Creating a Statue Puzzle

Written by Nathanial Benton for RPGMakerWeb.com

Hi there! My name is Nathanial. Welcome to my first tutorial on creating puzzles for RPG Maker VX!

In this tutorial, I assume you already know your way around the editor, and how to import character graphics via the Resource Manager. I will be showing you the following things major things:

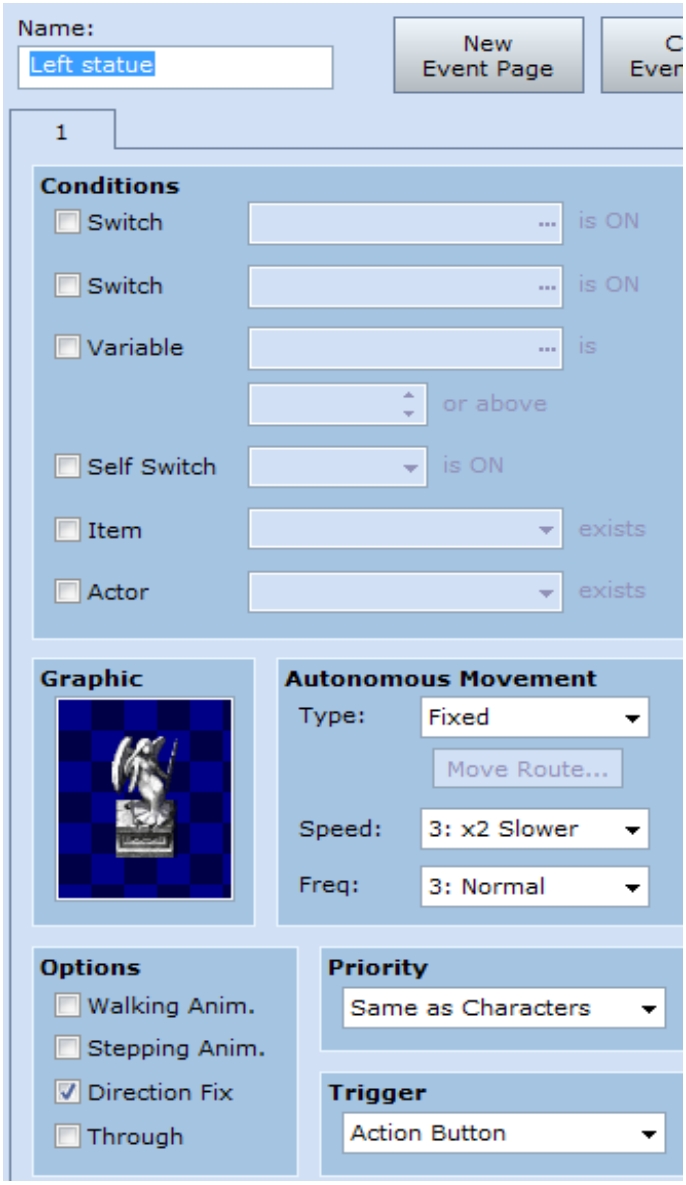
- The usage of variables to keep track of the position of two statues.
- Using several conditional branches to compare the value of of several variables.

First, we will draw a map. It doesn't need to be fancy, just something to get the job done for learning purposes. Notice the two tiles that seem sort of out of place out doors. The indoor floor tiles. They will be important in a little bit.

Okay, so you've drawn yourself a map. Let's add a closed door Event, that doesn't respond to the player. Be sure you also have a couple of markers like I do, with the indoor floor tiles as seen above. The next step is to place a couple of statue Events. I've created an image for you to use inside the program. Save it and name it something like !$Statues. Leave the extension .PNG. It's very important that you have the !$ at the front of the name. The image contains three of the statues found in the RPG Maker VX Run Time Package (RTP).

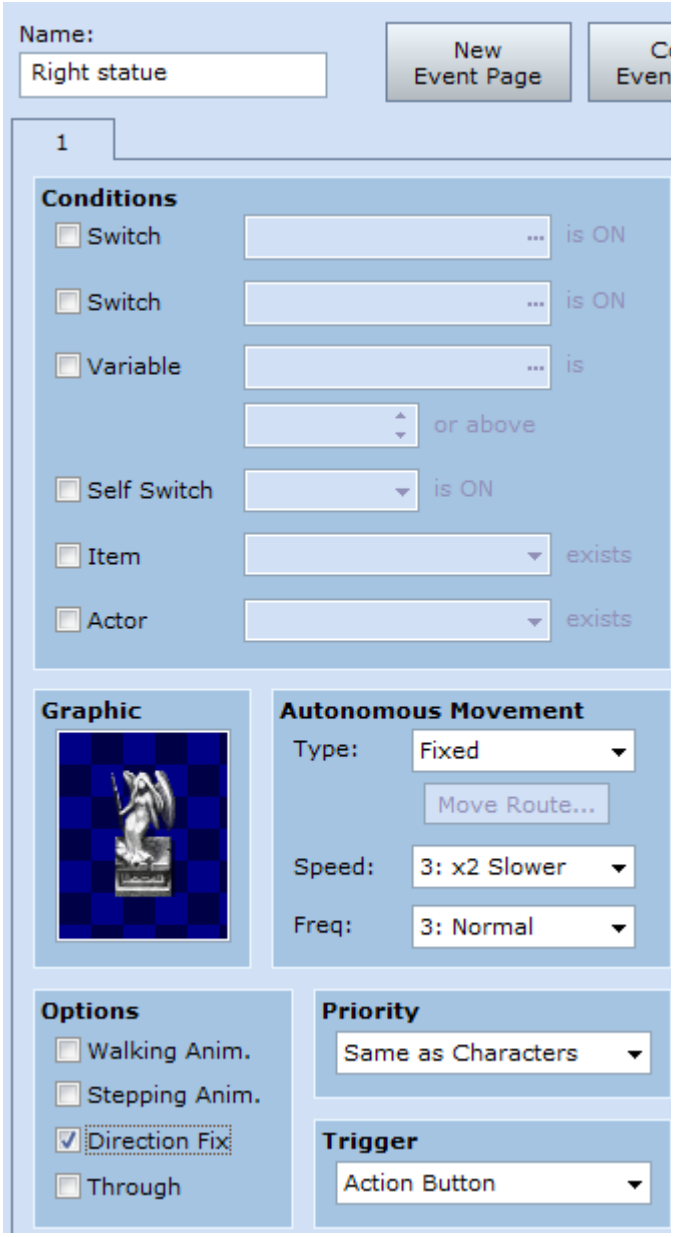
*The image contained in this PDF is not suitable to be used. You can download the graphic file off the RPG Maker Web website.

