

Humbird0's Rpg Maker Tutorial

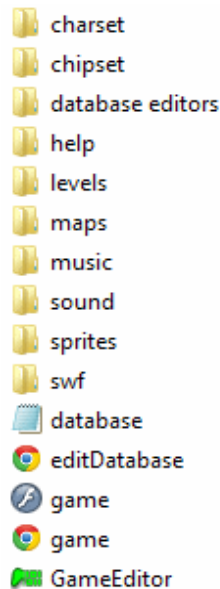
Introduction

First of all, I don't master entirely this Rpg Maker (RM), but I know some basic things that allowed me to make some basic things like using variables, create scenes, etc... You can see the few things I've done on the ulmf forum, in Humbird0's topic and mostly on the Legend of Krystal forum on the thread « Sex School Using Humbird0's RPG MAKER » by 'darkavenger'.

So I'll try to teach you the basics regrouped in this guide even if Humbird0 (dat genius!) already made some videos and a folder full of tips and advices to help you to progress. By the way, sorry for the typos and stuff, english isn't my natural language.

I-Where is your game ?

Before doing a game, you should know where it is put and where it is organized. Well it should look like this :

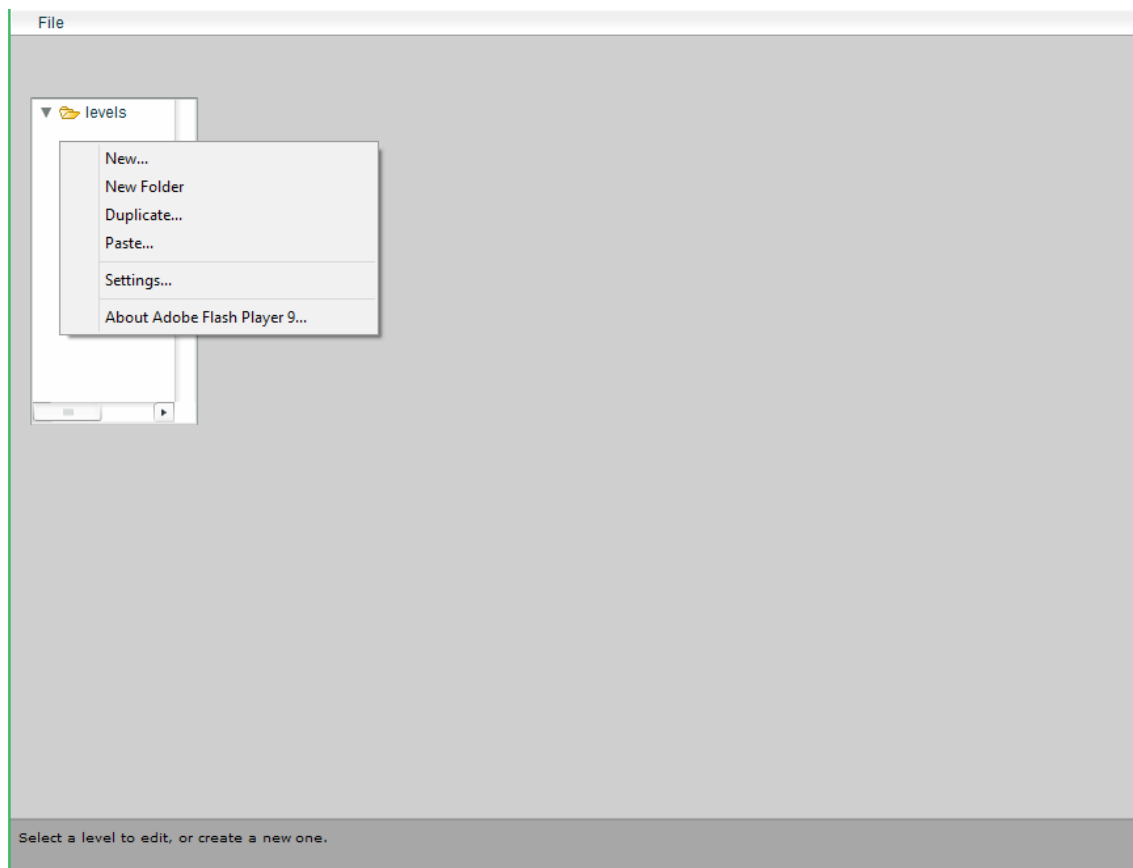


- **Charset** : This is the folder where you put your sprites-sheets, and by sprites, I mean character, doors, everything that can be animated, you'll see later how they are organized.
- **Chipset** : This the folder where you put the images that'll help you to make the maps, they are basically big pictures sliced in little tiles which one being a different texture.
- **Database Editors** : This is the folfer that contains your player sprite, starting place, starting screen. Don't go in there, you should use that with the RM.
- **Help** : The folder where all the Humbird0's tips are, just open it guys ! It's downloaded with the RM. He knows his stuff and he explain better than me. Open it by pressing F1 in the editor.

