THE EXPERT'S VOICE® IN GAME DEVELOPMENT

Make a 2D RPG in a Weekend

With RPG Maker VX Ace

CREATE YOUR FIRST RPG IN A WEEKEND

Darrin Perez

Apress®

For your convenience Apress has placed some of the front matter material after the index. Please use the Bookmarks and Contents at a Glance links to access them.

friendsof ED™

Apress®

Contents at a Glance

Introduction

Overview

Welcome to *Make a 2D RPG in a Weekend*! This book will take you through the process of using RPG Maker VX Ace (RMVXA) to make a six-level dungeon crawler within a single weekend. It will explain how to do what is necessary to create the game, so don't worry if you have had no prior RMVXA experience. With that said, if you have read my previous book, *Beginning RPG Maker VX Ace*, you'll have a leg up to work on this one. Here's a short list of things that were covered in my previous book relevant to this one.

- Events: How they work and how to create/edit them
- RMVXA's Map Editor: How to use it to add and edit maps
- RMVXA's Database: How to use it to add, delete, and edit entries.

Those are the bare essentials of using RMVXA. If you're anything like me, you'll probably learn how to do some or all of the above through mere experimentation. Anyway, let's move on to the next topic of interest in this introduction.

The Engine of Choice

You will have to use the full version of RMVXA to create this game. RMVXA (and a slew of related products, while we're on the subject) can be purchased from the official site at `www.rpgmakerweb.com`.

The exact link to download RMVXA is `www.rpgmakerweb.com/products/programs/rpg-maker-vx-ace`.

When not on sale, RMVXA usually costs $69.99, but you can try it for free for an entire month by grabbing the trial version. You will be asked for your name and e-mail address, if you attempt to download the full version of RMVXA (whether you pay for it at that time, wait until later, or decide not to keep using the product). If you use Steam, RMVXA is available through, and frequently on sale on, that platform as well.

Source Code Considerations

As you may have already noticed, this book's source code includes the full game and a few other things as well. Among the extras included are a set of blank maps for each of the six distinct dungeon levels covered in the book. It is my opinion that mapping is one of the more time-consuming aspects of RMVXA game design, so I included those maps as a shortcut if you want a head-start but don't want to be completely spoiled. You can see the game in its completed state and take as little or as much from it as you want. I merely ask that you take the time to go through the book and try to do as much of it as you can on your own. Making a game is like most other things in life: practice makes perfect.

Why Make a Dungeon Crawler?

A dungeon crawler is simple. The simplest dungeon crawlers break the standard role-playing game rule (if you will) of requiring a story. You can read some more considerations on the subject in the first chapter of this book.

Chapter 10

The book's final chapter exists outside the challenge posed by this book's title. Its proposes some interesting features that you can add to your game after you have officially completed it. Depending on your personality, time limits may be stressful, fun, or something in between. Thus, I didn't want you, the reader, to have to race to add to the game the extras discussed in Chapter 10.

CHAPTER 1

Laying Out the Framework

During the course of this chapter, we will be laying out the framework for our weekend game. This includes the story line, a general overview of each of our playable characters, and the basic foundation of our game's town. We will be creating the town area, but you can also find each of this game's maps in the book's source code.

Story

The first objective at hand in the creation of our turn-based, top-down dungeon crawler is to lay out our story. While dungeon crawlers are not usually pinnacles of complex storytelling, some still have at least a solid framework that carries their players through the game. Here are some examples of story lines (or absence thereof) in dungeon crawler role-playing video games, in no particular order of complexity.

- In *Rogue*, the classic dungeon crawler that inspired an entire genre (roguelikes), the player's only objective is to survive as long as possible. The dungeon is infinitely deep, and there isn't really a lore-related reason for the player's actions.
- In *Diablo* (technically an action role-playing game [RPG]), the player is an adventurer who arrives at the town of Tristram in an attempt to defeat the eponymous villain. Lore books within the dungeon lay out the backstory for the game's setting and context for certain areas.
- The remake of the very first *Etrian Odyssey* had a story mode that improved upon the mostly bare-bones plot of the original version. Even so, the basic framework of the plot is as follows: the player is a Highlander who arrives at Etria in search of fame, glory, and honor and eventually uncovers something that could jeopardize the entire town and even the rest of the world…
- *Eye of the Beholder* is a classic PC role-playing game that has a party of four adventurers trying to find and defeat a great evil under the sewers of the city of Waterdeep.

You may have noticed a common thread among most of those descriptions. Mainly, *the plot in a dungeon crawler is merely an excuse for the player to get into the dungeon and play!* Even so, that shouldn't stop you from adding plot to your dungeon crawler, as desired. Because most of the dungeon crawlers that I have played involve defeating a final boss, let's have one in which the player has to find an artifact hidden within the deepest levels of the dungeon. Here's a succinct blurb for our game's story line.

You are an adventurer from a faraway land who has arrived at the town of Eagle's Crossing. Legends speak of an ancient artifact buried deep within the town's catacombs. The artifact is said to have the power to heal any wound. Some even claim that it can raise the dead. It is owing to the last claim that you have made the arduous journey to the town, for you have lost **someone dear to you** recently.

