BEGINNER'S GUIDE TO RPG MAKER VX

How to easily get started with RPG Maker VX

BenKo

December 8th, 2008
V 0.5

INDEX

Introduction **4**

Why this guide?

The methodology

Game development **6**

Programmers versus designers

The team

Legend of Gladius **10**

The story

Locations

Quests

Items and weapons

Creating maps **12**

The world map

The village

Interiors

Basic events **18**

Starting position

Transfers

Doors

Decoration events

Treasure chests

Basic NPC's and dialogs 24

Sign posts

Inns

Dialog choices

Shops

Customizing texts

Walking NPC's

Game elements 36

Items

States

Skills

Weapons and armors

Jobs

Party members

Enemies and battles 52

Monsters!

Troops

Encounters

Story 60

Cut-scenes

Forbidden places

INTRODUCTION

WHY THIS GUIDE?

Truth is I wrote this guide mainly for myself. I downloaded **RPG Maker VX** and began to tests things until I got stuck with something and had to browse the Net for the solution to that problem.

Most of the times, I found a tutorial which taught what I wanted to do. Other times, however, I had to dig in some forums or read blogs. This system works but has some flaws:

> What if I forget something? Let's say I stop using RPG Maker VX for a couple of months… Will I be able to remember everything, considering that I'm a newbie? I don't think so. This leads us to...
> I don't want to lose my time! Everything I needed to know was scattered all over the Net. And not every answer was clear… Sometimes I was confused and didn't understand how something was explained until I learnt other stuff. This is because the "information" is not organized. And this implies that…
> My way of getting knowledge was wrong. Whenever you want to learn something, you must learn the basics first.

So, this guide is for me because I wanted to have all the basic stuff covered in ***just one place***, and ***organized***. This guide is also for my friends whom I try to convince to create games with me ;-) And, of course, this guide is for everyone who downloaded RPG Maker VX and doesn't know where to start.

In addition, I'd like to recommend you to these **forums**: http://www.rpgmakervx.net . I learnt a lot of stuff there and the community seems friendly and willing to help. At those forums you will find a tutorial called *RPG Maker School* (created by Andria), which is a RPG that will teach you a lot of useful things. You definitely want to download and play it!

If you have any comments or suggestions about this guide, you can send an **e-mail** to ladybenko@gmail.com.

THE METHODOLOGY

In this guide we'll be creating together, step by step, one mini-RPG. Attached to this guide you should have also got the RPG Maker VX project so you can take a look if you ever get lost.

In order to get the most of this guide, you should run RPG Maker VX and execute the instructions provided as you are reading the text. You will work much faster if you **print** a copy of this guide or use two video displays.

Each section of the guide explains a single concept and they are convenient titled and indexed, so you can find quickly this information later when you are working on your own project. However, chapters and sections are ordered so you can learn concepts **increasingly**. That's why you should read the chapters in order the first time you read the guide.

This guide contains a lot of screenshots detailed explanations at the beginning, but I tried not to repeat much stuff. Please don't get scared when you look at the number of pages of this document! Most of the space is filled with pictures, so reading should be quick.

Happy learning!

GAME DEVELOPMENT

This chapter does not contain information specific for RPG Maker VX, but you should read if you are new to game development.

PROGRAMMERS VERSUS DESIGNERS

Nowadays, developing a professional game takes a team working full time for months (in casual games like Bejeweled) or even years (in AAA games like Diablo). Each member of the team has a **role**: concept artist, animator, coder, tester, producer, designer, etc.

In amateur and casual game development teams are usually small (five or less people) and most of the times a member of the team has several roles assigned to him: designer, programmer and tester, for example.

Young people usually become interested in game development because they want to *make* a game. Then they learn that in order to do that, they need to "create" a program with a thing called "compiler" (they heard that C++ is the industry standard and start to learn that) and API's such as OpenGL or DirectX.

At the same time, they join a game development forum and post a message saying that they are going to create a RPG and that they need help and advice. The most common thing that will happen is that veteran members of the forum will laugh and discourage that newbie. And they are right: *you cannot develop a RPG on your own*, especially if you don't have previous experience.

However, you can *design* a RPG on your own. Being a programmer is different from being a designer. If you want to program a RPG, you will have to code the game engine, which handles stuff such as rendering meshes and sprites on the screen, collect input from the keyboard and gamepad, detecting collisions, the AI's core, etc. If you want to design a RPG you will have to create the maps and levels, set the AI for the enemies, balance combats, jobs, weapons and stats. You will also likely get to write the story and dialogues and design the main characters…

Do you want to be a programmer? A designer? Both?

There is no wrong answer here. Each person has his own interests and some people find coding very exciting and challenging and don't want to be bothered with stuff like level design. Some other people would rather spend their time writing a beautiful story or creating interesting maps and quests.

If you decide to become a programmer, buy a good C++ book and learn OpenGL. If you want to program 2D games you can also take a look at Microsoft's XNA and C#.

If you decide to become a designer, you have two options:

a) Recruit a team of programmers who will code a game engine for you.
b) Buy a pre-made game engine for some specific genre, such as RPG Maker VX.

The first option will cost you a lot of money and time. You might have some friends willing to do that for free but keep in mind that this will be their hobby and that they will be doing it in their free time. That implies that you will have to wait a couple of years until you get your engine done (if you're lucky and they don't give up the project). However, if you are successful with this, you will have a fully customized engine that will suit perfectly your needs.

The second option will save you a lot of time. Good news is that there are very affordable game engines. For example, RPG Maker VX will cost you only $60 and you will get some quality art and sounds assets. Keep in mind that this engine will have some restrictions and might not adapt entirely to your initial design: you will need to sacrifice some features in order to save time and money.

Again, be aware that there's no a best choice. Some people think that makers are for "losers" and that real men code their own engines. This is wrong. There are some successful games which are a modding (i.e. use a 3rd party game engine) of another game. For example, Counter Strike is a mod of Half Life: Counter Strike's developers created new levels and art assets and scripted the new game behavior for Counter Strike. Scripting is the medium which designers use to communicate with the game engine. This can be done by using an editor (this is the case of RPG Maker VX) and/or a specific script language.

To finish this section, let's read an extract of an interview to Amanda Fith[1], and indie game designer who has successfully released several games made with makers, all of them were sales hits in casual games sites such as Big Fish Games.

> **Q:** You've used some pretty high-end toolkits for creating your games - specifically RPG Maker for your two RPGs. Some people get a little funny about that. On the one hand, some people treat it as if it's somehow

[1] You can read the full interview at http://www.rampantgames.com/blog/2006/11/interview-with-amanda-fitchindie-rpg.html. It's really interesting.

cheating - that it is both overly constricting and somehow "too easy." Yet there are very few finished games out there using these engines, let alone polished, commercial-quality games of the quality and scope of *Aveyond*. So what does it really take to create a polished, commercial-quality game using a higher-level engine? If it's much harder than it sounds, is a higher-level engine even useful?

A: I knew that when I released *Aveyond* that I was stepping into the briar patch. I was very worried about how the development community would feel about my approach**. I repeated to myself over and over again that my goal was not to please developers, but to please players** […].

And I think she's right. Most players don't care about how games work on the inside. They care about the *outside,* because that's what they are interacting with.

The Team

Whether you chose to recruit a bunch of programmers to code an engine or not, you will still have to get a team: artists, music composers, other designers and so on.

Rule of thumb: **you get what you pay for**. If you want to create a game and make it look professional and polished, chances are you will be paying money to your team.

If you don't want to spend money and just want to create a RPG as a hobby or to learn new things, you can recruit a team of hobbyists to help you with the project. If this is the case, here there are some advices for you and your team:

- **Aim for quality, not quantity.** People would rather play a short, polished game than a long crappy one. Create good art. Design your maps with love and care. Write a thrilling story. Make your NPC's rich and interesting. Balance the combats with enemies. You can have all that in just one hour of gameplay. Reduce the scope of your game to be able to pay attention to every detail in it.
- **Assign a reasonable time span to the development.** If you are designing your game and estimate that it will take one year to complete it, then *don't do it*! Projects usually take twice, three times (or even more!) more months than expected. If you plan a game small enough to complete it in three months, you will be likely working on it for almost a year. That sounds pretty much reasonable and reduces the risk of someone leaving the team.
- **This is not *your* project.** If you work with a team made of people working for free, everyone should have his saying in every aspect of the game design and development. You are not their boss, you are their team mate.

Recruiting people can be hard at times. How do you make talented people (who are the most requested inside the community) get interested in your idea instead of working for some else's or their own?

Prove that they can trust you. Even if this is just a hobby, time is money and people don't like losing time. If someone joins your team and is working for a couple of months and then the project is abandoned, this person would have lose his time. Show that you can complete a project and get things done.

The best way to do that is to have a small portfolio of **completed projects**. Yes, it is hard to complete a project only on your own, but you can complete a very small project. Just make it look great.

In addition to that, you can start developing your game using existing art assets. If people get to play a **demo** and they like it, they might join your team to improve the game.

As you can see, the key here is to have real, tangible stuff to earn other people's trust.

LEGEND OF GLADIUS

All along this guide we will be creating a very simple and short RPG, titled *Legend of Gladius*. The purpose of this chapter is to introduce the story and design of this RPG. Of course, in a real project, you should write a good design document which describes the full story, characters, NPC's, items, tows, etc.

THE STORY

Salazar (a warrior) and Zach (a rogue) are two friends who leave their homeland looking for adventures. They both lack experience and money, but are brave and ambitious.

In their journey, they arrive to Evergreen, a small village in the island of Kretus. There, they meet Lime, a young sorceress. She tells them how the whole island suffers the tyranny an evil ogre. She asks Salazar and Zach to join her and kill the ogre together. They agree, and she tells them to prove their value by going deep into the Forbidden Forest and taking Gladius, a mythical sword. Once they get this magic sword, Lime will join the group and together they will go to the Ogre's lair and kill him.

LOCATIONS

Our world will be reduced to the island of Kretus. Inside this island, the player can travel to these different locations:

- ***Village of Evergreen***: this is Lime's village. The village is small, but has an inn and a shop.

- ***Forbidden Forest*:** this is a dungeon, inhabited by dangerous creatures such as slimes, spiders, bats, etc.
- ***The Ogre's Lair*:** this is a small dungeon. The final boss is inside, as well as some other minor monsters, like rats.

QUESTS

The game has only one main quest, but this quest can be split in two smaller ones, which must be completed in order:

1. Go to the Forbidden Forest and fetch Gladius, the magic sword.
2. Go to the Ogre's Lair and kill the evil Ogre, who will be the final boss of the game.

ITEMS AND WEAPONS

There will be these items in the game. All of them can be purchased in the village's shop, or found dropped by enemies after a combat.

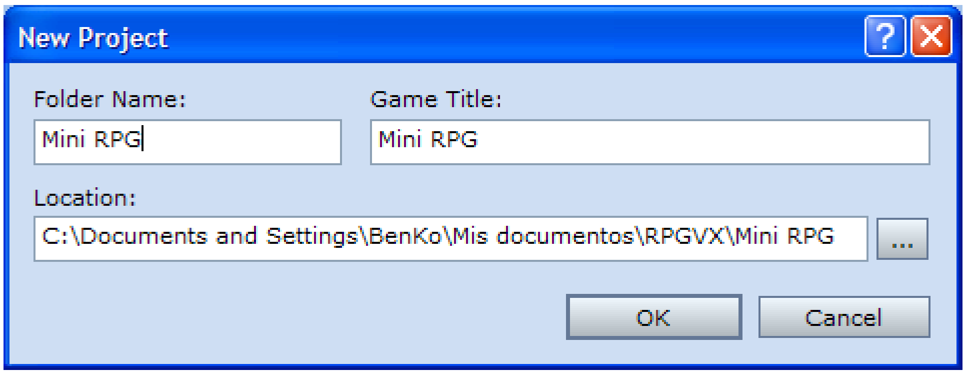
- ***Potion***: restores life
- ***Magic water***: restores mana
- ***Antidote***: heals poison
- ***Stimulant**: r*evives a character from KO

The party has established initial equipment which can't be upgraded. The exception is the sword Gladius, which will be used by Salazar.

CREATING MAPS

THE WORLD MAP

Setup a new project by selecting `File/New Project` in the main menu. Title it whatever you want.



New Project

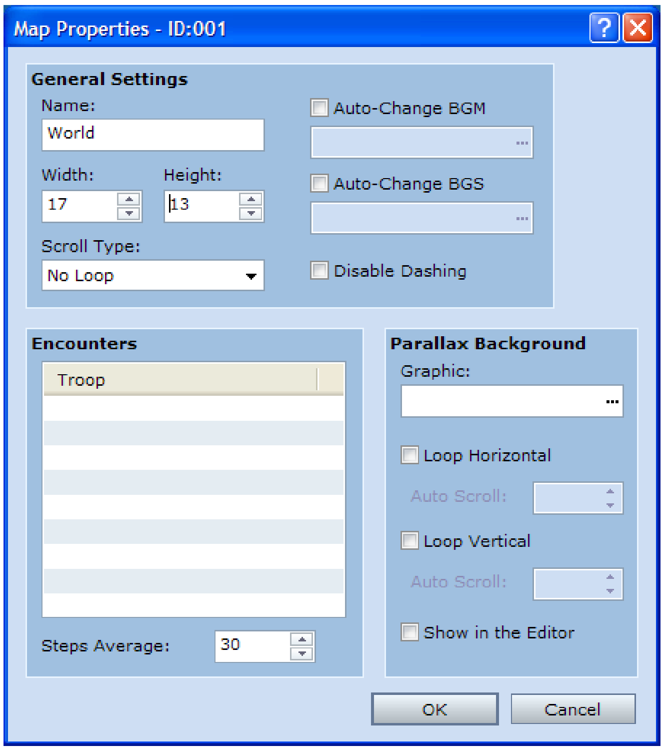
Folder Name: Mini RPG

Game Title: Mini RPG

Location: C:\Documents and Settings\BenKo\Mis documentos\RPGVX\Mini RPG

You can see now the main screen of RPG Maker VX. In the sidebar there are the tilesets you will use to draw your game maps, and a list of them. There's a map created by default, called MAP001. This will be our world map, but we have to adjust its properties.

Right click on it and select `Map Properties`. In this dialog you can configure all your map's parameters. For now, change its name and set its size to 17x13 (this is the minimum size for a map, by the way).


Map Properties - ID:001

General Settings

Name: World

Width: 17 Height: 13

Scroll Type: No Loop

Auto-Change BGM

Auto-Change BGS

Disable Dashing

Encounters

Troop

Steps Average: 30

Parallax Background

Graphic:

Loop Horizontal

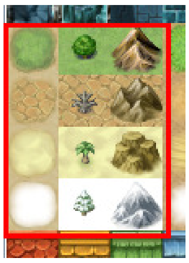
Auto Scroll:

Loop Vertical

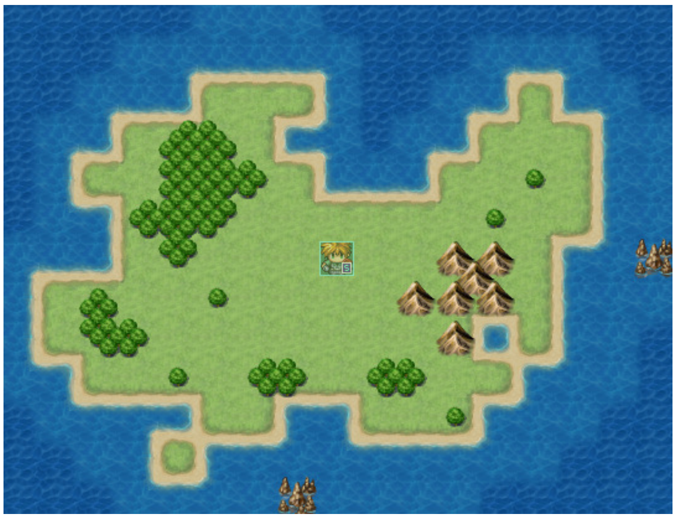
Auto Scroll:

Show in the Editor

You can start to **draw** your world map now! In our case, the world map is the island of Kretus. All floor and walls tiles are located in the tab A. The best tiles for world maps are these ones:

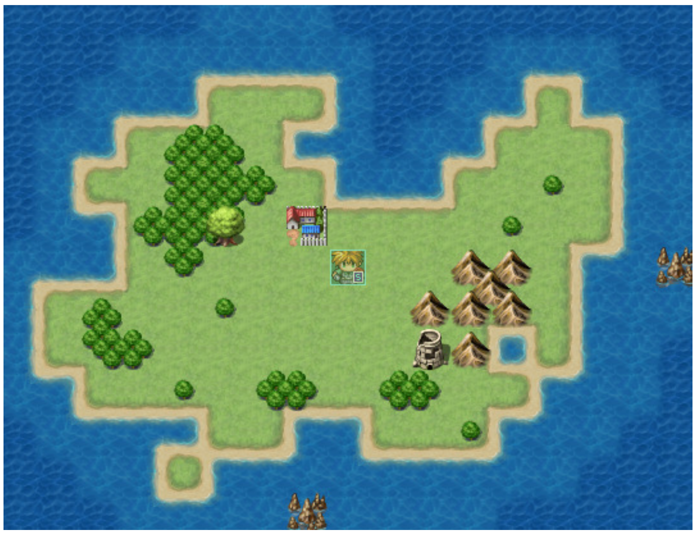


Draw your island with non-regular shapes and include some mountains and forests. Also you can use the deep ocean tile to make your sea more interesting. Here's what I came up with:



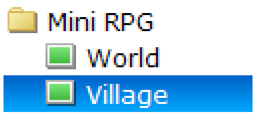
In tabs from B to E you will find tiles that will be drawn over the tiles from tab A: these are decorations such as trees, windows, etc. Choose some of these tiles to draw the Village of Evergreen, the Forbidden Forest and the Ogre's lair.

Here's the finished world map:



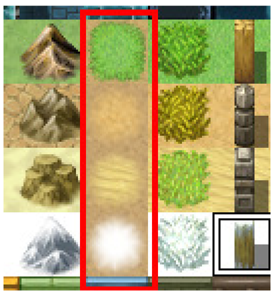
The Village

We have to draw the **village** in another new map. In order to do that, create a new map by right clicking on your project root's folder and selecting `New Map`. Choose a bigger size (for example, 25x20) this time and type an appropriated name for it. Your project's map tree should look like this now:

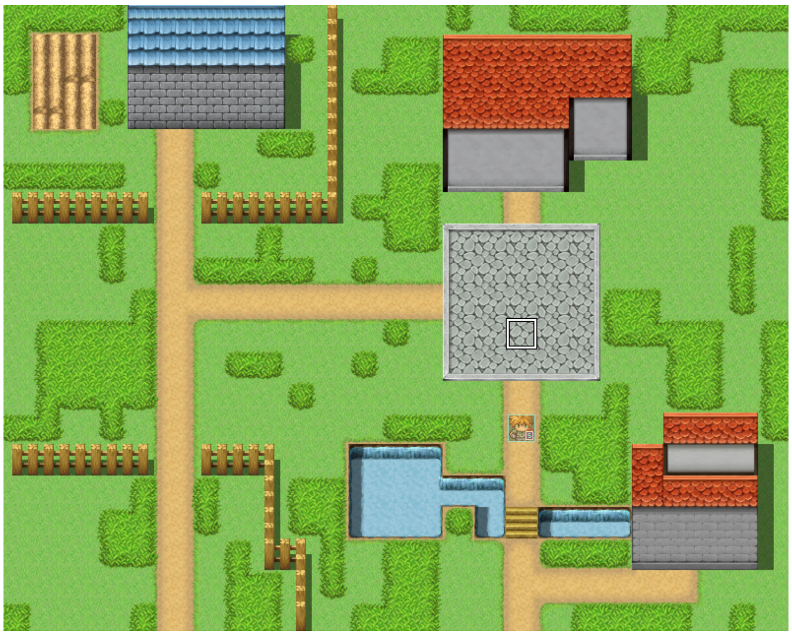


Mini RPG
World
Village

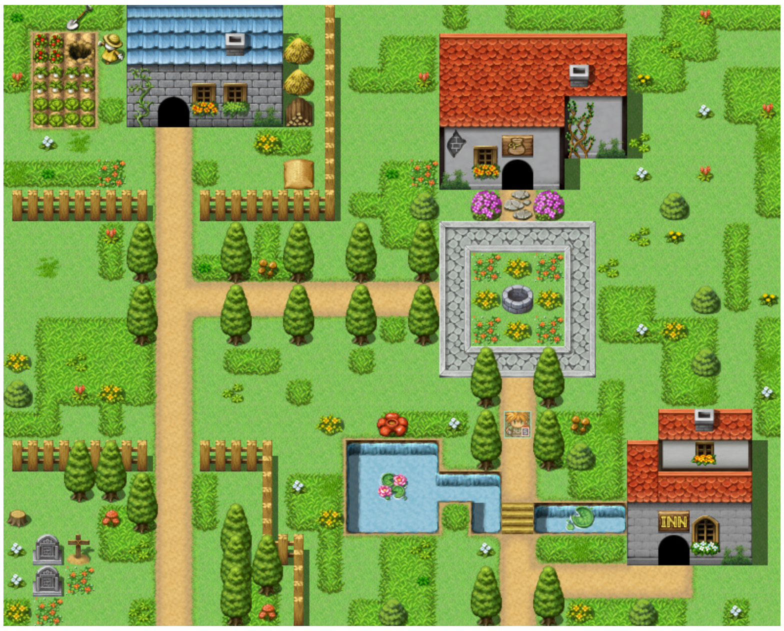
Choose a **terrain** tile among the following and use the paint bucket tool on the tool bar to fill the whole map with grass, earth, sand or snow. Then, use another other tiles to draw roads and fences. With the rest of tiles from this tab (A), you can also draw walls and roofs. Draw some houses (minimum of two: they will become later our inn and shop).



Try to draw irregular shapes: your map will look more realistic and natural. You can take a look at other people's maps and learn from them.



Now that we have a basic layout, it's time to add some **decorations**: pick plants, trees, windows and the like from tabs B-D.