After you have imported the image under the Characters section, create an Event and choose one of the statues facing left or right, as the graphic. Be sure everything under “Options” is unchecked except for “Direction fix”. This is very important to reduce the chance of an ugly bug.



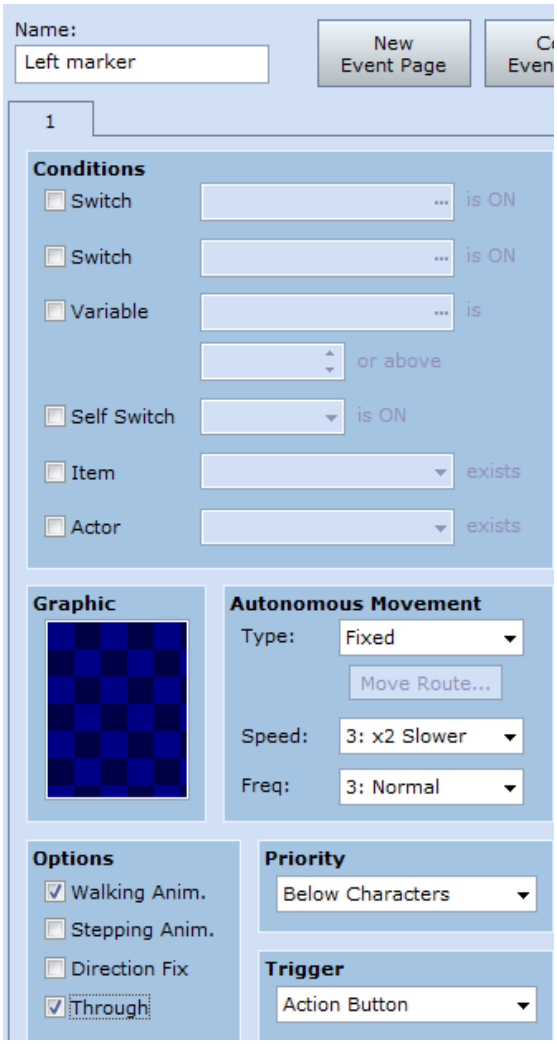
You might have noticed that I named this statue “Left statue” when it's facing right. The reason for this is, this is the statue that will be on the left hand side of the screen when completing the puzzle. If it's easier for you to name it “Right statue” I suggest you do so.

After creating the Event, repeat the process stated above, this time selecting the statue graphic facing the opposite way as the Event you created before.



Okay, great! We're on our way to making a neat puzzle. We still have a lot of Eventing to do. Remember the markers I had you place before? Those are going to come in handy now. What you need to do is create an Event on each of them. Be sure to name them a proper name, so you can tell what Event it is. I'm going to be naming the marker on the left hand side of the screen “Left marker” and of course, as you've probably guessed, the marker on the right hand side “Right marker”.

Leave the graphic blank, and make sure the priority is “Below Characters”. Under the “Options” portion of the Event window, make sure “Through” is checked. Here is a screenshot of my “Left marker” Event:



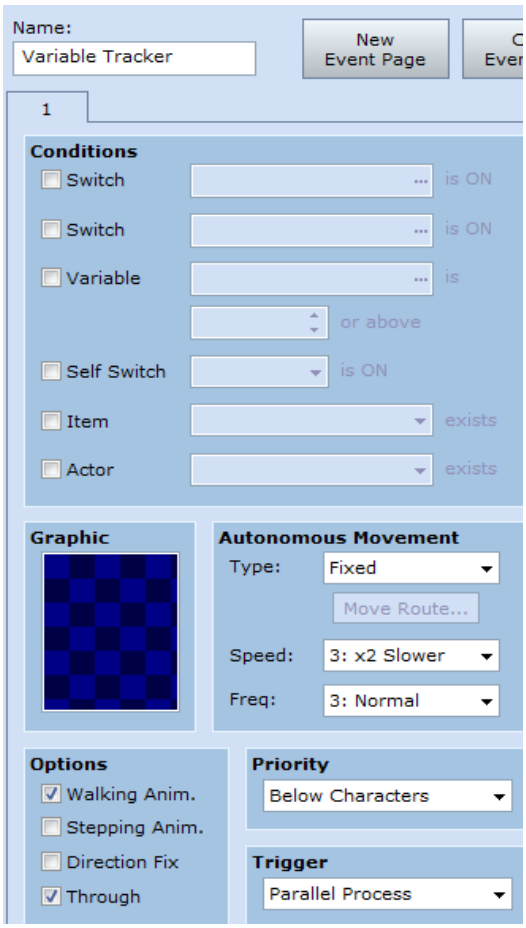
Here is a screenshot of my map so far, with all of the Events placed:

Nice and simple. That takes care of the easier part. Here is where things start to get slightly confusing. I will provide plenty of screenshots to help you along the way. You will notice that I will be using the "Comment" function a lot. The Comment function serves no function to the game at all. It's just a helpful feature for you, the developer, to keep track of what things do.