I bolded the former words for the purpose of mechanics: I plan to have multiple playable characters. Thus, in a game, if you have different characters with different motivations, it only makes sense that the certain someone be different for each character.

■ **Note** No one ever said you had to reveal *who* the character has lost. Maybe he/she lost multiple beings. Sometimes, the best plots are those that reveal only what is strictly necessary.

We'll honor the age-old tradition of classic dungeon crawlers and have three different characters representing classic fantasy archetypes.

- **Palnor** the *Warrior*: Palnor hails from the highlands directly south of the realm's northern mountains. His town was razed by one of the many mountain tribes. He seeks the ancient artifact in order to revive his kin and the rest of the people of his town. As a Warrior, Palnor has the widest weapon and armor selection of any of the three classes of characters, including the unique ability to wield two-handed swords and axes and heavy armor. However, he cannot master any spells.

- **Gust** the *Bandit*: Gust used to be as unsavory a character as one could imagine, and one most would rather not associate with, until the day he laid eyes on the princess of one of the desert clans. He swore thereafter to change his ways. However, his past life has caught up with him once again. He quests for the ancient artifact to revive the one person he has truly loved. As a Bandit, Gust is the only character who can use bows. His melee weapon and armor selection suffer a bit in comparison to those of the Warrior, but he makes up for it with the ability to cast low-level spells.

- **Feylia** the *Mage*: Graduated at the top of her class in the mystical arts, Feylia seeks the ancient artifact to revive her younger brother, who was lost in a terrible plague that struck the western lands when she was barely of age. As a Mage, Feylia has the worst weapon selection of any of the three characters. However, her spell book can carry any and all spells that she manages to find within the dungeon. Additionally, she is the only character who can wear magic robes. A well-equipped Mage is arguably the most fit for the dungeon's deeper levels.

With that out of the way, we pretty much have the skeleton of our plot. About the only thing left to do is provide more context to the town of Eagle's Crossing, as follows.

The town of Eagle's Crossing was created 411 years ago by a group of trading partners who wanted to live away from the various monarchies that would control them. Most of the olden kingdoms have since changed their methods of rule, but Eagle's Crossing remains the destination of choice for anyone who wishes to elude the less secluded parts of the world. Nestled against a mountainside that was carved out to form the town's catacombs, adventurers are now making their way to Eagle's Crossing in hopes of finding the artifact supposedly hidden within.

There you have it! The impetus for our players to start the game, summarized within a single page.

Basic Game Play Considerations

Dungeon crawler RPGs, more than any other type of RPG, live or die on their game play. After all, and as we have already demonstrated, you don't need much of a plot for a dungeon crawler. It is imperative, from the very start of your project, to think about what type of game play your dungeon crawler game will have. Here are some good questions to ask yourself.

- **Will I use RMVA's default battle system?** There's nothing wrong with using it. Then again, if you want to make a classic dungeon crawler, you might want enemies to move on the screen as you're fighting. In that case, you may have to tweak the system a little. Mind you, this does not necessarily have to involve heavy scripting.

- **How will the loot system work?** Many dungeon crawlers eschew the usual static treasure chests in favor of a more randomized system. This can result in some crazy things, such as getting the third-best sword in the game on the second floor. Such craziness has been largely circumvented over the years by the presence of loot tables that determine what can and cannot drop in a certain area. Another important consideration in the same train of thought is whether you will have the player find most of his/her items in the dungeon or have a town's shop stock some of them. In the latter case, gold (or whatever you would call your currency) gains value.

- **How will the leveling system work?** Most dungeon crawlers stay true to their RPG core. Thus, enemies offer experience, and the player's character (or characters) levels up once he/she has earned sufficient experience. On that note…
- **Will my dungeon crawler be a party-based or individual character game?** This is a really important question with no single correct answer. Citing as examples the games mentioned at the beginning of this chapter, *Rogue* and *Diablo* are controlled by a single character. *Etrian Odyssey* and *Eye of the Beholder*, on the other hand, are party affairs. As RPG Maker VX Ace (RMVXA) is designed to handle a battle party of up to four members, the easiest type of dungeon crawler to create with the engine is a party-based one.
- **Will my dungeon crawler have permadeath?** This is another important question, to which you should probably answer "no," unless you have a very specific reason to implement it in your game. *Permadeath* is short for "permanent death" and basically means that if characters die, they stay dead, with no way to be revived or otherwise allow you to use them in your party any longer. Most roguelikes include permadeath, which adds to their notorious levels of difficulty. They also have a single save game slot that deletes itself when the player dies, to prevent him/her from cheating death, as it were. Our game, on the other hand, will not have any of these mechanics.
- **Will the player be able to return to the town after leaving?** This correlates with earlier points of interest. If you want the player to live solely off of his/her own resources, you can bar access to the town after he/she leaves the first time. On the other hand, you wouldn't want to do this if you want the character to be able to use the town to sell his/her hard-earned loot and buy niceties.

Exactly how I implement game play will be explained throughout the book.

■ **Note** It's as good a time as any to point out that our game will not have a world map. The player starts in a town and goes into a linear dungeon that descends ever deeper into the bowels of the town.