- **Levels** : This is where your maps and scenes are, when you create a level in the editor, it goes there.
- **Maps** : It's where you can store maps and import them to the game.
- **Sprites** : DON'T GO IN THERE !!! I mean this folder is used to modify the objects you can create in the RM, that's what make the difference between Humbird0's pokemon and babysitter club 2, it had things like pokemon trainers, which at the difference of usuals RPGsprite can use the fight system.
- **Swf** : It's where are all the flash effects, like flames, lights and stuff.

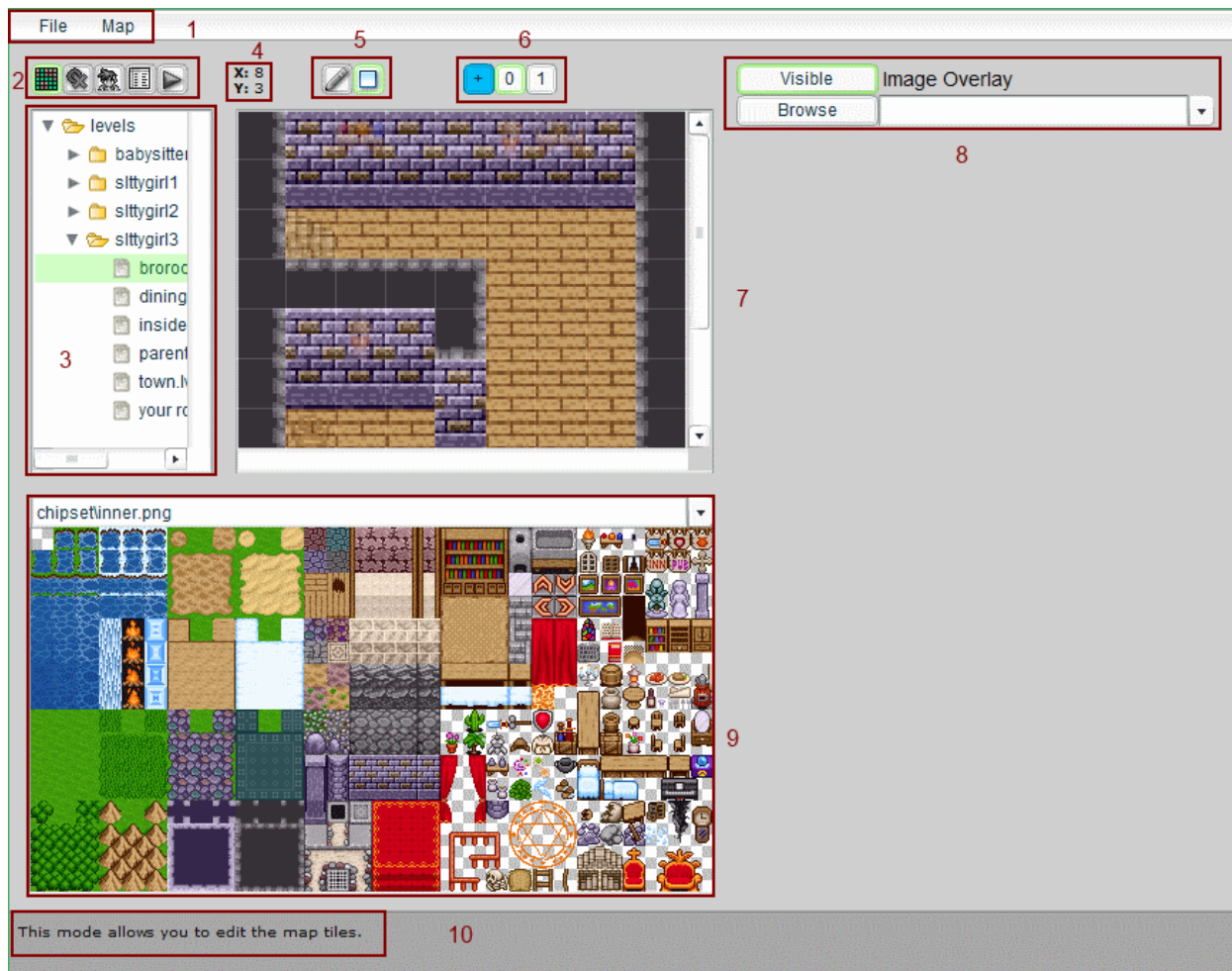
II-The RPG Editor !!!

