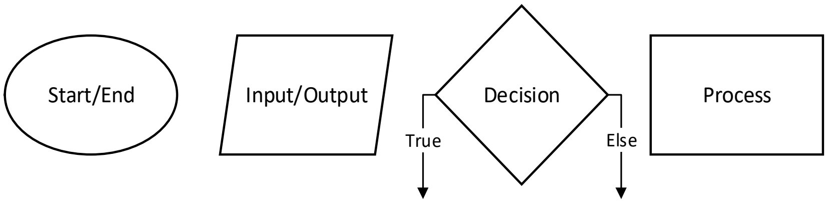
Name:________________________________

RPG Maker MV Workbook

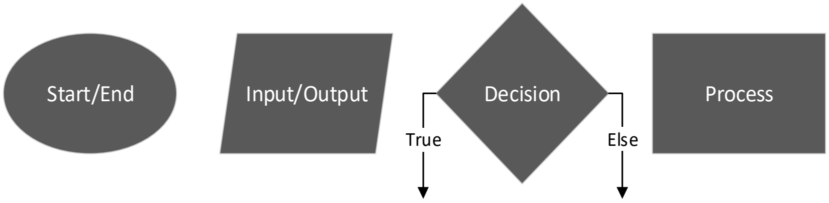
Flow Chart Shape “Color Code”

This workbook also uses shades of black to signify different things. White shapes are instructions for the programmer, whereas black shapes are instructions for the program.

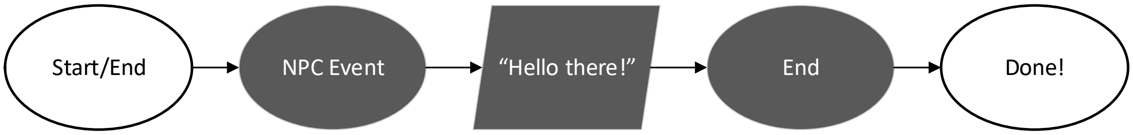
White shapes are instructions for you, the programmer:



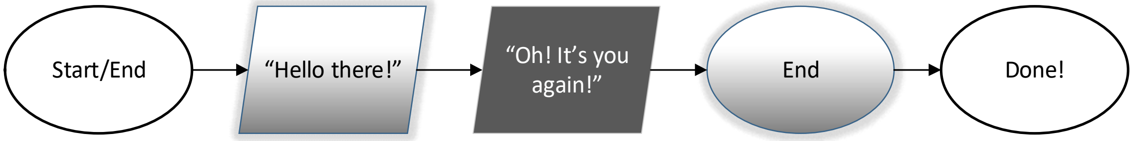
Black shapes are Command Events. The Event will be named in the “Start” oval.



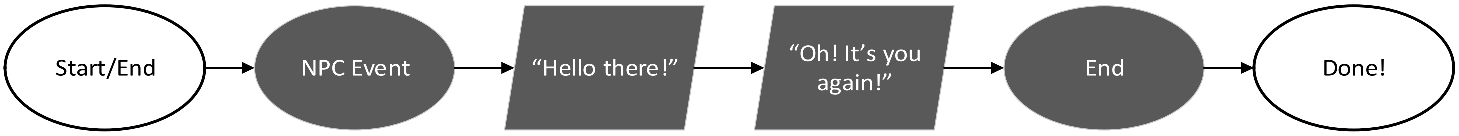
Gray/Faded shapes tell you that you need to insert code into an existing event. For example, if we have a flow chart like the one below:



…and a different flow chart tells you to insert a new step, like this:



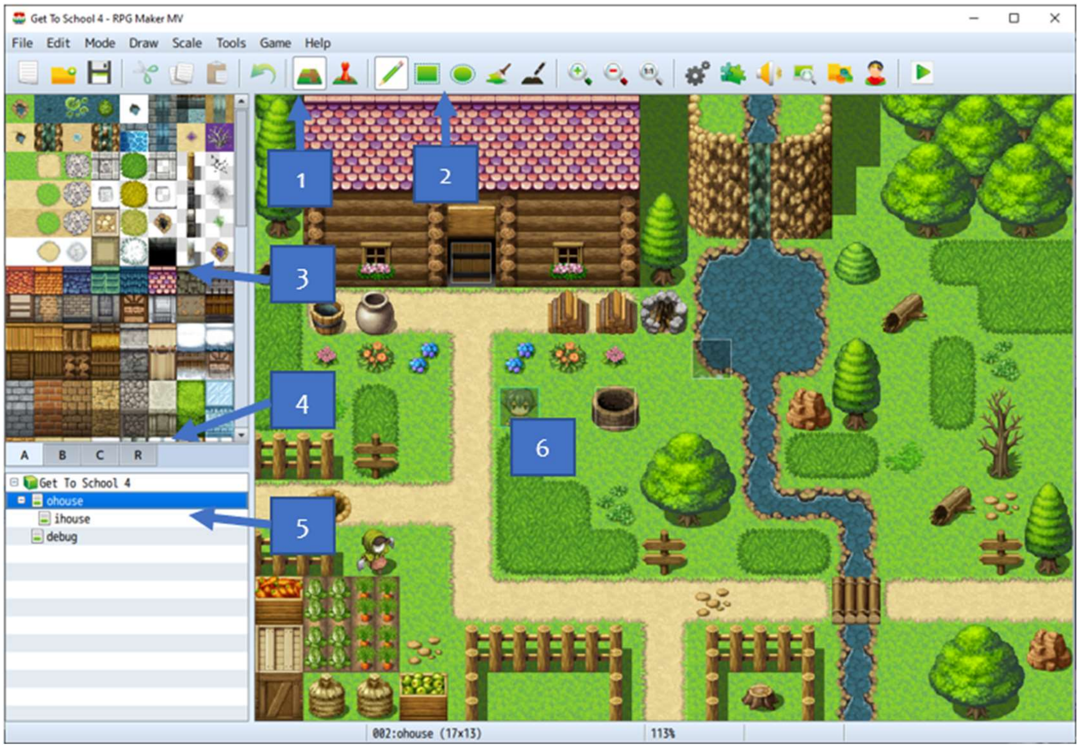
…then the original flow chart will look like this when you’ve inserted the new steps:



Create Your Home [Items, Text]

Create a well which your player can draw water from. This water will be an item the players can use later for healing and crafting.

