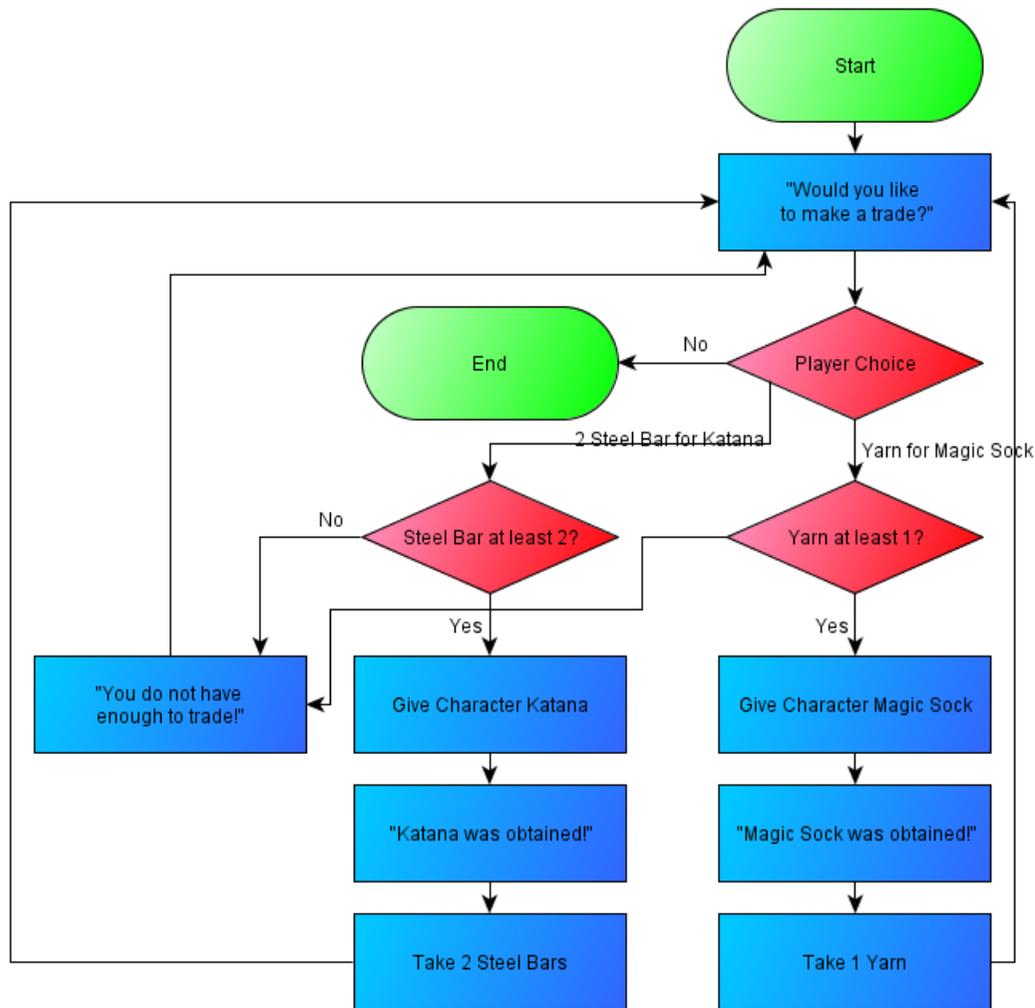
First things first, we need to add an additional branch to check whether the player has enough Gems. This needs to be after the check to see if you've already looted the chest.

Additional additions are the inclusion of taking 3 lightning gems from the player (otherwise it isn't a cost), and a message giving a hint to the player of how to open the chest. You could also add in a SFX that plays when you try to open the locked chest if you so choose.

Make Your Own Game Tutorial VIII: Events Part 1

Quiz! Answers (Cont.)

3.



This one is a little more complicated, but it really only takes combining the things we've learned in questions 1 and two.

First, we need to set up a question message and a player choice, just like we did in the "But Thou Must" quiz problem. After that, we have to set up some more branches to check if we have the proper materials. This is very similar to checking if we had the Lightning Gems in the Lightning Blade quiz problem.

When that is checked and shown to be false, tell the player that he doesn't have enough and scuttle back up to the starting question. You could loop it to the end from here as well, but that would make the character have to start the event again to do the other trade.