In this part I'll try to teach you the basics, I mean the base of the base, square one. Firstly, when you open it, there's no lvl, so create one



Then create a map on your level or import one from the 'Map' button next to the File one, you should come to the map editor page.

1-The Map editor



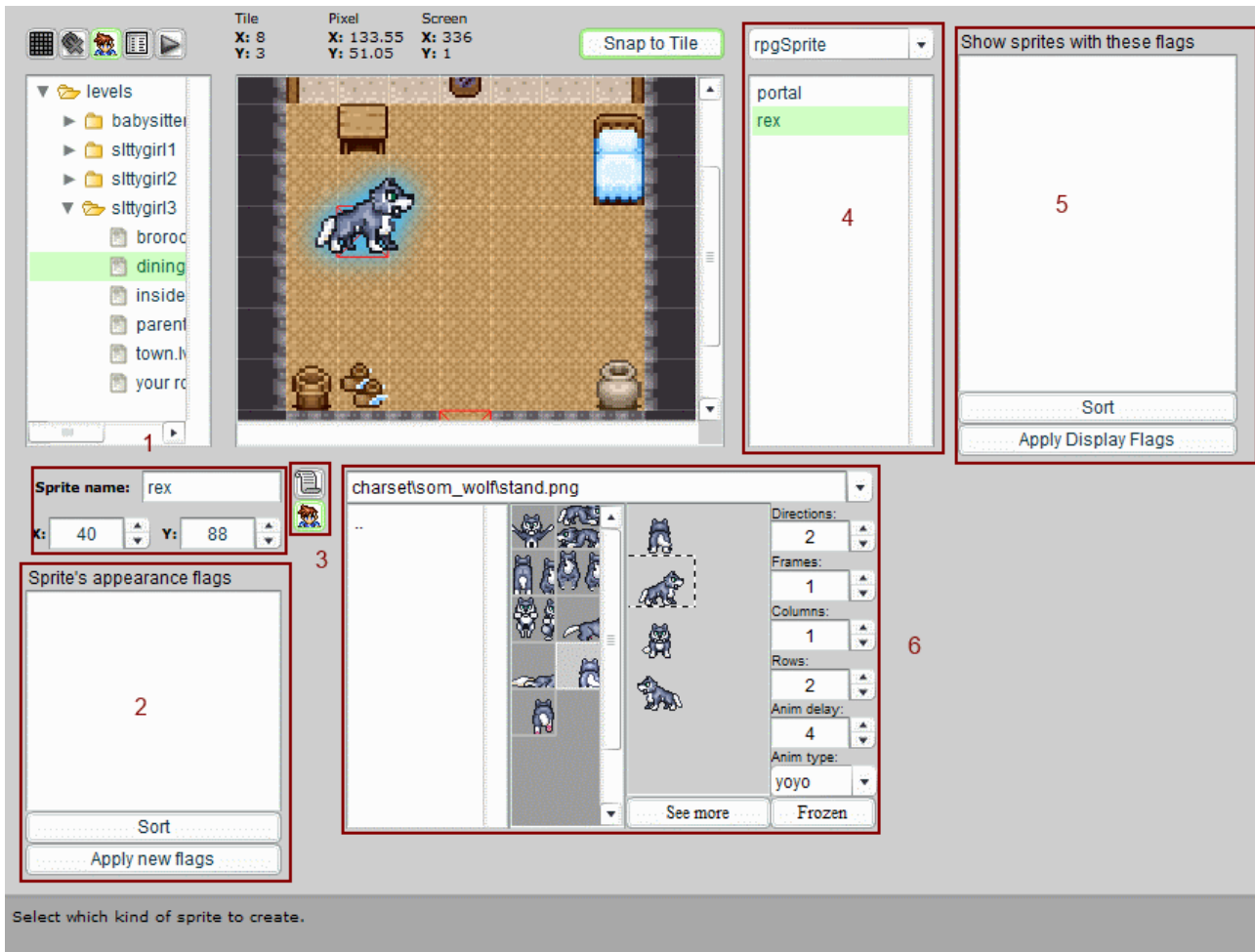
- 1 : The file button, where you can save open and stuff, the Map button where you can import or export maps.
- 2 : The editor buttons, they're used to switch to, from left to right, the map editor then the collision editor, the event editor, the database editor and the trying mode.
- 3 : The panel where you controls the levels.
- 4 : The position of your cursor on the screen, starting from the top left at $x=0$ and $y=0$.
- 5 : The mode of positioning of the tiles. The pen just drop one and the square make a zone full of the tile selected.
- 6 : The layer selection, basically 0 is the ground, you put a tile on that layer, you appear over it, the layer 1 is an object at the same 'level' as you, depending of your position, you can be seen over or under it. The layer 2 is something above you, on the screen you're always under it. I only use these 3 layers but you can use more.
- 7 : The screen panel, where you see your map, you can zoom in and out using the mouse wheel and pan on the screen by middle-clicking and drag.
- 8 : If you want to put an image as a map, from a snes game for example. It's like a tracing-paper, you can put your sprites to recreate it, but it won't appear in the game.
- 9 : The tileset, this is where you select the texture you want to use to create your map. Use the blank one, on the top left of this tileset to erase something.
- 10 : The most important, Humbird0's tips they're everywhere and they make your life easier.