The Town of Eagle's Crossing

As the sole representation of civilization present in our game, Eagle's Crossing will be our player's destination for rest, when he/she is not diving into the dungeon. The player will have access only to a specific part of Eagle's Crossing called the Adventurer's Quarter. It houses four buildings that provide for most of an aspiring dungeon crawler's needs. See Figure 1-1.

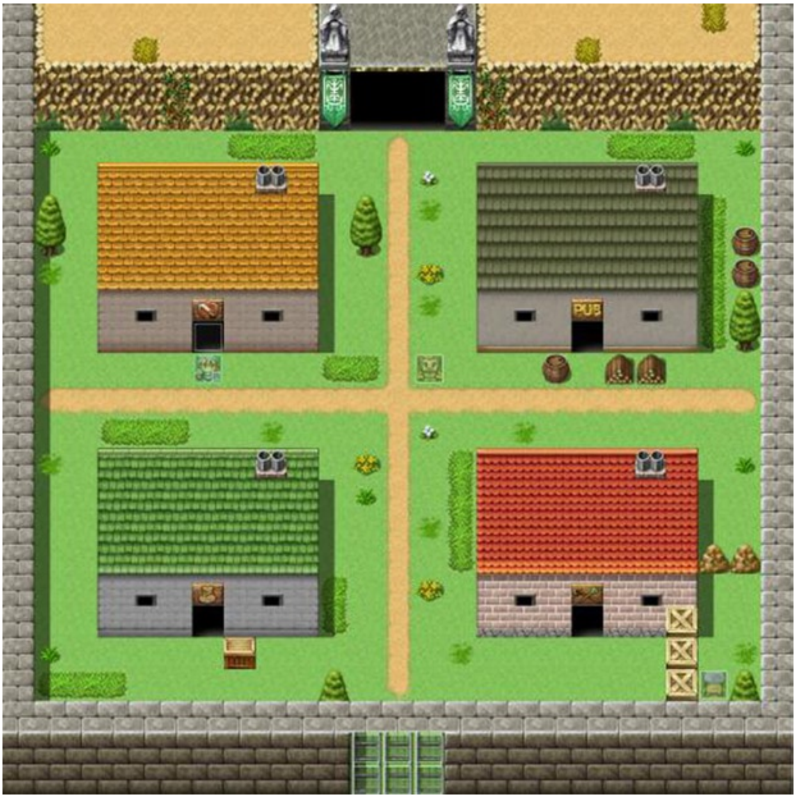


Figure 1-1. *A map of the Adventurer's Quarter in Eagle's Crossing, our game's safe zone*

The four buildings of Eagle's Crossing are (starting from the top left and going in a clockwise motion) are as follows:

- **The Equipment Shop**: Sells weapons and armor for use by the Warrior and the Bandit. The Mage is better served by shopping at the Magic Shop.
- **The Pub**: Every adventurer needs a place to kick back and relax when after returning from a particularly draining dungeon crawl. The pub serves as both a place to recruit companions and recover HP (health points) and MP (magic points).
- **The Magic Shop**: Sells equipment and spells for the Mage. The Bandit may also derive some benefit from visiting this shop, while the Warrior will probably give it a pass.
- **The Item Shop**: Sells consumable items and other miscellanea. All characters will want to visit this shop, if only to buy potions to restore HP and MP.

Perceptive eyes will also note the presence of a nonplayer character (NPC) at the lower-right corner of the town, seemingly blocked off from the rest of civilization. He'll be host to a particularly neat shop concept, but I'll get to that in the final chapter. The first order of business is to make the map, as displayed in Figure 1-1. Here's what you need to do:

- Create a new project, if you haven't already (I named mine DungeonCrawler), by clicking the File menu near the top of the application and then clicking New Project (you can press Ctrl+N, if you'd rather use hotkeys instead).
- Create a new map by right-clicking (or pressing Insert while hovering over) the project name at the lower-left corner of the screen (that part of RMVXA is the map list and is where our game's maps will go). Make sure that the new map has a width of 25 and a height of 25 and uses the Exterior Tileset. Name it "Eagle's Crossing."
- Verify that RMVXA is in Map Editing Mode (you can find the option in the Mode menu; alternatively, press F5).
- Directly below the menu bar, there are several series of icons. You'll want to find the series that has a pencil next to a square. Those are two out of five of RMVXA's map-drawing tools.
- Using either the pencil or the square tool (I prefer the pencil for single tiles and the square tool for larger groups of tiles, such as the roofs of our town's buildings), copy the map in Figure 1-1 (if the black and white is throwing you off, you can find blank versions of all of the maps in this game in the downloadable source code). Both tools are used by left-clicking after selecting the tile you wish to draw on the map. (With the square tool, you can also hold the mouse button and drag, to increase the size of your rectangle.)
- Once you are done with the exterior, you'll want to create the building interiors. With that said, take a look at the following note.

■ **Note** Using the sample maps provided within RMVXA is a great way to populate your game world. You can find them by right-clicking the map list at the lower-left corner of RMVXA and selecting the Load Sample Maps option.