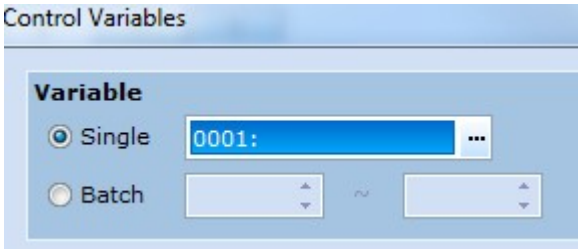
I strongly suggest when you see me add a comment, you do the same. This will help you keep track of what all these variables we're going to be adding do. So take a deep breath... and let's make a cool puzzle!

The next step for us to do is create an Event that is a “Parallel Process”. A Parallel Process is always running in the background. Whatever you set the Event to do, as long as it's a Parallel Process, it's going to loop. What makes it different from an Autorun Event, is that Parallel Process allows the player to move their character around and play the game. An Autorun Event stops the players movement, and disallows them to open the menu, among other functions.

As a Parallel Process, we're able to keep track of some variables silently in the background, to see if the player has finished the puzzle or not. So let's create the Event that's a Parallel Process, and make sure the actual Event square is somewhere the player cannot walk. Nice and out of the way for silent background tasks. Let's name the Event “Variable Tracker”.



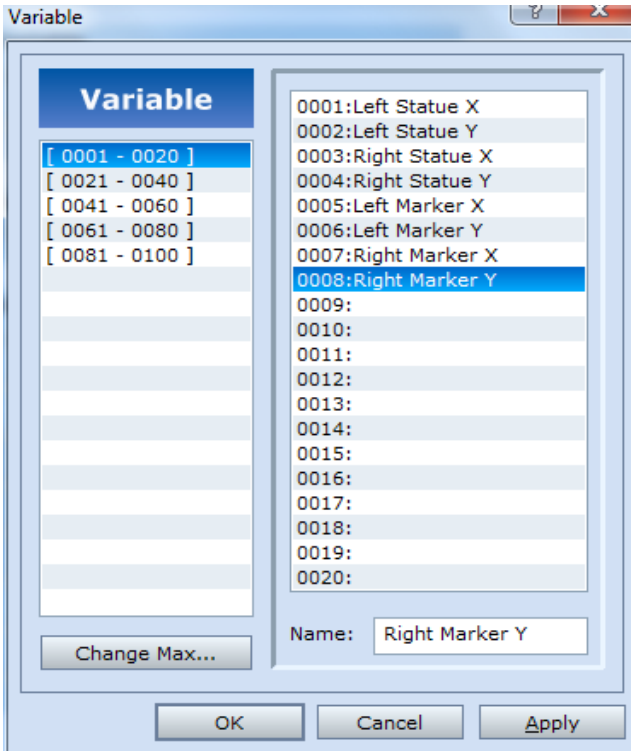
The next step is to create some variables. Under the “Event Commands” page, let's click the button “Control Variables”. Next click the ellipsis button on the right hand side of “Single”.



If you've ever used Switches before (which you should have by now) you'll find a familiar screen. We're going to name eight variables. We should name them something easy to identify with their function. So I'm going to name them the following:

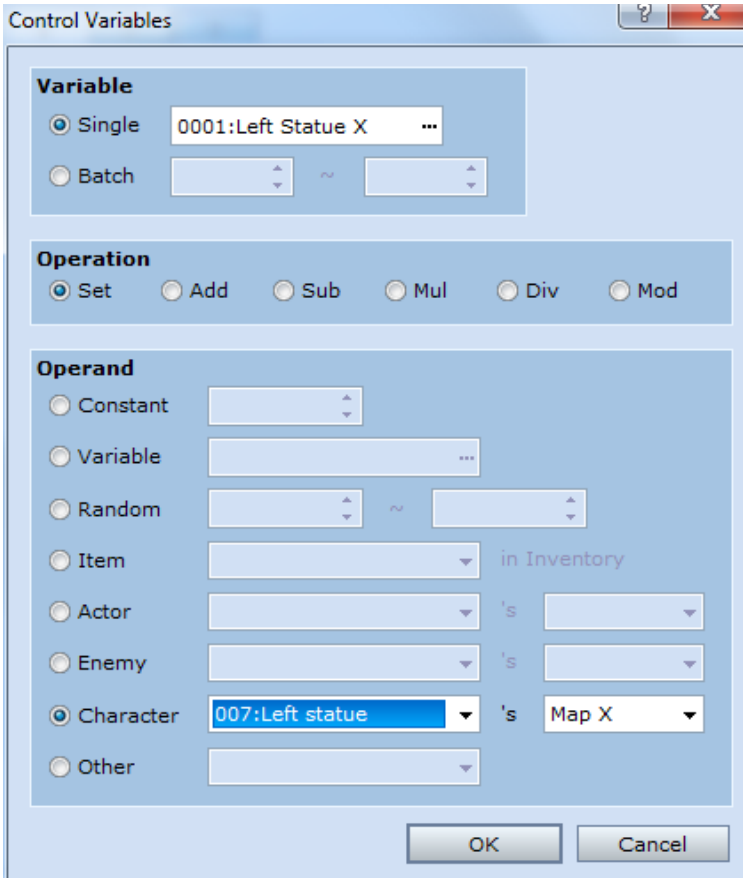
1. Left Statue X
2. Left Statue Y
3. Right Statue X
4. Right Statue Y
5. Left Marker X
6. Left Marker Y
7. Right Maker X
8. Right Maker Y

Just like so:



Now, let's select the first variable “Left Statue X” and click the OK button.

