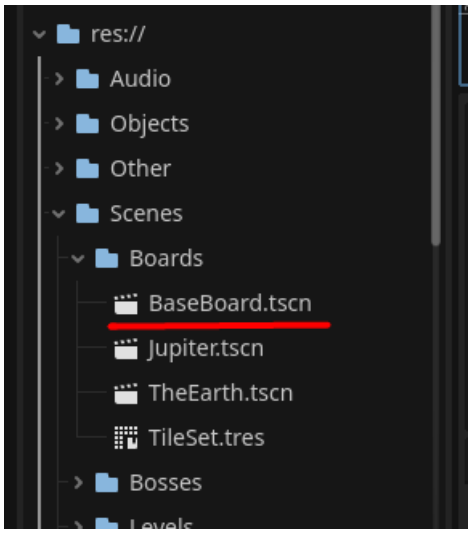
Contents

Creating the scene file

First, let's create a new scene for our new board.

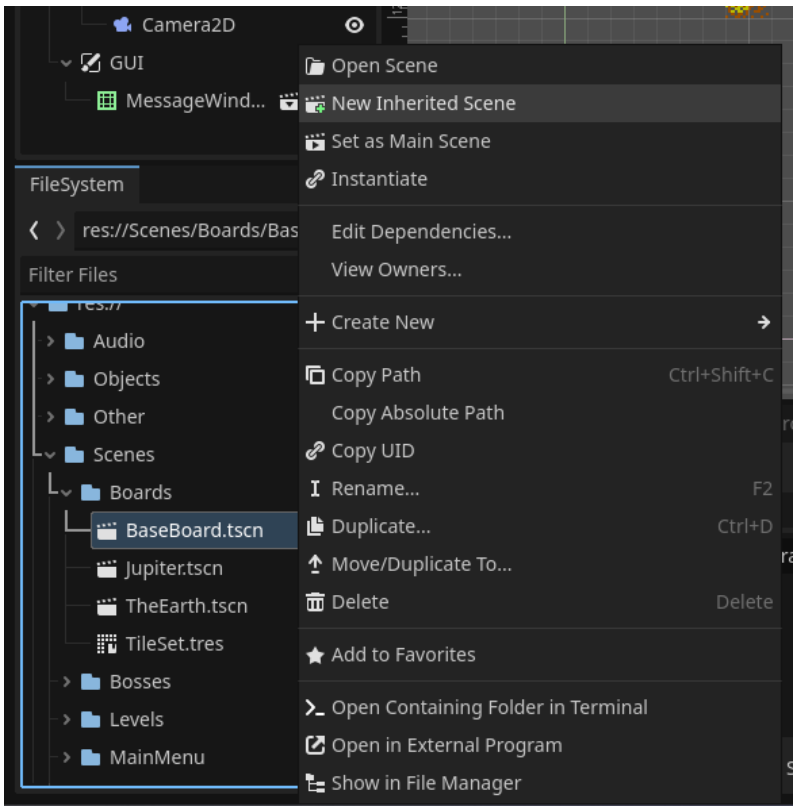
Find the base board scene file, "Scenes/Boards/BaseBoard.tscn"



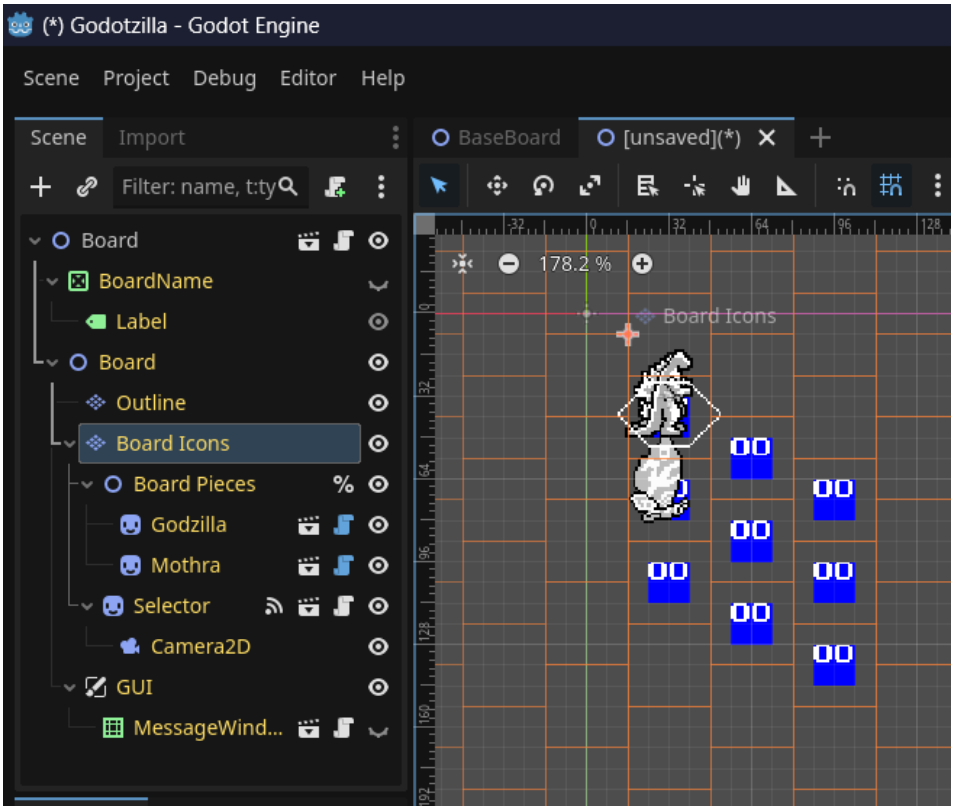
The base board scene contains objects that all boards require, such as board's name, board pieces, message window, etc.