That's precisely what I will be doing to create the building interiors. For a longer game, you'll probably want to dedicate a single map to the entirety of your town's interiors. Because this game will be designed in fewer than 20 maps (most of those will be consigned to the dungeon floors themselves), we can use one map slot per building. It doesn't matter much, all things being told. In any case, I tweaked the relevant sample maps, so that they would fit in a 17×13 tile space (this stops the map from scrolling while the player is on it). It gives the shops and the pub a neat little sense of coziness, in my opinion, anyway. Figures 1-2 to 1-4 contain the relevant screenshots. You'll notice the absence of the Item Shop among those screenshots, but I'm planning to do something neat with that near the end of the chapter. For the sake of consistency, I'll show the buildings in the same order as I noted them previously.

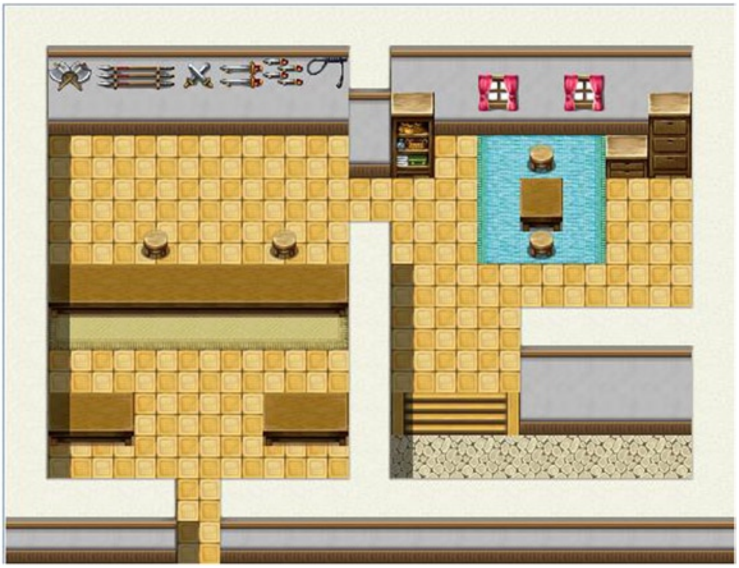


Figure 1-2. *A screenshot of the Equipment Shop*

Given that I wanted to condense the interiors into 17×13 tile spaces, the Weapon Shop sample map is better to work from than the Armor Shop (the Armor Shop is more spacious). I closed off the counter and added a second stool. This shop will have a pair of shopkeepers at the counter. One will deal in weapons; the other will deal in armor.

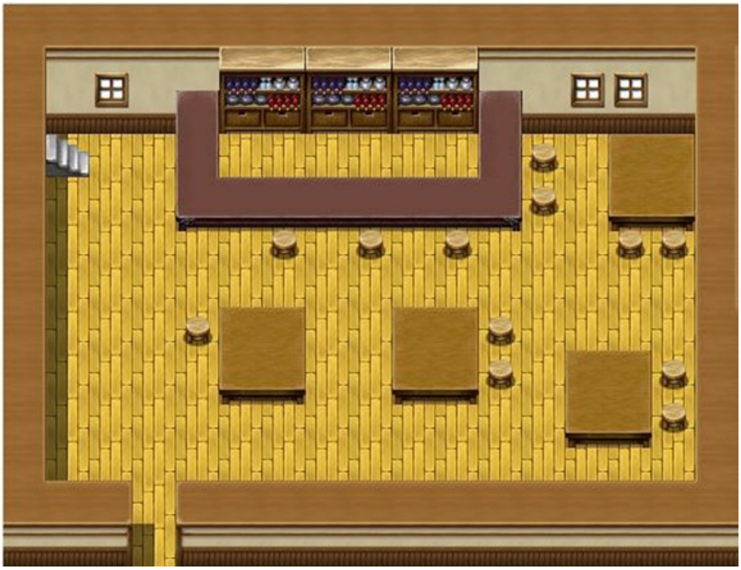


Figure 1-3. *A screenshot of the Pub*

The Pub is the map that I had to tweak most. I started with the InnF1F sample map and tweaked it to the state you see in the preceding figure. This building will also have two NPCs. One will offer companions for hire, while the other will act as the town's innkeeper, allowing the player to rest between dungeon crawls.

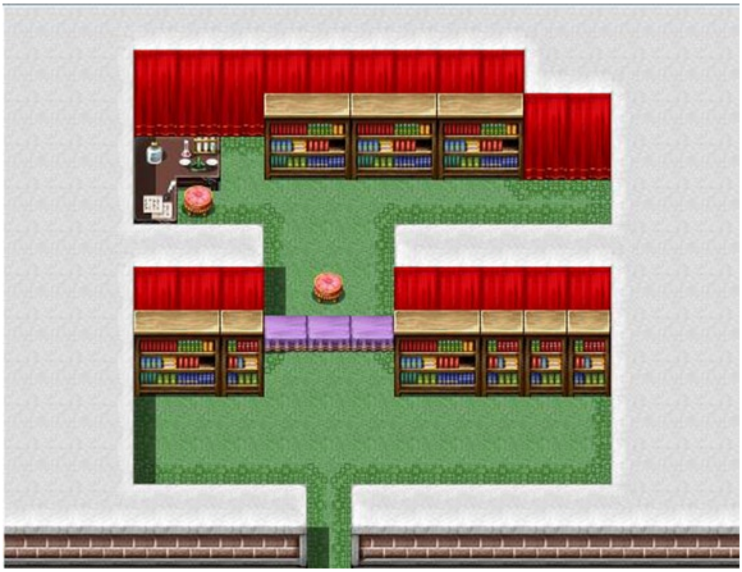


Figure 1-4. *A screenshot of the Magic Shop*

I used the Pencil Tool to move the more important parts of the Magic Shop to the center of the screen and then surrounded the shop with walls. Not much else to say about this one. A single NPC will be sitting on the stool closest to the entrance and will offer both spell scrolls and equipment best suited to the Mage.