The collision editor is almost the same except for the 6, they're 7 buttons, where each one have an

effect, I only use the '0' to make the ground 'no collide', and the '1' to block the way.

2- The Event editor.

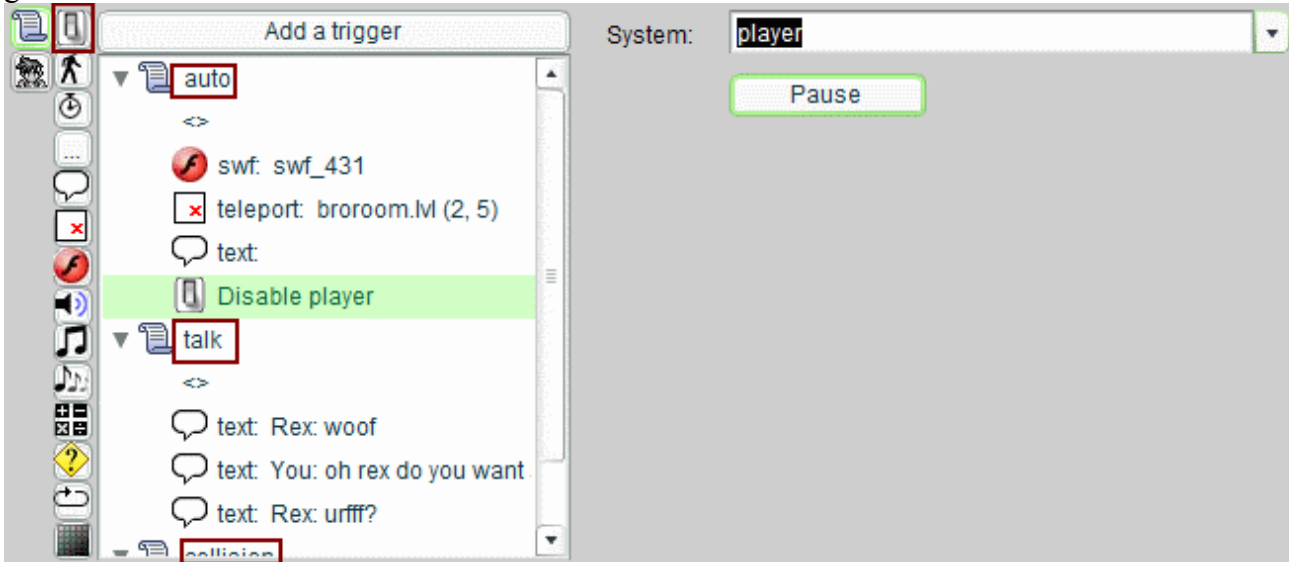
This editor allow you to create an event by double clicking on the map. An event is like an object you can program to make a scene, a dialog, anything ! Well anything virtual, you can't make food for example, unless it's virtual food, but I'm losing my train of thought.