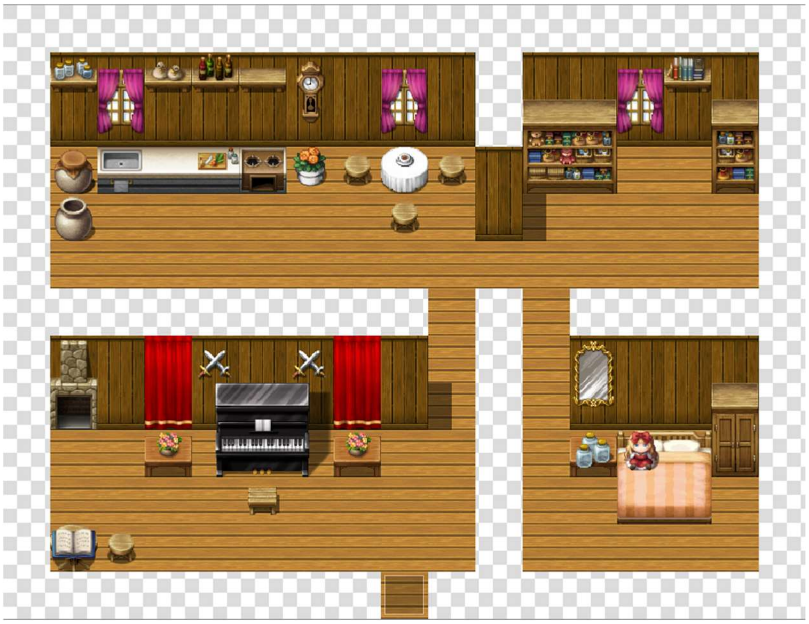
Every adventure starts at home. The next several challenges will involve adding amenities to your home to help you prepare for adventure. They will also give you a place to come home and rest when you're worn out from your travels. To create our home, we will create two new maps in the Maps List. Maps in the maps list can be easily organized like folders. I made my "ohouse" for the exterior (outside) of my home, and inside of that map in the Maps List, I place my "ihouse" for the interior (inside) of my home.


Get To School 4 - RPG Maker MV
File Edit Mode Draw Scale Tools Game Help
1
2
3
4
5
6
A B C R
Get To School 4
ohouse
ihouse
debug
002:ohouse (17x13)
113%

1. Map Edit Mode
2. Map Drawing Tools
3. Tiles
4. Tile Layer Selector
5. Maps List
6. Map Editing Area

Have fun designing the outside and inside of your home. Take time to learn how the tiles work. The Tiles menu has four tabs by default: A, B, C, and R. Don't worry about R, yet. Think of layer "A" as the ground layer. Layer "B" are things that go on top of the ground. Layer "C" are things that go on top of the things that go on top of the ground. For example, the floor of your house is Layer A, the table that goes on the floor is Layer B, and the tea cup that goes on the table is Layer C.

You cannot rotate tiles, but some tiles are "active tiles", meaning they will adjust and draw new shapes as you paint them onto the map. Common examples of these are animated water tiles, hill/mountain tiles, walls, and roofs. There are also extra tiles that *look* like active tiles, but are static. You can use these to create manual effects, such as multi-level hills with stairs.



Start thinking about your *design language*. This is how you communicate to players what they can do in your game, without pulling them out of the immersive experience of their game. For example, we will need three paths leaving your home for these challenges. I used dirt paths to show the player this, but I also blocked the sides of the path with fences, signs, and trees. This forces the player to have to walk down that path to use it, instead of stumbling upon it randomly.

Door Transfer [Animations, Movement]

Create a door which the player can open to be transported to another map. On the second map, create a region that will take them back to the first map when walked into.

Start by creating two new maps. For my first two maps I like to do the outside of my player's home, and the inside as the second map. On the first map (the outside), create a Door event as shown below. On the inside map, create an "exit" event.

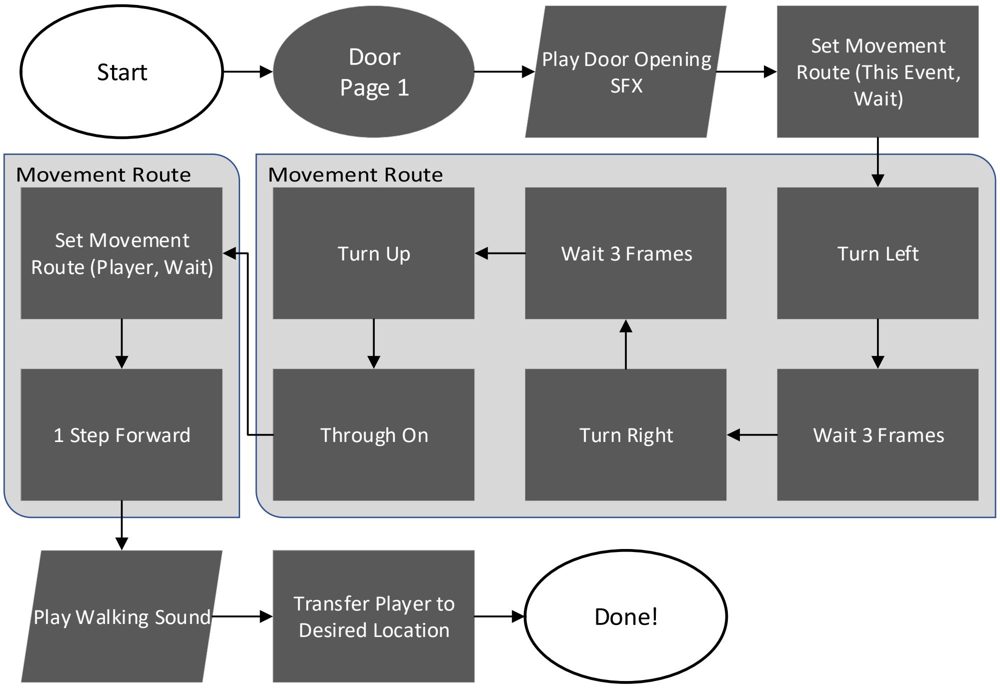
Create: Map Event

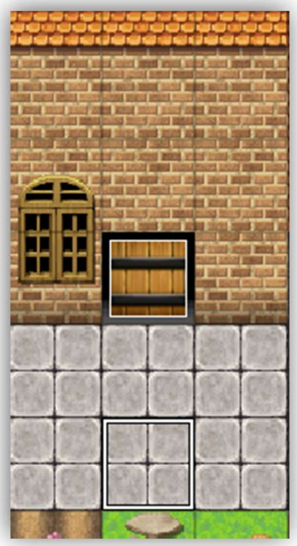
Name	Door
Priority	Same As Character
Trigger	Action

