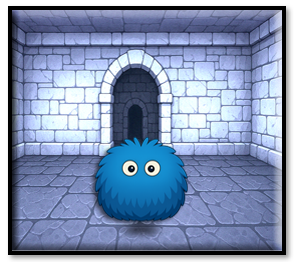
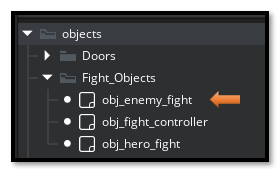
16 Creating Enemies



In this tutorial, we will be looking at creating an enemy. But basically our enemies will all be using the same enemy, or in other words, we will create just one fighting enemy object, and then this single object will be used by all of the enemies, which want to beat up our poor hero guy.

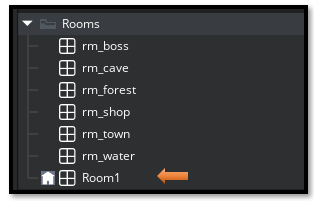


Oh, you can add different behaviors to each different fiend, but still the object that will be controlling it will still just be the same object.

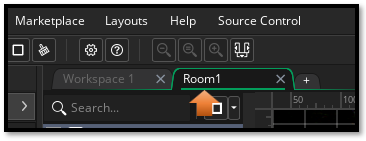
But, before wrestling with the creation of this creature, let’s take a look at the main dungeon floor that is being used inside of our maze. I tried adding brick or stone floors, but the elements did not show up very well, so I decided I need a darker floor. Yes, black marble isn’t exactly what is on most dungeon floors, but it just looked so cool, I decided to keep it.



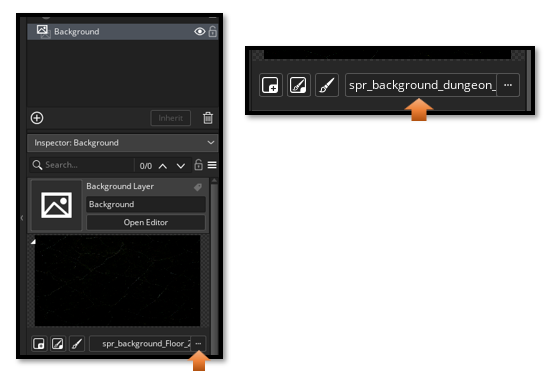
We need our floor to come in at a size of **1280 x 704**. This is really quite simple. Just create a sprite for it through the sprite panel, and then just assign the sprite to the room’s background layer. Make sure you have double clicked on Room 1 in the Asset Browser, if it is not already open.