■ **Tip** The easiest way to copy large swaths of terrain is by holding down the right button on your mouse and dragging it to encompass the area you require. The resulting rectangle of tiles can be placed by left-clicking the map. I recommend using the Pencil Tool for this.

While our town is currently devoid of life, that will be changing soon. We're going to add some NPCs to our world. You'll have to do the following:

- Switch from Map Editing Mode to Event Editing Mode. You can either use the Mode menu or press F6 (the hotkey for Event Editing Mode).
- Once you're in Event Editing Mode, you can right-click any tile of the map to bring up a small menu with several options. Head on over to the Magic Shop and right-click the stool behind the counter. The option we want to click is New Event.
- Afterward, you will be greeted by a large screen with many options. For now, just double-click Graphic and select the NPC graphic that most appeals to you for a Magic shopkeeper. (Most of the human characters are contained in the graphics sets that are named Actor and People.)
- Repeat this process on the other maps to add four other NPCs in the following locations: two in the pub (behind the counter), and two for the equipment shop (again, behind the counter).

That's all well and good, but now we have to link each of the building interiors with its exterior. To do this, we must use transfer events. You can create a transfer event automatically (via the Quick Event Creation submenu that you receive when you right-click a map in Event Editing Mode) or manually. Also, there's a way to make a single transfer event that covers multiple locations. As you may have noticed when you created your first event, the Create Event screen defaults to a certain set of parameters. Walking Anim. (short for "Walking Animation") is toggled; the event Priority is set to Below Characters (meaning players can pass over the event square); and the event Trigger is set to Action Button, meaning that the player can activate the event by pressing Enter while in contact with the event. For Below Characters and Above Characters, the player is in contact when he/she is directly above/below the event. For Same as Characters, the player is in contact with the event when he/she is directly next to and facing it.

With all of that said, when I list event code during the course of this book, I will list any differences in starting parameters and then write out the actual contents of the events themselves. To add an event command to an event, all you have to do is right-click the @> symbol in the Contents section and select Insert. You will receive a three-page list of event commands that you can plug into any of your events. For example, following is a transfer event that connects the interior of the Equipment Shop to its exterior. Note how the event's contents are displayed with the @> symbol at the beginning. This is exactly how it would look inside RMVXA.

```
Trigger: Player Touch
@>Play SE: 'Move', 80, 100
@>Transfer Player:[001:Eagle's Crossing] (006,011)
@>
```

■ **Note** I used Quick Event Creation to make the event on the previous page. Note how automatic transfer events default to Player Touch instead of Action Button. As the name implies, such an event triggers when the player walks onto it (or into it, in the case of Same as Characters Priority).

Using that transfer event as a template, make the other two interior transfer events. Once you're done, you'll want to make the events for the exterior. This time, let's use one of my favorite transfer event trimming devices: the Parallel Process trigger. Events with a Parallel Process trigger are always running in the background, waiting to be activated. Be careful not to crash/hang your game with them! Anyway, click one of the corner squares of Eagle's Crossing and create a new event. This meta transfer event, if you will, is going to do the following:

- Determine the player's location by polling his/her x,y position and storing the data into a pair of variables. The Control Variables event command will handle this.
- Via the use of Conditional Branches (another event command), check to see if the player is standing at the entrance to one of the buildings.
- If he/she is, transfer him/her to the appropriate interior map.

Without further ado, here's the event:

```
Trigger: Parallel Process
@>Control Variables: [0019:X] = Player's Map X
@>Control Variables: [0020:Y] = Player's Map Y
@>Conditional Branch: Variable [0019:X] == 6
  @>Conditional Branch: Variable [0020:Y] == 10
    @>Transfer Player:[002:Weapon Shop] (004,011)
    @>
  : Branch End
  @>
```

```
: Branch End
@>Conditional Branch: Variable [0019:X] == 18
  @>Conditional Branch: Variable [0020:Y] == 10
    @>Transfer Player:[003:Pub] (003,011), Up
    @>
  : Branch End
  @>
: Branch End
@>Conditional Branch: Variable [0019:X] == 18
  @>Conditional Branch: Variable [0020:Y] == 19
    @>Transfer Player:[004:Magic Shop] (007,011), Up
    @>
  : Branch End
  @>
: Branch End
@>
```

■ **Tip** Another way to go about this is through the use of regions. You would have to use Get Location Info in conjunction with a third variable, to store the region value of the square the player is standing on. Then, you use a distinct region for each building entrance. I will be using regions in a later chapter, but for the most part, you'll want to pick up my *Beginning RPG Maker VX Ace* (Apress 2014) to read more about them.