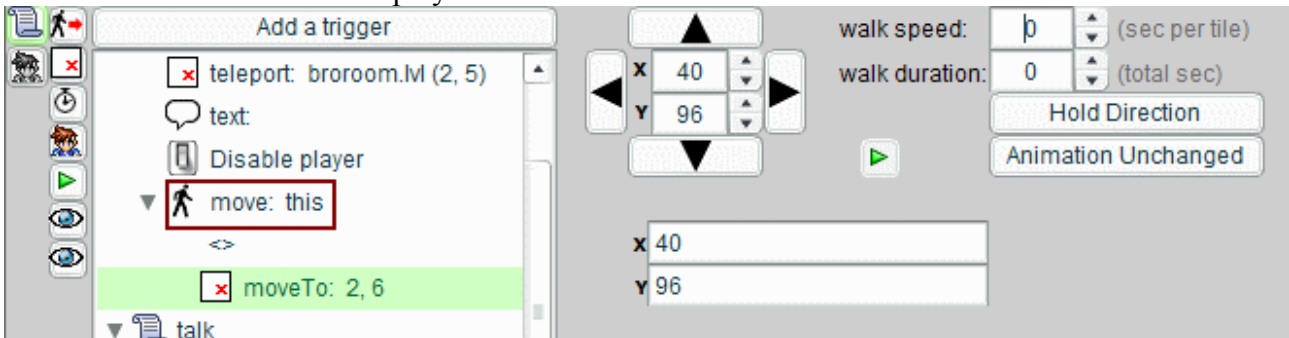
- 1 : The name and position of the event, note that you have to validate the name by pressing enter on your keyboard, and that the event can be positionned out of tile by pressing the snap to tile on the upper part.
- 2 : This is the sprite appearance flag's board (Captain Obvious is on duty) , it allows you to make the event visible and activable depending of the variable you write on this very board. For instance, if you want to make a event only executable once, put a variable, let's call it 'end', different of a certain value, i.e 'true', by writing 'end!=true' (!= → different of). At the beginning of the script, end is undefined, (end=undefined so end!=true, so the event is visible), then, create the variable end at the end of the script event, and give it the value true. The script will end and the event will disappear. You can write multiple condtions as long as they are on a different line.
- 3 : These button allows you to switch from the appearance of the event panel and the script panel, the first one allows you to change what the event looks like not activated , the second one allows you to change what the event does.
- 4 : This is the list of event in the level just select the one you want to edit by clicking on it.
- 5 : This panel is used to see what the screen would look like when it is in a certain environnement i.e when a certain set of variables are at certains values.

3-The Script editor

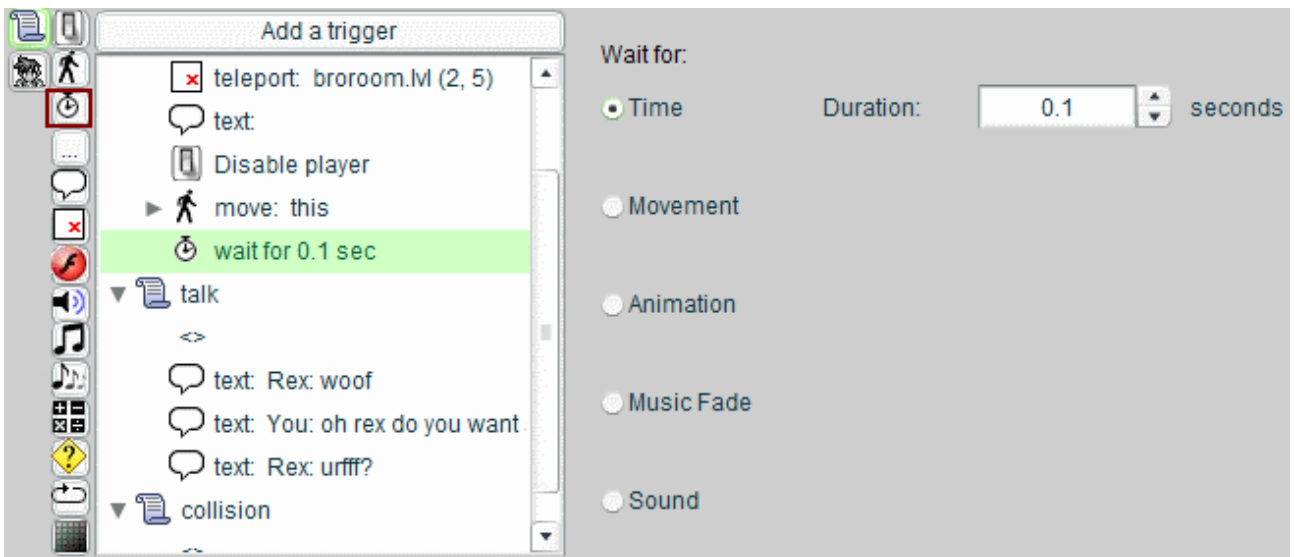
This one's a big one, it's a part of the event editor but it deserve a chapter, this is the base of the game.