Create: Map Event

Name	Exit
Priority	Same As Character
Trigger	Player Touch

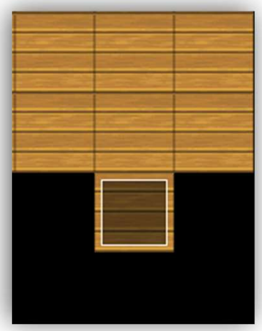
Follow this flow chart to allow the door to swing open and make the player walk through before transferring them to the new map:



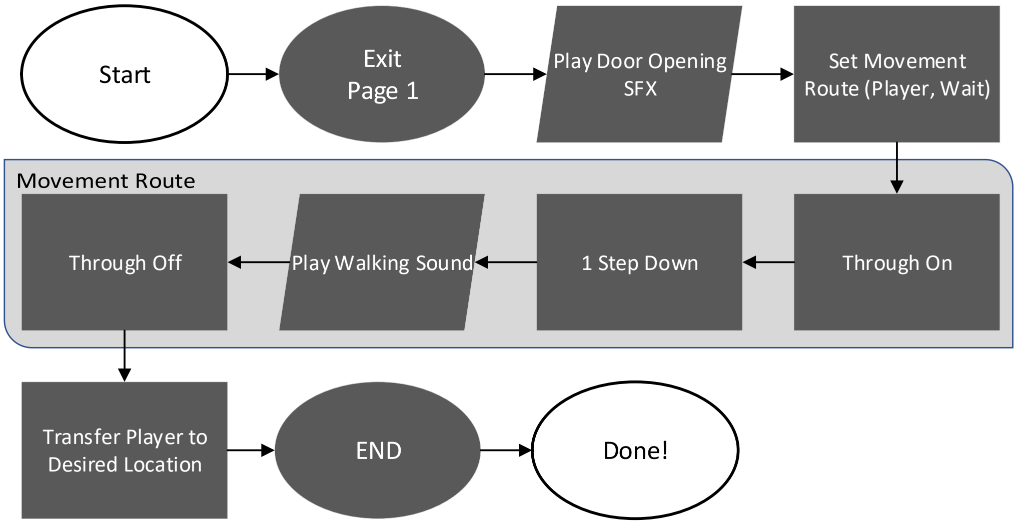


On the first map, be sure to leave a hole in the wall containing the door, and put a doorway tile (Page B) in the doorway.

Then follow this flow chart to create a exit area on the other map, that when stepped on, moves you back to the first map.



In the interior of the building, I like to add an extra floor tile to dhow the player where the exit is. Touching it will transfer them back out of the building onto the first map.



Testing: *Does your door swing open? Then does your player step into the doorframe before transferring? If you touch the exit (not step on top of it), does it make you walk into the doorframe before transferring the player? Do all of the sounds play correctly?*

Well [Items, Text]

Create a well which your player can draw water from. This water will be an item the players can use later for healing and crafting.

Start by creating a Well map event on the map you want the players to be able to get Water. Then create an item in the database to represent the water:

Create: Map Event

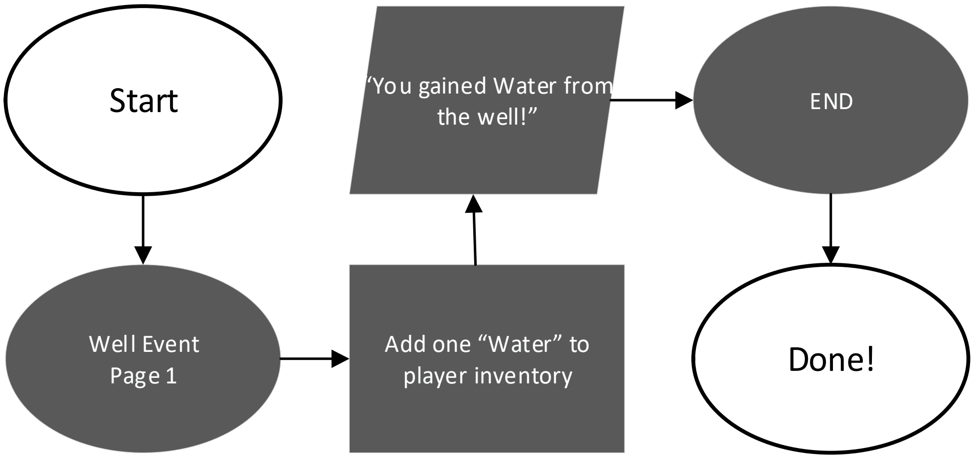
Name	Well
Priority	Same As Player
Trigger	Action Button

Create: Item

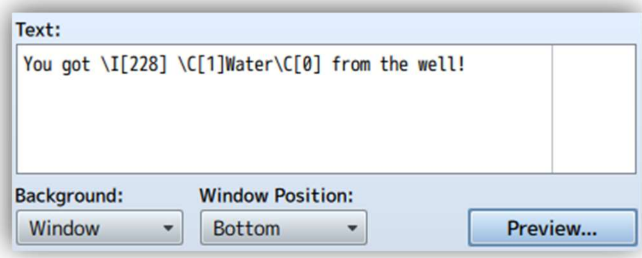
Icon: 228

Name	Water	Type	Regular Item
Consum.	No	Scope	None
Occasion	Never	Effect	None

Follow the simple flow chart below to program the well to give the player water:



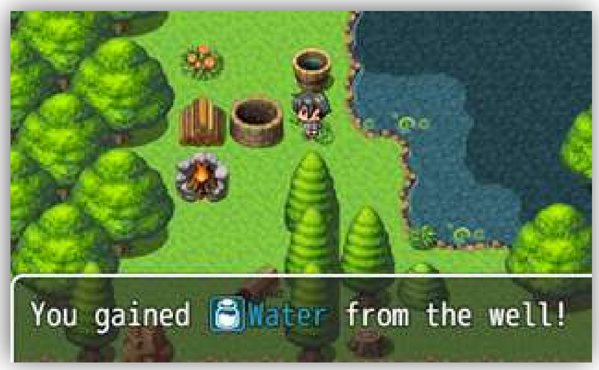
Start
Well Event Page 1
Add one "Water" to player inventory
'You gained Water from the well!"
END
Done!



```
You got \I[228] \C[1]Water\C[0] from the well!
```

When creating the text telling the player they got Water, you can use inserts. Hold your mouse still over the Text box to get a tooltip showing you the options. In this case, we display the **Icon** with \I[xxx], where 'xxx' is the number of the Icon you used for the Water Item. We also use \C[x] to switch the color of the text to blue (1) and switch it back to white (0).