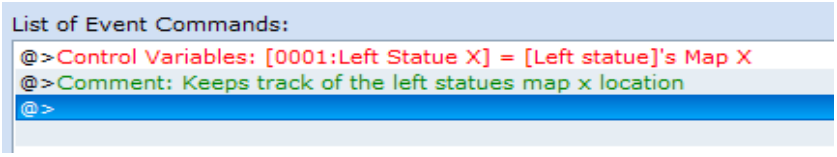
Next select the radio button “Character” and choose “Left statue” from the drop down menu. Make it set to “Map X”. Just like so:



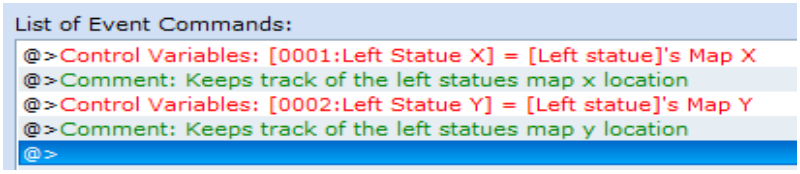
Click the OK button. Next I'm going to open the Event Commands window again, and select “Comments”.

In the comment box, let's type something like “Keeps track of the left statues map x location”. Click OK

You will now see something like this:



We're now going to repeat the above process for the other variables we created. So open the Events Command window, and select Control Variables, click the ellipsis button, and choose “Left Statue Y”. Choose the radio button “Character” again, and choose the Left statue Event again. This time, from the drop down menu on the right, choose “Map Y”. Click OK. Add a comment such as “Keeps track of the left statues map y location”.

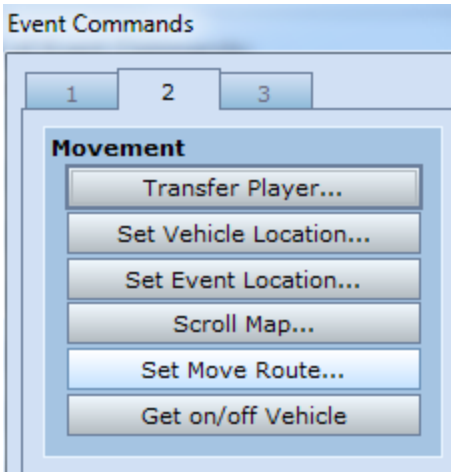


Repeat this process for each other variable we've created. The Right Statue X variable will be tied to the Event Right statue's map x, etcetera. Here is what my Event looks like now:

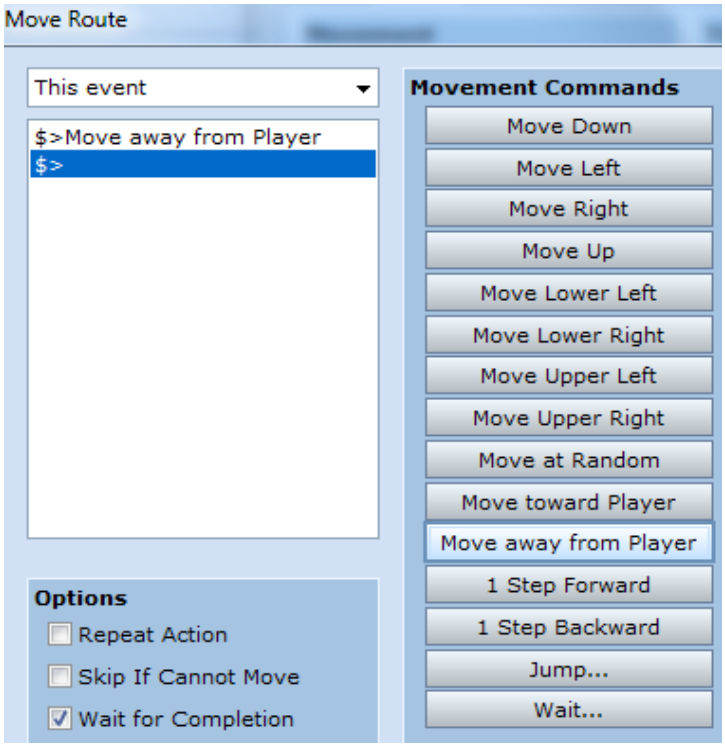
What we've established so far, is an Event that runs silently in the background, without the players knowledge. It's consistently keeping track of the X and Y coordinates of the Left statue Event, the Right statue Event, the Left marker Event, and the Right marker Event. If the Left statue Event is pushed later, then the game will automatically update the X and Y coordinates of the statues location. We're going to use these variables, which are always updating the coordinate locations of the statues, as a way for the game to know when the statues are pushed on the correct tiles, and facing each other properly for the door to open.