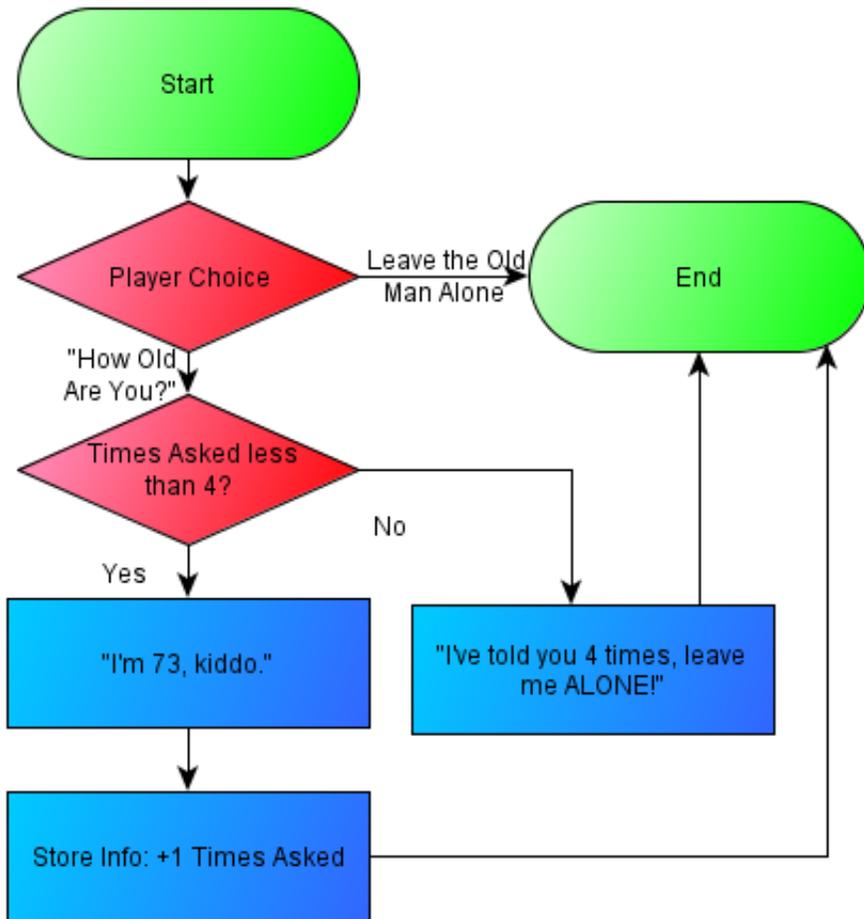
Handing out the items is handled the same way as in the Lightning blade quiz problem. Give the item, tell the player, take the trade items.

Since the event should happen as many times as the player wants, there is no reason to put in a condition to end it immediately after start.

Make Your Own Game Tutorial VIII: Events Part 1

Quiz! Answers (Cont.)

4.



On problem 4, I threw a little bit of a curve ball at you. Its a relatively simple event, but it requires a bit of information I hadn't covered yet.

Did you intuit the simplest way to accomplish this event? The key is knowing that we can store more than just yes or no information.

We can also store numbers. And not only can we store numbers, we can manipulate them using basic math. We use this to just add +1 to the Times Asked every time the Player asks him how old he is.

Quiz! Wrapup

So how did you do on the Quiz? Were you able to create flowcharts that were similar to the ones I made? If they didn't look like mine, did they still work?

Don't worry too much if you didn't do well. Eventing logic just takes practice! Keep charting new event ideas and you will get the hang of it!

Make Your Own Game Tutorial VIII: Events Part 1

Practice! Practice! Practice! and Moving into Scripting

With Eventing logic, the key to improving both efficiency and how complicated of events you can make is nothing but practice and pushing your boundaries.

This process of design and charting is the same used to teach people computer programming. This is because eventing is basically programming simplified.

You use a lot of the same concepts: Get input, Store Information, Manipulate Information, Retrieve Information, show output. The difference is that with eventing, you have a number of defined event commands to control your events with, allowing you to worry less about about language syntax.

The good news though is that all this information is also applicable to programming. The better you get at eventing, the more you will understand programming logic. Good eventers can generally move into scripting with ease.

Just remember, KEEP PRACTICING.

Tutorial Wrapup

In this tutorial, we spent most of our time going over theory. Even though we didn't touch the editor itself, we spent a lot of time learning the processes by which events flow.

We also went over the idea of storing information for later use. Both Yes/No and numbers.

We learned how to diagram this flow using flowcharts, an invaluable tool to organize what you are going to put down in the editor later.

You also got the opportunity to try out the knowledge with a fun little quiz.

Next Tutorial Preview

In the next tutorial, we will take this theory, and use it to create events, mapping the plain English descriptions to Event Command. We will also learning about using Event Pages vs Conditional Branches, as well as Switches vs Variables.