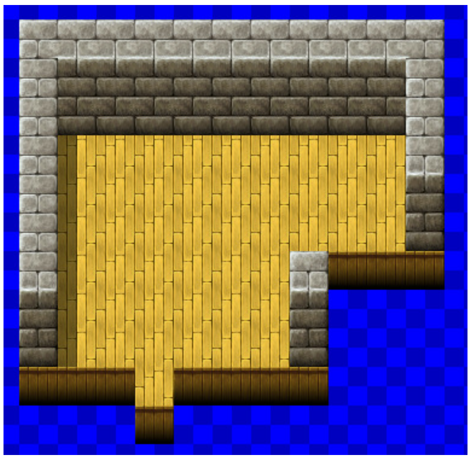


Don't forget to add some information signs to the inn and the shop. They will help the player to locate them.

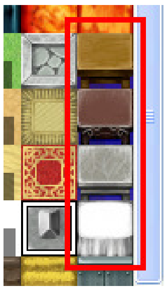
INTERIORS

There are three buildings in the village and the player will be able to enter into two of them: the inn and the shop. Let's start with the shop by creating a new map of size 17x13. This time don't right click on the project's root, but in the village's map. This way you will have your maps organized (very useful when you have a large game).

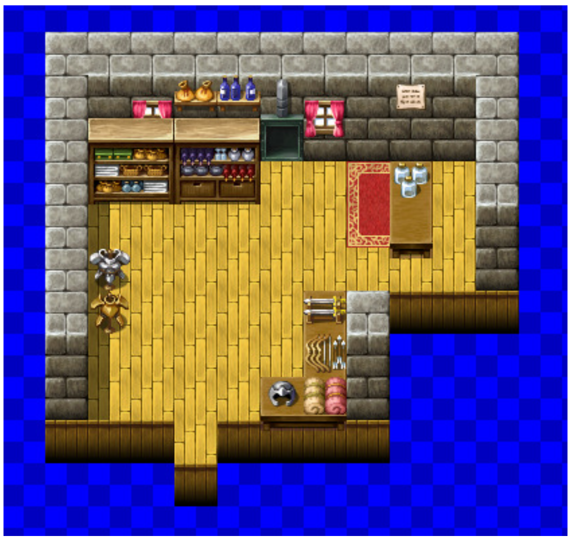
Draw the walls and the floor. Keep in mind that you don't have to fill the whole map area. Transparent areas will be rendered with black color or with a custom background.



Once you have the building's structure, add the furniture and decorations. Some tiles are special: the **counters**. Counters are useful because you can interact with Non Player Characters (NPC's) even when there are counters in between. You can use these tiles as counters (from tab A):

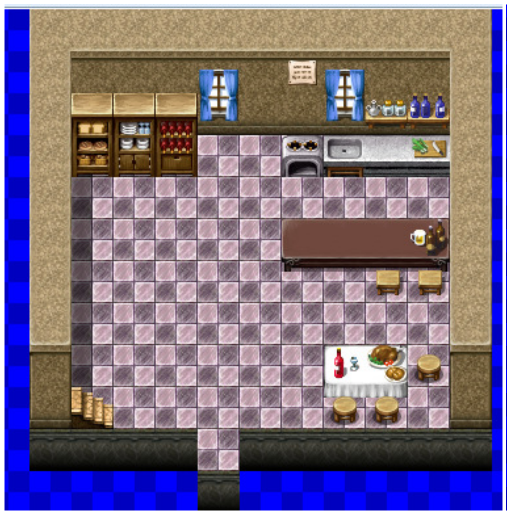


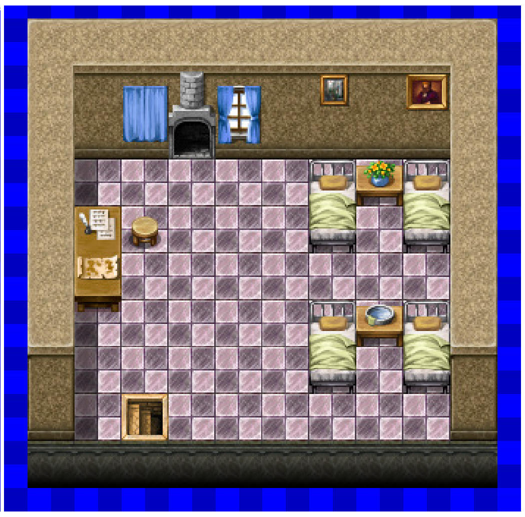
And here's how my own shop looks like:



It's time for the inn now. Our inn will be a two-story building, so we're going to need two maps. Create them (minimum size again) and proceed just like we did with the shop. Don't forget to add some stairs to connect the floors!

And here they are: first and second floors of our beautiful inn.





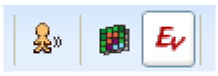
BASIC EVENTS

Events are a core feature of RPG Maker VX: with them, your game comes alive. Events are an entity with an **attached behavior** and/or a **graphic representation**. For example, NPC's are events: they have a picture associated with it (graphic representation) and they move around or talk to the player (behavior). Some events haven't got a graphic representation (for example, an event which will launch a battle when you enter some location) and some haven't got a behavior (for example, a glowing animated candle on a table).

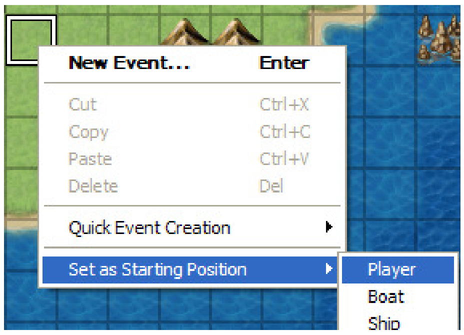
In this chapter we will begin to learn how to create simple events, such as doors or chests.

STARTING POSITION

The **starting position** of the main character when the game begins is not an event but it is created as such. To set it up, you just have to enable the Event mode by clicking the correspondent button at the toolbar. The map will darken slightly and a square grid will also appear when changing to Event mode.



Now right click on some tile of your world map and select `Set Starting Position / Player`.



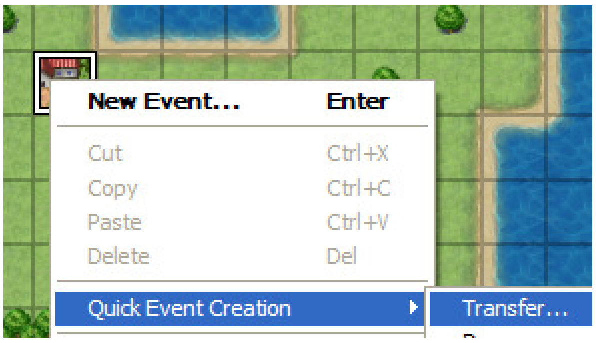
TRANSFERS

Transfer events allow us to set **connections** between maps: they teleport the main character from one location to another one (this location can be part of the same map, or be in a different one).

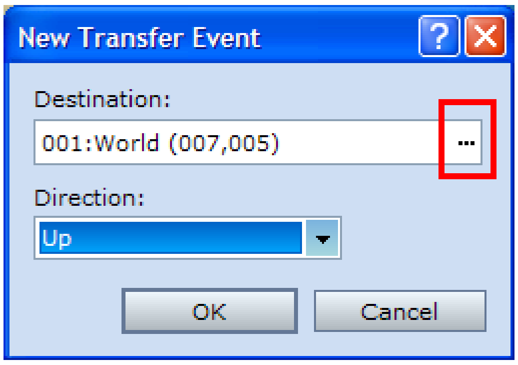
In our case, we'll set up a connection between the world map and the village, and make this connection bi-directional.

Show up the world map and change the editing mode to

Right click on the village tile in the island and choose `Quick Event Creation/Transfer`.



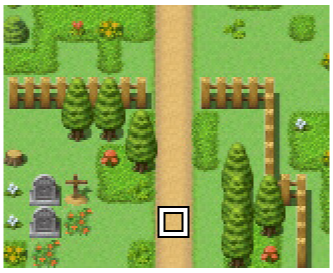
At the New Transfer Event dialog, set the direction to `Up`. This direction will be the one that the main character will be looking at. Since the character will later appear at the road located at the bottom of the village, `Up` seems the most appropriated direction. Change this if you draw your map in a different way. Once you've set the direction, click on the ellipsis to select where the character will teleport.


New Transfer Event

Destination: 001:World (007,005)

Direction: Up

Select the map for the village and then click on the tile you want the main character appear. It will look more natural when playing if this tile is a road.



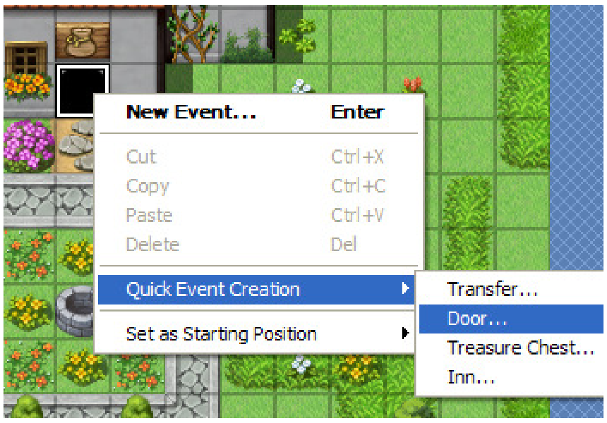
Accept both open dialog windows and you are done! The player can now enter the village from the world map. Now we have to make this connection bi-directional: the player should also be able to leave the village and enter the world map. To do that, repeat the steps to create a transfer event on a tile at the border of the village's map.

Now you can **playtest** your game by clicking the `Playtest` button. Try to go from one map to the other.

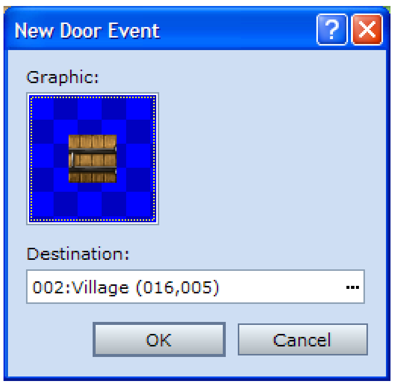
Doors

Doors are just transfer events with an associated animated door image. Creating doors is as easy as transfers, so let's start!

Go to your village's map and enable the Event mode. Right click on the empty door space in the shop's building and select `Quick Event Creation/Door`.



A new dialog window will appear. If you double click on the door's image, you'll be able to select a different picture for the door.

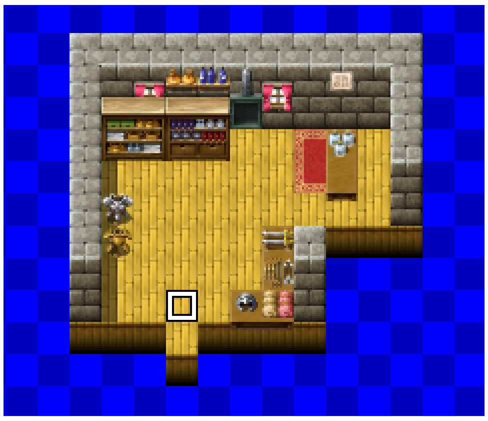


New Door Event

Graphic:

Destination: 002:Village (016,005)

Choose one that suits your building and click on the ellipsis to change the destination. Select a tile from your shop's interior map as the door's destination and you are done!



Once thing you might have noticed is that I didn't choose as destination the tile which later will be used as the shop's exit. Why? The teleporting will work anyway, but if the player makes a mistake and enters the shop without willing it, he would have to step off that tile and stepping on it again in order to leave the shop. It's just a matter of taste; you can set the same tile as your building's entrance and exit.

Now that we have set up the door, the only thing left is allowing the player to leave the shop. Create a teleporting event back to the village's map and playtest your map to see if everything is working as it should.

Repeat the steps to create a door for the inn. Don't forget to create also a transfer event to leave the inn and two more transfer events to go upstairs and downstairs.

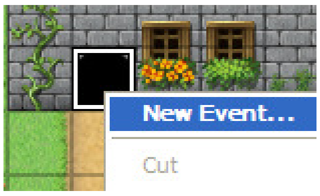
Remember that if you ever make a mistake creating events, you can **delete** them by right clicking on them and selecting `Delete`.

DECORATION EVENTS

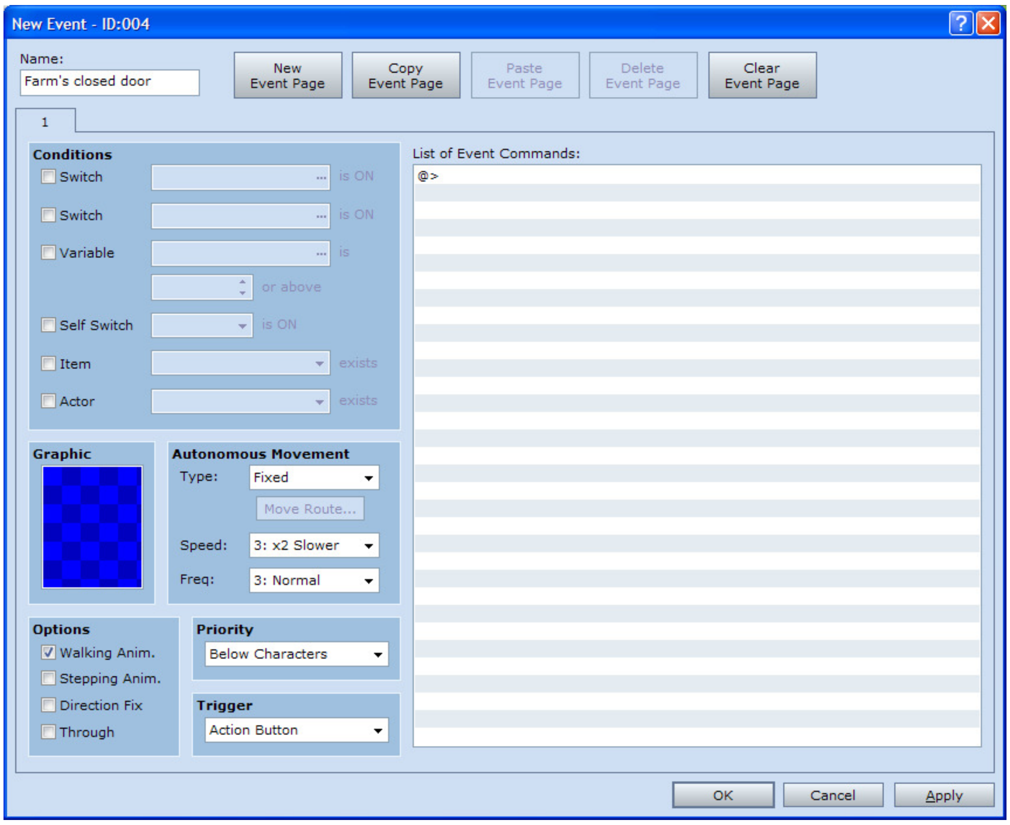
Those quick events we have created until now have a behavior attached to them. Now we are going to create some **behaviorless events**, which will be only used for decoration purposes.

In the village (well, at least in mine) there is some building without an interior map associated. The player, therefore, won't be able to enter on it. If you browse all the tile tabs, you will notice that there are not closed doors as tiles. However, there are some sprite sheets that can be attached to events: you have already seen them with the doors events.

So, in order to put a **closed door** in our map, we have to create a new event. Right click on the doors empty space and select `New Event`.



The new event dialog will appear. It's quite large, but don't panic!



Name: Farm's closed door

Conditions
- [ ] Switch ... is ON
- [ ] Switch ... is ON
- [ ] Variable ... is ... or above
- [ ] Self Switch ... is ON
- [ ] Item ... exists
- [ ] Actor ... exists

List of Event Commands:
@>

Graphic

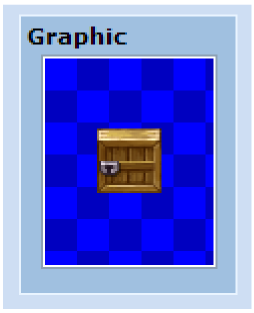
Autonomous Movement
Type: Fixed
Speed: 3: x2 Slower
Freq: 3: Normal

Options
- [x] Walking Anim.
- [ ] Stepping Anim.
- [ ] Direction Fix
- [ ] Through

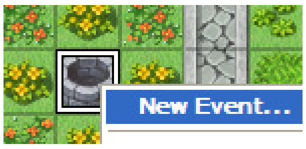
Priority: Below Characters

Trigger: Action Button

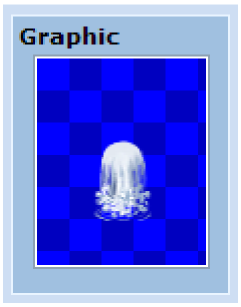
First things first: give your event an appropriated name. This will keep stuff organized and will help you greatly in the long run. Now double click in the graphic area and select a door you like. No more work here, we're finished!



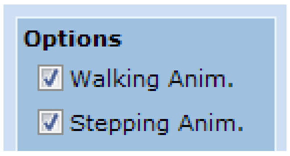
Behaviorless events can also have an **animated image**. Let's create a **fountain** for our town. Right click on the center of the town's square (I put a well there) and select `New event`.



The same dialog window will appear. Double click on the graphic area and choose a stream of water.



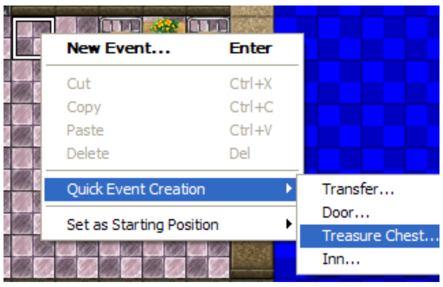
Right now this is just a non-animated event. To force the image to be animated even when the event is not moving, check the `Stepping anim` checkbox (you can uncheck the Walking anim checkbox as it is right now, it doesn't matter).



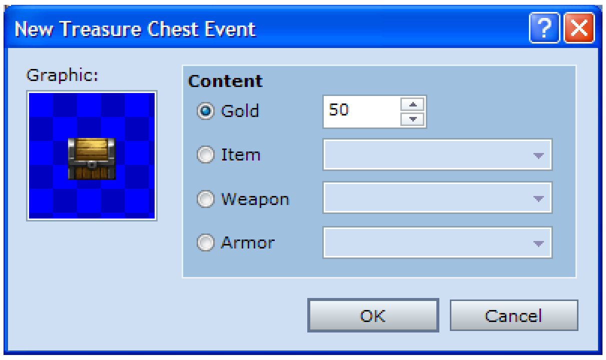
Playtest your game and watch our wonderful fountain!

TREASURE CHESTS

Chests can be created with the quick event creation. We are going to put a chest containing some gold at the second floor of the inn. Right click on a tile of that map and select Quick Event Creation/Treasure chest.



Select an image you like and click on the Gold radio button and set the amount of money you want to reward the player with.


New Treasure Chest Event

Graphic:

Content

- [✓ Gold] 50
- Item
- Weapon
- Armor

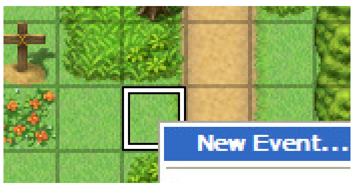
And that's all! Play and get the gold to see it working.

NPC'S AND DIALOGS

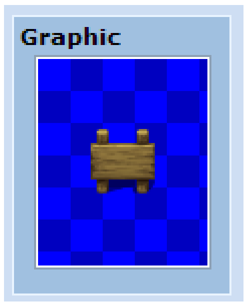
Non Player Characters (NPC's) and dialogs are also created via events (like almost everything!). In this chapter we will learn how to create dialogs and setup fully functional shops and inns!

SIGN POSTS

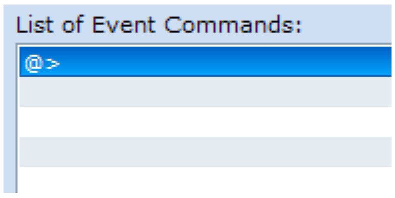
The first thing you should know is that dialogs can be triggered by any kind of event, not only NPC's. A **sign post** is just an event with an appropriate graphic with a dialog attached to it. To create one, right click on the entrance of your village and select `New event`.



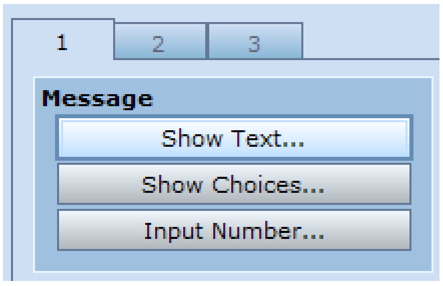
Give it a name and double click in the graphic area to select an image for this sign.



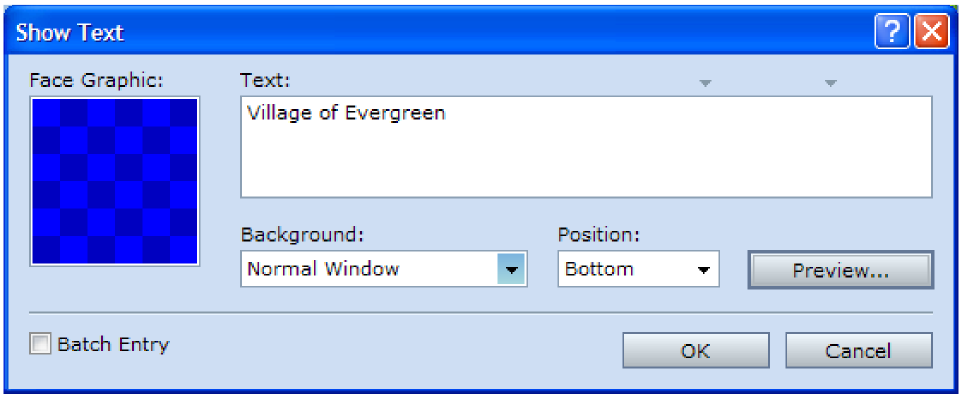
Right now this is just a decoration, behaviorless event. But we don't want it to be like that. To add custom behavior, double click on the `List of Event Commands` area.



A new dialog will appear, with tons of tabs and actions. Fortunately, this dialog is quite well organized, so it is likely you won't have any problems with that. The command we want to execute is called `Show Text` and is located inside tab 1.



Click on it and you will get to configure the text box that will appear. Type the name of your village and accept the dialogs.