If we were to use a region for the same effect, part of the Parallel Process event would look something like the following code (keep in mind that you need to actually mark the Equipment Shop entrance with the correct region, or nothing will happen):

```
@>Control Variables: [0019:X] = Player's Map X
@>Control Variables: [0020:Y] = Player's Map Y
@>Get Location Info: [0018], Region ID, Variable [0019][0020]
@>Conditional Branch: Variable [0018:Region] == 1
  @>Transfer Player:[002:Weapon Shop] (004,011)
  @>
: Branch End
@>
```

As long as you're not planning to use the regions for something else (and, given that you have 63 possible regions, this should not be an issue), it could be a nice way to handle area transitions. Now, what happens if you want to embrace even more minimalism in your game and don't want your buildings to have actual interiors?

The Adventurer's Quarter in Eagle's Crossing—A Minimalist Item Shop

That's a bit of a mouthful for a section title, isn't it? Anyway, this section will tackle the opposite approach. We're going to create an Item Shop that consists merely of a single event that handles what would normally be handled by NPCs inside the building. On that same note, the Item Shop won't have an interior. We want this event to do the following:

- Make the player's sprite disappear (to simulate the effect of going into the building).
- Bring up a description of the shop and its keeper(s). For subsequent visits, you could make it so that said description is abbreviated a bit.
- Give the player a list of choices. In the Item Shop, the choices would be Shop and Leave.
- Once the player is done, have him/her leave the shop and make his/her sprite reappear.
- Otherwise, revert to the previous list of choices, so that the character can continue shopping.

If you want to try your hand at the event, here is what it will require in terms of raw event commands and setup.

- Change Transparency
- Show Text
- Show Choices
- Shop Processing
- Set Move Route
- Jump to Label/Label
- The event should have a Below Characters priority and a Player Touch trigger.

That shouldn't be too hard to work out. If you figured it out and want to check against what I did, or are otherwise stumped, take a look at the following to see the relevant event code. Do note that we have yet to design what kinds of items our game will have. As that will be covered later on, I'm using comment placeholders for the event. You'll want to place this event directly on top of the entrance to the Item Shop.

```
Trigger: Player Touch
@>Change Transparency: ON
@>Text: -, -, Normal, Bottom
:     : You enter the shop and are instantly overwhelmed by
:     : the variety of equipment available. You look at your
:     : gold bag as you make your way to the counter, where
:     : an older shopkeeper awaits.
@>Label: MainMenu
@>Text: -, -, Normal, Bottom
:     : What can I do for you?
@>Show Choices: Shop, Leave
: When [Shop]
  @>Comment: Shop Processing Goes Here.
  @>Jump to Label: MainMenu
  @>
: When [Leave]
  @>Set Move Route: Player (Wait)
  :               : $>Turn Down
  :               : $>Transparent OFF
  :               : $>1 Step Forward
  @>Text: -, -, Normal, Bottom
  :     : You leave the shop.
  @>
: Branch End
@>
```

There you have it! The label is used to ensure that the player is not kicked out of the building as soon as he/she leaves a shop. By doing it this way, we make it so that the player has to consciously leave the shop. You could do the same thing for each of the other buildings as well, if you're so inclined.

Of Blank Maps

As mentioned at the start of the chapter, the source code for this book includes the entire game. If you're ever stumped by how I implemented X or Y, feel free to take a look. Additionally, if you'd rather not draw up the area maps yourself, I provide blank versions of every single map used for the game in the same project file. Drawing maps for your game is good practice but one of the most time-consuming aspects of working with RMVXA. Because the intent of this book is to create and complete a game within a weekend, don't hesitate to use the blanks to accelerate your progress, if need be. That concludes this chapter.

Summary

During the course of this chapter, we started work on our game, establishing the basic plot of the story, the playable characters, and creation of the basic skeleton of Eagle's Crossing. The town will be populated in Chapter 4. In the next chapter, we will create our game's main characters and a way for the player to select which to play the game as.

CHAPTER 2

The Characters of Our Game

This chapter is all about our three characters and the many systems related to them. During the course of this chapter, I will cover subjects such as the stat system that we will use for our game, a list of weapons and armor that will be available, and a way for the player to pick which character he/she would like to use.

Overview

The first order of business is to decide how we want our stat system to work in our dungeon crawler. By default, RPG Maker VX Ace (RMVXA) favors high numbers. For example, a character can have up to 9999 HP (health points) and 999 MP (magic points). However, most of the dungeon crawlers I mentioned in the previous chapter run with much lower numbers. For instance, *Diablo*'s characters have one primary stat that can be increased to 255 and three secondary stats with lower caps. So, in the interest of differentiating our game a little more from RMVXA's default system, let's lower our stats by a lot. Doing this will also help us simplify the game in ways that possibly won't be apparent for quite a while. First, head to the Database and erase the preexisting entries for everything in the Classes and Actors tabs of the Database. You can find the Database from the Tools menu at the top of the application or by pressing F9.

Tip A really easy way to purge a Database section is by changing the section's maximum entries to 1. You can do this by clicking Change Maximum near the bottom of most Database tabs. Then, you need only reset the value to the maximum of your choice, right-click the lone remaining entry, and then click Clear (or press Del) to delete it.