The first one is the enable button it allows to enable or disable a system object like the player, I only use it to diable and enable the players movement. I don't know what else it can do.

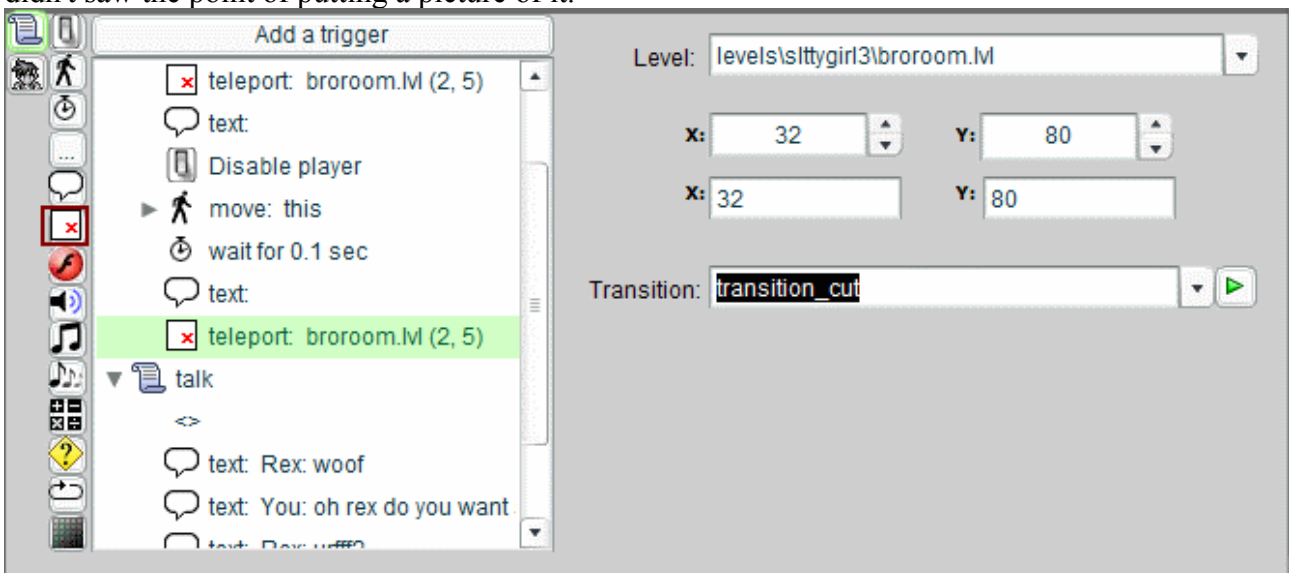


The second is the move button it allows to change the state of this event, another event or even the player. It can make it move with the 2 button at the top, it can make a pause, change the look of an event or the player, or make it look in a certain direction. In this picture, you see the second deplacement tool, where you select the place where the event must go, the speed or duration of its movement, if it have to be animated or the direction changed, and the button in the middle show you a preview.



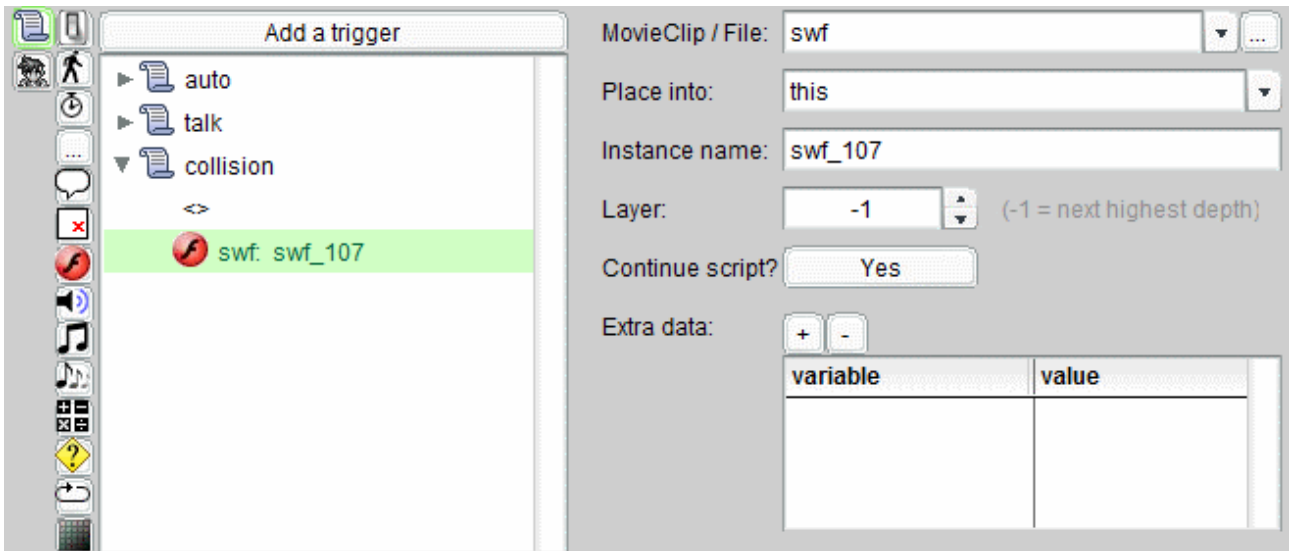
This one allow you to make a pose in your script, depending on time, movement or animation of an event, the fading of the background music, or wait for the end of a sound.