Show Text

Face Graphic:

Text: Village of Evergreen

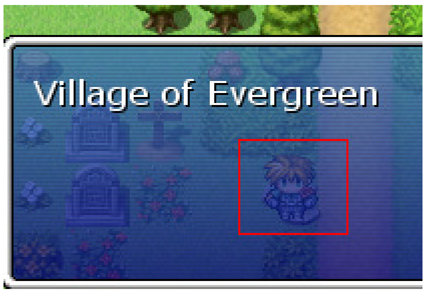
Background: Normal Window

Position: Bottom

There is still also a small, but very important, change to make. However, to understand it better playtest your game right now. Go to the sign post and press the Action button (key Enter). See? Nothing happens! Why?

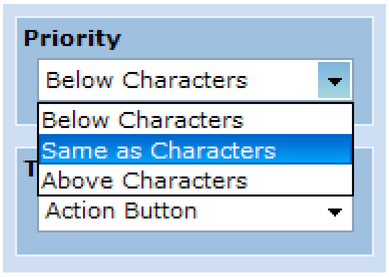
Let's do a short test. Try to change the image of the sing post for another one. Pick a tile your character could walk through if you drew it on a map. For example, these rocks:

Playtest and try again. This time you will notice that you can trigger the dialog if the character is right over the event.



However, this is not a correct behavior. What we want to do is trigger the dialog when the player is by the side of the sign post, not over it! Furthermore, since the sign post image we chose is a tile that won't let the character walk over it, the dialog will never be triggered.

Of course, we can change this. There is a panel named `Priority` with a dropdown list. With this, you can configure how the character will interact with the event. If you set it to `Same as characters`, the event will be like a physical entity with the same "rank" as the main character. That is, you can "talk" to the sign post as if it were a person!



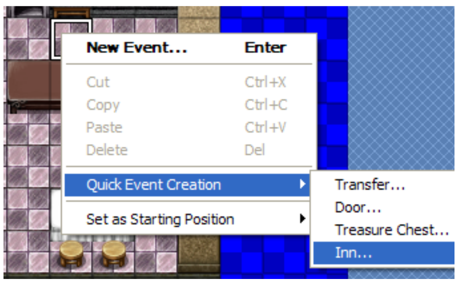
Other possible values are `Below characters` and `Above characters`. The first one is for events which should be rendered below the main character; the second one is for events which should be rendered above the main characters (like birds, for example).

Don't forget to set the sign post image again and playtest the game. It should work now as expected.

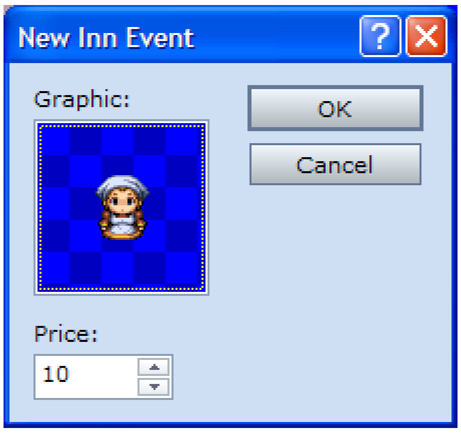
INNS

Creating an **inn** event it's very easy because it can be made with the `Quick Event Creation`. An inn event will let the party members to restore all their life and mana points in exchange for money.

Enable Event mode, choose the inn's map (first floor) and right click on a tile behind the counter and select `Quick Event Creation/Inn`.

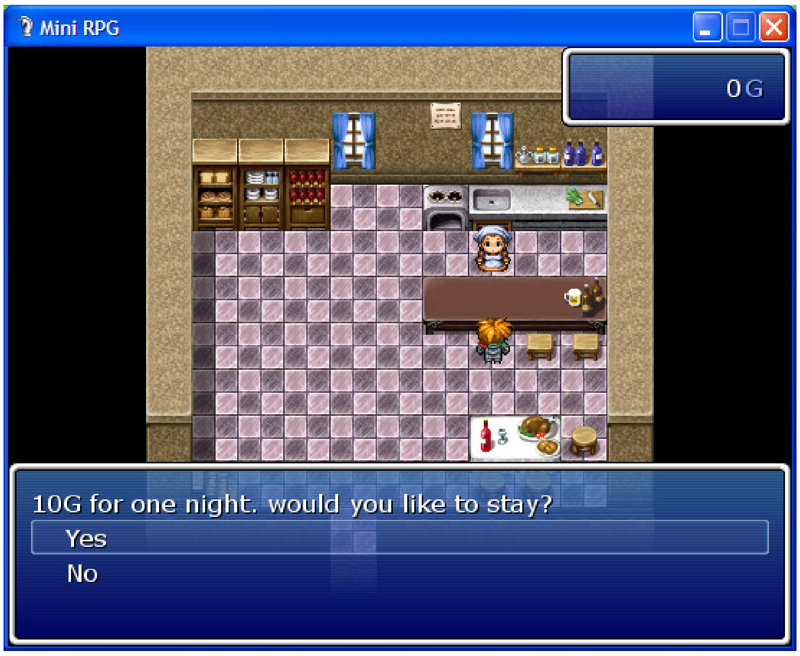


In the next dialog you get to choose a graphic for the innkeeper and the price to stay one night. Set these options as you like and… that's all!



Price: 10

Playtest your brand new inn and see how it looks. Cool, ain't it?

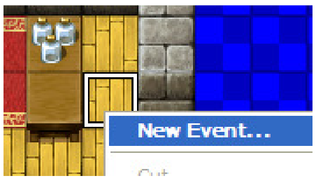


10G for one night. would you like to stay?
Yes
No

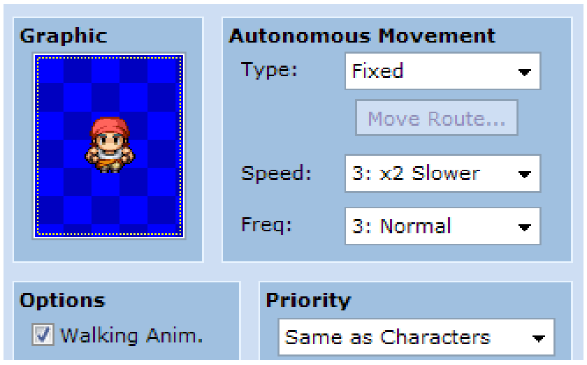
Dialog choices

We have already created some simple "dialog" with the sign post. But **dialogs** can get more complex than that: you can assign an avatar to every dialog box and also offer the player several **choices** and branch the dialog from there.

Let's create now some dialog with choices. It will be the start of our shop. In Event mode, right click on a shop's tile and select `New Event`.

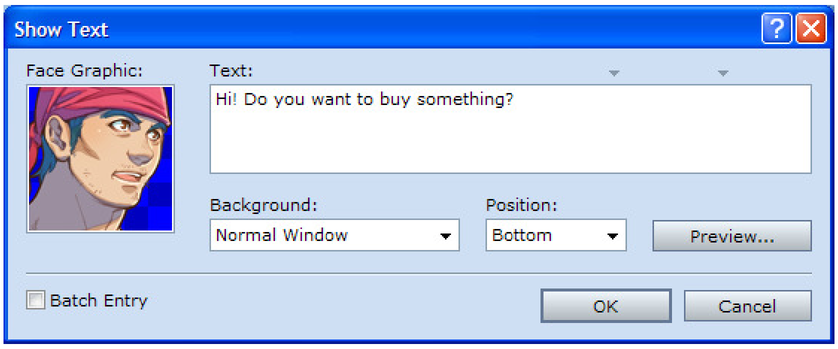


Name the event accordingly and choose a graphic for the NPC (this will be the shopkeeper). Also, make sure that `Priority` is set to `Same as Characters`.

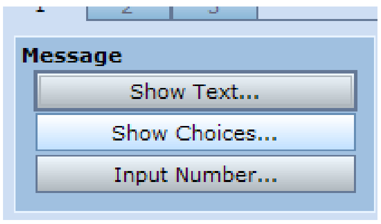

Graphic
Autonomous Movement
Type: Fixed

Speed: 3: x2 Slower
Freq: 3: Normal
Options
☑ Walking Anim.
Priority
Same as Characters

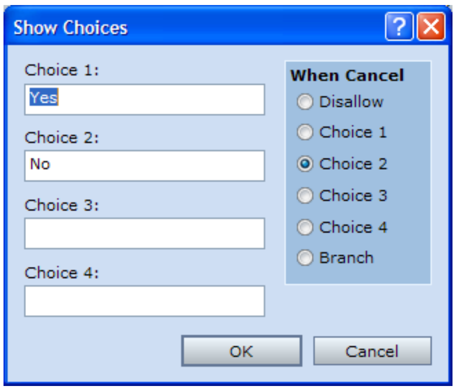
Now double click on the `List of Event Commands` to insert a new command. Choose `Show Text` to display a greeting message. Type some text and choose a face for the NPC's avatar.


Show Text
Face Graphic:
Text:
Hi! Do you want to buy something?
Background: Normal Window
Position: Bottom

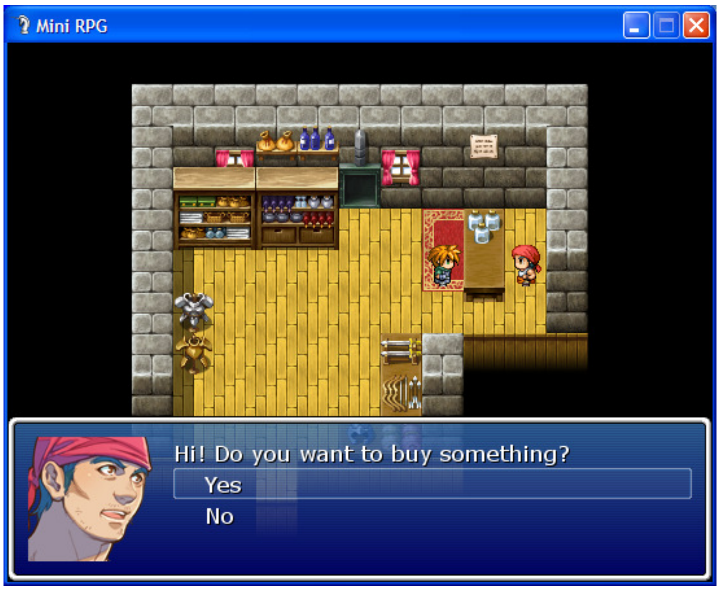
To display the choices, double click again on the List of Event Commands and select Show Choices (tab 1).



You can set up to four different choices in the next dialog. The `When Cancel` panel let's you configure which choice will be selected if the player press the Cancel button (key Escape). Leave the defaults (`Yes` and `No`) and accept the dialog.

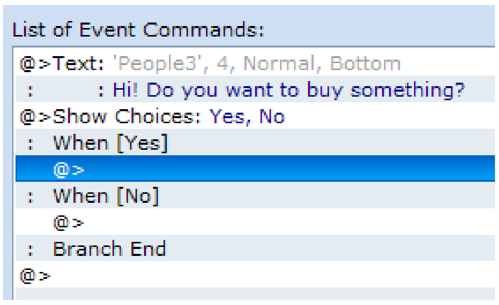

Show Choices
Choice 1: Yes
Choice 2: No
Choice 3:
Choice 4:
When Cancel
○ Disallow
○ Choice 1
● Choice 2
○ Choice 3
○ Choice 4
○ Branch

If you playtest the game now, you will be able to trigger the greeting and the choices, but nothing will happen when you select one of them.



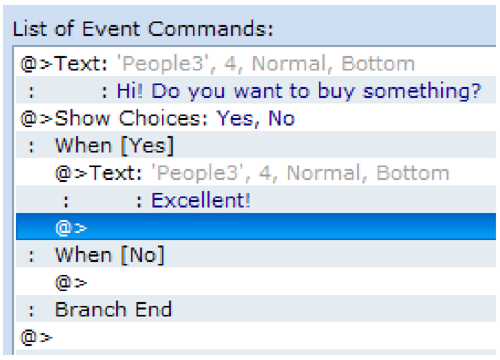
Let's fix that. Right click on the event and select `Edit event`. The event's dialog will appear and you can continue to make changes to it. Take a look to the `List of Event Commands`:

You can see a bifurcation with two branches: `When Yes` and `When No`. These are our options. Inside the `Yes` branch you can insert the commands that will be executed when the player selects `Yes` in the dialog. Double click inside this branch, select a Show Text command and create some test message.



```
List of Event Commands:
@>Text: 'People3', 4, Normal, Bottom
:        : Hi! Do you want to buy something?
@>Show Choices: Yes, No
:  When [Yes]
   @>
:  When [No]
   @>
:  Branch End
@>
```

Now you should have something like this:

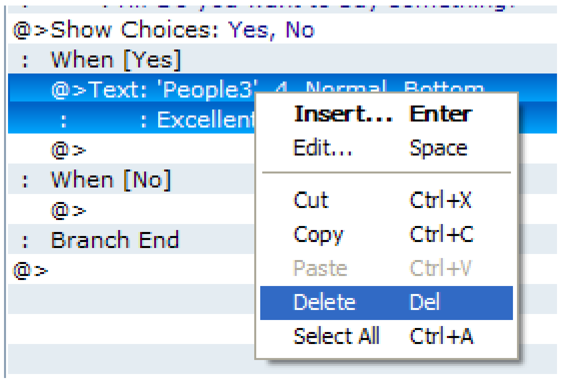


```
List of Event Commands:
@>Text: 'People3', 4, Normal, Bottom
:        : Hi! Do you want to buy something?
@>Show Choices: Yes, No
:  When [Yes]
   @>Text: 'People3', 4, Normal, Bottom
   :        : Excellent!
   @>
:  When [No]
   @>
:  Branch End
@>
```

Playtest it and check everything is working fine.

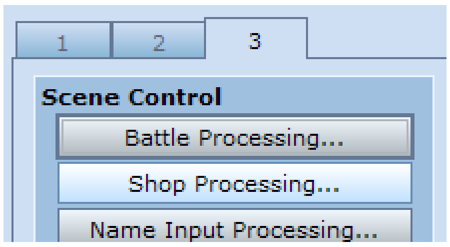
SHOPS

Let's now finish the **shop**. Edit the shopkeeper event and delete the test message we created to test the Yes branch. You can do this by right clicking on the command and selecting `Delete`.

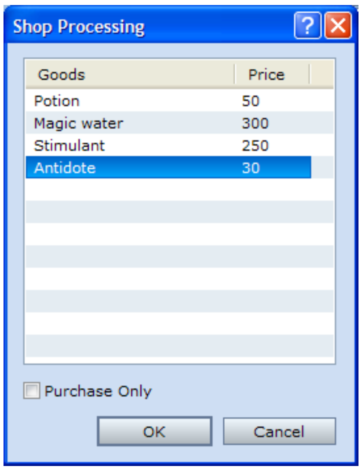


```
@>Show Choices: Yes, No
 :  When [Yes]
    @>Text: 'People3'
     :         : Excellent
    @>
 :  When [No]
    @>
 :  Branch End
@>
```

Fortunately for us, a shop is just another command. So double click inside the `Yes` branch and choose `Shop Processing` in tab 3.



In the new dialog that will appear you can set up all the items that will be sold in this shop. If you check the `Purchase only` checkbox, the player won't be allowed to sell items from the inventory. Double click on the panel to pick an item. Insert *Potion*, *Magic Water*, *Antidote* and *Stimulant*. This shop will only sell items, neither weapons nor armors.

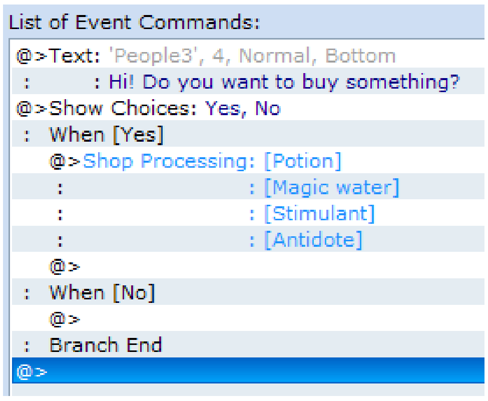


Shop Processing

Goods	Price
Potion	50
Magic water	300
Stimulant	250
Antidote	30

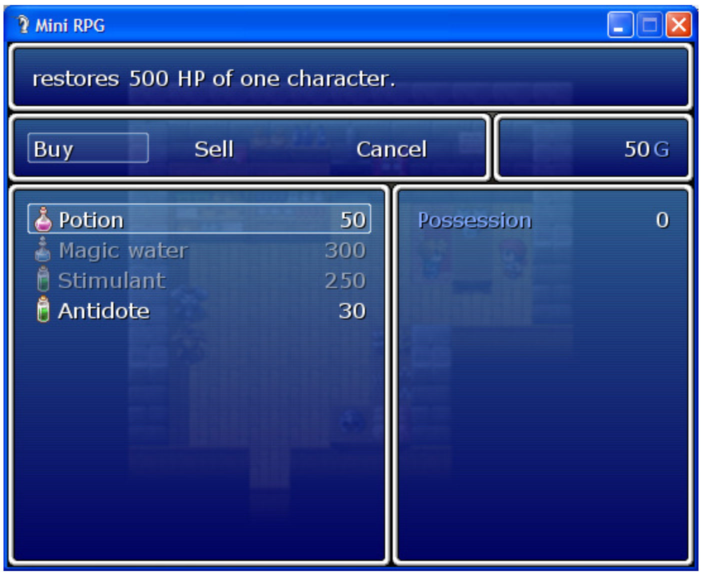
☐ Purchase Only

Your list of event commands should be like this one:



```
List of Event Commands:
@>Text: 'People3', 4, Normal, Bottom
:        : Hi! Do you want to buy something?
@>Show Choices: Yes, No
:  When [Yes]
   @>Shop Processing: [Potion]
    :                 : [Magic water]
    :                 : [Stimulant]
    :                 : [Antidote]
   @>
:  When [No]
   @>
:  Branch End
@>
```

Play the game to see the shop in action. If you want to buy something, you can get some money in the treasure chest that is located at the inn's second floor.



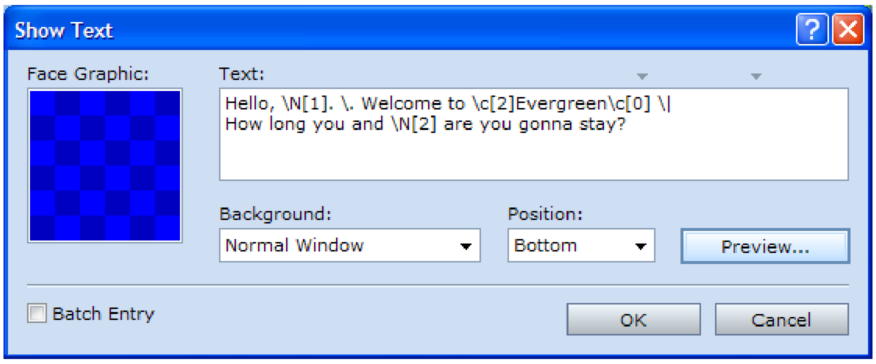
```
Mini RPG
restores 500 HP of one character.
Buy        Sell        Cancel        50 G
Potion          50     Possession     0
Magic water    300
Stimulant      250
Antidote        30
```

CUSTOMIZING TEXTS

The text you type in dialog boxes can be **customized** to display the character's name or to appear written with a different color, for example. The way to do this is via some markup tags (like those you use in some forums to put text in bold, for example). Some of these tags are:

- `\.` stops for 0.25 seconds
- `\|` stops for 1 second
- `\c[x]` changes the color, where x is the number of the color we want to use. Default is 0 (white)
- `\G` opens an additional text box displaying the money the party has got.
- `\N[x]` displays the actor's (a member of the party) name, where x is the number of that actor.

Let's try this with an example. Edit your sign post event and add a new command to the List `of Command Events`. Choose `Show Text` and type the following message:

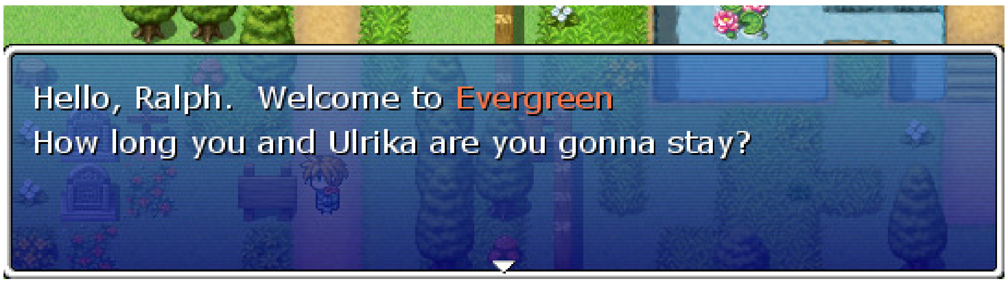


Hello, \N[1]. \. Welcome to \c[2]Evergreen\c[0] \|
How long you and \N[2] are you gonna stay?

Background: Normal Window
Position: Bottom

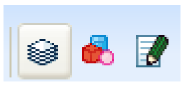
```
Hello, \N[1]. \. Welcome to \c[2]Evergreen\c[0] \|

How long you and \N[2] are you gonna stay?
```

Playtest the game and trigger the sign post event. You should see something like this (pay attention to pauses made with `\.` and `\|`).

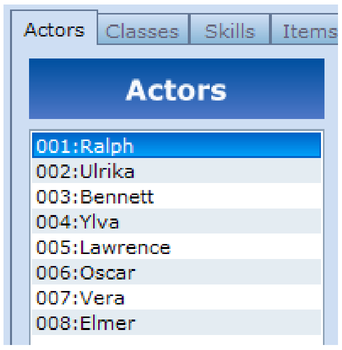


Hello, Ralph. Welcome to Evergreen
How long you and Ulrika are you gonna stay?

From this example you can learn that, in order to write text with a different color, you have to use \c[x] *and* \c[0] to write in white color again. Also, you will notice that characters (actors) number 1 and 2 are called Ralph and Ulrika. How do we know which character corresponds to which number? Easy, just click on the `Database` button in the toolbar.



In the Actors tab you will see the characters that RPG Maker VX creates by default. They can be changed, of course (and you don't have to use exactly 8 characters), but we will do it later. Character number 1 is always the main character, so it's pretty safe to use that number.