We won't make a copy of it but rather make a new inherited scene from it! It makes it so no object duplication occurs and when we make new changes to the base board (for example, we want a new game mechanic for all boards), these changes will automatically appear in other scenes! Awesome!

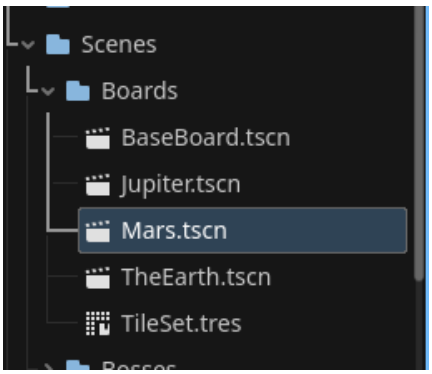
So how do we create an inherited scene? Right click on "BaseBoard.tscn" file and click on "New Inherited Scene".



This will create a new tab in the editor and make all the node names appear yellow

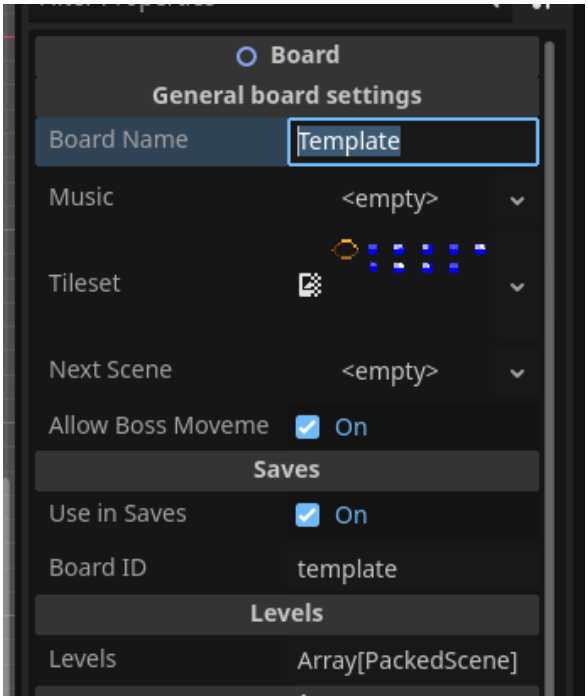


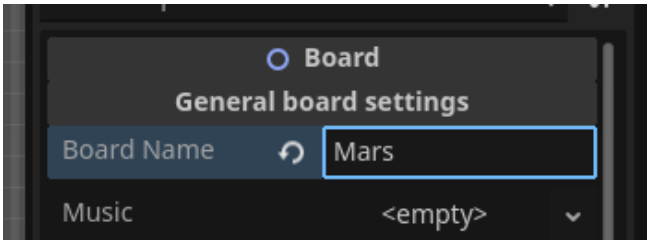
Yellow node names means that node is inherited from the base scene, in our case, BaseBoard.tscn. Let's save this newly created scene, for example, to "Mars.tscn".



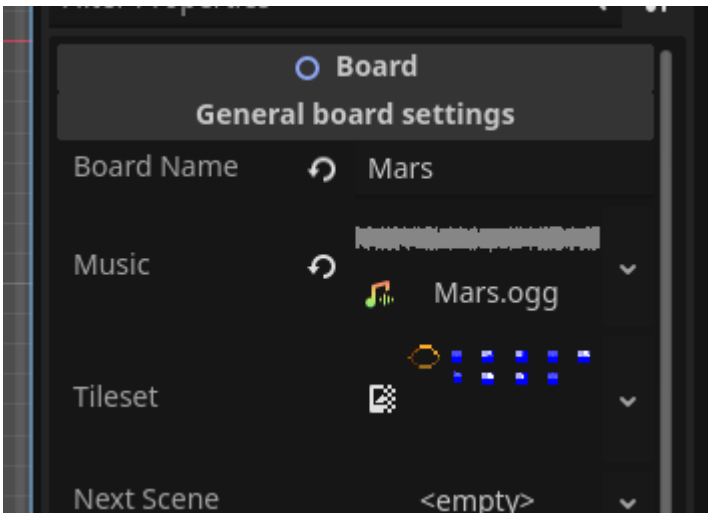
Changing common properties and creating assets

Great, now let's do some customizations. First, it's very easy to change the board's name in-game, you can just click on the root node "Board" and change the name in its properties:

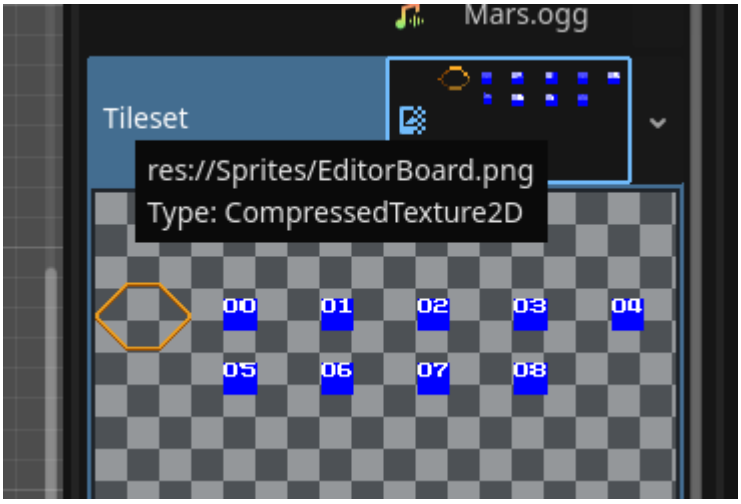




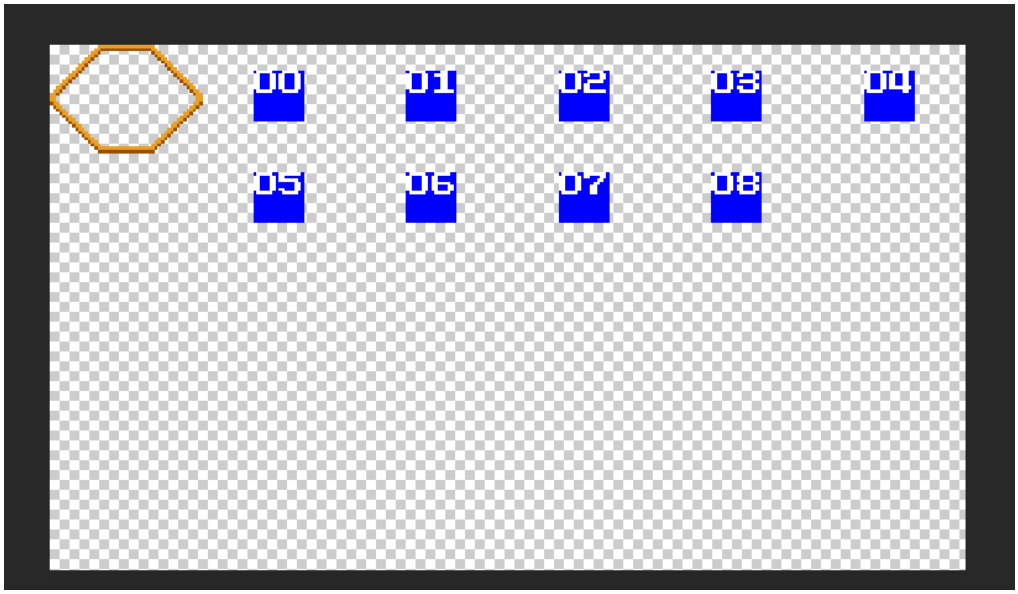
You can notice that the music property says "<empty>", which means there will be no music playing if we tested the board. Let's drag and drop some music file into that property!



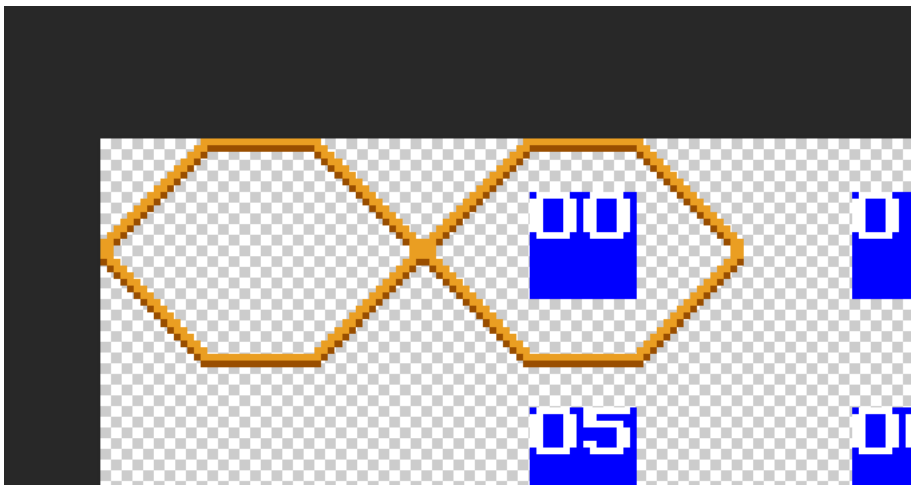
Great, now let's look at the tileset image:



We can see that it contains the board's placeholder icons and the outline, let's create our own image.



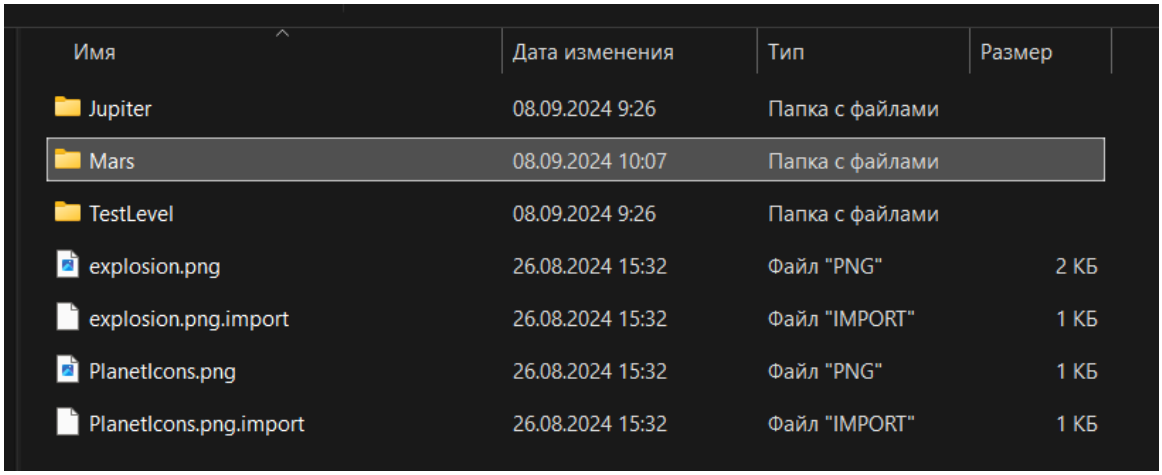
You can see how the icons would look like on the board by copying the outline sprite and putting it underneath the icons like this:



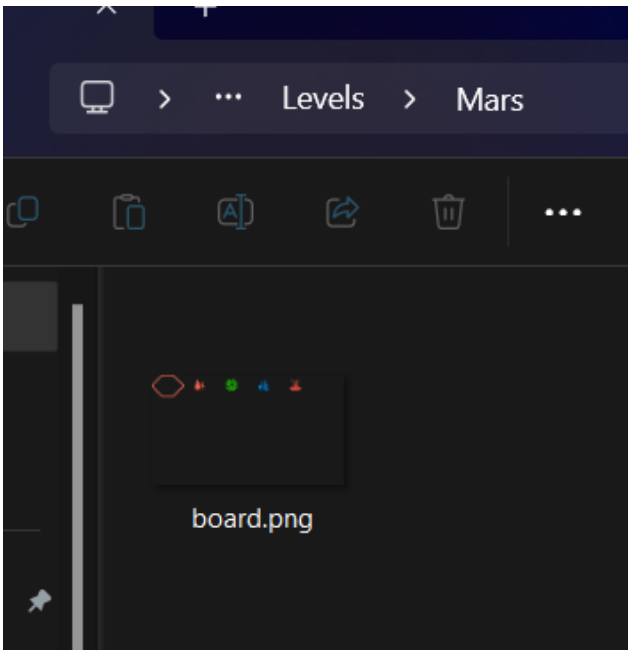
So… Let your imagination have some fun, I guess! :D

I used the test level's board sprites as a base. Yeah… Maybe I'm not that good at imagining things lol. But this should work for a tutorial!

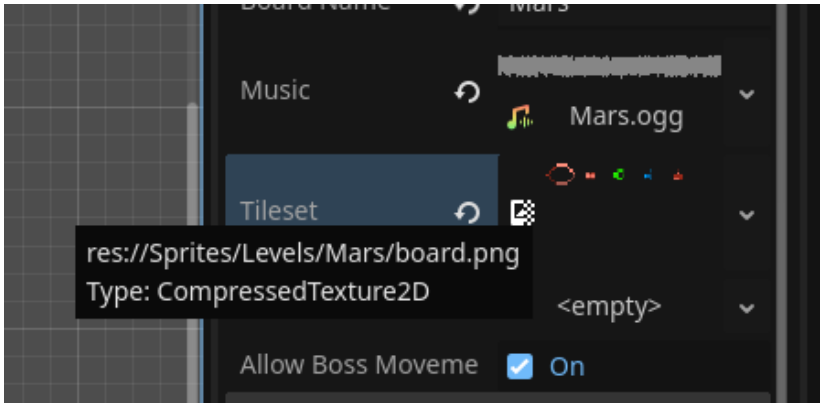
Now let's create a new folder for files for our board (and the future levels that will be on that board, but that's going to be in another tutorial). So the path for the test board's sprites is "Sprites/Levels/TestLevel", we can now create our own folder in "Sprites/Levels" folder.



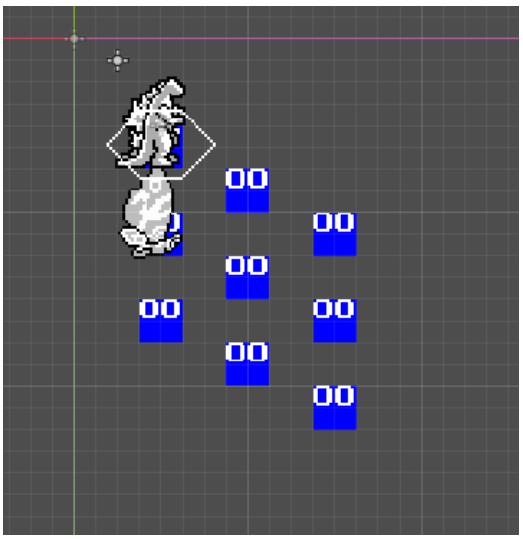
Now let's save our board sprites there.



Nice, now let's make the game load this file. Drag and drop this file into the tileset property.