After you are done with the purge, head over to the Classes tab. Let's create the classes for our three playable characters. First, we need to figure out each class's starting stats, as well as how much of each stat we want them to gain at each level. As I mentioned, I want the stat system to have *low* numbers. Right now, all of the stat curves are high.

Take a look at Table 2-1 to see how low I'm going. MHP and MMP are abbreviations of Maximum HP and Maximum MP respectively. The other abbreviations are as follows:

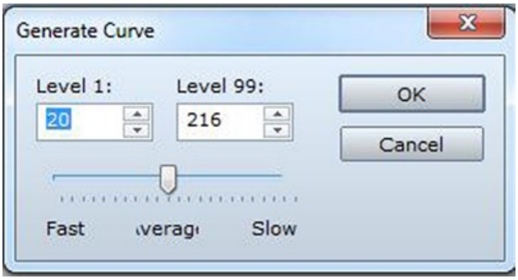
- ATK – Attack
- DEF – Defense
- MAT – Magic Attack
- MDF – Magic Defense
- AGI – Agility
- LUK - Luck

Table 2-1. *List of Base Stats for Each Class and the Stats That They Gain on Each Level Up*

Class	MHP	MMP	ATK	DEF	MAT	MDF	AGI	LUK
Warrior (base stats)	50	0	3	3	1	1	1	1
Warrior (stats per level)	6	0	1	1	0	0	0.5	0.5
Bandit (base stats)	35	15	2	2	2	2	2	2
Bandit (stats per level)	4	2	0.5	0.5	0.5	0.25	1	1
Mage (base stats)	20	30	1	1	3	3	1	1
Mage (stats per level)	2	4	0.25	0.25	1	1	0.5	0.5

As you can see, the Warrior starts with the most HP but no MP, while the Mage starts with the least HP and the most MP. The Bandit's stats are between those of the other two classes. Note how each class differs in its stat gains and its initial stats. Taking the Warrior class as an example, here's how you can set up/change its stats:

- Double-click any of the eight parameter curves, and a new window will open, containing the base curve. Keeping in mind that RMVXA can plot a custom stat curve, all you need to know is the character's stat value at 1 and 99, based on stat growth.
- Click Generate Curve, and you'll get a window much like Figure 2-1 (which illustrates the Mage's HP curve). In the Warrior's case, he has 50 HP at Level 1. For Level 99, he will have leveled up 98 times for 6 extra HP a level, or 588 more HP. So, the Warrior will have 638 HP at a hypothetical Level 99.



Generate Curve

Level 1: 20

Level 99: 216

Figure 2-1. *A screenshot of the Generate Curve window for the Mage's MHP stat*

■ **Note** The slider bar below the curve values determines the speed at which the class gains that particular stat. You can make classes that gain the bulk of a stat early in their career and others that gain it later on. For the purposes of this game, leave all of these sliders on the default setting of Average.

- Plug in 50 for Level 1 and 638 for Level 99, and you're set! Now, I could have you crunch all of those other max parameters, but that would be mean. So, instead, see Table 2-2 for a list of the maximum stats for each of the three classes at Level 99.

Table 2-2. *The List of Maximum Stats for Each Class at Level 99*

Stats at Level 99								
Class Name	**MHP**	**MMP**	**ATK**	**DEF**	**MAT**	**MDF**	**AGI**	**LUK**
Warrior	638	0	101	101	1	25	50	50
Bandit	329	218	51	51	51	26	100	100
Mage	216	422	25	25	101	101	50	50

With those numbers in hand, take some time and fill out the stat blocks for each of the three classes. After that, we have one last curve to work on: the experience curve. By clicking the button marked "..." below Exp Curve, you'll bring up a screen with various parameters that can be changed with scrolling bars. Minimize all of them (which drops Base Value and Acceleration B to 10 and the other two parameters to 0) for the three classes and then note how the experience required per level has dropped.

■ **Note** A quirk of dropping all four parameters to their minimums is that experience requirements to gain a level beyond level 84 *drop* rather than increase. This isn't significant, given our game's level cap; it's just a neat little bit of trivia that I noticed.

The last thing we have to do in the Classes tab is set up the features of each class. See Table 2-3 for a list of features that each class should have.

Table 2-3. *A List of Features for Each of the Game's Three Classes*

Class Name	HIT%	CRI%	Equip Weapon	Equip Armor	Add Skill Type
Warrior	85	5	Axe, 2H Axe, Dagger, Sword, 2H Sword	Light Armor, Medium Armor, Heavy Armor	No
Bandit	85	5	Axe, Bow, Dagger, Sword	Light Armor, Medium Armor	Magic
Mage	75	5	Dagger, Staff	Light Armor	Magic

■ **Tip** You can add and edit class Features by right-clicking a preexistent feature or, in the empty slot directly below the last feature in the list (the very first slot, if the Feature list is completely empty), by selecting Edit. Double-clicking works as well.

Now, flip over to the Actors tab and create your three characters. First, add their sprites and portraits, as seen in Figure 2-2.