It might sound intimidating, but it really isn't! So let's continue on.
Next let's make our player be able to push our statues! Let's open up the Event

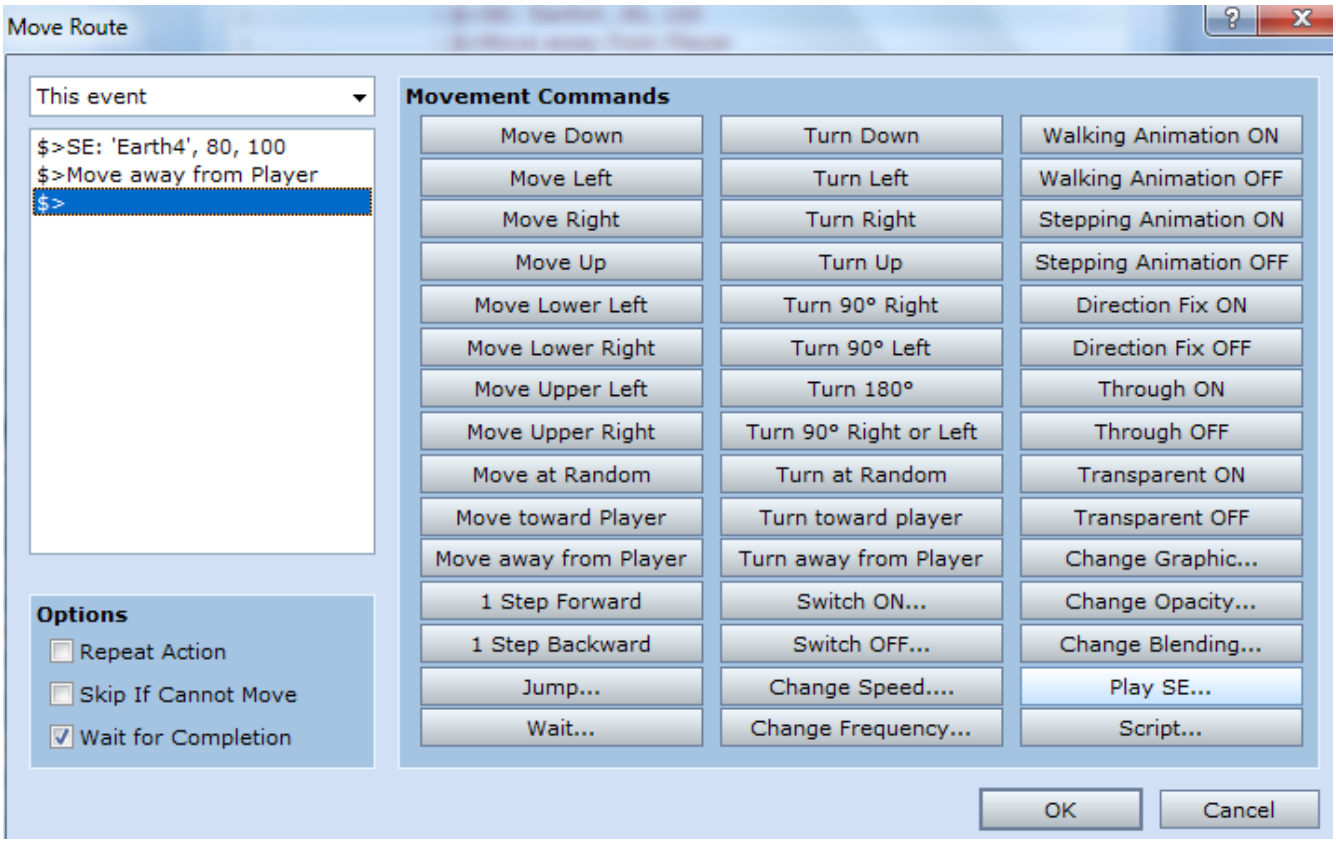
page for our Left statue Event. On page two of the Event Commands page, choose the option “Set Move Route”.



On the left hand side of the “Move Route” window, select “This Event” from the drop down menu. Next click the button “Move away from player”.



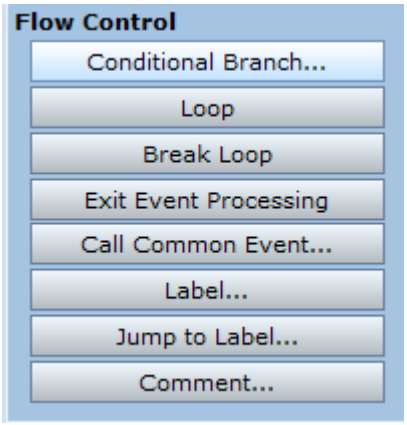
Be sure to keep “Wait for Completion” checked, and then click the OK button. Now, if you tested your game, you'll notice that when you press the Action Button while facing your statue Event, the statue will move away from the player, as if the player is pushing the statue! Neat. Let's edit the Event again, to play a nice background noise though, shall we? So let's edit the move route command, and click the “Play SE” button. Let's choose a sound such as the “Earth4” sound.



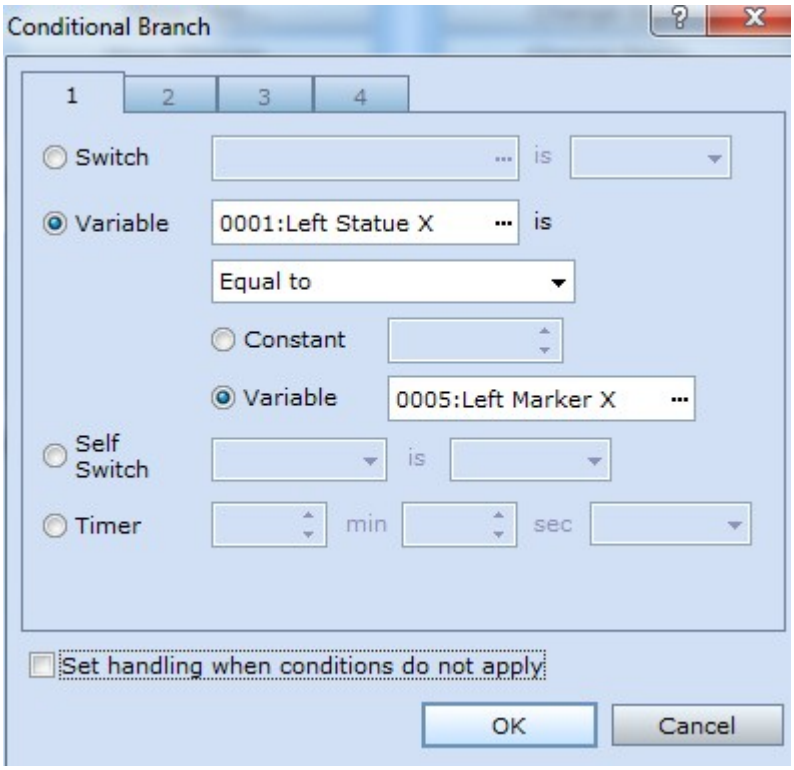
Be sure that “SE: 'Earth4', 80, 100” line is **above** the “Move away from Player” line, or the effect won't be correct.