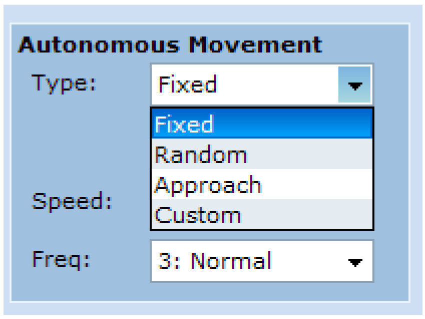


001:Ralph
002:Ulrika
003:Bennett
004:Ylva
005:Lawrence
006:Oscar
007:Vera
008:Elmer

Now you can delete this last command, for it makes no sense in a sign post.

WALKING NPC'S

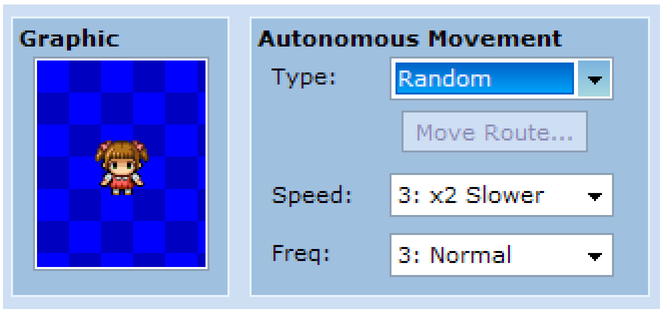
NPC's (and events in general) don't have to be static in a tile: they can move across the map. There are different types of **movement**; all of them correspond to options in the Autonomous Movement panel.



- `Fixed`: no movement at all (default)
- `Random`: movement in random directions
- `Approach`: makes the event chase the main character
- `Custom`: let's you customize the movement

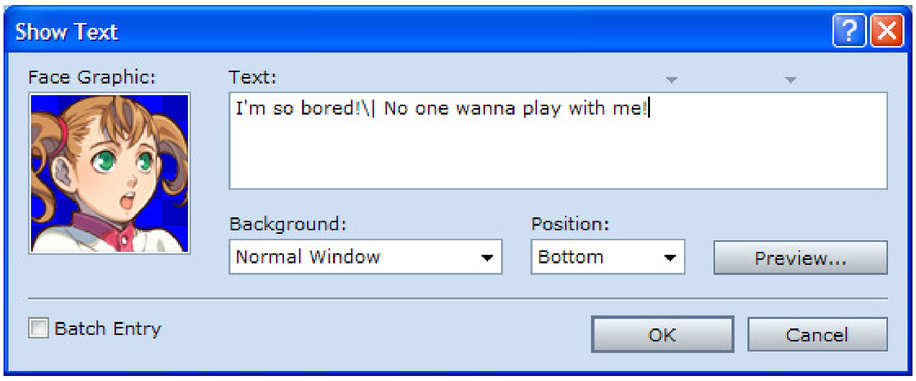
Also, you can set the Speed (how fast the event moves) and the frequency (how often the event moves).

As an example, let's create a kid who lives in the town and just plays around. Create a new event in the village's map (or maybe he's the son of the innkeeper, that's up to you), set a graphic for it and select the type of movement `Random`.

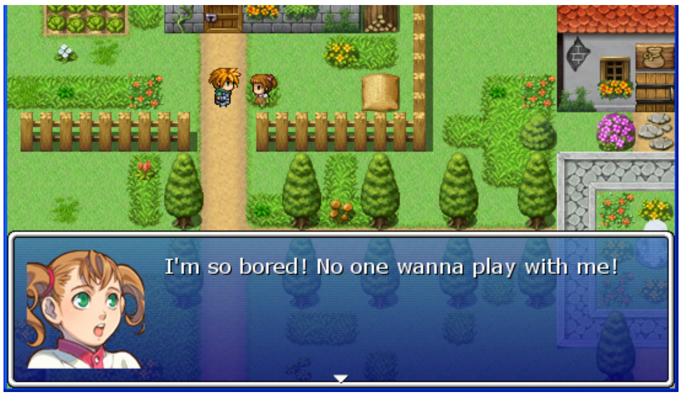

Graphic

Autonomous Movement
Type: Random
Move Route...
Speed: 3: x2 Slower
Freq: 3: Normal

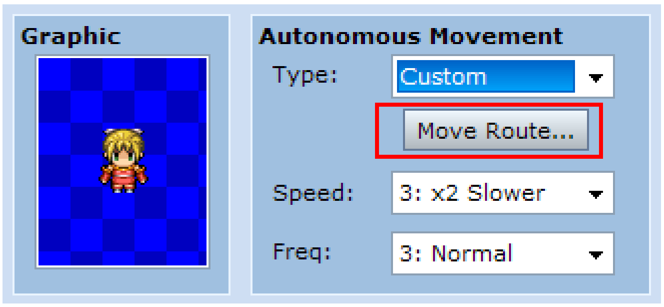
In addition, make her to say some words when the main character talks to her.


Show Text
Face Graphic:
Text:
I'm so bored!\| No one wanna play with me!
Background: Normal Window
Position: Bottom
Preview...
Batch Entry
OK
Cancel

Playtest the game to see this new NPC in action. Experiment with different values for both `Speed` and `Frequency` and see what happens.

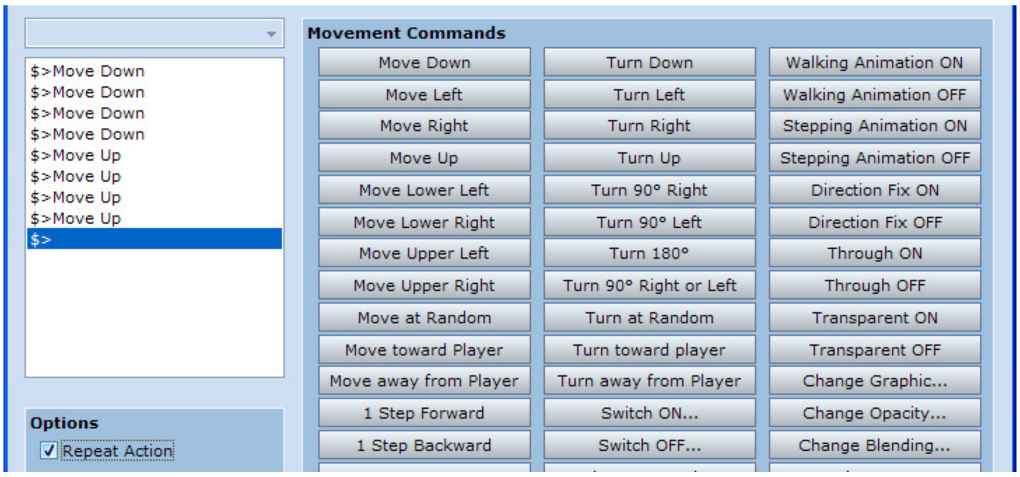


Let's create another walking NPC: this time it will be **Lime**, our sorceress. Create a new event, assign a name and a graphic to it and set her movement to `Custom`. Then click on the `Move Route` button.



From this new dialog you can change the character's route step by step. There are a lot of options: you can make the NPC move towards any direction and also turn. You can also make him/her jump, approach the main character, run away from him, etc.

For now, give her some simple route which she can follow (this depends on how you built your map and where the event is placed). In my example, I made her move four tiles down and then return to her original position.



```
$>Move Down
$>Move Down
$>Move Down
$>Move Down
$>Move Up
$>Move Up
$>Move Up
$>Move Up
$>
```

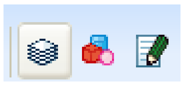
Repeat Action [✓]

If you check the Repeat Action checkbox, the NPC will repeat this same route forever. Playtest his now and experiment with the different movement commands you can assign to the NPC’s route.

GAME ELEMENTS

We are going to make a break here to customize our **game elements**: items, weapons, jobs, and such. This configuration is vital for your game —it will really make a difference! So in your own games you have to think all this stuff very carefully.

You can customize your game elements in the Database Manager. To open this window, click on the `Database` button in the toolbar.

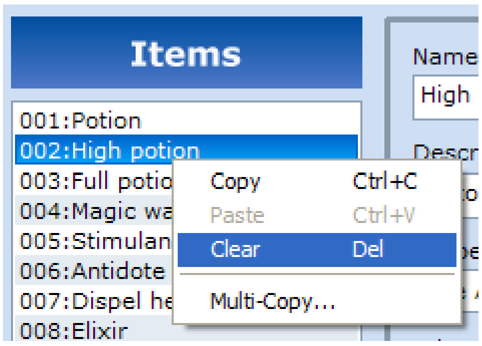


GETTING RID OF JUNK

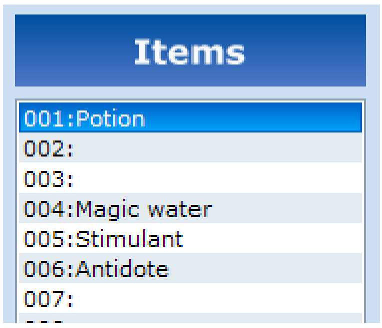
Items are objects that the party members can use. Some items can be used always, but some of them can only be used under some circumstances (for example, while in battle). You can view all the items in your game in the `Items` tab. You will see a bunch of them, since RPG Maker VX creates some items by default.

We are not having all of these in our mini-RPG, so let's **delete** all items we don't need. The same very steps apply to weapons, armors, jobs, etc.

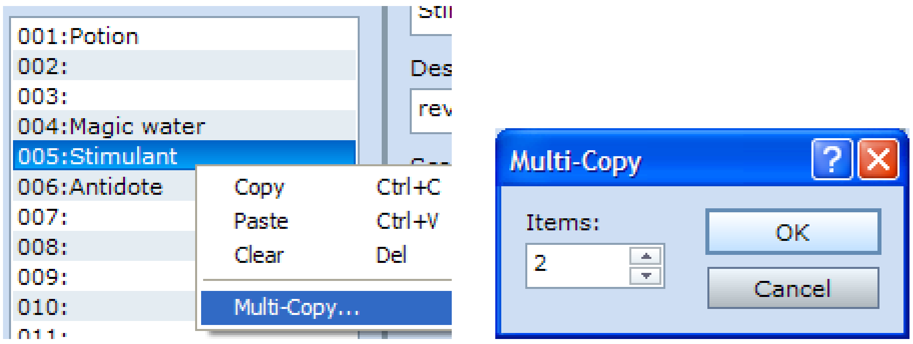
To delete an item, right click on its name in the list and select `Delete`. You can also left click on it and press `Supr` key.



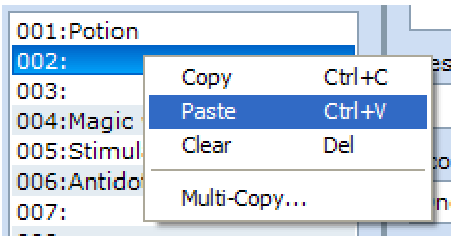
Proceed to delete all items but *Potion*, *Magic Water*, *Stimulant* and *Antidote*. Notice that deleting an item only deletes its data, but its slot still remains on the list.



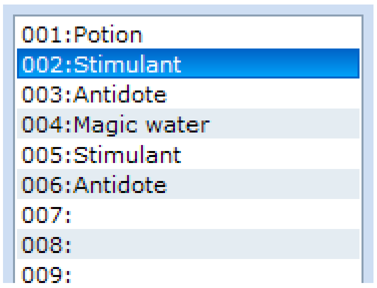
You have to fill the slots which cause **gaps** on the list. In our case, we have a two-slot gap caused by slots number 2 and 3, so we have to put *Stimulant* and *Antidote* in that gap. To do that, right click on *Stimulant* and select `Multicopy`. In the new window, type 2 as a number of items to copy and click OK.



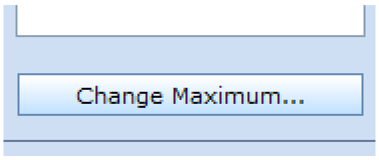
Now right click on the slot number 2 and select `Paste`.



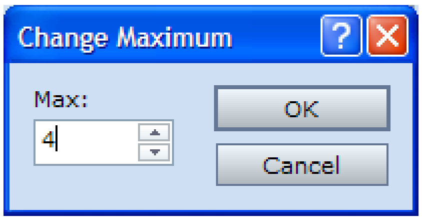
And here is how our items list looks now:


001:Potion
002:Stimulant
003:Antidote
004:Magic water
005:Stimulant
006:Antidote
007:
008:
009:

To get rid of the duplicates, click on the Change Maximum button at the bottom of the dialog window.

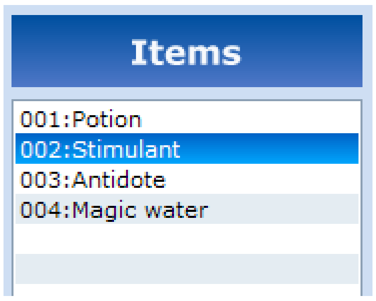


Now type the numbers of items you want to keep (4 in our case):



Max: 4

And *voilà*!


Items
001:Potion
002:Stimulant
003:Antidote
004:Magic water

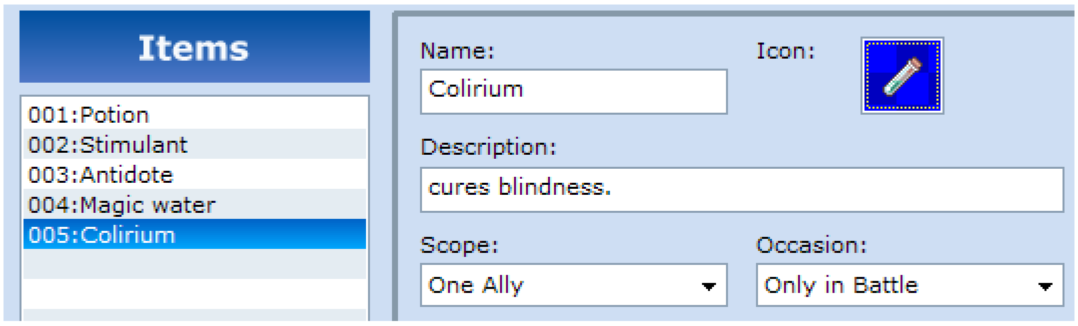
ITEMS

Now that we know how to delete all the junk we don't want, we will learn how to create *our own* junk. We will start by **creating a custom item** for our game: a *Colirium* to cure *Darkness* state[2].

To do so, go to Items tab and click the `Change Maximum` button. Increase by 1 the number that will appear in the window (in our case, type 5). This will create a new empty slot for our own item. Left click on the new slot to customize its behavior.

[2] *Darkness* state cause a character to miss most of his/her physical attacks. In some games is called *Blindess*.

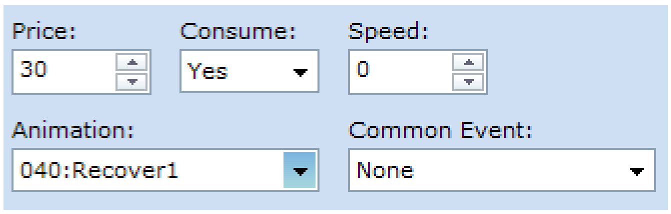
Start by given the new Item a `name` and a `description`. The `Scope` configures *who* can use the item. In our case, only members of the party can use this item, so set it to `One Ally`. `Ocassion` sets *when* the item will be able to be used. Since *Darkness* is a state that occurs only in battle and is healed afterwards, select `Only in Battle`. Choose also an icon for your item.


Items

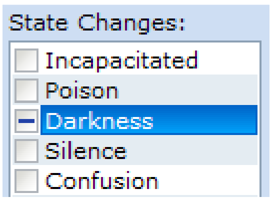
001:Potion
002:Stimulant
003:Antidote
004:Magic water
005:Colirium

Name: Colirium
Icon:
Description: cures blindness.
Scope: One Ally
Occasion: Only in Battle

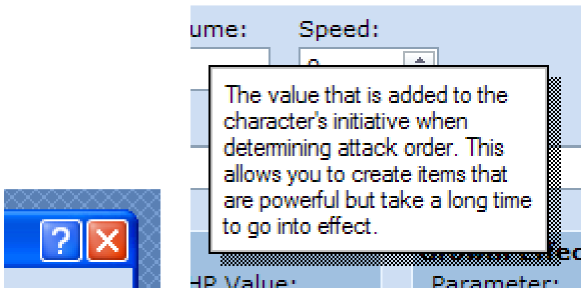
Now set its price (30 money units, like the *Antidote*) and the animation that will be displayed when used.


Price: 30
Consume: Yes
Speed: 0
Animation: 040:Recover1
Common Event: None

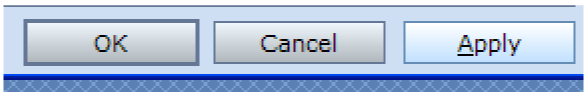
The other panels in the window are used to customize the item's effects. This item will only cure Darkness, so go to the State Changes panel, and click on the Darkness checkbox until a minus sign appears:


State Changes:
☐ Incapacitated
☐ Poison
⊟ Darkness
☐ Silence
☐ Confusion

This will remove the *Darkness* state from the item's target. If you are curious about what does some option, click on the interrogation icon in the title bar and then click on an element:


ume:
Speed:
The value that is added to the character's initiative when determining attack order. This allows you to create items that are powerful but take a long time to go into effect.
HP Value:
Parameter:

To finish this item, click on the Apply or OK buttons.



STATES

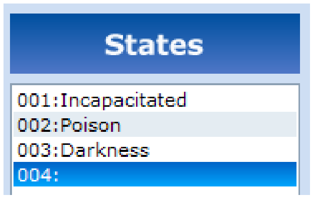
States are conditions which affect a character (being a party member or an enemy). States can change temporally the amount of damage that a character does per attack, drain his life, paralyze him, increase his defense, etc. Some states only take effect when in battle (and are cancelled afterwards) and some of them still take effect when the character is moving around a map.

In our small RPG, we will be having these states:

- ***Incapacitated***: a character is knock out ("dead") and cannot participate in the fight.
- ***Darkness***: a character misses 75% of his physical attacks. This state is healed after battle.
- ***Poison***: a character loses life in every action he does.
- ***Berserker***: a character does more damage than usual, but he attacks physically (and automatically) the enemies. This state is healed after battle.

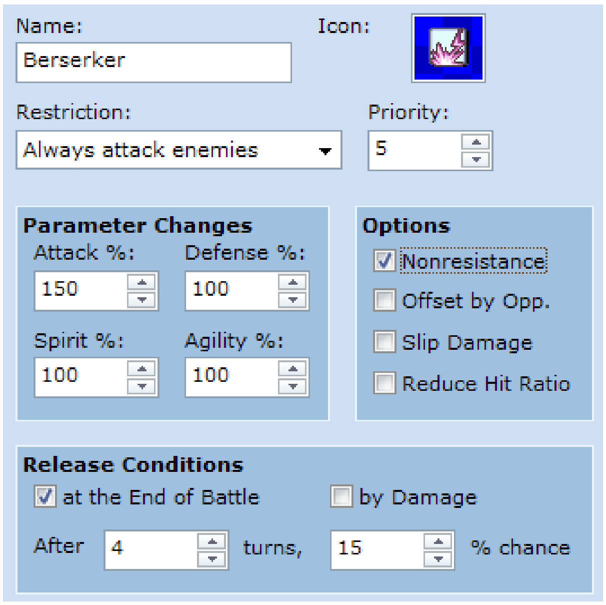
The Berserker state will be induced thanks to Gladius, so only the main character will be able to go on berserk.

All of these states are already created by default in RPG Maker VX, except *Berserker*. So, first of all, delete all the other states and create a new slot for the *Berserker*. You can access to the different states in the game from the `States` tab in the `Database`.


States

001:Incapacitated
002:Poison
003:Darkness
004:

And here is the configuration for this state.


Name: Berserker
Icon:
Restriction: Always attack enemies
Priority: 5

Parameter Changes
Attack %: 150
Defense %: 100
Spirit %: 100
Agility %: 100

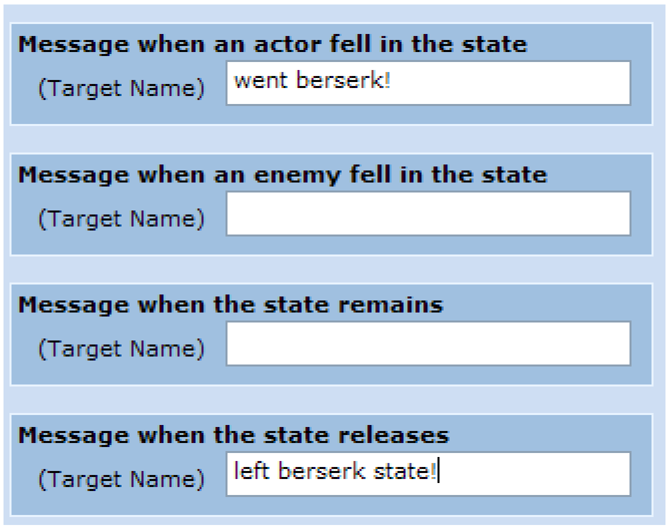
Options
[✓] Nonresistance
[ ] Offset by Opp.
[ ] Slip Damage
[ ] Reduce Hit Ratio

Release Conditions
[✓] at the End of Battle
[ ] by Damage
After 4 turns, 15 % chance

It has a custom icon and `Always attack enemies` as a restriction. This emulates the "blood thirst" of this state, so the player won't be able to select actions for this character when in berserker: the character will always attack the enemy. In the `Parameter Changes`, a bonus has been set for `Attack`: the character will cause 50% more damage than usual.

Furthermore, with the `Nonresistance` checkbox enabled in `Options`, changes to the berserker state will always be successful (some states can be cancelled by a character's resistance or luck). `Release conditions` specify how the state will be "healed". We have set it to `End of Battle` and *also* after 4 turns with 15% chance.

Now we just have to fill the information messages that will be displayed in battle regarding this state. Since we only will inform the player when a character has gone berserk or not, we only have to fill two message boxes (enemies won't be able to go berserk).



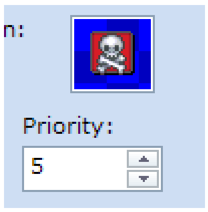
Message when an actor fell in the state
(Target Name) went berserk!

Message when an enemy fell in the state
(Target Name)

Message when the state remains
(Target Name)

Message when the state releases
(Target Name) left berserk state!

The only thing left is configuring the **priorities** for the icon's display. In RPG Maker VX only one icon can be displayed to indicate the character's status, so you have to choose which states are more important, so their icons will be displayed instead of others. You can set priorities in the Priority control of each state (the higher the number, the more priority)



n:

Priority: 5

Our states priority is: Incapacitated → Poison → Berserker → Darkness.