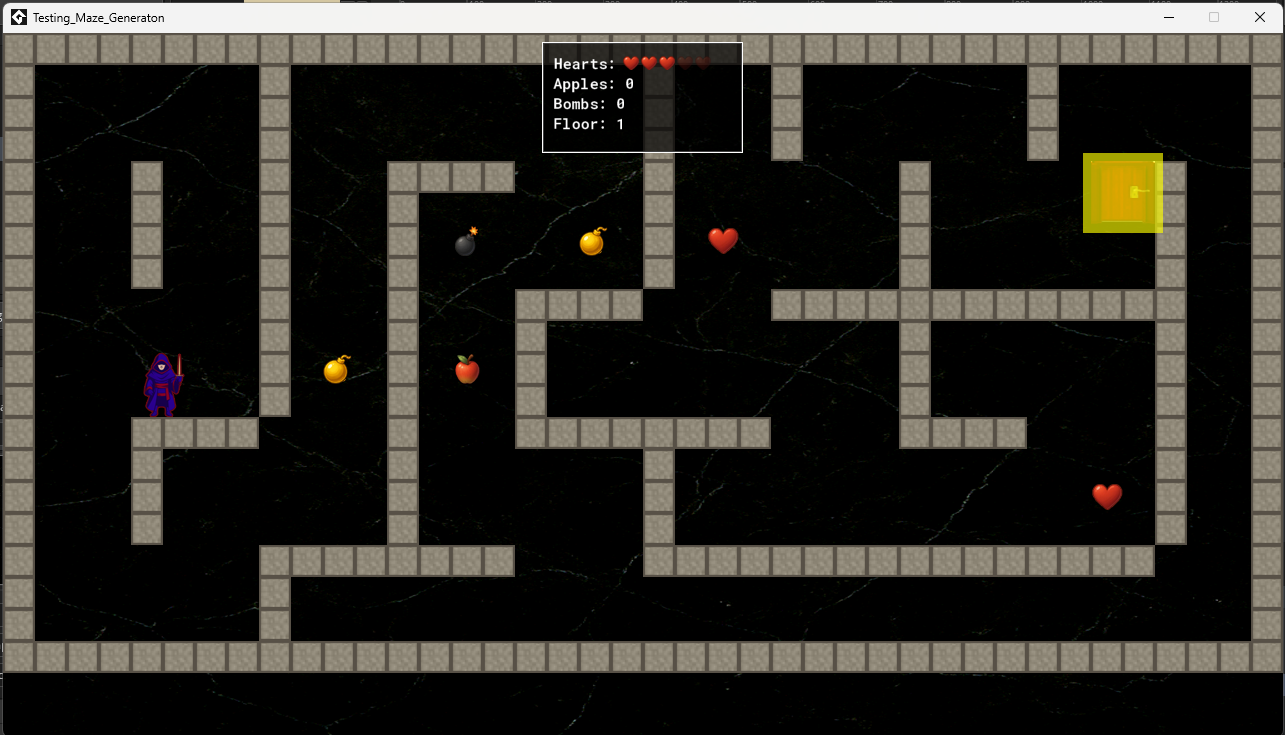


Or just click on the tab for the room at the top of the middle screen, if you are in the Workspace 1 tab.



This is so we can have access to Room 1 properties in the left hand panel. Click on the Background layer and then scroll down to find where you can bring your image for the floor in, Hit the three dots, browse to the sprite file for the background image, and just import it.





# Creating the Enemies

Folder Structure for the Sprites will be first. We have one main folder called Enemies, inside that folder, we have a folder for each type of enemy, then inside of each enemy folder, you need 4 folders each for each film strip.

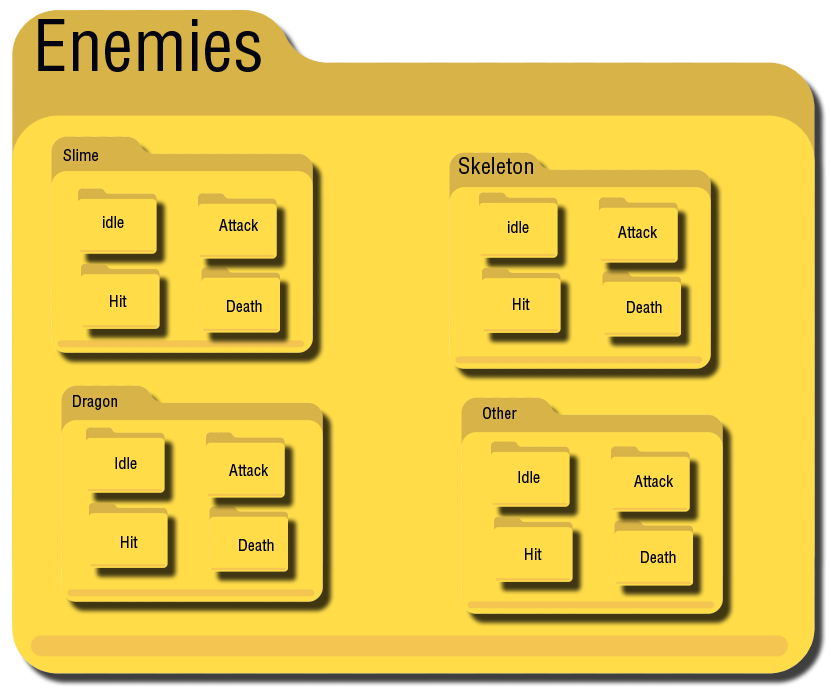
We need a different film strip for each phase of monster behavior.