The next button is to add a commentary in your script, it doesn't modifying it's just a note tool, I didn't saw the point of putting a picture of it.

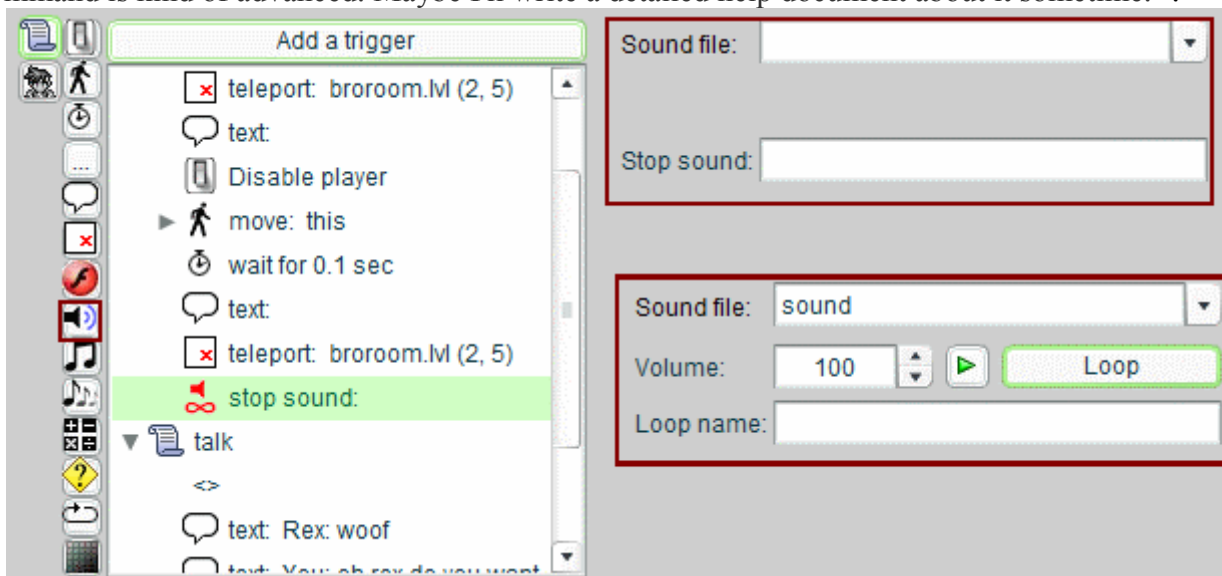


This one allows you to teleport your player even at another level. You first chose the level, then the position and finally the transition.

It is useful to make doors, or change the player position at the end of a scene.



I'm not familiar with this one, I only use swf like mood_dark which allows you to make a filter that make the light like it's the night, and I place it into overlay. For the rest, like yesNo, QCM and stuff, just copy Humbird0's games, that's how I do. You can also enter code instead of swf, like menu systems and other effects, but Humbird0 will explain that better than me. To quote him : "This command is kind of advanced. Maybe I'll write a detailed help document about it sometime. "