Repeat this process for the other statue Event you have created. Test your game, and you'll notice that you'll be able to push your statues, and that they make a neat sound effect when they move. We're getting close! Unfortunately, the most confusing part is coming up. We're now going to create a second Parallel Process Event. This will be another “background” Event that the player doesn't interact with, but just helps the game know when the player has completed the puzzle.

On page one on the Event Command page, choose “Conditional Branch” under the Flow Control category.



Click the “Variable” radio button, and make sure Variable 0001 Left Statue X is selected from the drop down menu. For the second drop down menu, make sure “Equal to” is selected. For the second radio button choose “Variable”. Click the ellipsis button, and choose “Left Marker X”. Uncheck the button “Set handling when conditions do not apply”.

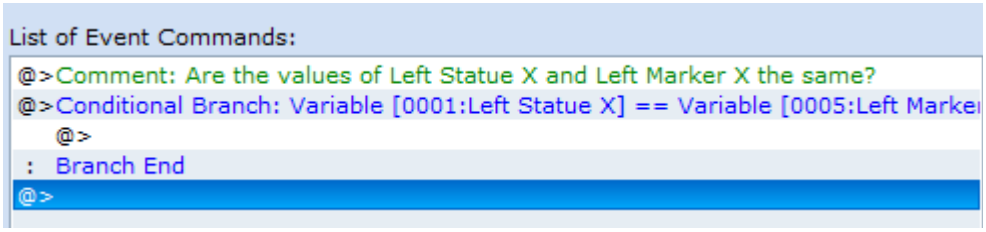


Let me explain what we just did. As I stated before, this Event is a Parallel Process, so it's constantly running. In this case, we're having the Event constantly checking a conditional branch. A conditional branch is the game asking a question. In this case, the game is constantly asking (in the background) “Does the variable 'Left Statue X' equal the same value as the variable 'Left Marker X'?”.

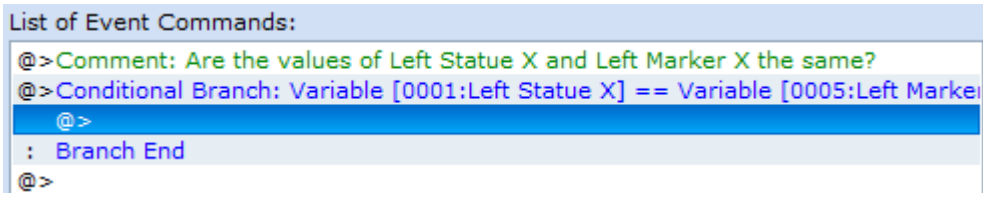
It can be slightly confusing at first, but we're just making our game ask a simple question. Just try to continue thinking of a conditional branch as a way of the

game asking a question.

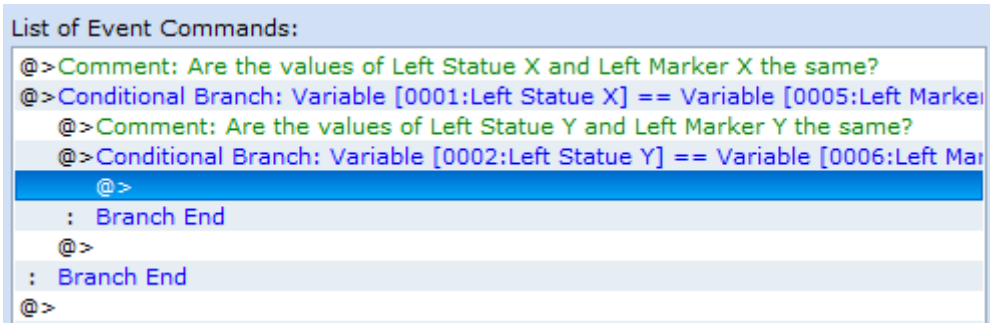
So let's add a comment explaining what this does, so that we can keep track of which conditional branch does what (there will be many).



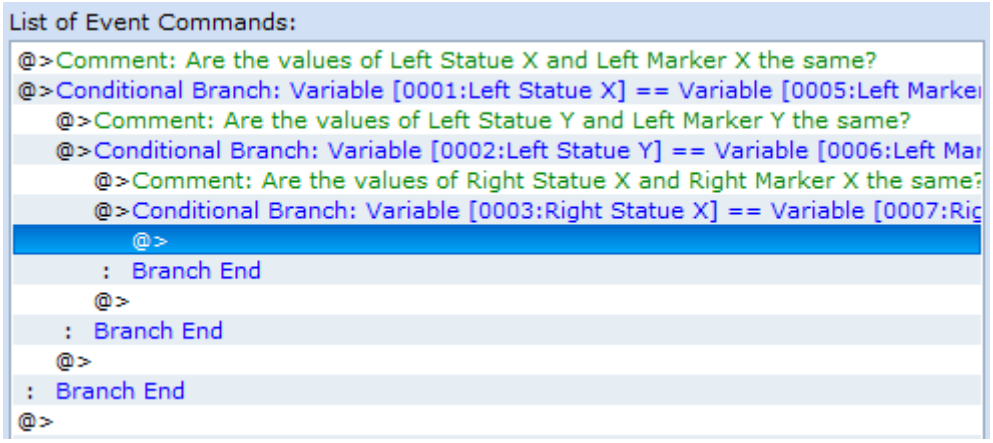
Next, we're going to want to click on the @ that is indented such as below. Notice how the blue cursor highlights “inside” the conditional branch.