WEAPONS AND ARMORS

In our game, the party will not be allowed to upgrade their equipment, except to equip the sword *Gladius*. We will have then a reduced set of weapons and armors.

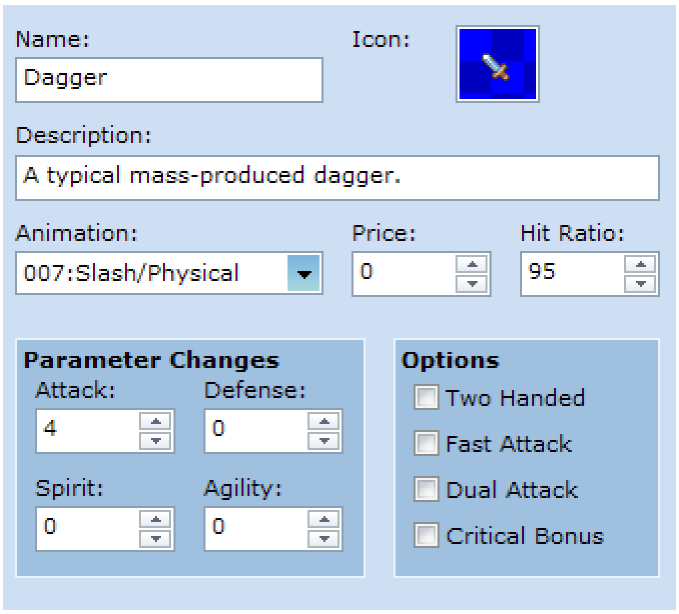
- **Salazar**: long sword, leather shield, leather breastplate.

- **Zach**: two daggers, leather breastplate.
- **Lime**: robe, rod

In the `Weapons` tab you can configure all the weapons in the game. Delete all the default weapons, unless *Long Sword*, and change the maximum amount of weapons to 4.

First, we will create the ***Dagger*** for the rogue in the party. If you look at the *Long Sword* attributes, you will see that has an attack of 5. We want the dagger to be less powerful, so we will set it to 4. Other parameters are very similar.

In the animation list we get to choose how this attack will be displayed while in battle. You can browse the different animation sin the `Animations` tab. The Hit Ratio is the percentage of success when hitting. 95% is the standard, but you might want to lower it for weapons such as great axes, for example.

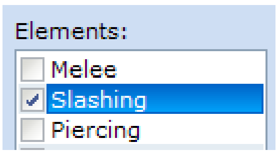


Name: Dagger
Description: A typical mass-produced dagger.
Animation: 007:Slash/Physical
Price: 0
Hit Ratio: 95

Parameter Changes
Attack: 4
Defense: 0
Spirit: 0
Agility: 0

Options
☐ Two Handed
☐ Fast Attack
☐ Dual Attack
☐ Critical Bonus

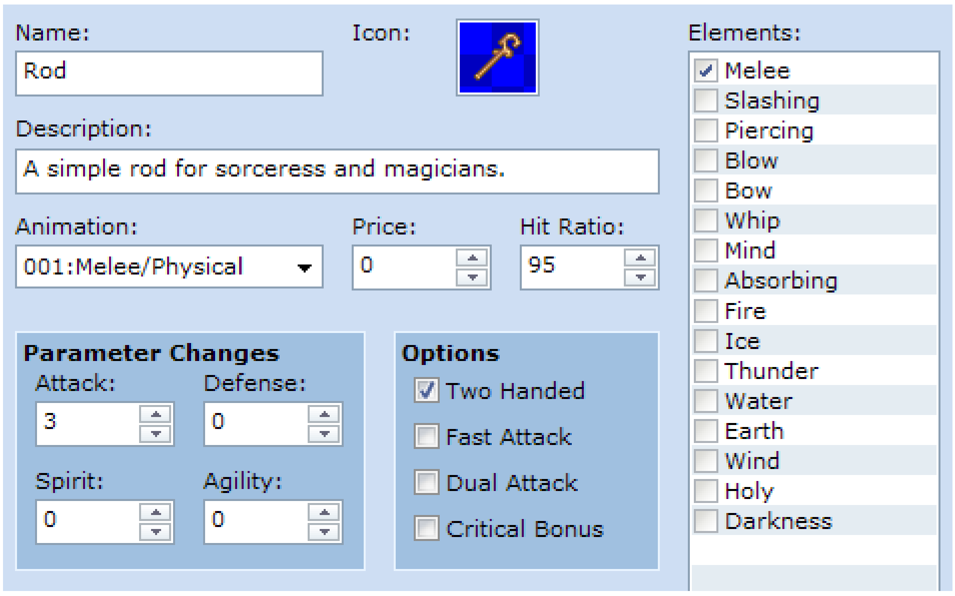
Indicate also which kind of damage does your brand new weapon. In our case, it will be `Slashing`.



Elements:
☐ Melee
✓ Slashing
☐ Piercing

Regarding the `Price`, set it to 0 to avoid this weapon being sold in shops. Don't forget to set to 0 the price for the *Long Sword*!

Let's create now the ***Rod*** for the sorceress. It will be a weak weapon, since the real power of the sorceress is in her spells. It will be a two-handed weapon.



Name: Rod

Icon:

Description: A simple rod for sorceress and magicians.

Animation: 001:Melee/Physical

Price: 0

Hit Ratio: 95

Parameter Changes

Attack: 3
Defense: 0
Spirit: 0
Agility: 0

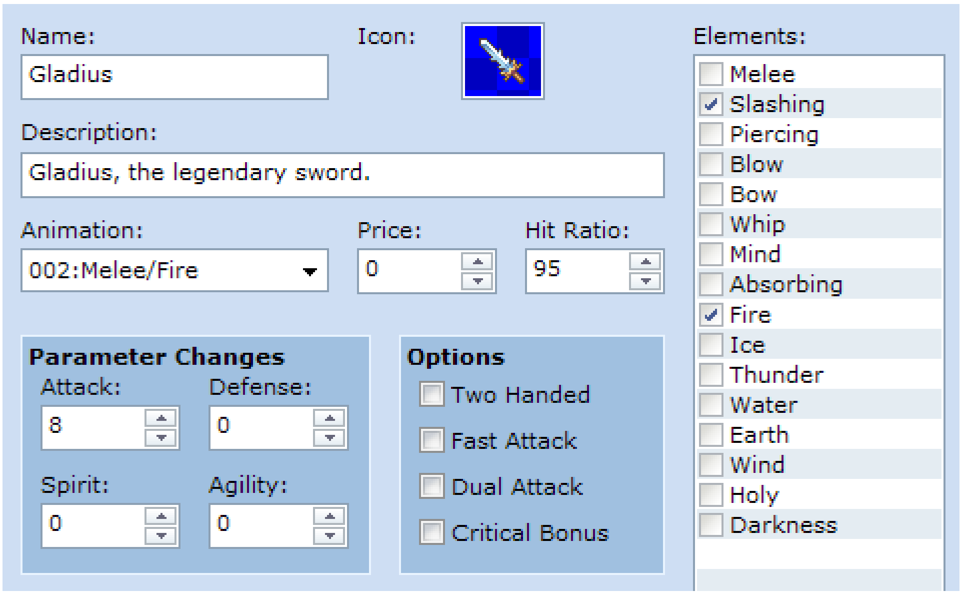
Options

- [✓] Two Handed
- [ ] Fast Attack
- [ ] Dual Attack
- [ ] Critical Bonus

Elements:

- [✓] Melee
- [ ] Slashing
- [ ] Piercing
- [ ] Blow
- [ ] Bow
- [ ] Whip
- [ ] Mind
- [ ] Absorbing
- [ ] Fire
- [ ] Ice
- [ ] Thunder
- [ ] Water
- [ ] Earth
- [ ] Wind
- [ ] Holy
- [ ] Darkness

And this is ***Gladius***, the legendary sword. Notice how this sword does Fire damage and this gets reflected on its animation.



Name: Gladius

Icon:

Description: Gladius, the legendary sword.

Animation: 002:Melee/Fire

Price: 0

Hit Ratio: 95

Parameter Changes

Attack: 8
Defense: 0
Spirit: 0
Agility: 0

Options

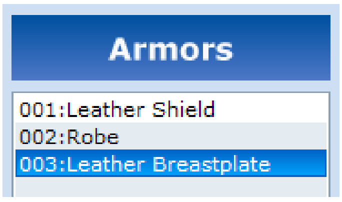
- [ ] Two Handed
- [ ] Fast Attack
- [ ] Dual Attack
- [ ] Critical Bonus

Elements:

- [ ] Melee
- [✓] Slashing
- [ ] Piercing
- [ ] Blow
- [ ] Bow
- [ ] Whip
- [ ] Mind
- [ ] Absorbing
- [✓] Fire
- [ ] Ice
- [ ] Thunder
- [ ] Water
- [ ] Earth
- [ ] Wind
- [ ] Holy
- [ ] Darkness

Now that all weapons have been created, let's go for the **armors**. In order to save time, we are using some armors that come by default: *Robe*, *Leather Shield* and *Leather Breastplate*. Remove all armors but those from `Armors` tab.

Don't forget to set their price to 0 to forbid the player to sell them in shops.



Armors

001:Leather Shield
002:Robe
003:Leather Breastplate

The `Armors` tab is really similar to the `Weapons` tab, so we are not digging in it. Use the `Help` button when in doubt!

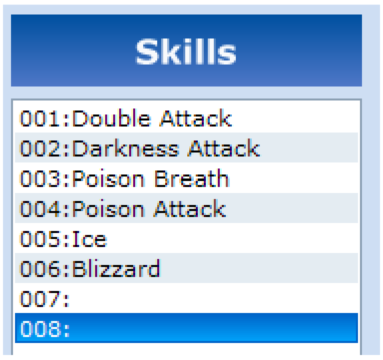
SKILLS

Skills are powers or abilities that a character can use, such as fireballs, healing, a special attack, etc. Skills can be used in battle and/or in maps.

In our game there will be these skills (keep in mind that *enemies* can use skills too):

- ***Darkness attack***: adds Darkness state to a common attack
- ***Poison attack***: adds Poison state to a common attack
- ***Berserk:*** goes on Berserker state
- ***Bash***: single, powerful attack
- ***Poison breath***: adds Poison state to all enemies
- ***Double attack***: attacks two random different enemies
- ***Ice***: causes ice damage to one enemy
- ***Blizzard***: causes ice damage to all enemies

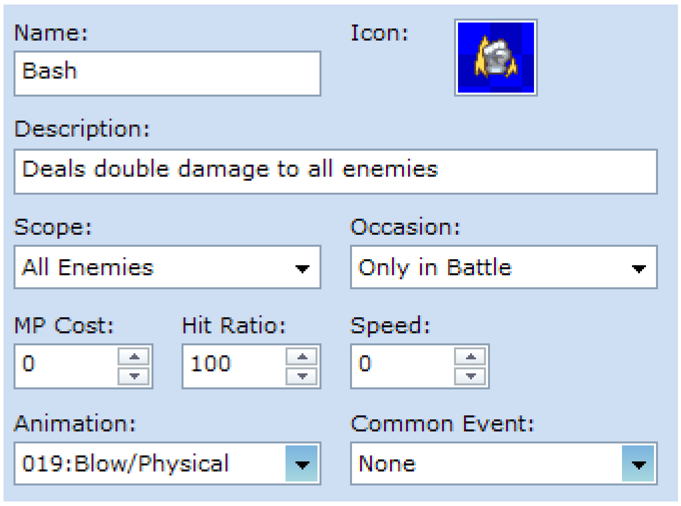
Some of those skills are already created by default, so go to Skills tab and delete all skills but these ones:



Skills

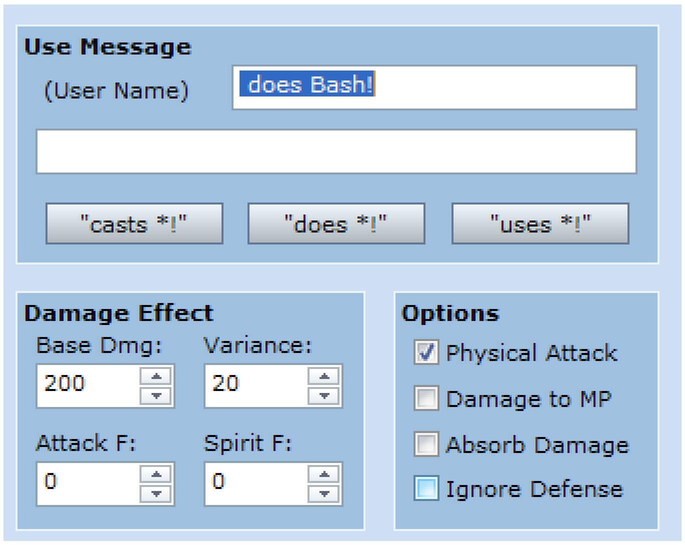
001:Double Attack
002:Darkness Attack
003:Poison Breath
004:Poison Attack
005:Ice
006:Blizzard
007:
008:

Let's create the remaining skills! We will start with ***Bash***, which will be a special ability for the final boss: the Ogre. This skill will be a powerful attack (200% damage) that will affect all enemies. Here you can see that this skill requires no magic points to be casted, and that can be used only while in battle. The rest of the parameters are pretty self-explanatory.



Name: Bash
Icon:
Description: Deals double damage to all enemies
Scope: All Enemies
Occasion: Only in Battle
MP Cost: 0
Hit Ratio: 100
Speed: 0
Animation: 019:Blow/Physical
Common Event: None

The `Use Message` is just an informative text that will be displayed in the fight to tell the player what's going on. The `Damage Effect` specifies the power of the attack: with 200 of `Base Damage`, the damage will be the double as usual. If you ever want to create a **healing** skill, enter a negative value here (for example, -100). Last, with the `Physical Attack` checkbox enabled, we are telling RPG Maker to treat this as a physical attack and not a spell.

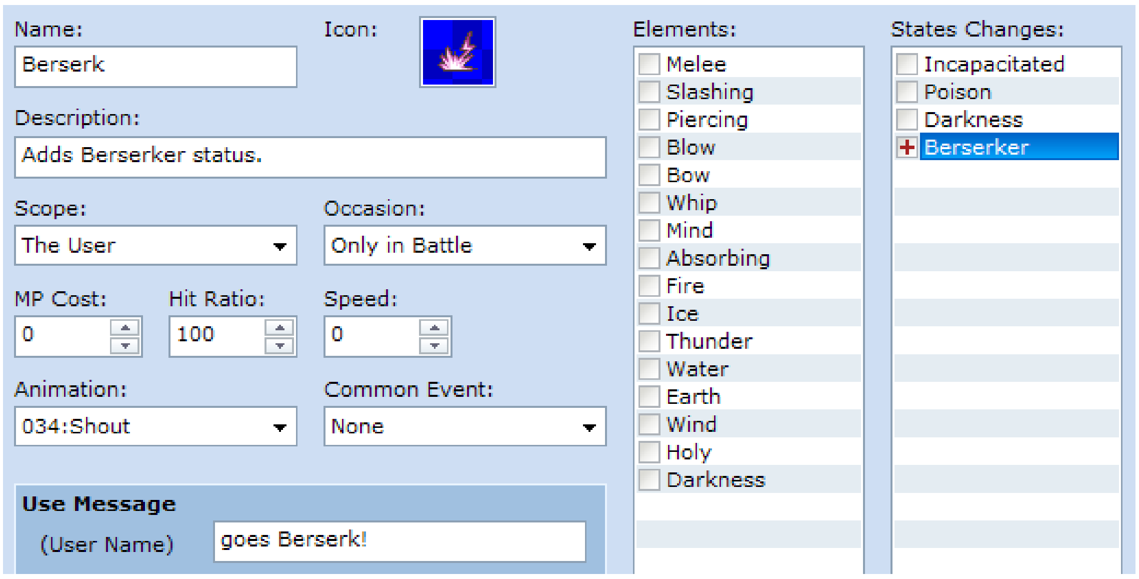


Use Message
(User Name) does Bash!
"casts *!" "does *!" "uses *!"

Damage Effect
Base Dmg: 200
Variance: 20
Attack F: 0
Spirit F: 0

Options
[✓] Physical Attack
[ ] Damage to MP
[ ] Absorb Damage
[ ] Ignore Defense

The other skill, ***Berserk***, is pretty simple. It just changes the user's state:



Name: Berserk
Icon:
Description: Adds Berserker status.
Scope: The User
Occasion: Only in Battle
MP Cost: 0
Hit Ratio: 100
Speed: 0
Animation: 034:Shout
Common Event: None

Use Message
(User Name) goes Berserk!

Elements:
[ ] Melee
[ ] Slashing
[ ] Piercing
[ ] Blow
[ ] Bow
[ ] Whip
[ ] Mind
[ ] Absorbing
[ ] Fire
[ ] Ice
[ ] Thunder
[ ] Water
[ ] Earth
[ ] Wind
[ ] Holy
[ ] Darkness

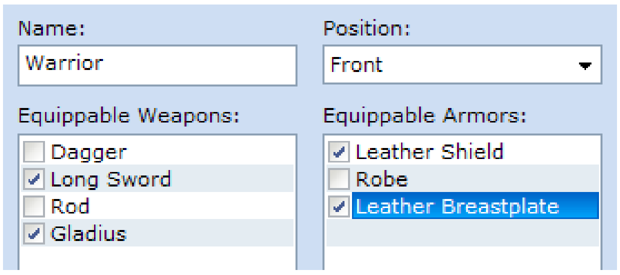
States Changes:
[ ] Incapacitated
[ ] Poison
[ ] Darkness
[+] Berserker

CLASSES

Classes are the jobs available for the characters. Each class different skills and equippable weapons and armors.

We are gonna have three different jobs in our game, and we will be creating these from zero, so delete all classes and get only three cleared slots. Name these slots as *Warrior*, *Rogue* and *Sorceress*.

We will begin by creating the ***Warrior*** class. Set its name and also its equippable weapons and armors. The `Position` list is for setting the relative position in the battlefield. Characters at the front line will be more likely to be selected as enemy targets. Jobs such as warriors, paladins and the like should be in the front line.



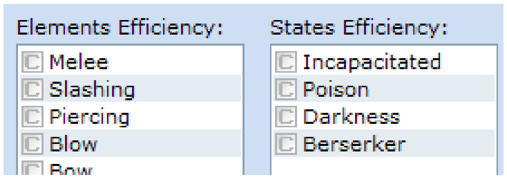
Name: Warrior

Position: Front

Equippable Weapons: Dagger, [✓ Long Sword], Rod, [✓ Gladius]

Equippable Armors: [✓ Leather Shield], Robe, [✓ Leather Breastplate]

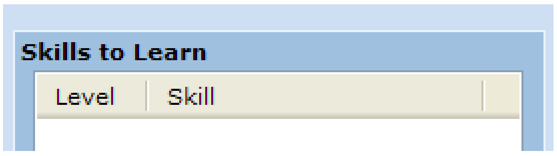
In the `Elements Efficiency` and `States Efficiency` panels you get to configure the job weak and strong points regarding element damage and state changes. You can tweak it, but in my own project I left the default values for everyone.



Elements Efficiency: C Melee, C Slashing, C Piercing, C Blow, C Bow

States Efficiency: C Incapacitated, C Poison, C Darkness, C Berserker

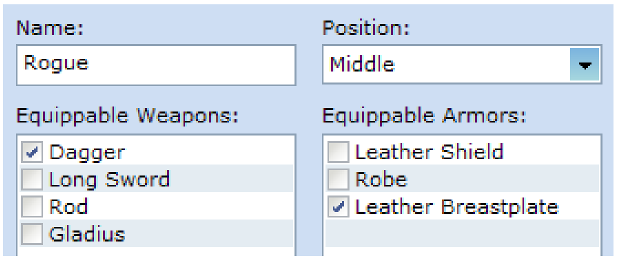
In the `Skills to Learn` panel we can set which skills the character with this job will learn at a given level.



Level	Skill

The only skill that our main character will learn is Berserker, and he will get it when he gets the sword *Gladius*, so we will leave this panel blank.

It's time for the ***Rogue***. Set the `Position` and `Equippable` items this way:



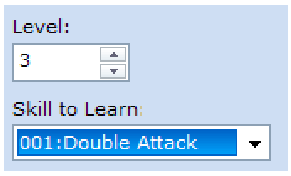
Name: Rogue

Position: Middle

Equippable Weapons: [✓ Dagger], Long Sword, Rod, Gladius

Equippable Armors: Leather Shield, Robe, [✓ Leather Breastplate]

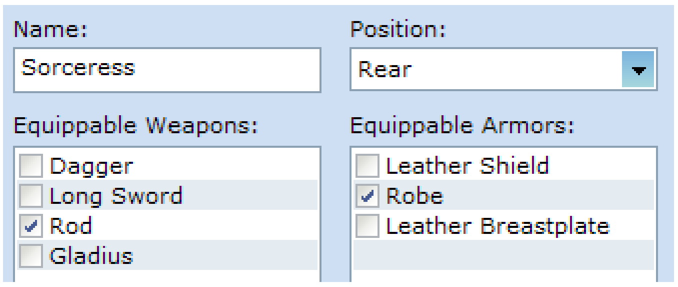
As for skills, the rogue will learn how to perform a double attack when he reaches level 3. To do that, double click in the `Skills to Learn` panel and a new window will appear:



Level: 3

Skill to Learn: 001:Double Attack

As for the Sorceress, the configuration would be this one:



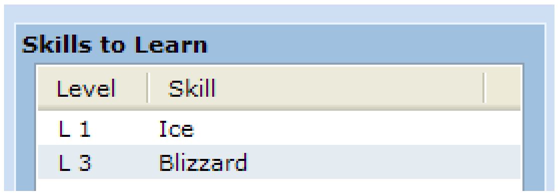
Name: Sorceress

Position: Rear

Equippable Weapons:
- ☐ Dagger
- ☐ Long Sword
- ☑ Rod
- ☐ Gladius

Equippable Armors:
- ☐ Leather Shield
- ☑ Robe
- ☐ Leather Breastplate

And the sorceress will have these skills to learn at levels 1 and 3:



Skills to Learn

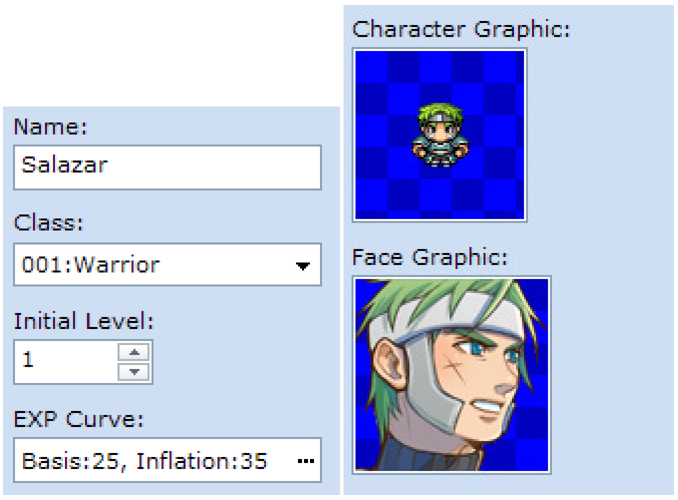
Level	Skill
L 1	Ice
L 3	Blizzard

PARTY MEMBERS

You get access to the **party members** at the `Actors` tab in the `Database`. There you can find some actors created by default. Delete them all and create three empty slots.