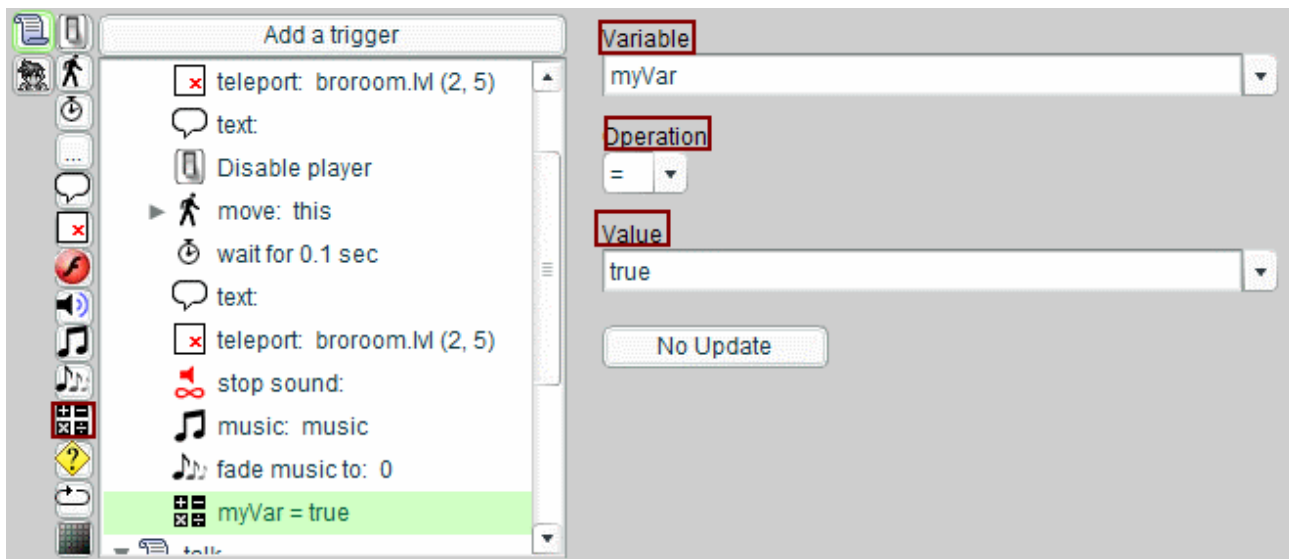


This one is about the sound. Actually, you only obtain one panel, the lower one, where you can make a sound once or loop it. If you put it once, you can enter a 'wait name' which allow you to put a timer waiting for the sound to finish. But if you put loop, you'll be asked a loop name. If you empty the sound file field, you can enter a stop sound allow you to stop the looping sound with the same name .

The next one is the music button allowing you to change the background music.

Then there's the fading music button, where you choose a new volume of the background music and the duration of the transition.

These two are pretty easy to use



This one is a tough one, the button variable allows you to create a variable, it's used to define an environment, some actions can only be realized under certain condition, this is what define conditions. It's like you ask the game to remember something and to give this something a value. You can change the fact that the game update the screen according to the changement of variable right away or when you reenter the level. This tool allows you to create and change the value of a variable, just look at the '2' of the 'Event editor' part to see how I mostly use it, but it have a lot, A LOT, of other possible utilisations. You can use it with the next button. Normally, the changes done when a variable is updated happens when you re-enter the map, but by pressing th « update sprite flags » button, it happens right away. This tool allow you to enter command : by default every variable is stock in RAM, but you can change the parameters of another object, like SPRITE for instance to temporarily put the player half translucent, just write "SPRITE.player._alpha=50". I don't know all these commands but I'll tell you where you can find them in another update. The [] tell it to look up the value of whatever is between them. If it's the name of a variable, then it'll use whatever is stored in that variable. If you also put () at the end of the variable name, it will instead treat it as a function call instead of a variable. Another use for this is manually calling scripts. [SPRITES.girl.talk()] will trigger the "talk" script of a character named "girl" on the map (if such a character exists) This can be useful for telling doors to open during cutscenes.

The next one is a 'if' function, when you create it, it shows a field and realize the actions you write in this field only if the condition of the 'if' is valid. The condition is the often the value of a variable.

The next one is the loop, I never used it, never ever. But it allows to repeat a certain action until a certain condition isn't valid. It's like the if but instead of just execute the script, it repeat it until the condition is no longer valid.

The last one is the fade in/out button allowing you to make a fade in/out and to wait or not the fade to execute the next part of the script.

Conclusion

So to finish, I'd like to say that with this script goes a little example to see practically how it works, I'll be waiting for your questions on the LoK forum, this is not a definitive version of this tuto, I'll add thing I will understand later or if something is missing or not clear enough I'll change it.

I hope this will help you to understand this compacted version of RpgMaker, it's not difficult to use it, but hard to master, I didn't past that much time on it, but I could make some games.

Finally thanks Humbird0, honestly, bravo, this is worth the 6 months you past on it.