Idle

Attack

Hit

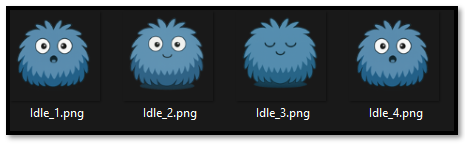
Death



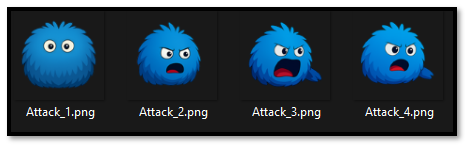
# Blue Furry Slime

This can get intense, so let us start with just a Blue little furry cute slime guy, to begin. The size of the blue furry slime guy is 128px.

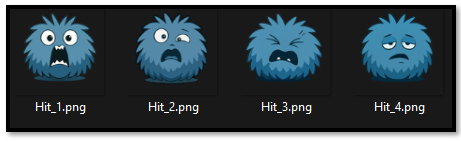
## Idle



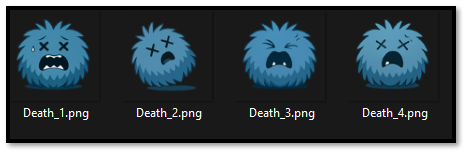
## Attack



## Hit

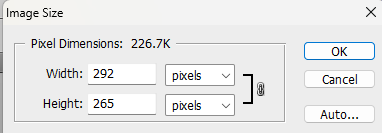


## Death

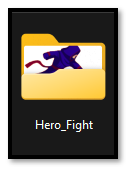


# The Hero

I started out with the hero being the same size as the slime, but realized that he was being drarfed by the enemy object, and I wanted my hero to be much more fierce looking in the battle, so the size of the hero ended up being 292 x 265. Since our fight will take place in a separate room, we are not restricted to the same rules in setting the sizes for our objects. We can set them to any size. In the dungeon, it is restricted by the code, and maze building, and walls, and such.



We will need to first create a new folder in the sprite folder for our hero.



## Idle

## 

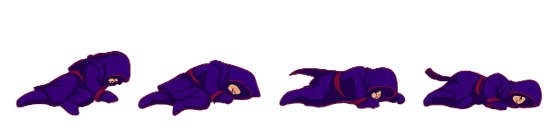
## Attack

## 

## Hit



## Death



# Creating a Special Fighting Room

As, I have already mentioned, we will be using this special fighting room that the characters in this battle can dual inside of. It will be a separate room from the room that holds the dungon itself. Every time the hero encounters an enemy, both of them will be sent into this room to work out their differences. So, the first thing that we need to do is to create the image for our room. We will create the fighting room at the same dimensions as the dungeon room size. So, our dimensions for this room will be **1280px X 704px**.