Inside this tab you can configure the appearance of your characters, their stats, initial equipment and other details. Let's start with our main character, ***Salazar***.

Give him a name and set his `Class` to *Warrior*. You can also select his avatar and his sprite to be displayed on maps.



Name: Salazar

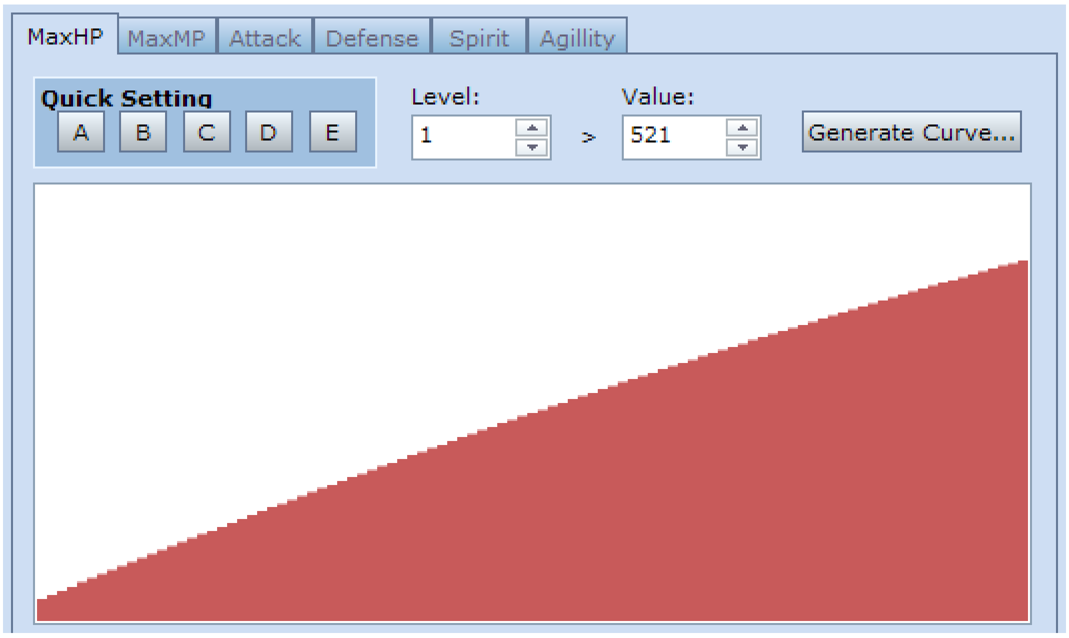
Class: 001:Warrior

Initial Level: 1

EXP Curve: Basis:25, Inflation:35

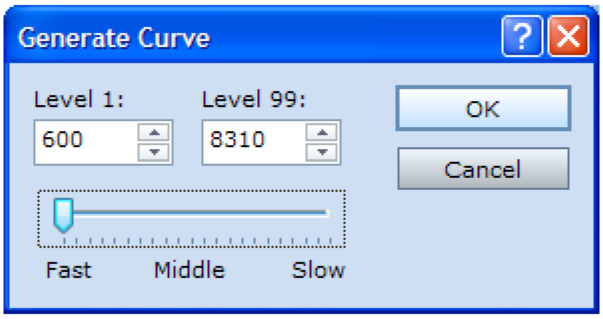
At the right, there are his `Parameter Curves`, which configure the **evolution of his stats** (agility, attack, etc) depending on the earned experience.

If you click of one of those curves, the following window will appear. With the Quick Setting buttons you can automatically generate different curves. If you click several times on one button, you will see variations of that curve.



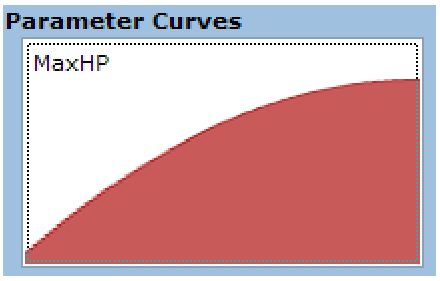
Level: 1 > Value: 521

Another quick method of generating curves is to click on the Generate Curve button. In the new window you can set the initial and end values of that stat and how fast/slow is the transition.



Level 1: 600
Level 99: 8310

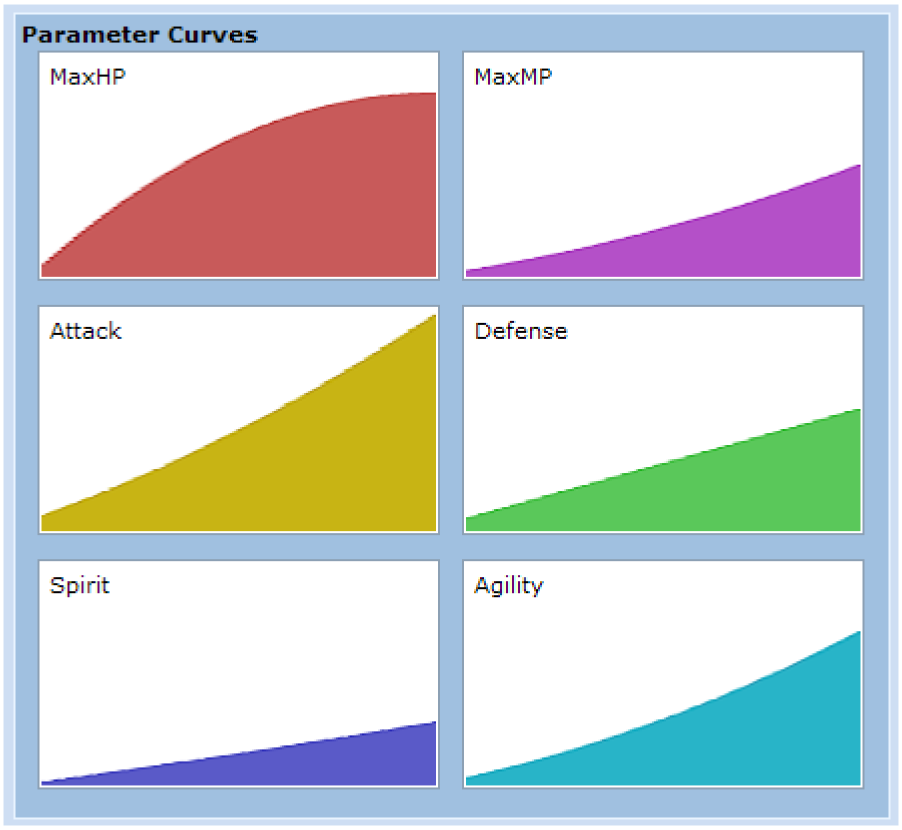
For example, the configuration displayed above generates this curve.



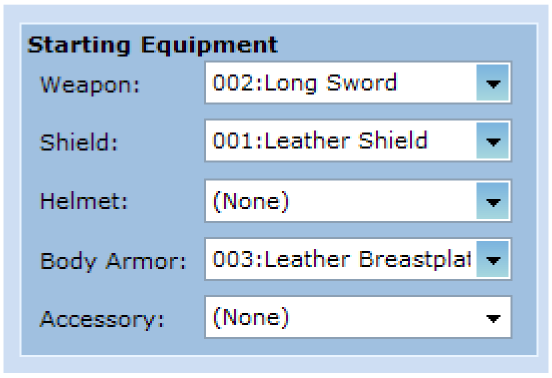
You can also hand-draw your own curve, although is much easier to generate it using the methods we've just described.

Since Salazar is a warrior, create some parameter curves which make sense to his job (i.e. lots of attack and life points, but less spirit, for example).

Here there are mine:

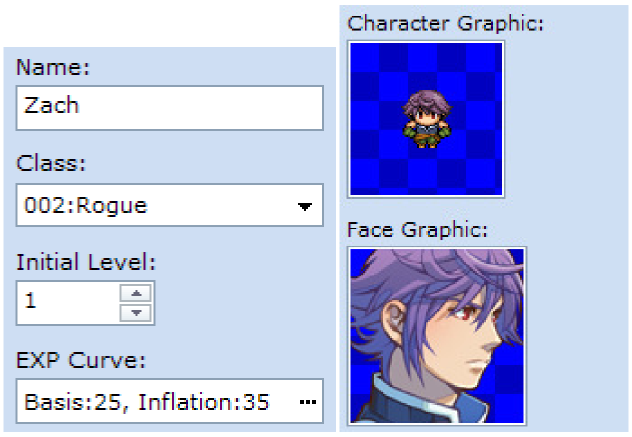


At the bottom of the screen you can configure the initial equipment of that character. Give *Salazar* the long sword and the leather shield and breastplate.


Starting Equipment

Weapon:	002:Long Sword
Shield:	001:Leather Shield
Helmet:	(None)
Body Armor:	003:Leather Breastplat
Accessory:	(None)

Now we'll create Zach, our ***Rogue*** companion. Set his name, class and graphics this way:

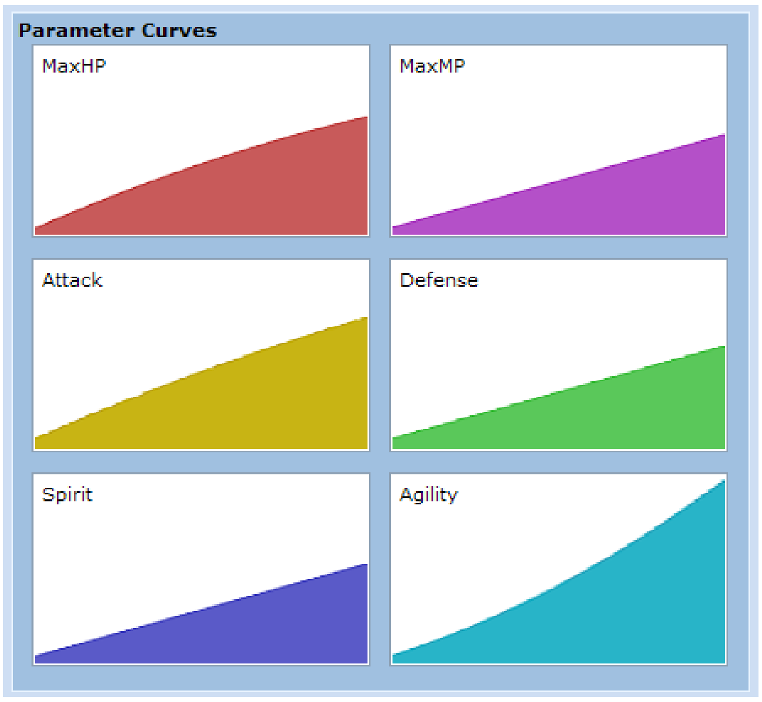

Name: Zach

Class: 002:Rogue

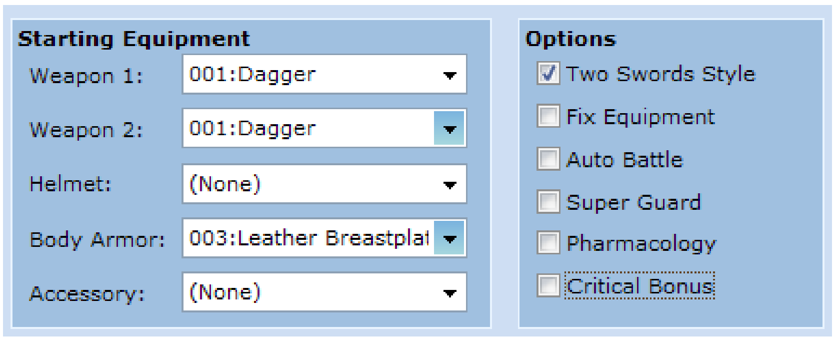
Initial Level: 1

EXP Curve: Basis:25, Inflation:35

Now, create some parameters according with his job. For example:



And about his equipment, check the `Two Swords Style` checkbox in the `Options` panel so he can use two weapons in combat instead of one weapon and a shield. Give him two daggers and the leather breastplate.



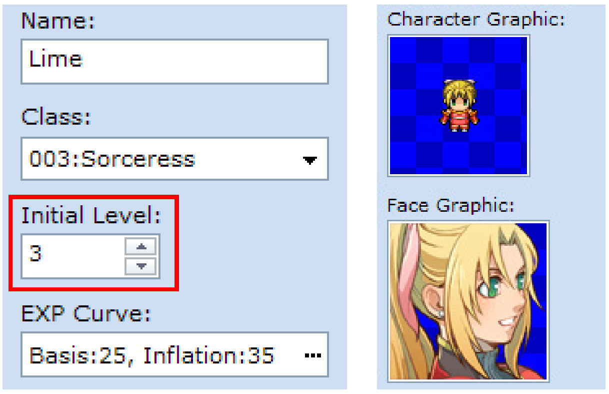
Starting Equipment

Weapon 1:	001:Dagger
Weapon 2:	001:Dagger
Helmet:	(None)
Body Armor:	003:Leather Breastplat
Accessory:	(None)

Options

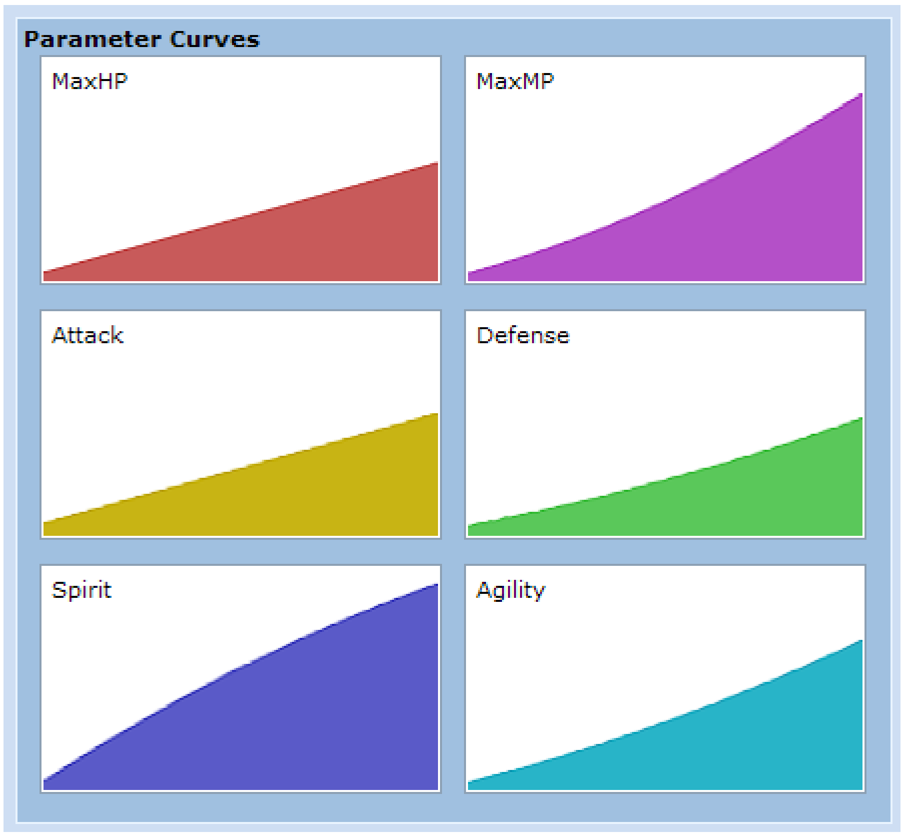
- ✓ Two Swords Style
- Fix Equipment
- Auto Battle
- Super Guard
- Pharmacology
- Critical Bonus

Lime is the only character left. As always, start by giving her a name and a job. Notice how her initial level is set to 3. This is because she will join the party later, not at the beginning of the game.

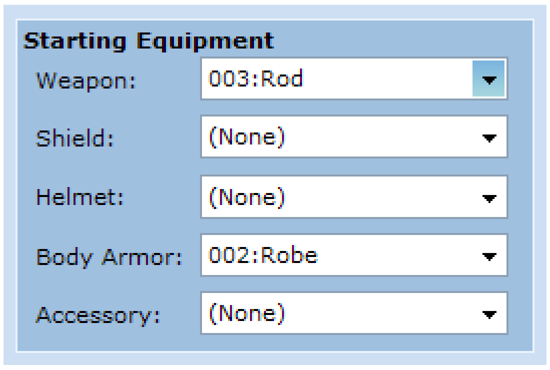


Name: Lime
Class: 003:Sorceress
Initial Level: 3
EXP Curve: Basis:25, Inflation:35

Set her parameters to something similar to this:



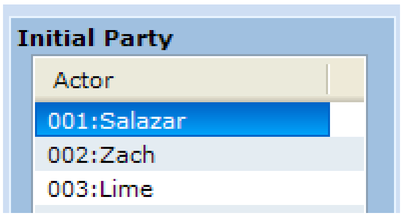
And don't forget to equip her!



Starting Equipment

Weapon:	003:Rod
Shield:	(None)
Helmet:	(None)
Body Armor:	002:Robe
Accessory:	(None)

Since Lime won't be in the party since the beginning, we have to remove her from the **initial party**. To do that, go to `System` tab and remove Lime from the `Initial Party` list.



Initial Party

Actor
001:Salazar
002:Zach
003:Lime

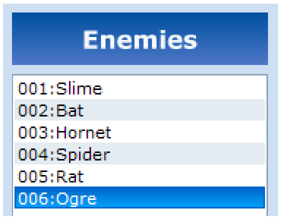
We are done with some of our game elements configuration. In this category would also fall enemies and troops, but I think they deserve a whole chapter in their own. So keep reading!

Enemies and Battles

Monsters!

In the `Database` manager you can create all the monsters and **enemies** that will populate your world. They are located inside `Enemies` tab.

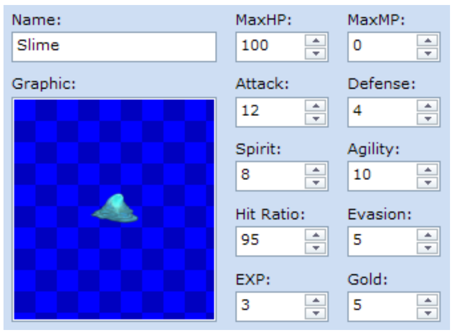
Balancing the game it's a hard task, so we will build up our work from some pre-set enemies. Delete all enemies in the list but *Slime*, *Bat*, *Hornet*, *Spider*, *Rat* and *Ogre*.



Enemies

001:Slime
002:Bat
003:Hornet
004:Spider
005:Rat
006:Ogre

We are going to customize these enemies. Let's start with the *Slime*. The part of the window displayed below configures the enemy's main parameters.

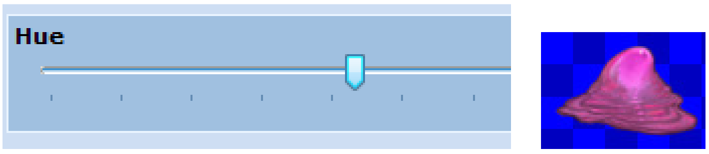


Name: Slime
Graphic:
MaxHP: 100
MaxMP: 0
Attack: 12
Defense: 4
Spirit: 8
Agility: 10
Hit Ratio: 95
Evasion: 5
EXP: 3
Gold: 5

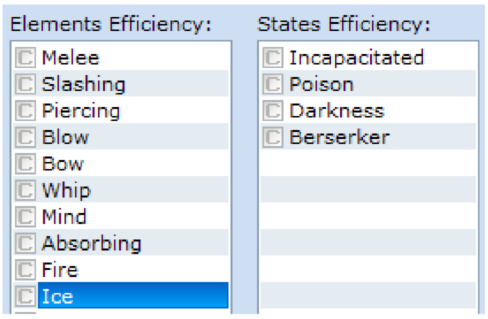
You can see the stats for the slime and also how much experience and gold will be rewarded to the player when he defeats this enemy. Notice that you can also configure which items the enemy might drop when defeated.

We are going to leave those stats just like they are, but we'll change the **image** for the *Slime* a little bit. Double click on the picture area.

With the `Hue` slider you can tint the image for that enemy. Choose a color you like.

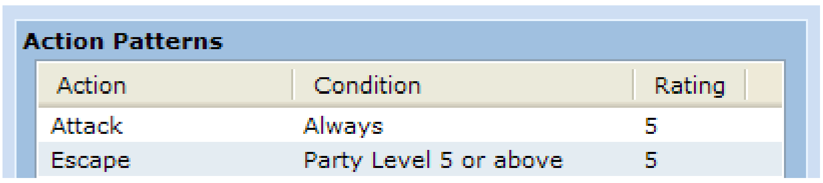


At the `Elements` and `States Efficiency` panels you can configure the weak and strong points for this enemy. The *Slime* is especially vulnerable to *Ice*, since it has an `A` for that element. In our game, the *Slime* won't have such vulnerability, so click on it until you choose `C`.



Elements Efficiency:	States Efficiency:
C Melee	C Incapacitated
C Slashing	C Poison
C Piercing	C Darkness
C Blow	C Berserker
C Bow	
C Whip	
C Mind	
C Absorbing	
C Fire	
C Ice	

In the `Action Patterns` panel you can set the enemy's behavior. Each action has a **rating** associated to it. The highest rating will be the standard action and every action with the same rating will have exactly the same odds to be executed. Ratings two points below the highest one will get its action executed too, but with much less probability.