You can always press the "Preview" button to see how your text will look in-game. It won't show variables or gold correctly, however, so don't rely on it 100%



You gained Water from the well!

Using Icons and color in your text can greatly help your players understand what is an Item, what kind of item it is, and alerts them that this is something they should pay attention to. *The Legend of Zelda: Ocarina of Time* uses this to great effect. Have you played it? You should!

Testing: *When you use the Action Button on the Well, does it tell the player that they got water? When you go to your inventory, does it show that you have one more Water item than you did before?*

Fishing Spot [Variables, Items, Animations, Loops, Branches]

Design a fishing spot that goes over a body of water. Activating this spot gives you a random chance of catching a fish.

Start by creating a Small Fish item in the database. This fish will heal a party member if they "eat" it. Then create invisible event over a body of water to act as a fishing spot.

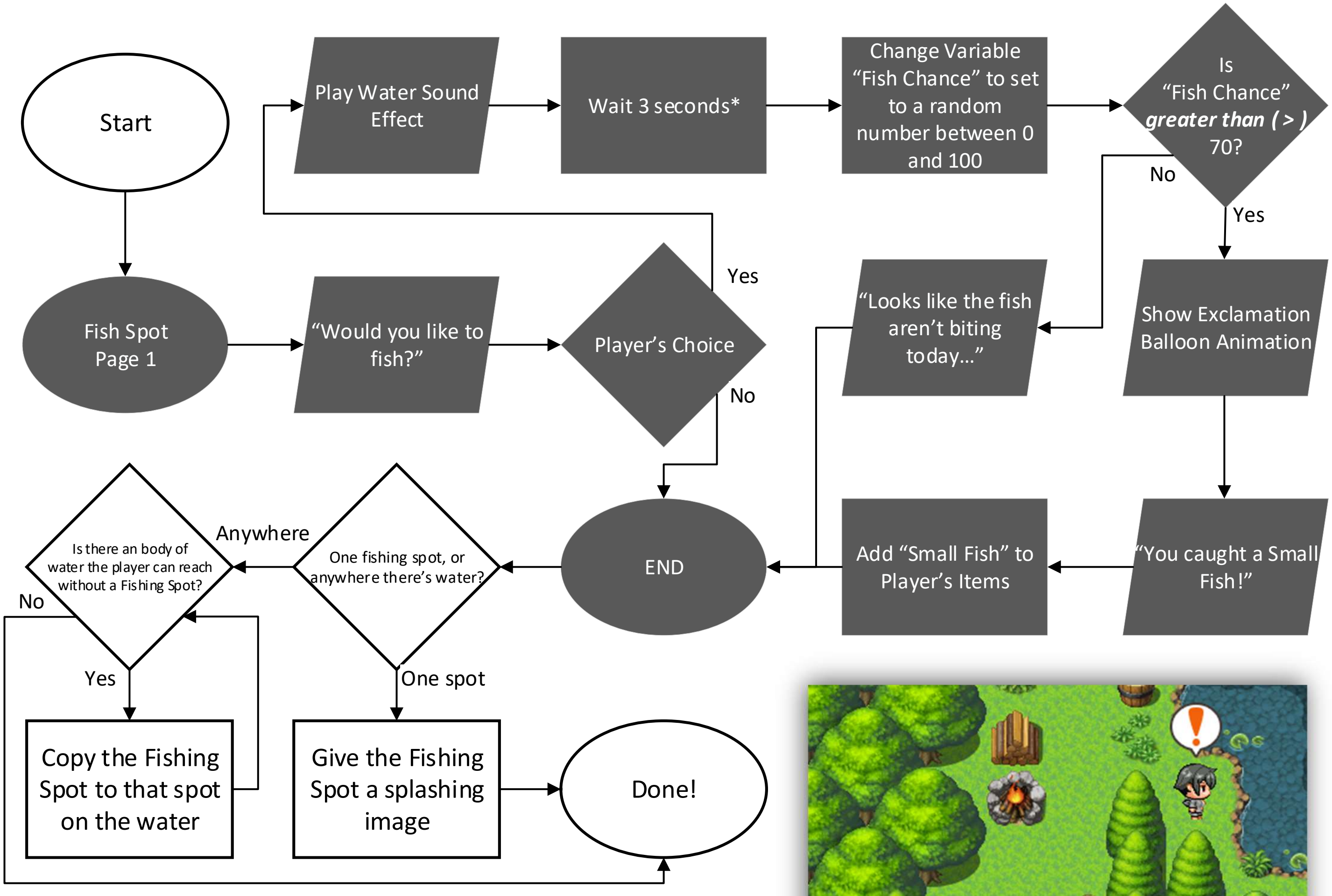
Create: Item

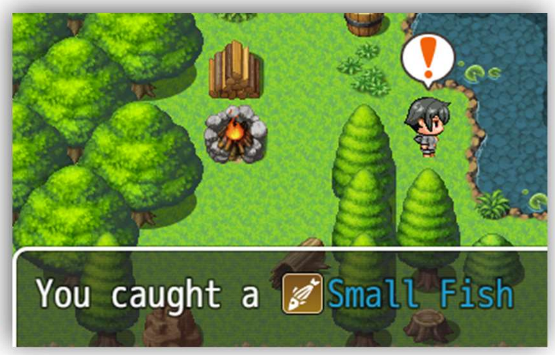
228				
	Name	Small Fish	Type	Regular Item
	Consum.	Yes	Scope	1 Ally
	Occasion	Always	Effect	Recover 50 HP

Create: Map Event

Name	Fishing Spot
Priority	Same As Player
Trigger	Action Button

Follow this flow chart to program the game to let the player fish. Note that near the end, you as the programmer will be asked to make a decision. If you want there to just be one spot the player can fish at, you will have to add a splashing character image where the Fishing Spot is, so the player knows to fish there. Otherwise, you can decide that ANY body of water the player can reach can be fished at. If you do, you will have to copy and paste this event every there is water. This flow chart pattern is what we call a "for" loop. "***For every*** blank space of water the player can touch, copy the Fishing Spot to it."