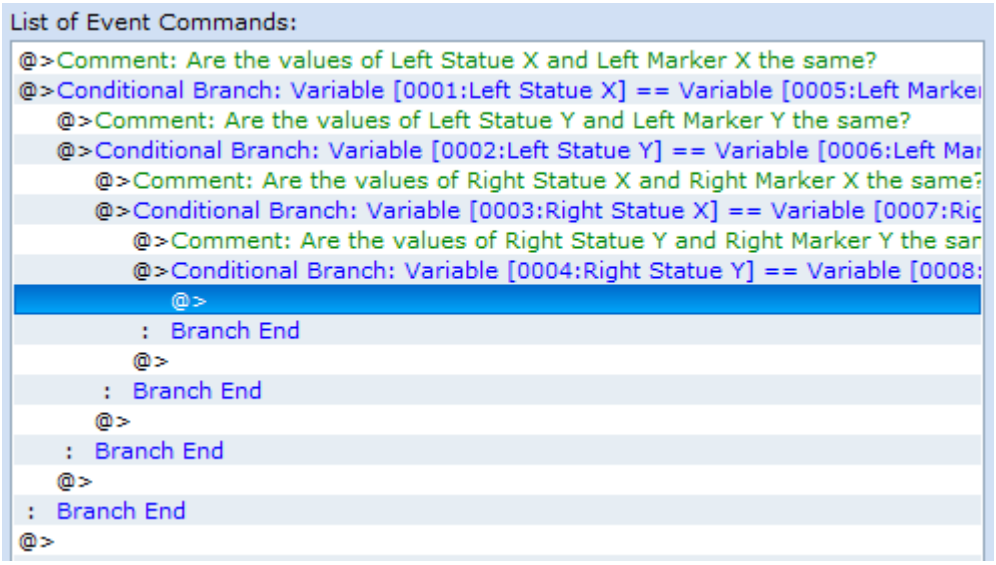
Double click, and let's repeat the process of adding a conditional branch, this time comparing if the variable Left Statue Y is Equal to Left Marker Y. Let's also be sure to add a comment!



Looking good! We're almost done. We're going to do two more conditional branches. You've probably already guessed what. Next we're going to make sure the value of Right Statue X is the same value of Right Marker X. Be sure that you're placing your conditional branch “inside” the second conditional branch we just created!

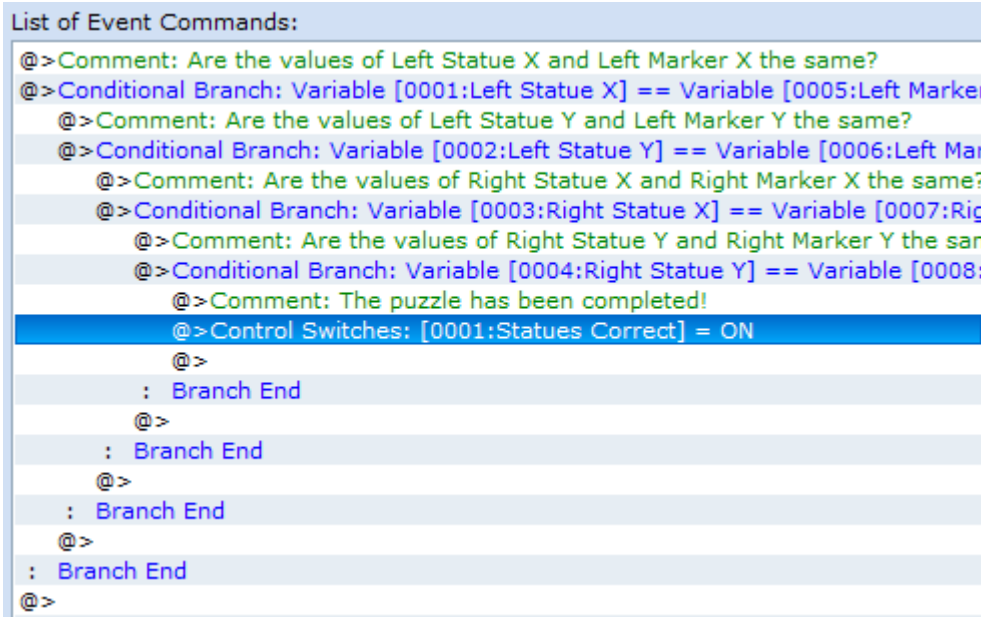


Our branches are starting to get really squished, huh? One more! Let's make sure Right Statue Y is the same value of Right Marker Y. Again, be sure to add it "inside" the third branch you just created!



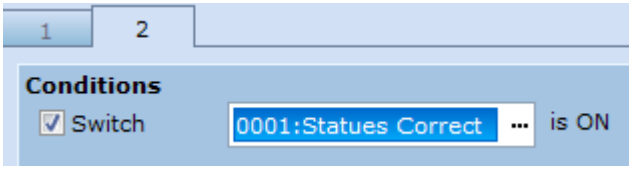
Double check that your work matches my own. If it does, pat yourself on the back. If you've made it this far, you've completed the most confusing part of this tutorial!