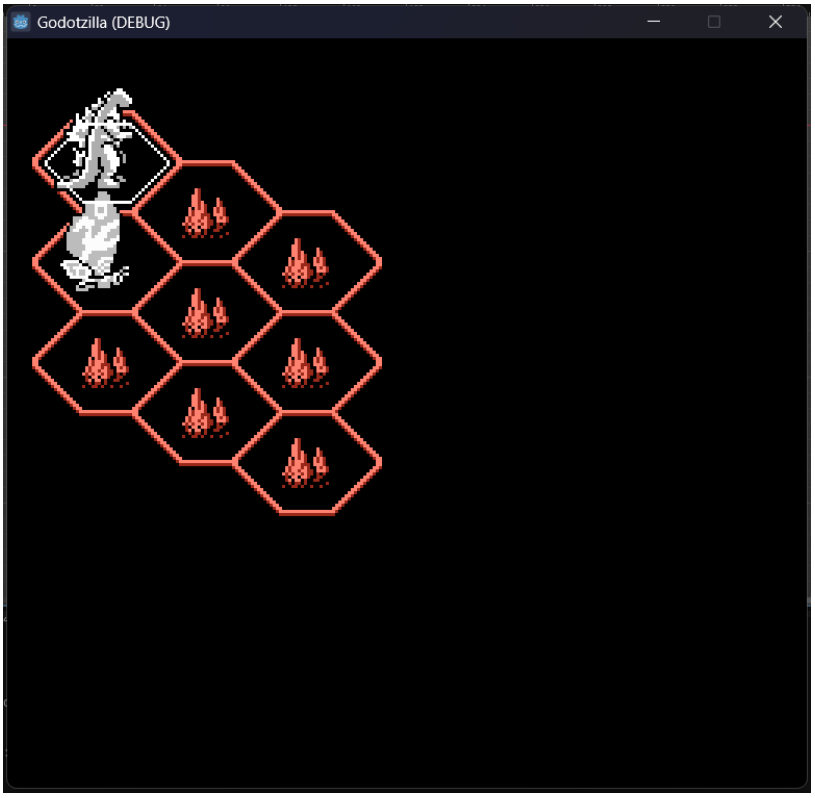
But you can notice that the icons in the editor did not change:



Don't worry, the sprites only change in-game and their placeholders are displayed in the editor (Believe me, I wanted a better way to make this work, to make the

selected tileset work in-editor, but there were numerous bugs that happened with that approach, for example, when I set the tileset image on one board, in the base board the tileset image was just gone, for some reason).

Now we're ready to test the board in-game! In the "How to create a new character" tutorial there is a section on how to change the first scene that opens up when you play the game, you can use that section and change the first scene to the new board.

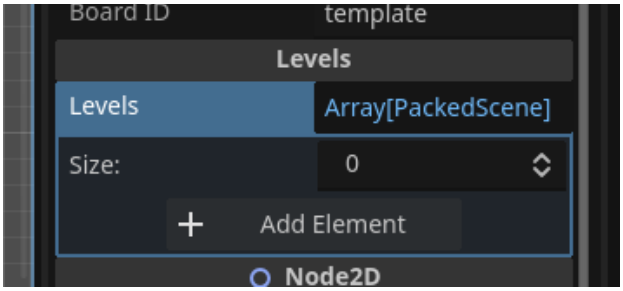


Our new tileset indeed showed up, not the placeholder images!

But when you try to play the levels with the characters, the levels are immediately skipped, that's because we haven't configured the levels array yet.

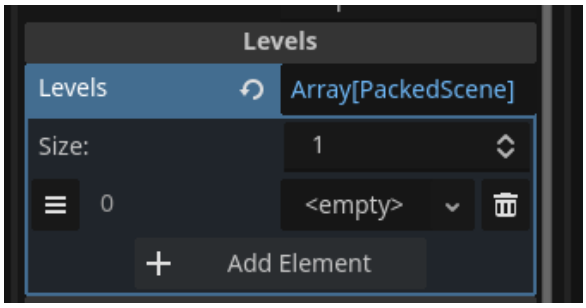
Adding levels to the board

In the board's properties you might have noticed the section about levels:

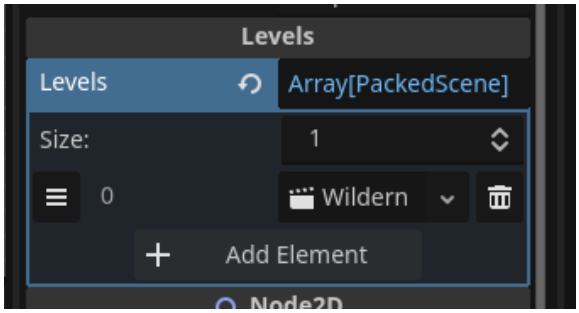


But it's empty, that's why the levels were skipped: there are no levels on the board.

So let's add some! Click on the "Add Element" button.

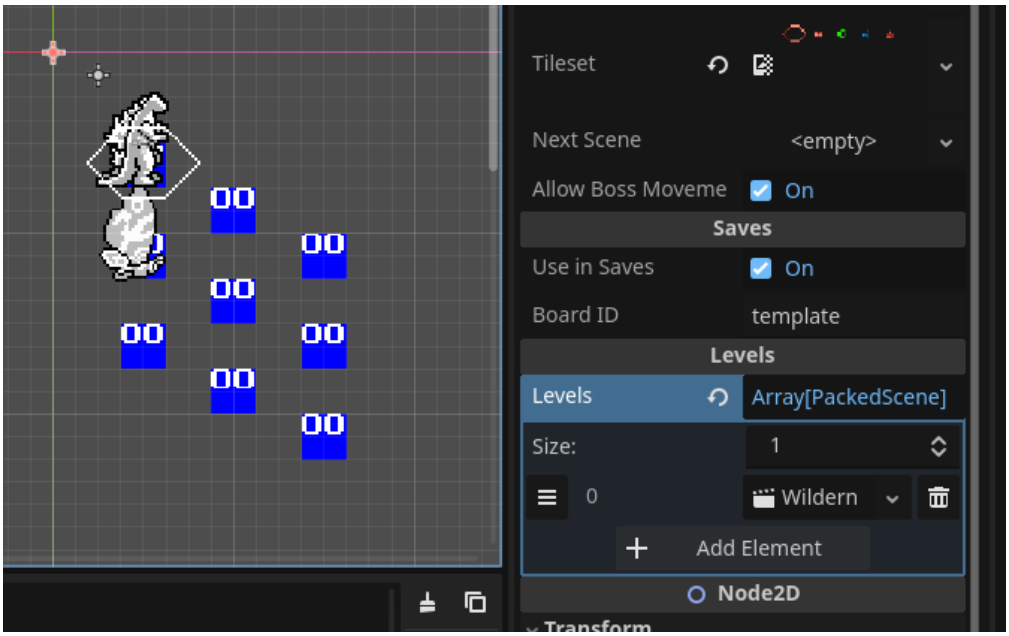


A new entry appeared, you can drag and drop a level into it, for example, you can use the default "Scenes/Levels/Wilderness.tscn" level.