Start
Fish Spot Page 1
"Would you like to fish?"
Player's Choice
Yes
No
Play Water Sound Effect
Wait 3 seconds*
Change Variable "Fish Chance" to set to a random number between 0 and 100
Is "Fish Chance" greater than (>) 70?
No
Yes
"Looks like the fish aren't biting today..."
Show Exclamation Balloon Animation
"You caught a Small Fish!"
Add "Small Fish" to Player's Items
END
One fishing spot, or anywhere there's water?
Anywhere
One spot
Is there an body of water the player can reach without a Fishing Spot?
No
Yes
Copy the Fishing Spot to that spot on the water
Give the Fishing Spot a splashing image
Done!


You caught a Small Fish

There are three kinds of animation used in this challenge. The first is choosing a character image for the fishing spot if you want it to be visible. It should be a whirlpool or bubbles, with "Stepping" checked in the options. You can make splashing effects over the Fishing Spot with "Show Animation", and you can show player expressions by using "Show Balloon Icon". Try them all out!

Testing: *When you use the Action Button on the Fishing spot, does it ask the player if they want to fish? If they do, does it play all of the correct animations and sounds to make it look like you're waiting for/catching fish? Does it add the fish to your inventory* ***only*** *if you actually "caught" the fish? Do you really have a random chance of catching the fish? Does using the fish heal you?*

Cooking / Crafting [Items, Branches] Requires “Well” and “Fishing Spot”

Make a campfire that can be used to cook Fish Soup using Water and a Small Fish. After this challenge, you can create lots of recipes and add them to you game using the same concepts.

Start by creating a Fish Soup item in the database. This soup will heal a party member if they “eat” it, and should heal more than a Small Fish by itself. Then create a fire event over top of a campfire tile.

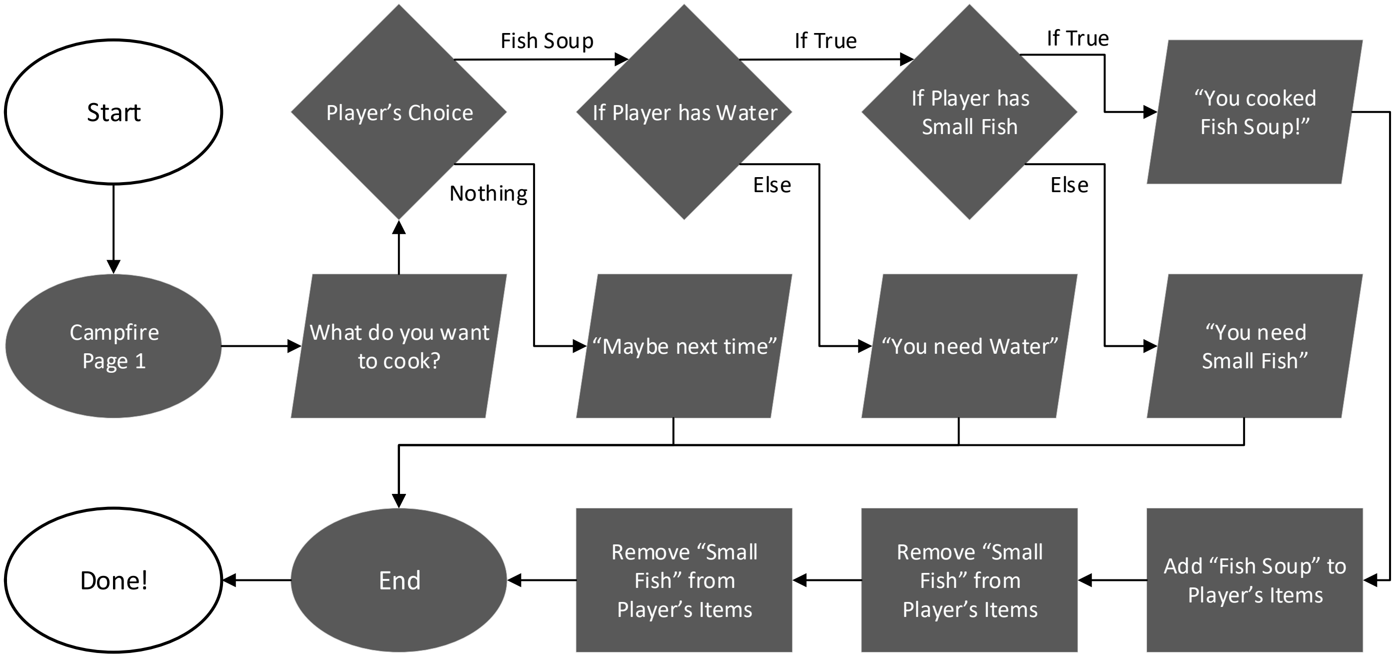
Create: Item

211	Name	Fish Soup	Type	Regular Item
	Consum.	Yes	Scope	1 Ally
	Occasion	Always	Effect	Recover 50 HP

Create: Map Event

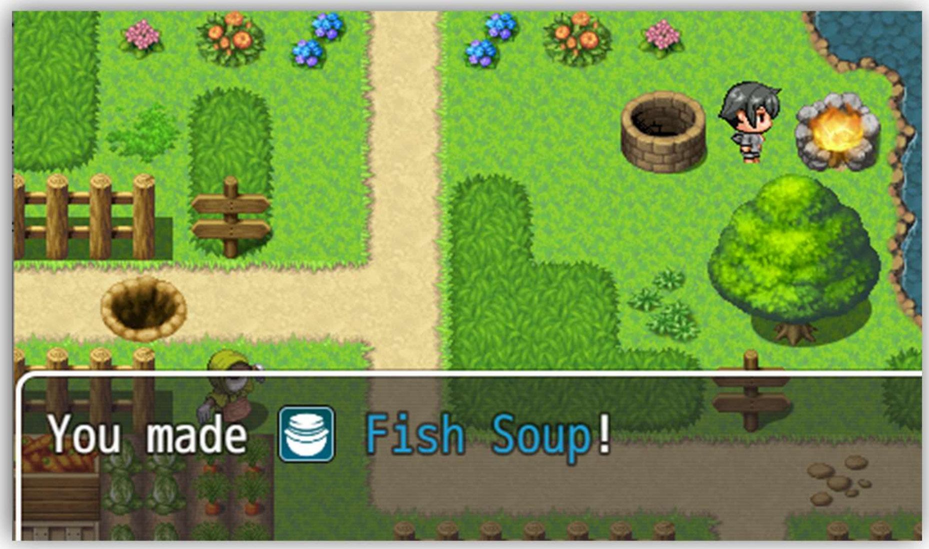
Name	Campfire	Options	Stepping
Priority	Same As Player		Direction
Trigger	Action Button		Fix

Follow this flow chart to program the game to let the player cook the soup:



There are two types of Decision Steps here. One is a “Show Choices”, with the option for many outputs. The other two are a normal “Conditional Branch”, each with an “Else Branch”.