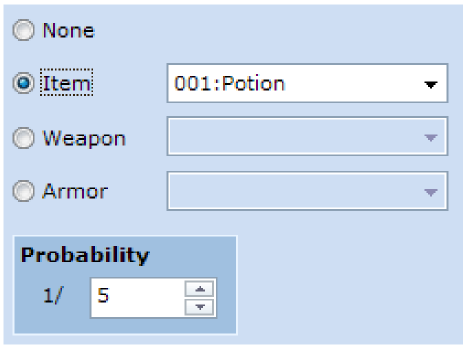
Action Patterns

Action	Condition	Rating
Attack	Always	5
Escape	Party Level 5 or above	5

The default *Slime* would always attack the party. But if the Party Level is 5 or above, the *Slime* will have a 50% chance of fleeing in its turn.

With that in mind, let's customize our other enemies. Select the ***Bat***. If you look at its Element Efficiency panel you'll see that the *Bat* is vulnerable to *Wind* and *Holy* elements. Since these elements are rated with a `B`, any spell or weapon causing *Wind* or *Holy* elemental damage will get a 50% bonus.

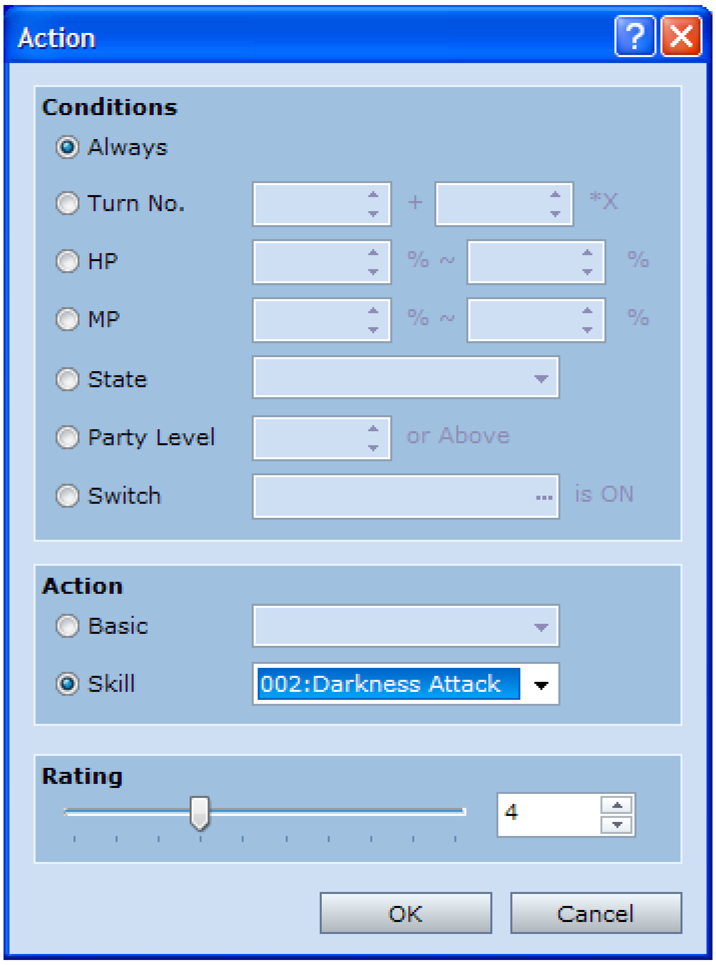
Now click on `Drop Item 1`. We are going to make it drop one Potion with 20% of probability. The configuration would be this:



None
[✓ Item] 001:Potion
Weapon
Armor

Probability
1/ 5

In the `Action Patterns` list there is an *Ice* action. It makes no sense, but it is there because we messed with the default skills. Change that action (double click on it) for a *Darkness Attack* and give it a 4 rating. That way it will have less chance to be chosen than the common attack.



Action

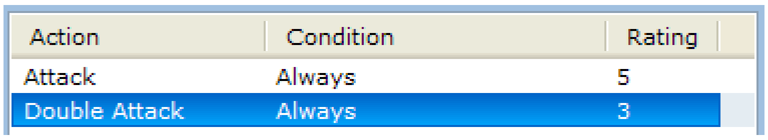
Conditions
[✓ Always]
Turn No. + *X
HP % ~ %
MP % ~ %
State
Party Level or Above
Switch is ON

Action
Basic
[✓ Skill] 002:Darkness Attack

Rating
4

OK Cancel

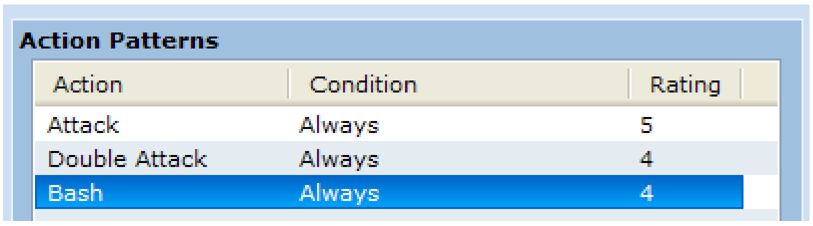
Leave the ***Hornet*** and the ***Spider*** as they are by default and go for the ***Rat***. Change its second (and empty) action for a Double Attack with a rating of 3.



Action	Condition	Rating
Attack	Always	5
Double Attack	Always	3

As for the dropped items, set them as you like. Later, when you test your game, you can adjust the loot to balance the game.

The ***Ogre*** is our final boss. It has pretty high stats, so we will have to adjust them to give a chance to the player for win. However, until we don't play through the game we won't know certainly which level will reach the players when they get to the *Ogre's lair*. That's why we'll leave his stats they way they are for now, and focus on the Action Patterns.


Action Patterns

Action	Condition	Rating
Attack	Always	5
Double Attack	Always	4
Bash	Always	4

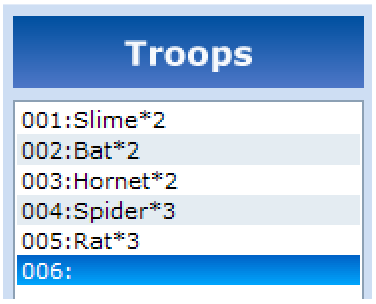
You will also see that the *Ogre*, by default, is vulnerable to *Fire* and can perform critical hits (like the party members). That sounds pretty reasonable for a final boss, but you can change it if you don't like it.

TROOPS

The `Enemies` tab is some kind of bestiary. In order to make combats with the Party Members, you have to organize your enemies in **troops**.

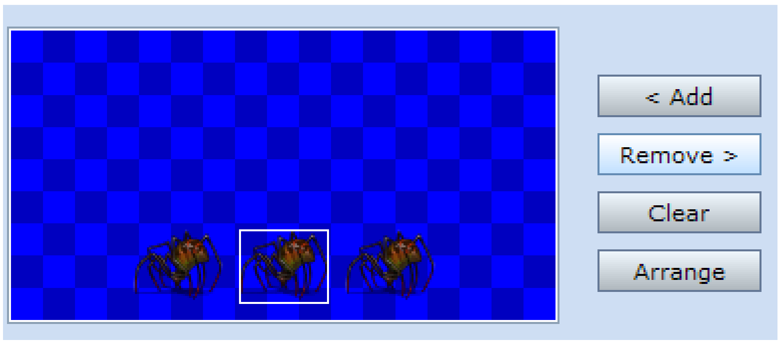
Troops are groups of enemies (though a single enemy can appear on his own). You can set up the troops in your game in the `Troops` tab.

Delete all troops but *Slime*2*, *Bat*2*, *Hornet*2*, *Spider*3* and *Rat*3*. Create also a new empty slot for the *Ogre*.

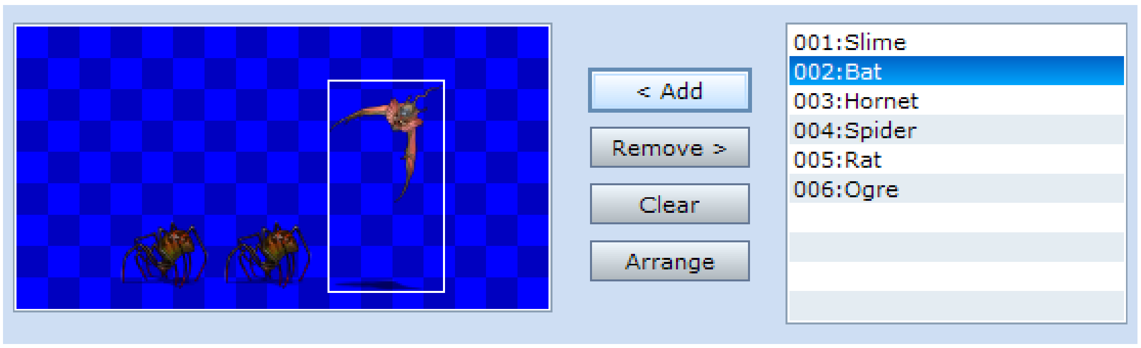

Troops

001:Slime*2
002:Bat*2
003:Hornet*2
004:Spider*3
005:Rat*3
006:

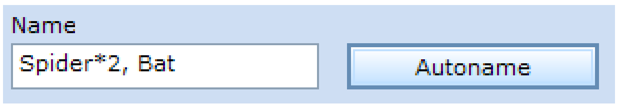
The *Spider*3* and *Rat*3* seem quite hard troops for the party, so let's change them. Select the *Spider*3* troop. Click on a spider and then on the `Remove` button to delete that monster from the troop.



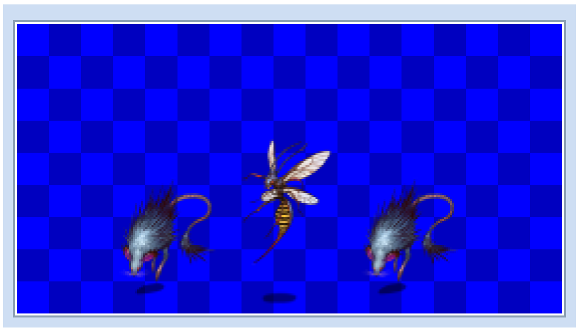
Now select a *Bat* from the list and click on the `Add` button.



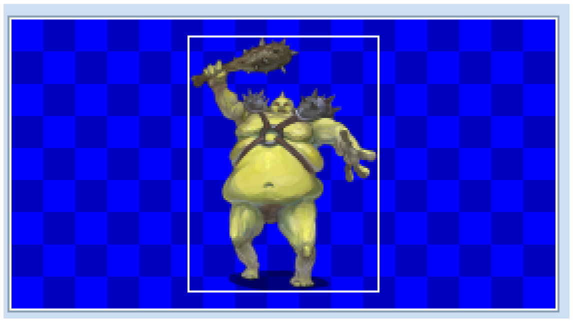
You can drag the *Bat* picture to put in wherever you want in the battle screen. Click on the `Autoname` button to rename this troop.



Repeat the steps with the *Rat*3* troop to change it for two *Rats* and one *Hornet*.

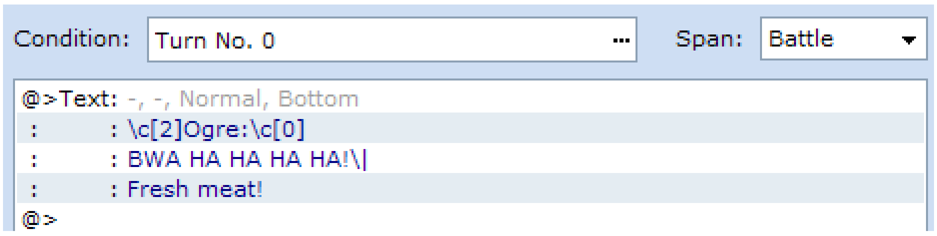


The *Ogre* troop is quite simple. Just select the empty troop slot and add the ogre at it. Don't forget to autoname this troop too!



At the bottom of this screen you will see some kind of **Event editor**. That's right; you can have events in battle! This way, you can add dialogs or make a magician summon monsters, etc.

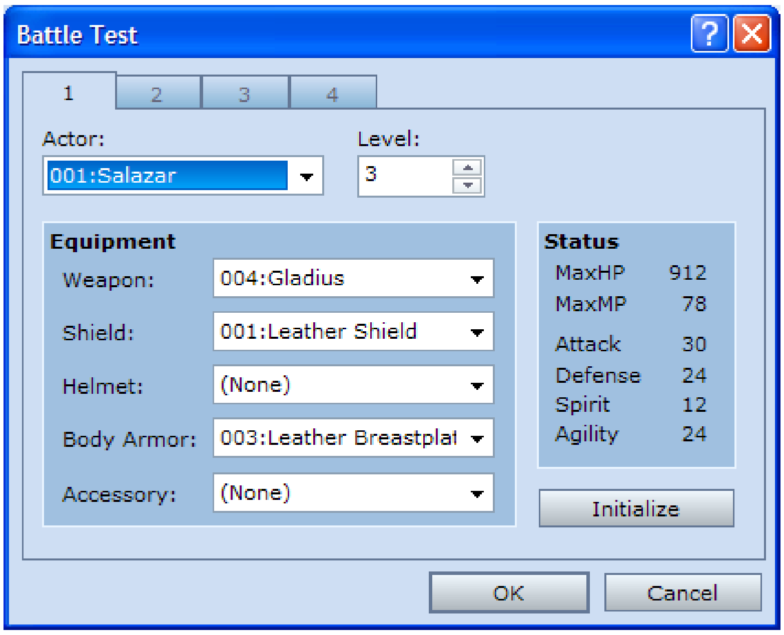
For example, let's make the *Ogre* say some lines when the battle has just begun. Double click on the list of event commands and insert a `Show Text` command. As `Condition`, set it to `Turn number 0` (that's the beginning of the battle) and choose `Battle` as `Span`. The `Span` controls how often the event checks for the condition. With `Battle`, the event will only be triggered once per battle.



Condition: Turn No. 0 Span: Battle

```
@>Text: -, -, Normal, Bottom
 :        : \c[2]Ogre:\c[0]
 :        : BWA HA HA HA HA!\|
 :        : Fresh meat!
@>
```

To test if it works, click on the `Battle Test` button at the top of the screen. You will get to configure the party members and equipment. You can leave the defaults and click on `OK`.



Battle Test

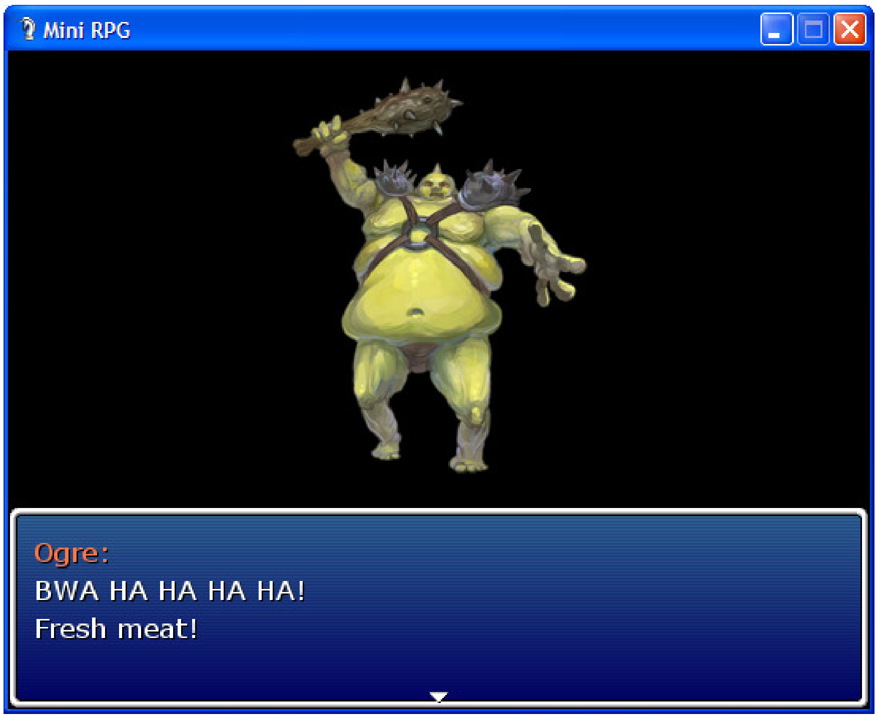
Actor: 001:Salazar
Level: 3

Equipment
Weapon: 004:Gladius
Shield: 001:Leather Shield
Helmet: (None)
Body Armor: 003:Leather Breastplat
Accessory: (None)

Status

MaxHP	912
MaxMP	78
Attack	30
Defense	24
Spirit	12
Agility	24

A battle simulation will begin! If you have set up properly the event, you will see something like this when the battle starts.

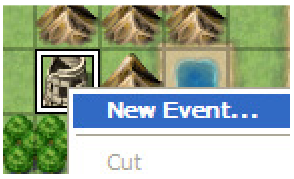


Ogre:
BWA HA HA HA HA!
Fresh meat!

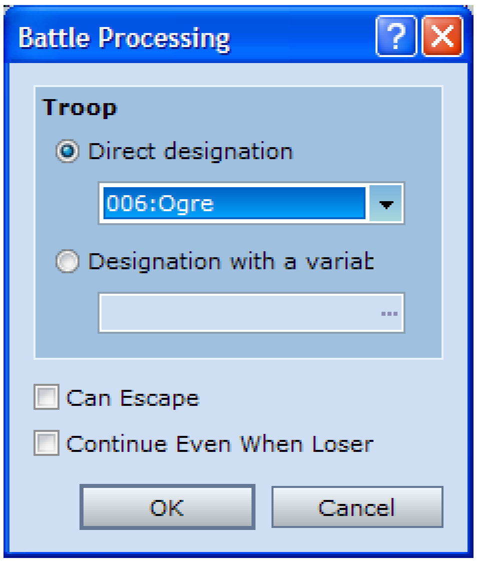
ENCOUNTERS

Once that you have enemies and troops you can create **encounters** which will lead to battles. These encounters can be either random or triggered by an event.

For example, let's create an encounter with the *Ogre* when the player goes over the tower in ruins (the *Ogre's lair*). In the world map, create a new event at the lair.



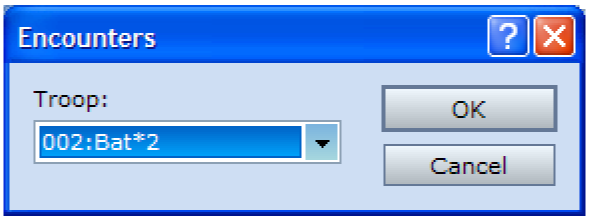
Insert a new command called `Battle Processing` (in tab 3) and configure it to fight the *Ogre*. If you want to prevent the party from fleeing, uncheck the `Can Escape` checkbox. Change the event's `Trigger` to `Player Touch` so the player won't have to press the `Action` button to fight the *Ogre*.



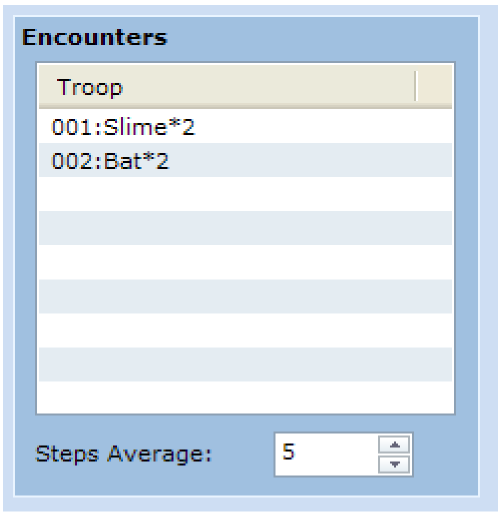
Playtest the game to see if it works as it should.

Random encounters are very popular in RPG's. Let's create some random battles with weak enemies at the world map. Right click on the world map at map's list and select `Map Properties`.

If you double click in the `Encounters` panel you will be able to choose a troop to battle to. Create encounters with *Bats* and *Slimes*.



Also, change the Average Steps to trigger one encounter, since our world map is very tiny.



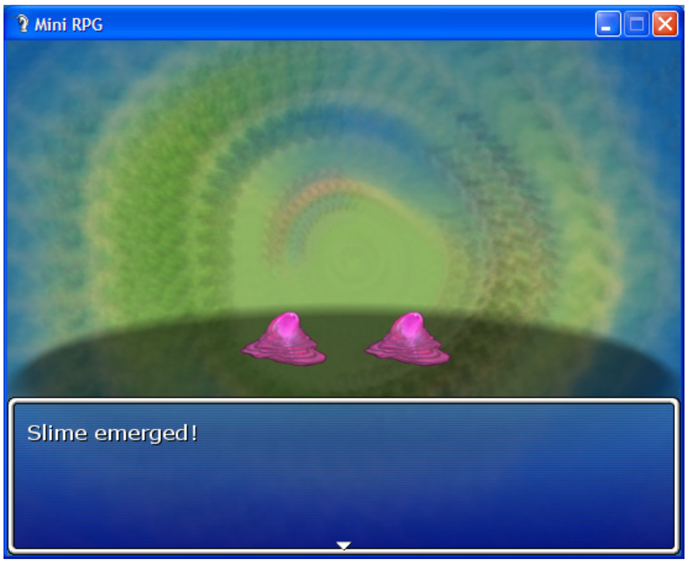
Encounters

Troop
001:Slime*2
002:Bat*2

Steps Average: 5

If you ever want a troop to appear more frequently than the others, you will have to repeat the encounter twice or several times.

Go and play the game now to make sure it works.



Slime emerged!

STORY

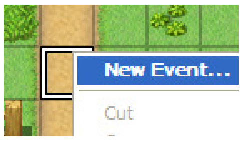
In this chapter we will learn some techniques to tell a story to the player and making the game "advance" through quests.

CUT-SCENES

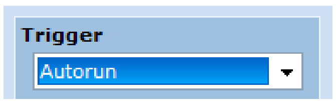
Cut-scenes are the key to tell a story. In them, the player acts like a TV spectator and watches some scene with action and/or dialogs. In RPG Maker VX, cut-scenes are created with events.

For example, let's assume that our game begins with the party entering the village of Evergreen. We could arrange things so when the game starts the two characters on the party have a conversation. This cut-scene should be only played once in the whole game, so we will introduce an important concept in RPG Maker VX: switches.

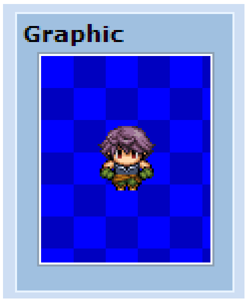
Set the starting position for the player somewhere in the village's entrance. Now create a new event two tiles north from it.