We have a couple of more things to do. Make sure your blue line is inside your fourth conditional branch, and under Event Commands choose "Control Switches". Let's turn a switch named "Statues Correct" on.



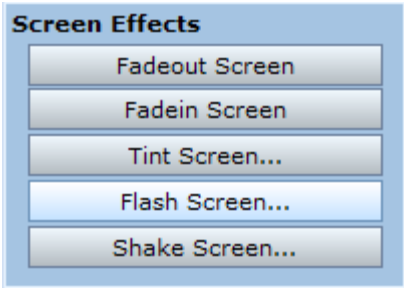
Let's click OK out of the Event, and save our project. We're approaching the final steps!

Let's head back to our door Event that doesn't respond to the player. In the door Event, click the “New Event Page” button, and check the checkbox “Switch”. Make sure it points to your “Statues Correct” switch. Make the Event “Autorun” and keep the graphic set to the closed door.



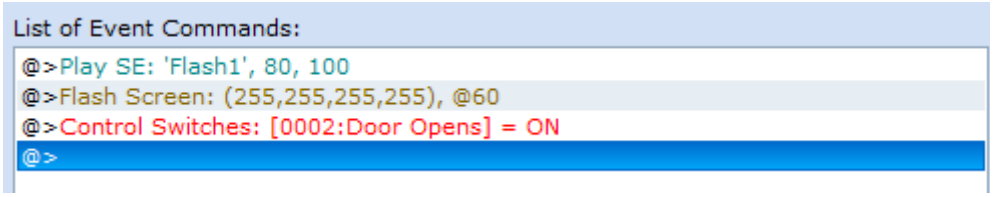
So the player has completed our tricky puzzle, and we're rewarding him with the door opening. Let's also give them a nice flashy effect. Under our new page two Autorun tab, let's navigate to page two in the Event Commands window and select the button “Play SE”. Choose a nice sound effect to play for a screen flash effect.

Next, let's add the flash screen animation. Let's navigate to page two of the Events Command window again. This time choose “Flash Screen” under Screen Effects.

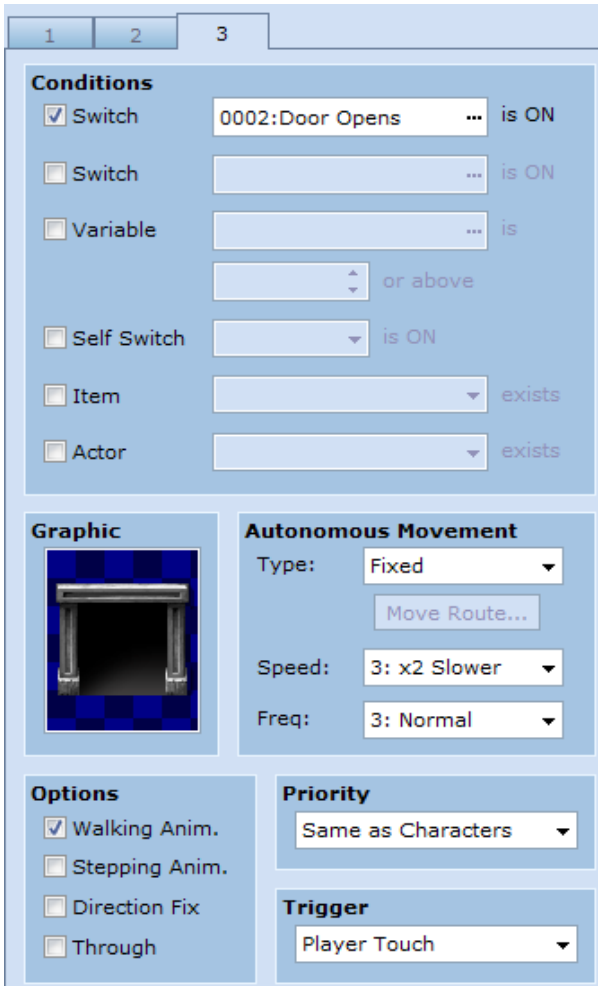


Leave all the values as default, except uncheck the “Wait for Completion” checkbox. Click OK.

Next let's set one more switch to On. Name it something like “Door Opens” perhaps ;)



Add one more page to this Event, and make the switch that triggers it “Door Opens”. Set the graphic to the doors opened graphic. Add any map transfers you'd like, and set it to “Player Touch”.



Now we just have a few more things to wrap up. To make things look better, let's reopen our statue Events, and add new pages that are triggered by the "Door Opens" switch being set to on. Leave the graphic blank, and page blank. This way, during the screen flash, the statues vanish when the door opens. We don't want the player to be able to continue to push the statues around.

There's just one last thing we should do. We should add another special tile somewhere on the map and create an Event over it set to "Player Touch". Under page two of Event Commands select "Play SE" and choose a sound effect of your choice. Next, add another Flash Screen Event. Be sure to uncheck the "Wait for Completion" box!

Next, under page two of the Event Commands window, choose the option "Set Event Location".

Choose one of your statues under the drop down menu, and make sure the radio button is set to “Direct designation”, then click the ellipsis button to the right. Choose the same exact tile the statue Event is currently sitting at, and click OK.

Repeat this process for the second statue Event.

What we've done is created a special tile, so that when the player walks on it, the statues are reset to their previous position! That way, if a player accidentally pushes a statue into a position where they can no longer move it to it's correct destination, they can simply “reset” the puzzle.

Save your project, and test out all your hard work! If everything has been done correctly, you should have a fully functional puzzle!

Pat yourself on the back! You deserve it.

I hope you have enjoyed this tutorial!