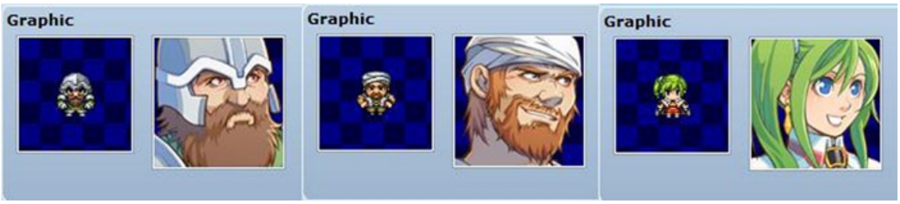


Figure 2-2. *(From left to right) Palnor's sprite and portrait, Gust's sprite and portrait, and Feylia's sprite and portrait*

■ **Note** You can find Palnor's sprite in the Actor3 sprite set and his portrait in the similarly named portrait set. Gust's graphics are located under Evil, while Feylia's are in Actor2.

Next, we will fill out the General Settings for each actor as follows:

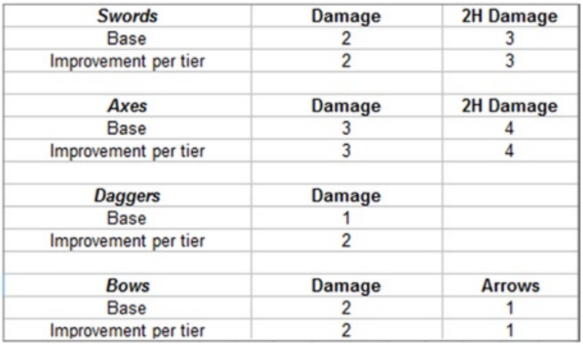
- **Palnor**: Class: Warrior—Initial Level: 1—Max Level: 20—Description: He seeks the ancient artifact in order to revive his kin and the rest of the people of his town.
- **Gust**: Class: Bandit—Initial Level: 1—Max Level: 20—Description: He seeks the ancient artifact in in order to revive the love of his life.
- **Feylia**: Class: Mage—Initial Level: 1—Max Level: 20—Description: She seeks the ancient artifact to revive her brother, lost to a plague years before.

In filling out the General Settings, you may have noticed the Starting Equipment section of the Actors tab. I'll get to that in a bit.

Equipping the Characters

For now, we'll want to come up with the armor and weapons that the player will find and use during the course of a play-through. It's not necessary to add all of the items to the Database just yet, but your game will benefit greatly from thinking about this and adding at least the first set of equipment. For starters, this will help you balance the stats of any enemies that the player will face in the first few levels of the dungeon. With that said, Table 2-4 covers the various weapon types that the Warrior and Bandit will be able to use (the Mage's staff-type weapons [staves] are a bit different, and I'll cover them momentarily).

Table 2-4. *The Four Types of Weapons Available for Warriors and Bandits to Use*



Swords	Damage	2H Damage
Base	2	3
Improvement per tier	2	3
Axes	Damage	2H Damage
Base	3	4
Improvement per tier	3	4
Daggers	Damage	
Base	1	
Improvement per tier	2	
Bows	Damage	Arrows
Base	2	1
Improvement per tier	2	1

The table bears a little explanation. Mainly, our game will have six types of materials for weapons. Those materials are (in order of damage potential): wood, copper, bronze, iron, steel, and mythril. The base potential damage of a weapon is the amount of ATK (attack) given by a wooden version of that weapon. For each tier higher than Wood, you add the improvement value. So, a mythril sword would give the wielder 12 ATK. 2H is shorthand for "two-handed." (It's a common nomenclature in role-playing gaming communities, so you might have already known that.) Only Palnor the Warrior is allowed to use two-handed swords and axes, which have roughly 50% more ATK than their one-handed cousins. Only Gust the Bandit is allowed to use bows, and they inflict extra damage if you have an arrow equipped in the off-hand slot (the one used for shields by default in RMVXA). Feylia's staves are a bit different in that they increase her ATK like other weapons, but they also increase her MAT. Additionally, there are only four types of distinct staff-type weapons. Obsidian is in the same tier as iron is for other weapons, while crystal is the same as for mythril. See Table 2-5.

Table 2-5. *The Four Types of Staff Weapons That Feylia Can Use*

Staves	ATK	MAT
Wood	4	2
Stone	6	6
--	--	--
Obsidian	10	14
--	--	--
Crystal	14	22

Armor is a little more complex than weapons. For one, mythril, in our game (and in certain fantasy settings throughout the years), is a magical metal that's unusually light. Thus, it is possible to make from it a tunic that even Feylia can wear. In the same vein, Palnor can wear any type of armor. Gust can wear light and medium armor, while Feylia can only wear light armor. The first five tiers of heavy armor are full plates (that is to say, they're a full suit of armor). Thus, Palnor cannot wear a helm while wearing any of those suits. Adamantine, in our game, is an exceedingly heavy metal that offers unparalleled protection, should you have the strength to wear it. No one was/is crazy enough to try and make weapons out of that particular metal. Adamantine comes in both chestplate (which allows a helm to be worn) and full plate varieties. See Table 2-6 for a full list of DEF values for our game's armor.