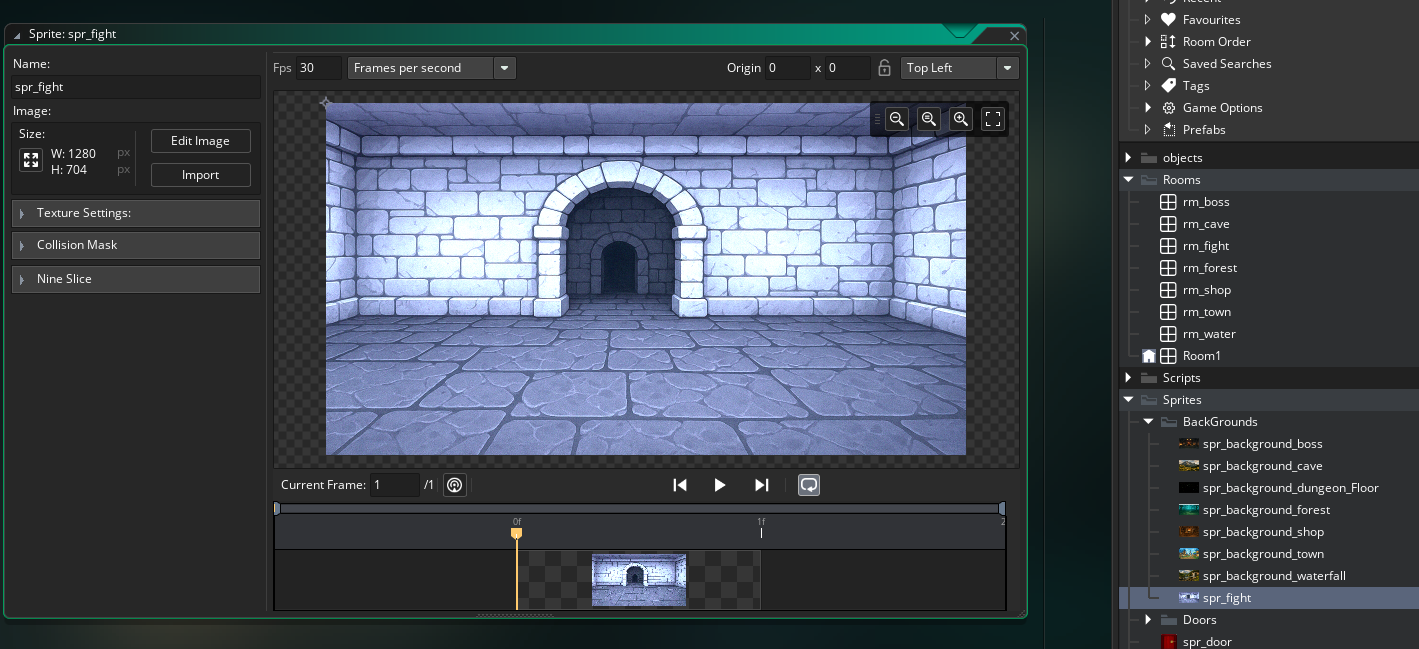


This is what we will be working toward, but we won’t really get there until we do a bit, (actually a whole lot) of coding first.



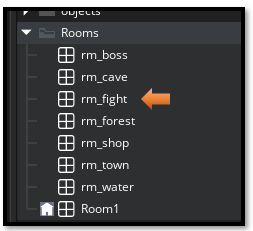
After creating the image at the right size, we will need to bring the background into the game. Now remember for a background to work, all we need to do is to create a sprite, we do not need to create an obj for it.

spr\_fight



Then we need to create the room itself. Go to the **Rooms** folder where we have all of our rooms. Name this room:

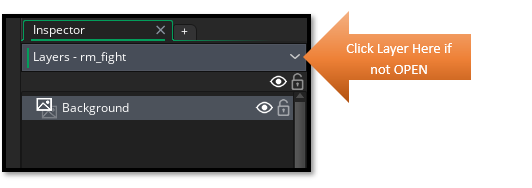
**rm\_fight**

****

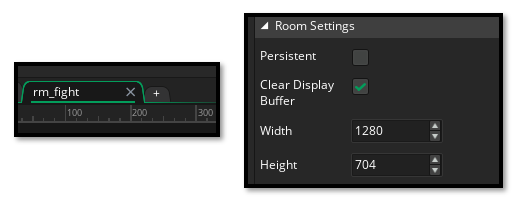
You must be in the **tab** for the **rm\_fight**, to see the properties for it.