You may want to experiment with adding fire animations over the Campfire when the player can successfully cook something. You can do this on your own, though. I won’t tell you how it’s done.



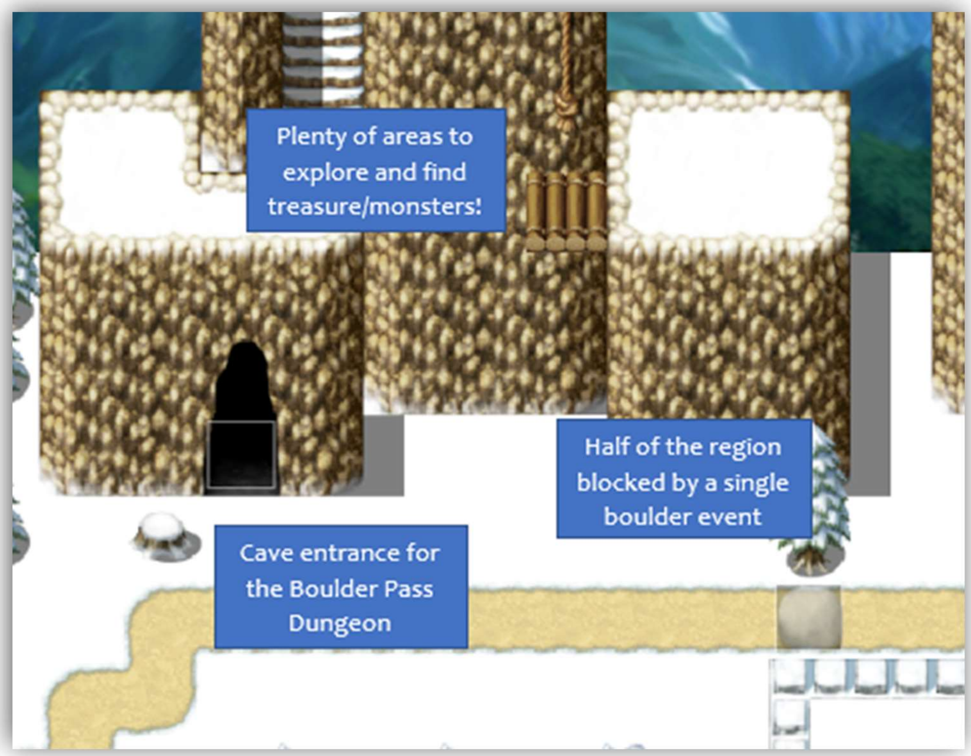
You made Fish Soup!

Testing: *When you use the Action Button on the Campfire, does it ask the player what they want to make? If you choose Fish Soup without Water and/or Small Fish, will it fail? Does cooking Fish Soup add 1 Fish Soup to your inventory, and remove 1 Small Fish and 1 Water from your inventory?*

Create the Boulder Pass

Design the first "dangerous" region of the game. Players will explore this area, finding treasure and monsters, then defeat the local dungeon to progress.

Now that we have our home ready to go, we can work on the first region our players can adventure in. We will design the "Boulder Pass", a rocky region whose path leads to Atown, the town our school is in. I made mine snowy in the image below, but you can make it grassy, rocky, or even sci-fi. The important part is there will be a nearby "Dungeon" that holds the treasure we need to move a boulder, or other heavy object.


Plenty of areas to explore and find treasure/monsters!
Half of the region blocked by a single boulder event
Cave entrance for the Boulder Pass Dungeon

A classic Adventure RPG gameplay loop is one like the Zelda series, which typically goes like this:

1. Find new region
2. Explore the region, noting places you can't get to yet
3. Get to the region's dungeon
4. Fight the Dungeon's Boss
5. Get the Dungeon's Treasure
6. Use the treasure to get to the places you couldn't earlier
7. Repeat

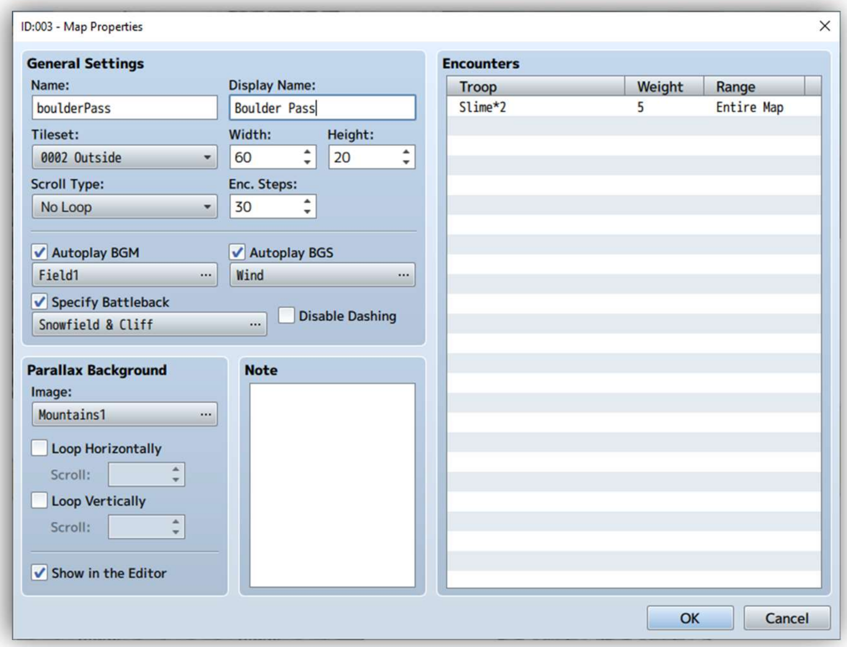
We left our home to discover the Boulder Pass. We can go a few places, but half of the map is blocked off by a boulder. If we enter the Boulder Pass Dungeon, defeat the Boss, and get the treasure (the Power Bracelet), we can move the boulder and continue to the next region in the game.