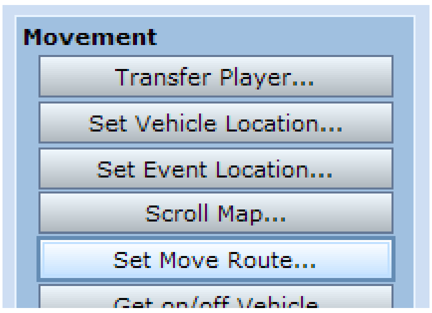
First things first: put the `Trigger` on `Autorun`. This way, the event will be triggered automatically and the player won't be allowed to do anything until it ends.


Trigger
Autorun

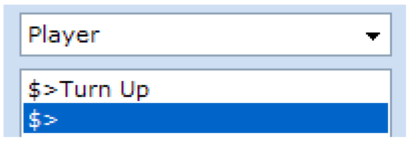
Be aware that if you don't end the event somehow, the player will be stuck in an infinite loop! Now, change the event's graphic for the picture of our *Rogue*.



In order to establish a proper conversation, the characters should be looking at each other. We'll have to change then the direction the main character is facing. Create a new event command `Set Move Route`.

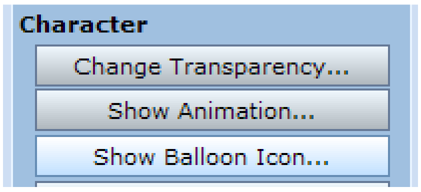


You will see a lot of options now. Note that you can set the move route not only for this event, but for the player's character and every other event. For now, make the player look at North by selecting `Turn Up`.

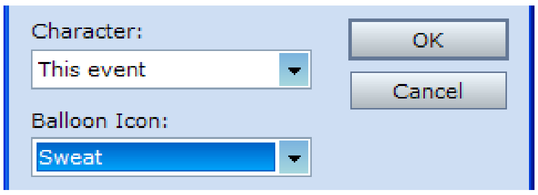


```
Player
$>Turn Up
$>
```

It's time to add some dialog with Show Text. If you want to make your dialog more expressive, you can insert some balloon icons in it. To do that, choose the command Show Balloon Icon.



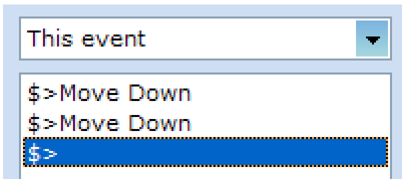
You get to choose then the character will display the balloon and what kind of expression will be inside it.



Character: This event

Balloon Icon: Sweat

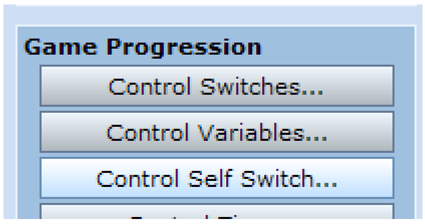
Once your dialog is finished, you want the secondary character to disappear. Instead of making him vanish, let's make him "join" our main character by walking towards him (like in Final Fantasy VII). Since the Rogue is placed two tiles North from our main character, just insert a `Set Move Route` command and make the event move down twice.



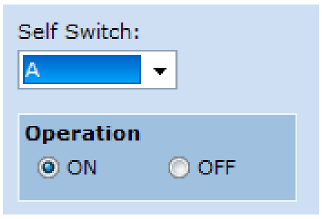
```
$>Move Down
$>Move Down
$>
```

Now the cut-scene is finished. How do we make *now* the Rogue to disappear? How do we cancel the playing of this cut-scene every time the player enters the village? This is what **switches** are for. A switch holds the ON or OFF status. There are switches local to events (**self switches**), but there are also switches which can be accessed from anywhere (**control switches**). Most of the times we will be using self switches and this is one of those occasions.

The trick is that some events can condition its execution to the status of some switch. So, if we create a new page for the event and make this page execute when a switch is ON, we will stop the cut-scene. But first, let's insert a command that turns ON a self-switch at the end of the cut-scene. This command is called `Control Self Switch`.

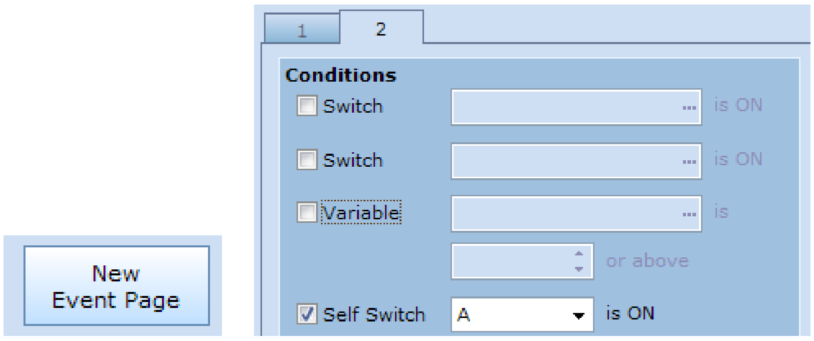


There are a few self switches for each event, called by a letter. In this command, turn ON the switch A.


Self Switch: A

Operation: [✓ ON] OFF

Now create a new page clicking on the `New Event Page` button and condition the execution of this page to the ON status of self switch A.



Conditions

Switch is ON

Switch is ON

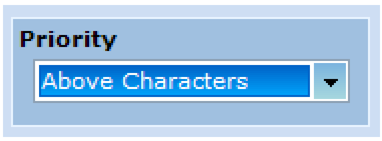
Variable is

or above

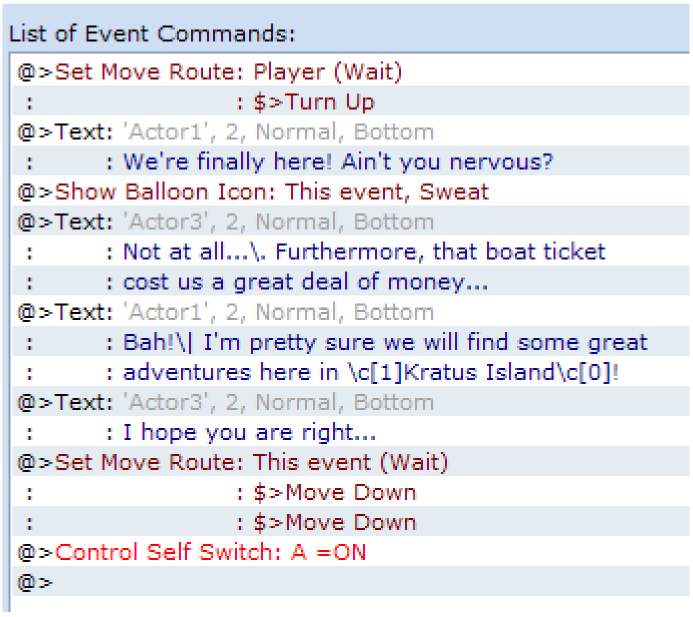
[✓ Self Switch] A is ON

Don't put any image for this event's page and that will make the character disappear. And that's all!

Well, if you actually execute the game now you will detect a nasty bug. The cut-scene never ends because the Rogue can't "join" the main character by walking over him. To do so, set his priority (remember, this priority belongs to the event page number 1!) to either `Above Characters` or `Below Characters`.

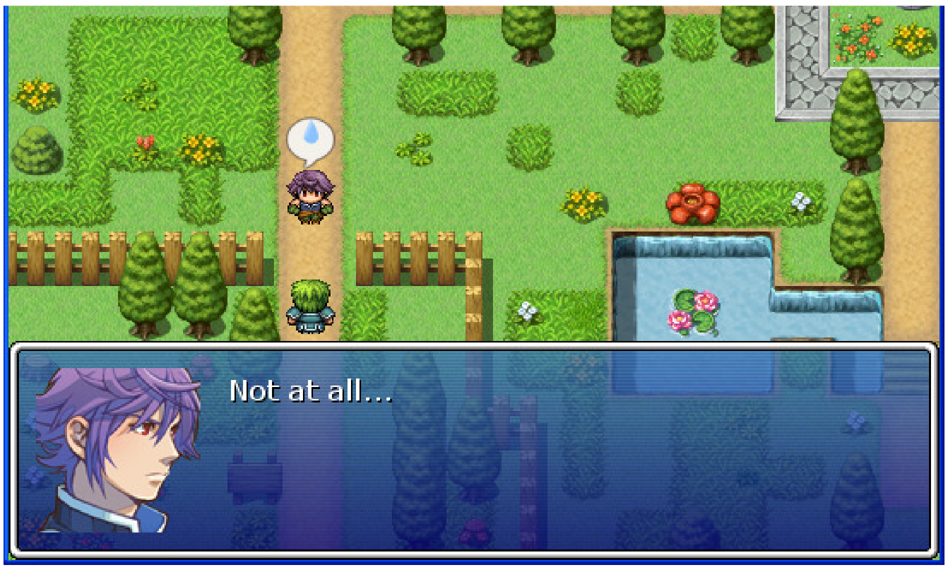

Priority
Above Characters

With that fix it should work now. By the way, this is how my cut-scene turned out:


List of Event Commands:

```
@>Set Move Route: Player (Wait)
 :                : $>Turn Up
@>Text: 'Actor1', 2, Normal, Bottom
 :      : We're finally here! Ain't you nervous?
@>Show Balloon Icon: This event, Sweat
@>Text: 'Actor3', 2, Normal, Bottom
 :      : Not at all...\. Furthermore, that boat ticket
 :      : cost us a great deal of money...
@>Text: 'Actor1', 2, Normal, Bottom
 :      : Bah!\| I'm pretty sure we will find some great
 :      : adventures here in \c[1]Kratus Island\c[0]!
@>Text: 'Actor3', 2, Normal, Bottom
 :      : I hope you are right...
@>Set Move Route: This event (Wait)
 :                : $>Move Down
 :                : $>Move Down
@>Control Self Switch: A =ON
@>
```

And… ta-dah!

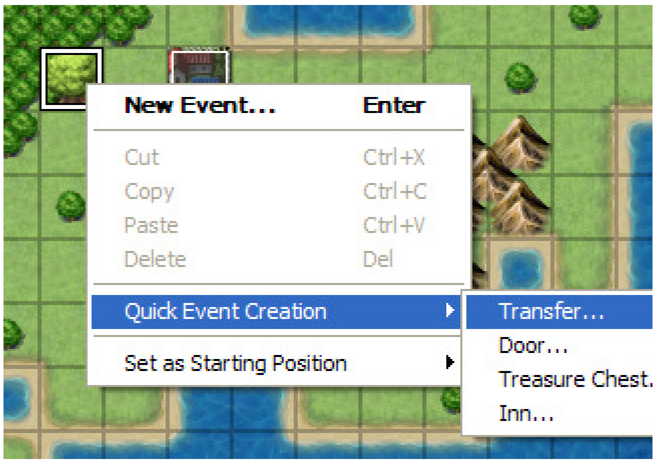


FORBIDDEN PLACES

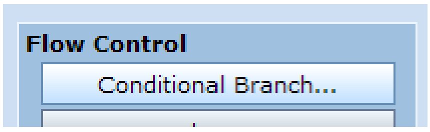
Often, we will have to **restrict the access** to certain areas of our game until the player has completed a quest, or possess some item or has talked to some NPC. There are several ways to achieve this; one of them is using **control switches**.

For example, let's prevent the player to enter the *Forbidden Forest* until he has talked to *Lime* in the village. To do that, create a new map and name it `Forbidden Forest`. You don't have to draw this map yet, so fill it with grass for now.

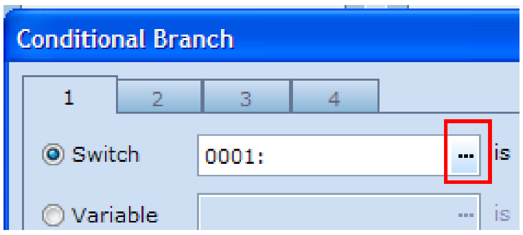
Now, create a new transfer event to that new map from the world map.

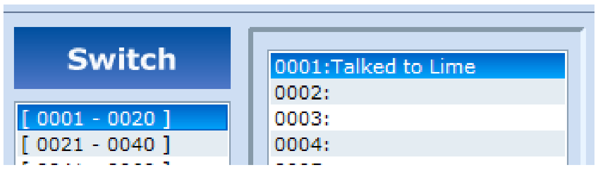


Once created, right click on it and select `Edit Event`. Now you have to condition the execution of the transfer to the value of a control switch. Either create a new conditional branch or a new event page. This time, we will use the conditional branch.

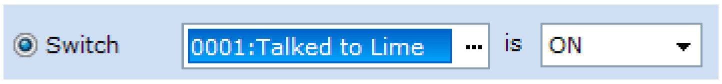


Click on the ellipsis to manage switches. Right now you have empty slots. It's better to give them proper names because in a non-trivial project like this, you will be using a LOT.

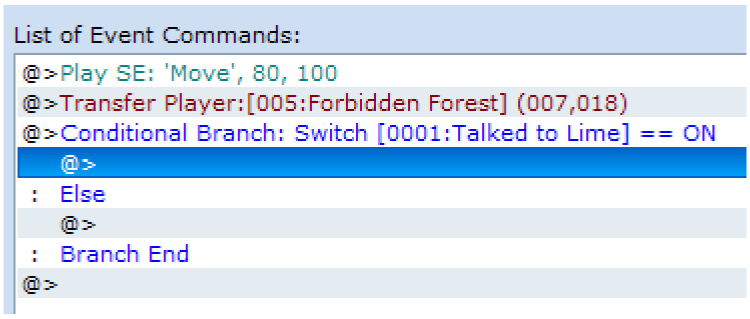




Now you can fill the condition for the branching. Set it to check if the Talked to Lime new switch is ON.

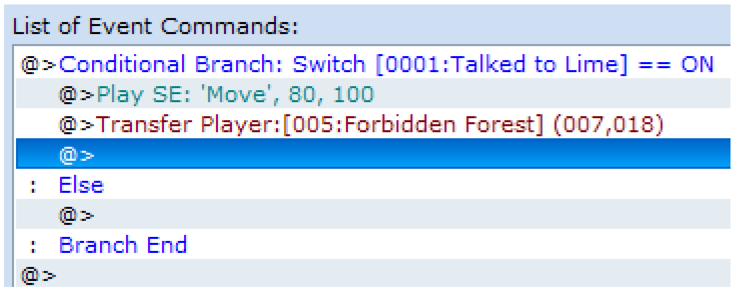


Your branching will look like this:



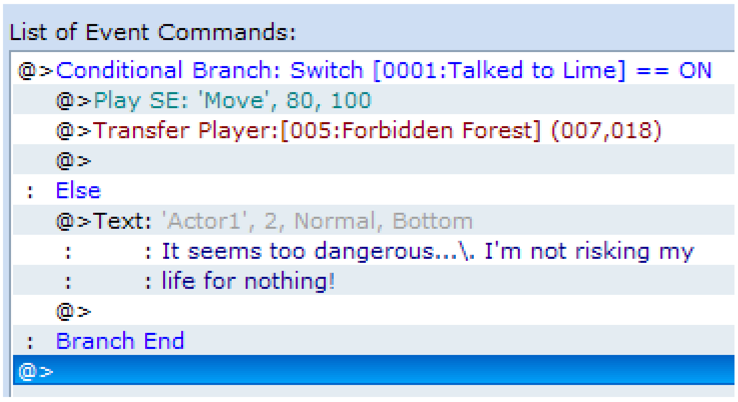
```
List of Event Commands:
@>Play SE: 'Move', 80, 100
@>Transfer Player:[005:Forbidden Forest] (007,018)
@>Conditional Branch: Switch [0001:Talked to Lime] == ON
  @>
 :  Else
  @>
 :  Branch End
@>
```

Now cut the transfer event and paste it the true branch (the one before the `Else`). Now it should look like this:



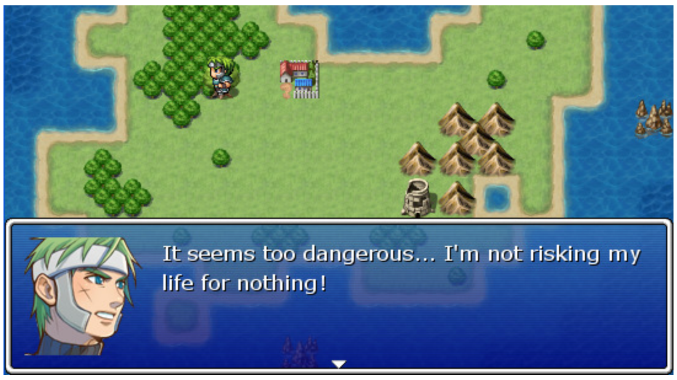
```
List of Event Commands:
@>Conditional Branch: Switch [0001:Talked to Lime] == ON
  @>Play SE: 'Move', 80, 100
  @>Transfer Player:[005:Forbidden Forest] (007,018)
  @>
 :  Else
  @>
 :  Branch End
@>
```

To tell the player what's going on, insert some dialogue lines in the `Else` branch, like the main character saying "It seems too dangerous and I have no reason to go in!"

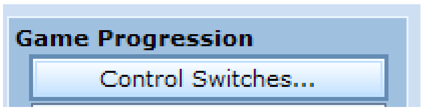


```
List of Event Commands:
@>Conditional Branch: Switch [0001:Talked to Lime] == ON
  @>Play SE: 'Move', 80, 100
  @>Transfer Player:[005:Forbidden Forest] (007,018)
  @>
 :  Else
  @>Text: 'Actor1', 2, Normal, Bottom
   :        : It seems too dangerous...\. I'm not risking my
   :        : life for nothing!
  @>
 :  Branch End
@>
```

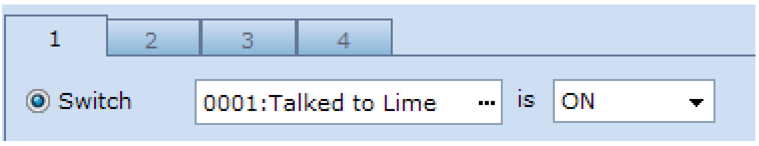
Since all switches are OFF by default, if you play the game and try to enter the forest, you will get the main character complaining. Go and check it.



Now we have to turn that switch ON once the player has talked to *Lime*. Go to *Lime's* event in the village (remember that NPC following a fixed path?) and edit it. Add some dialogue lines first and insert a Control Switch ON command at the end of it.

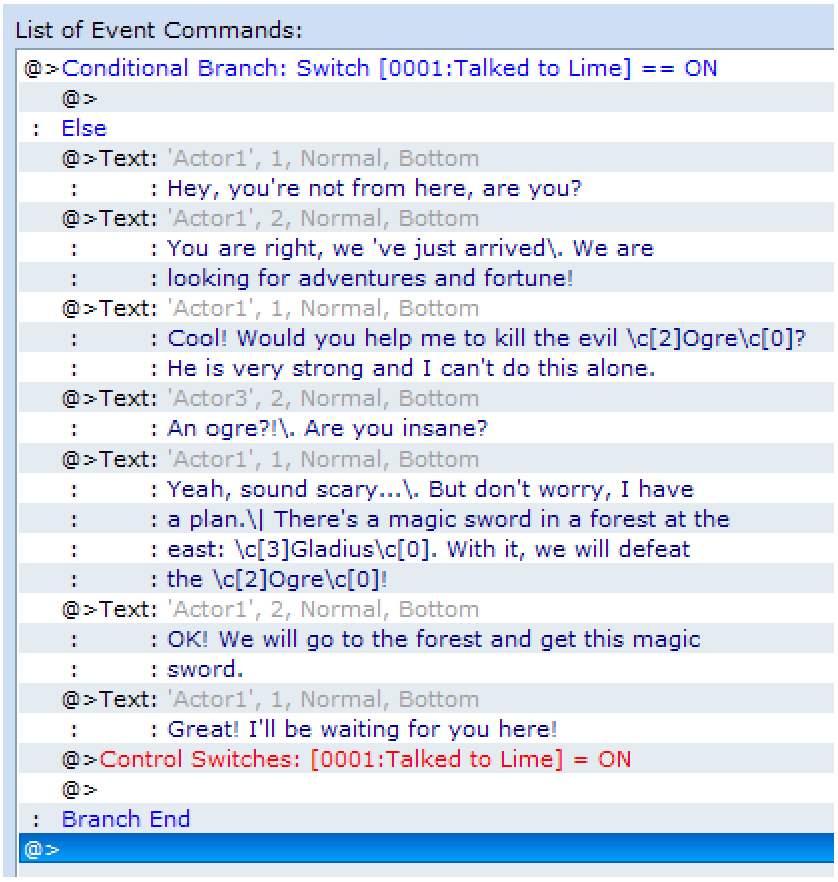


Now, we are going to avoid repeating the same dialog again and again. Create a new branch like this one:



Switch 0001:Talked to Lime is ON

And paste the entire dialog in the `Else` branch. Like this:



```
List of Event Commands:
@>Conditional Branch: Switch [0001:Talked to Lime] == ON
  @>
 :  Else
  @>Text: 'Actor1', 1, Normal, Bottom
   :      : Hey, you're not from here, are you?
  @>Text: 'Actor1', 2, Normal, Bottom
   :      : You are right, we 've just arrived\. We are
   :      : looking for adventures and fortune!
  @>Text: 'Actor1', 1, Normal, Bottom
   :      : Cool! Would you help me to kill the evil \c[2]Ogre\c[0]?
   :      : He is very strong and I can't do this alone.
  @>Text: 'Actor3', 2, Normal, Bottom
   :      : An ogre?!\. Are you insane?
  @>Text: 'Actor1', 1, Normal, Bottom
   :      : Yeah, sound scary...\. But don't worry, I have
   :      : a plan.\| There's a magic sword in a forest at the
   :      : east: \c[3]Gladius\c[0]. With it, we will defeat
   :      : the \c[2]Ogre\c[0]!
  @>Text: 'Actor1', 2, Normal, Bottom
   :      : OK! We will go to the forest and get this magic
   :      : sword.
  @>Text: 'Actor1', 1, Normal, Bottom
   :      : Great! I'll be waiting for you here!
  @>Control Switches: [0001:Talked to Lime] = ON
  @>
 :  Branch End
@>
```

In the true branch, insert some dummy dialog for now. We will work on it later. Go and try to enter the forest once you have talked to Lime. It should work properly!