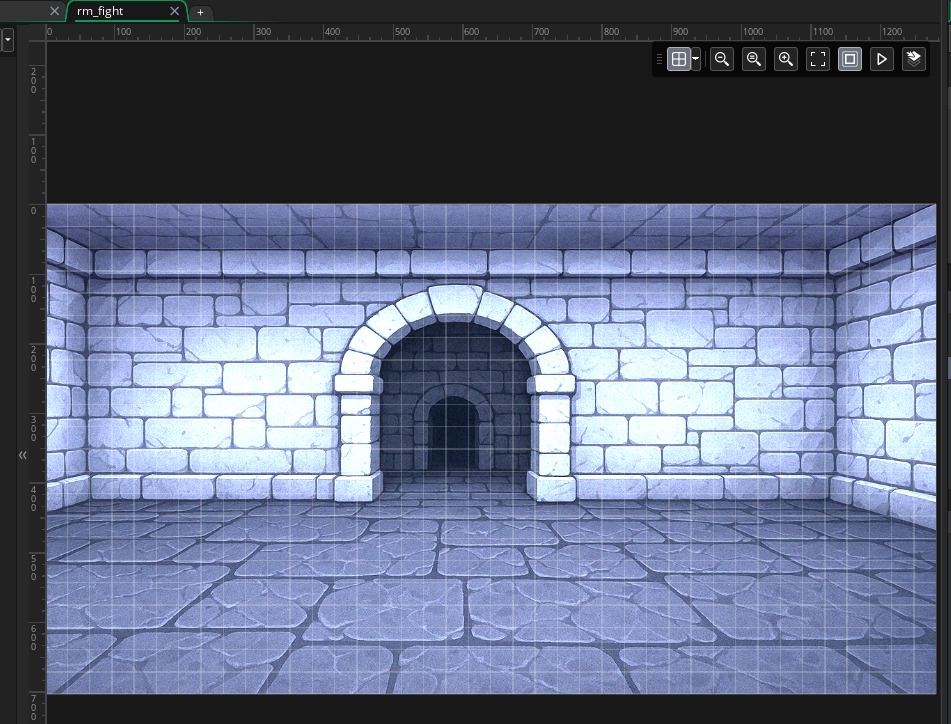
Go to the Left Panel, and click on layers if it is not open. There you will see the background Layer. If you do not see the Background layer, just like this, make sure you have actually clicked on the rm\_fight tab to display the room in the middle section of the workspace.



Double Check your dimensions for the room, it needs to be set to 1280px X 704px. You will need to click back on the rm\_fight tab at the top of the middle section to see the section to set the dimensions.



Your room background should now look like this.



# Creating Your Sprites to Work as a short movie

Since, we do not actually just want a static image during the fight scene, and we want the fighters to actually look like they are doing some sort of magic foot work, we will need to create our sprites just a bit differently. We need to create a sprite with multiple frames. This way our fighters will be animated.

1. **Create a Sprite with Multiple Frames**

* Go to **Sprites** → **Create Sprite**.
* Click **Import** and select all your PNGs (Idle\_1.png, Idle\_2.png, etc.).
* GameMaker will automatically add each image as a separate frame in the sprite.

Create the Enemy Sprites

spr\_enemy\_idle\_128

spr\_enemy\_attack\_128

spr\_enemy\_hit\_128

spr\_enemy\_death\_128

spr\_hero\_idle\_128

spr\_hero\_attack\_128

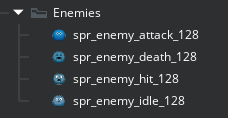
spr\_hero\_hit\_128

spr\_hero\_death\_128 (optional)

Remember to add the correct sprite multi-image sequence for each sprite. It is easy, just grab all the images when browsing for it and Game Maker will put them into a strip, instead of a single image.



Do this for all of the different states for the enemy fight scene.