Let's pay closer attention to our Map Properties this time (right-click on the Boulder Pass map you made in the Map Viewer and select "Edit…"). Because we will be fighting random monsters, we need to "Specify Battleback", which is the background used in combat. If you do not, the game will just default to a picture of the game map without events.

You can add monster groups to be randomly encountered by double-clicking the first empty spot of the "Encounters" list, and adding a "Troop" in. Slimes are a classic first monster for Adventure RPGs, so I used that.

We will have to make the map a little larger as well. If you want, you can use a "Parallax Background", which will show behind transparent or missing tiles in your map.

Finally, don't forget to add a Transfer event from your home to the accessible part of Boulder Pass, and a Transfer event the sends you back to your home.



ID:003 - Map Properties

General Settings

Name: boulderPass
Display Name: Boulder Pass
Tileset: 0002 Outside
Width: 60
Height: 20
Scroll Type: No Loop
Enc. Steps: 30
☑ Autoplay BGM: Field1
☑ Autoplay BGS: Wind
☑ Specify Battleback: Snowfield & Cliff
□ Disable Dashing

Parallax Background

Image: Mountains1
□ Loop Horizontally
Scroll:
□ Loop Vertically
Scroll:
☑ Show in the Editor

Note

Encounters

Troop	Weight	Range
Slime*2	5	Entire Map

Power Bracelet [Items, Conditional Branches] Recommended: Create the Boulder Pass

Design a classic game mechanic to push "boulders" that block our path once we've obtained the magic item "Power Bracelet".

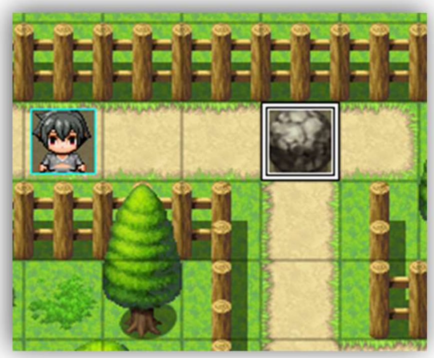
A *classic* RPG item is the Strength Booster. Usually, this item has no real effect on your character's actual Attack score or anything else in the game, but it *does* allow you to push movement blocking events out of your way. In our case, we will use the Power Bracelet to push a Boulder out of the way. Start by creating the Item and Event:

Create: Item

145	Name	Power Bracelet	Type	Key Item
	Consum.	No	Scope	None
	Occasion	Never	Effect	None

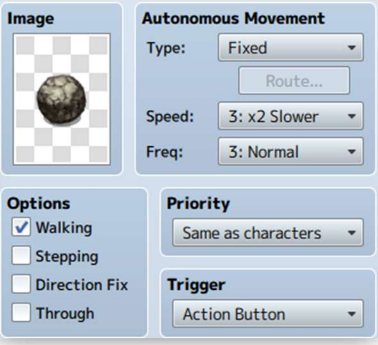
Create: Map Event

Name	Boulder
Priority	Same As Player
Trigger	Action Button



If this is your first time using a Decision Step (Diamond), be sure to explore the "Conditional Branch" command event in its entirety. You do not need to edit the Event's "Conditions" section.

When complete, you can copy/paste the boulder event around your game.



Image

Autonomous Movement

Type: Fixed

Route...

Speed: 3: x2 Slower

Freq: 3: Normal

Options

☑ Walking

☐ Stepping

☐ Direction Fix

☐ Through

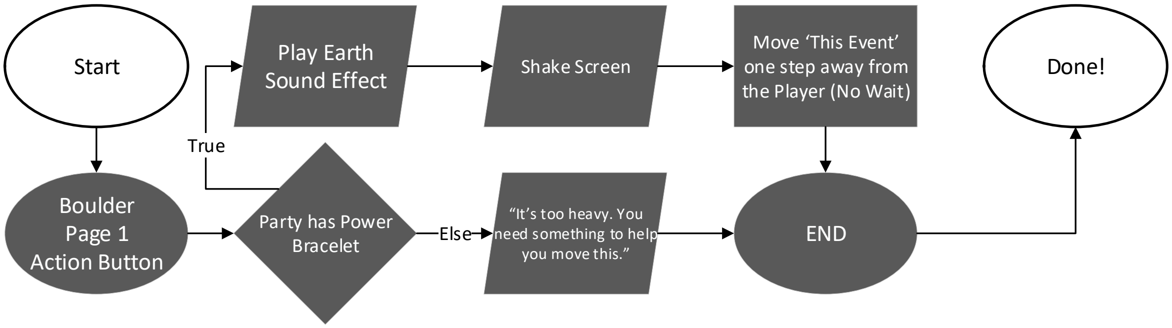
Priority

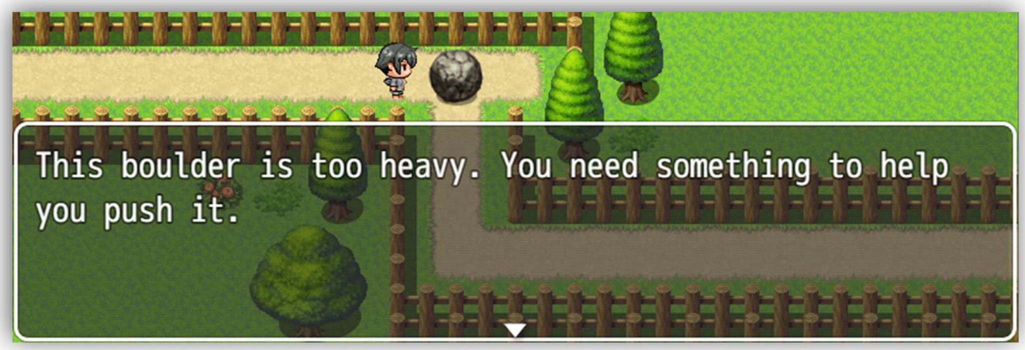
Same as characters

Trigger

Action Button

Then follow the flow chart below to complete the challenge. This boulder will walk/move one step away from wherever the Player is standing when they use the Action button while possessing the Power Bracelet. Do not remove, teleport, or move the boulder in a fixed direction.


Start
Boulder Page 1 Action Button
Party has Power Bracelet
True
Play Earth Sound Effect
Shake Screen
Move 'This Event' one step away from the Player (No Wait)
Else
"It's too heavy. You need something to help you move this."
END
Done!