# Create the Hero Sprites

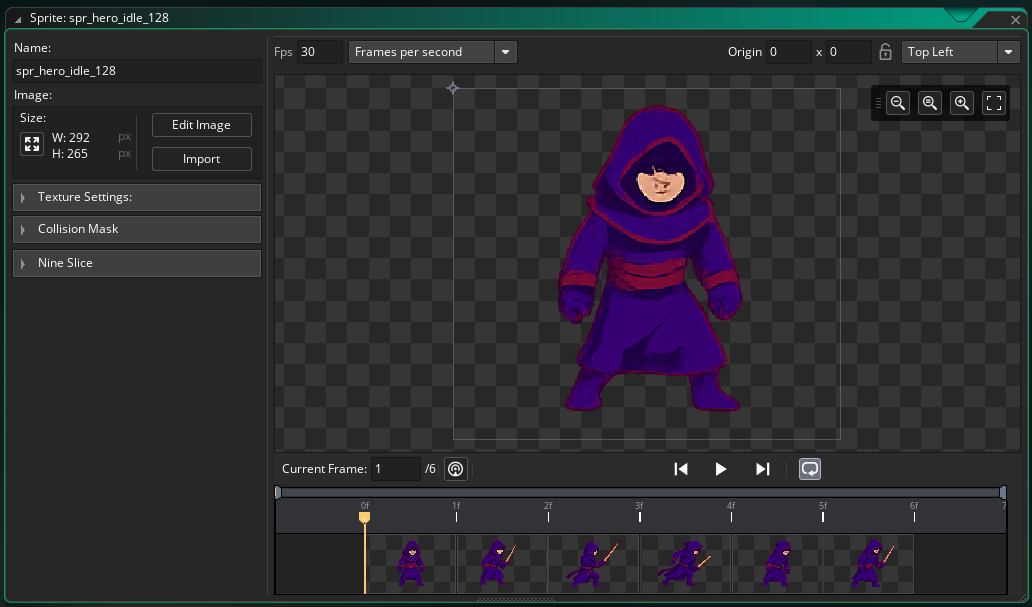
Keep the 128, at the end of the name, even though the hero is larger than 128, he will be using this exact name in the code, so just keep it like this.

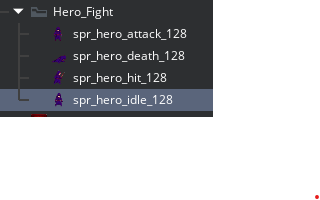
spr\_hero\_idle\_128

spr\_hero\_attack\_128

spr\_hero\_hit\_128

spr\_hero\_death\_128

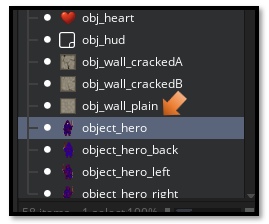




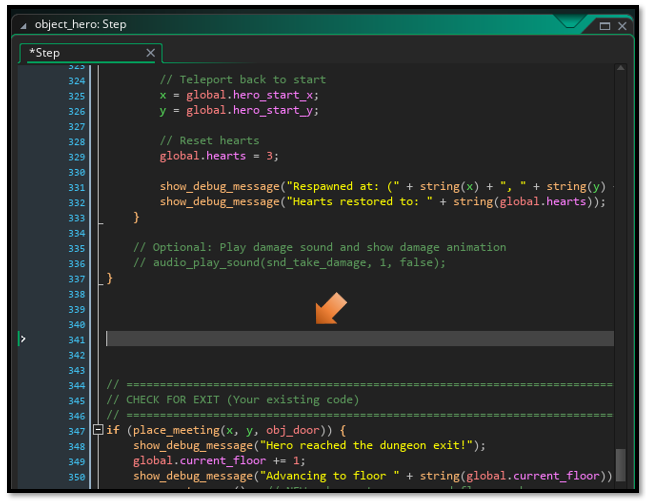
# Fixing our Hero Step Event

But before we can begin writing code for the each object in our fight scene, we need to do a bit of back tracking and make some coding changes to our Hero object.

Open the **object\_hero**



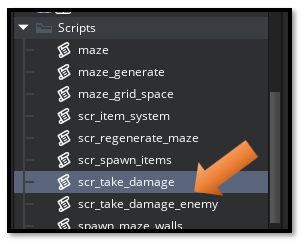
Double click on it to open up the Object code. Go into the step event. Right before the exit code,



**1**

Code for the Hero Step object. Go [here](Html_Code_Files/1_Hero_Object_Step_Event.html) for the code to write inside of the Hero Step Event.

# 2 Scripts



Name these two Scripts just like this:

scr\_take\_damage

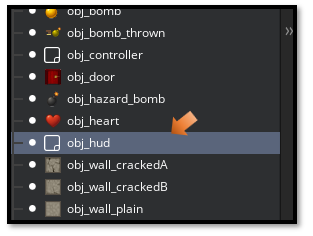
scr\_take\_damage\_enemy

**2**

Go [Here](Html_Code_Files/2_Scripts_for_Taking_Damage.html) to find the code for these two TAKING DAMAGE Scripts

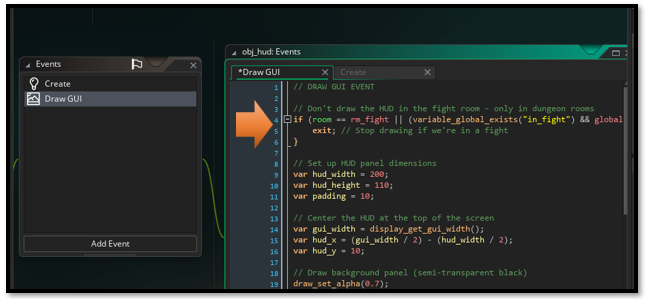
# Fixing the HUD Code

Open your **obj\_hud**. in order to get the score board in the fight room from showing up inside of the dungeon (which makes no sense, because you are not fighting any monsters there, and is just cluttering the maze) you will need to make some changes to the **obj\_hud** object. So, double click on it, to open it up and make changes to its code.



You should already have the **DRAW GUI EVENT** there, we will just make changes to it.

You are basically adding this top code to it.

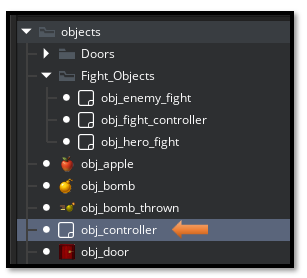


**3**

Go [Here](Html_Code_Files/3_The%20HUD_Draw_GUI_Event.html) for the Full code for the Draw GUI

# Getting the Bomb throwing action to still work in the Maze

Open the Main Controller, we need to set a variable in there. This is because we want our bomb throwing to still be active in the maze. We shut it off in the fight room, so that we could use the space bar to control the hero’s sword fighting instead.



To fix this, we basically only need to change one line of code. Add this one line of code to the bottom of the list of global variables, that we are setting in the **Create** event of **obj\_controller**.

global.in\_fight = false; **// Initialize fight state flag**

Go here for the full Code.

**4**

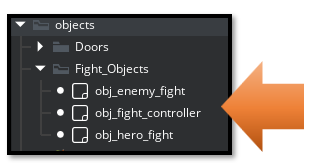
# The Code for each of the 3 Fight Objects

In this next part we will be dealing with writing the actual code for each of the 3 fight objects.

The **enemy fight object, the hero fight object, and the controller fight object**, which will control the code inside of this fight room.

# Creating the Objects

Ok, we have all of these sprites, but when we go about creating the objects, we will only need 3 objects for the fight room. Create a Group called Fight\_Objects to put them in, inside of the Asset Browser.



You can go to the next 3 hyperlinks to be able to see this code.

**5,6,7**

1. [obj\_enemy\_fight](Organized%20Script%20Files%20for%20Article/Enemy%20Fight/4%20Code%20for%20the%20Enemy%20Fight%20Object.docx)
2. [obj\_hero\_fight](Organized%20Script%20Files%20for%20Article/Enemy%20Fight/5%20Code%20for%20the%20Hero%20Fight%20Object.docx)
3. [obj\_fight\_controller](Organized%20Script%20Files%20for%20Article/Enemy%20Fight/6%20Code%20for%20the%20Controller%20Fight%20Object.docx)

Each of these objects will have 3 events.

1. Create
2. Step
3. Draw

I know this enemy system, for our furry slime, consisted of a lot of code. Just take your time with it, and when you get done. Test your game to make sure that it works.

And so this brings us to the end of this chapter. Next, we want to try and get our blue furry slime in the maze to be a bit smaller, and perhaps even have a bit of a bouncing action as he moves around the halls, in search of attacking his favorite hero guy.