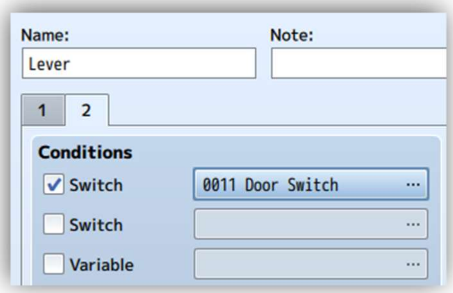
This boulder is too heavy. You need something to help you push it.

***Bonus Challenge:** Once you have this challenge complete, can you edit doors in your game to play an effect and **explode** when you try to open them? The Power Bracelet must be giving you **too** much power.*

***Testing:** Does the boulder move AWAY from the player, towards whatever direction the player is facing? If you push the boulder against a wall or a corner where it can't move, does it freeze the game? Can the player push the boulder with no input after the initial Action Button press?*

Remote Door Lock [Switches, Event Pages] Recommended: Boulder Pass Dungeon

Design a door that can only be opened by pulling a lever elsewhere in the game.


Name: Lever
Note:
1 2
Conditions
☑ Switch 0011 Door Switch ...
☐ Switch ...
☐ Variable ...

Another classic mechanic is many types of games, especially adventure games, is the remote door lock. Mechanically, this can serve a lot of purposes. Sometimes, it is used to make sure the player has made it to a certain place in the world before unlocking a cutscene or new region. Another very common use is to unlock a door near the beginning of the region, so the player doesn't need to backtrack when they have explored everything. In this case, we will use it to force our players to explore the Boulder Pass Region before entering the Boulder Pass.

Start by creating two map events, a door and an lever. Each will have two Pages, one for open and one for closed. In my game, I placed the lever high on a mountaintop, and the door inside the Boulder Pass dungeon cave. It will be the first door the players see before actually entering the dungeon.

Create: Map Event

Name	Lever
Priority	Same As Player
Trigger	Action Button
Page	1
Condition	None
Options	Direction Fix

Create: Map Event

Name	Lever
Priority	Same As Player
Trigger	Action Button
Page	1
Condition	None
Options	Direction Fix

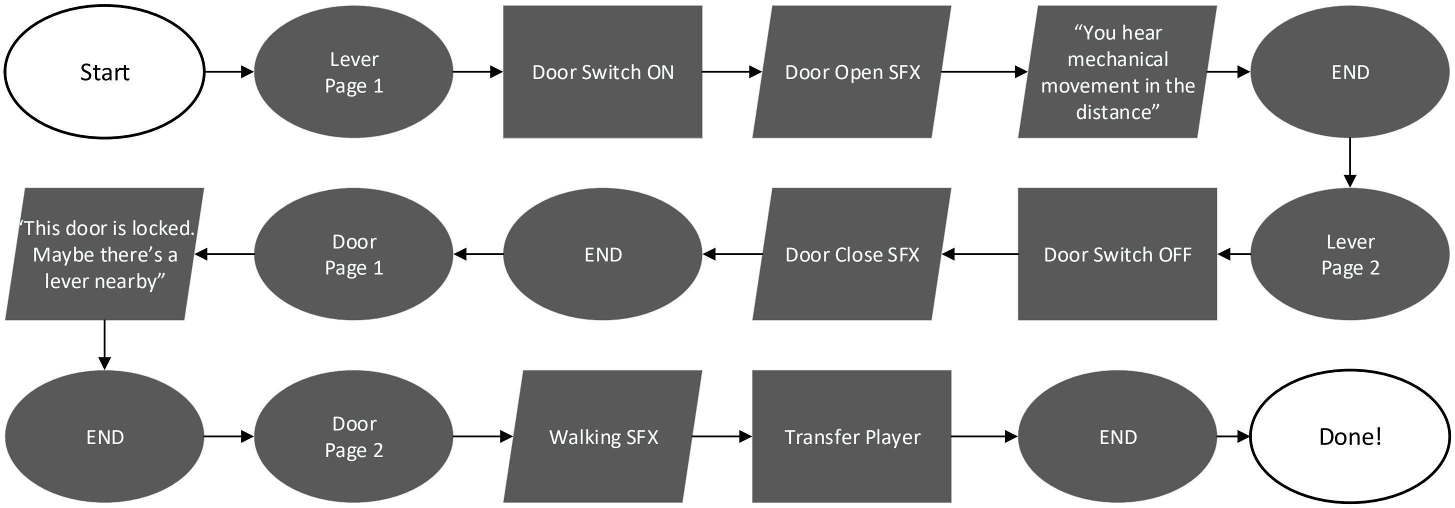
Create: Map Event

Name	Lever
Priority	Same As Player
Trigger	Action Button
Page	1
Condition	None
Options	Direction Fix

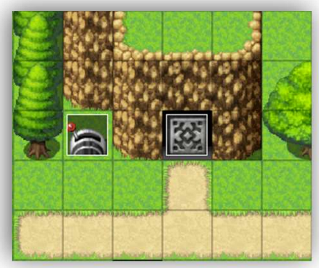
Create: Map Event

Name	Lever
Priority	Below Player
Trigger	Player Touch
Page	1
Condition	None
Options	Direction Fix

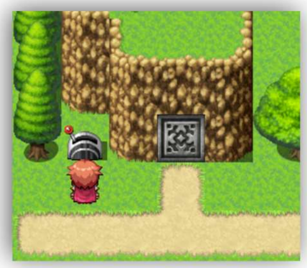
Then follow the flow chart below to complete all of event pages and the challenge. You can also add your own screen shake and text messages to enhance the actions of the door/lever.



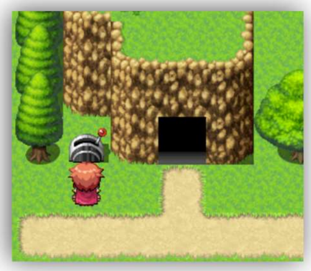
Create Events



Lever is off. Door is closed.



Lever is on. Door is open.



The *most important* thing to remember from this challenge is that **the Event Page with the highest number and all of its conditions met will be the active Event Page.** When the Door Switch is ON, the Page 2 of each event has all of its conditions met. When it is OFF, Page 1 is the only one with all of it's conditions met.

Testing: *Does the lever and the door start off/closed? If you flip the lever, is the door open? If you flip it back, does the door close?*