Also notice how it shows "0" instead of "1" as the first level, that's because in most programming languages (GDScript included) the first element in the array has index 0, the second element has index 1, etc. Think about it like this: we have this array in memory, we grab it's beginning, and how many steps should we make until we get into the first element? 0! The second element? Move to the right just once, so 1! (I hope it makes sense)

And that's also why the placeholder images on the board show "0", they will contain that first level in the array.

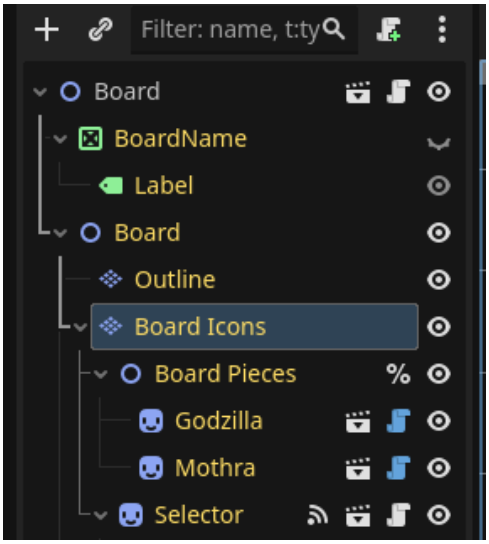


Try to play the game again and you will see that it indeed works.

Changing the board layout

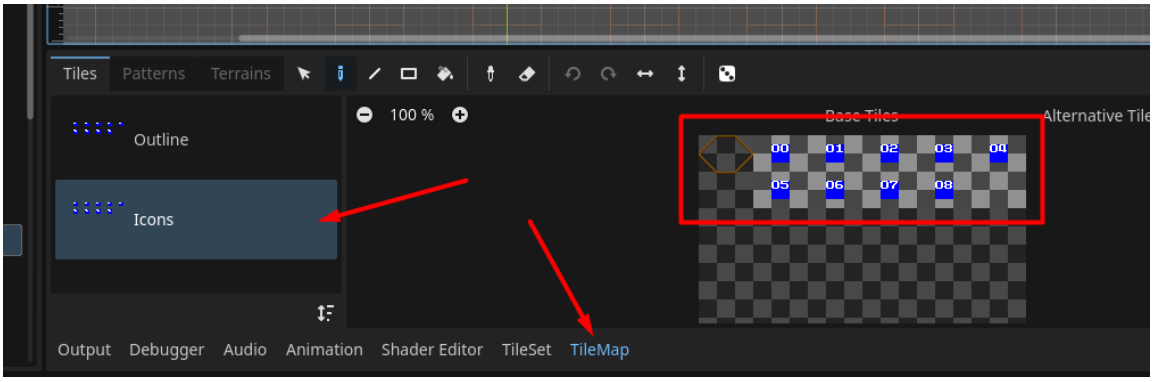
So how do we actually change the board layout?

Notice this particular node in the scene:

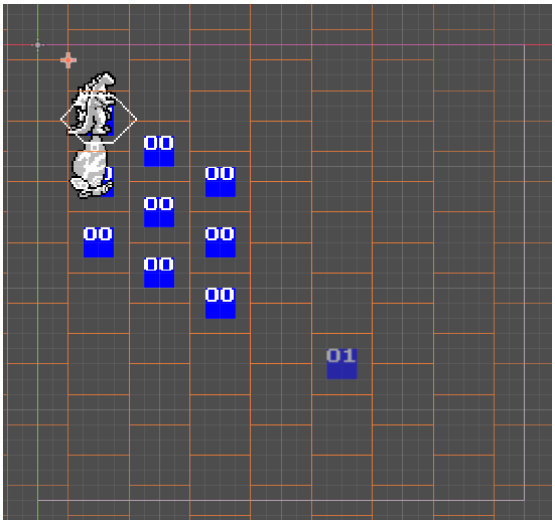


That's where our layout is! (You can make sure it's the correct node by temporarily making it invisible with the eye button to the right of the node's name)

Now that we selected the node, we can now select the tiles we need:

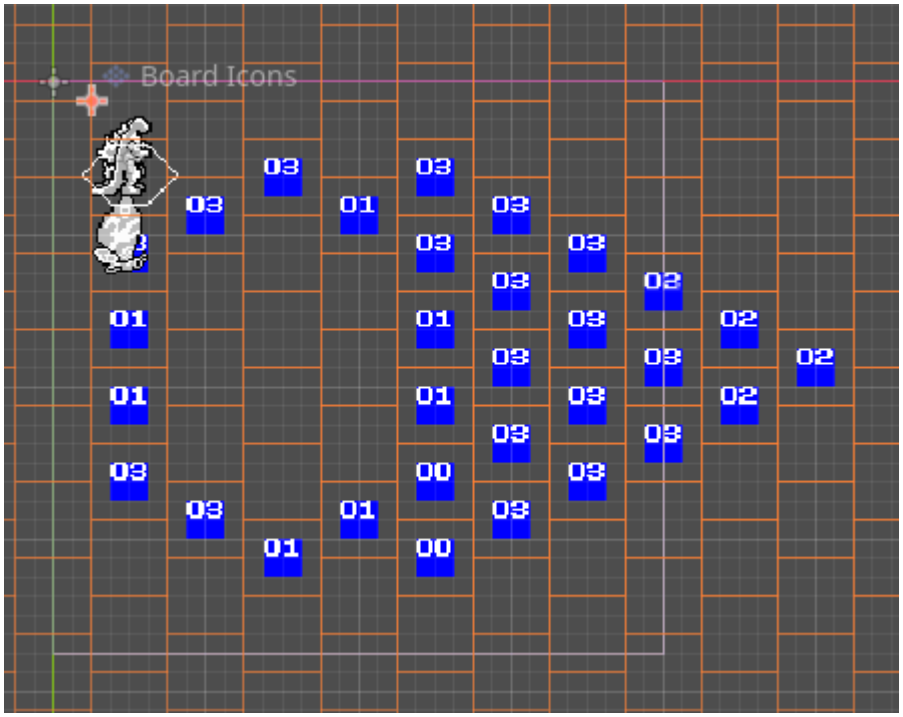


For example, select the tile "01" and move the cursor to the board.

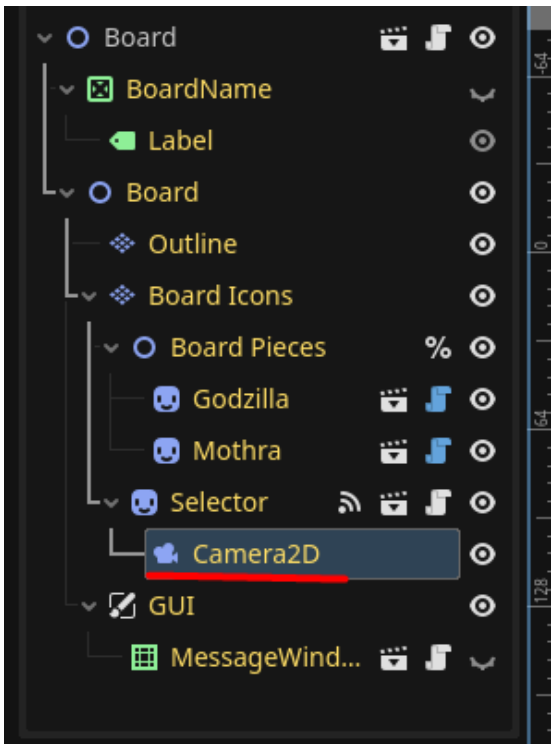


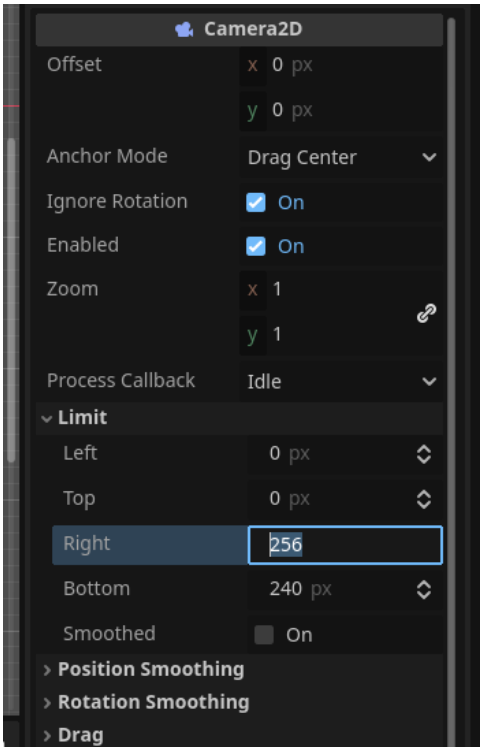
We can now start building our layout! You can also select any other tile with that method, but sometimes won't show up on the board (the ones that don't exist on your tileset you selected in the board's properties).

You can also remove the icons by holding the right mouse button.



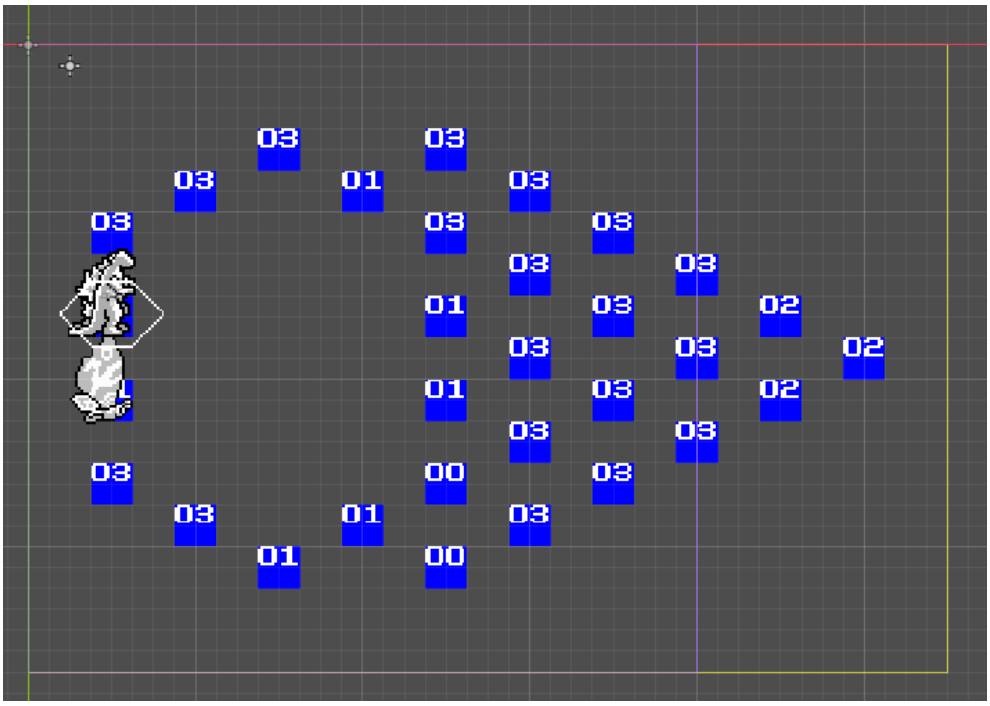
Nice! But when you try it in-game half of the board is inaccessible, that's because there is a camera limit, you can change it this way:





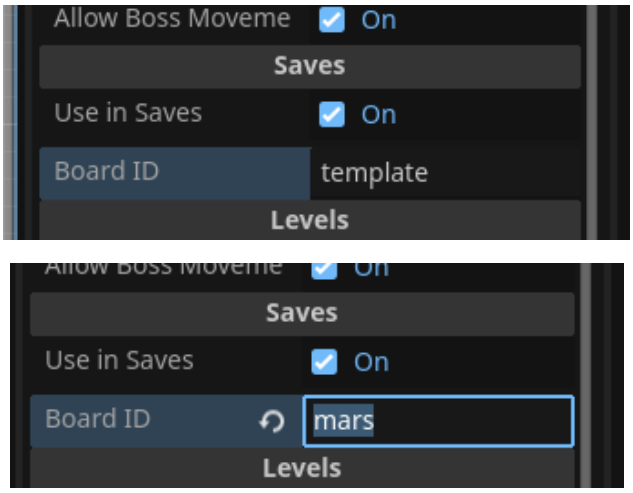
Now it should work fine in-game.

And of course, you can move the character objects on the board (“board pieces”) and the selector to properly position the characters.

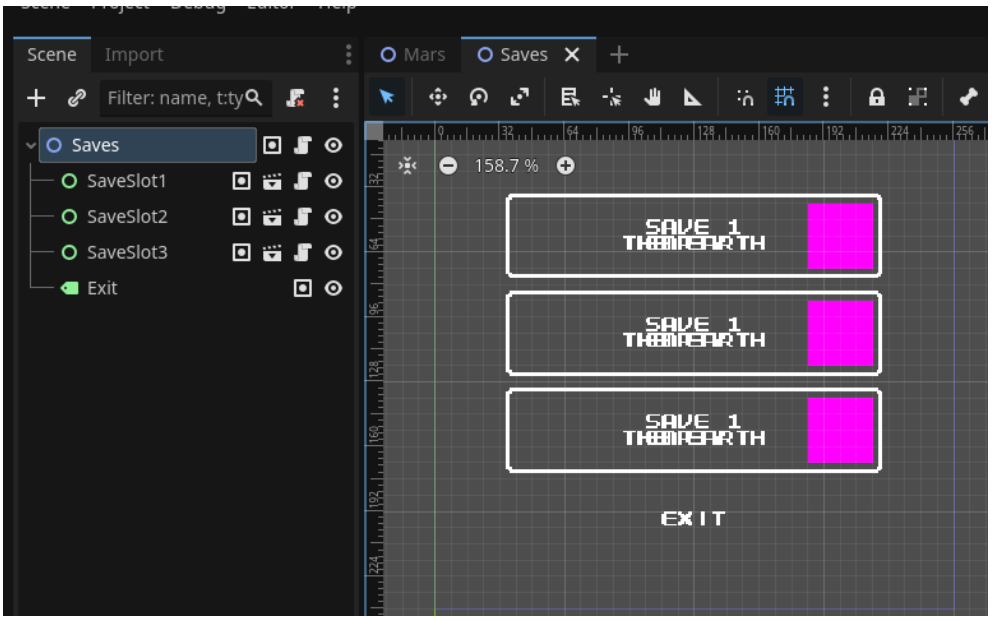


Using the board in the game saves

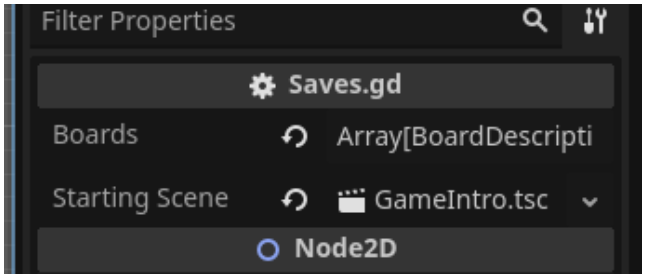
First of all, let’s change the board’s ID for saves:



Nice, now let’s open the save menu, which is located in “Scenes/MainMenu/Saves.tscn”

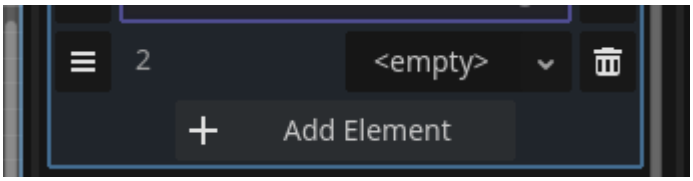


By opening the root node's properties you can see the "Boards" property:



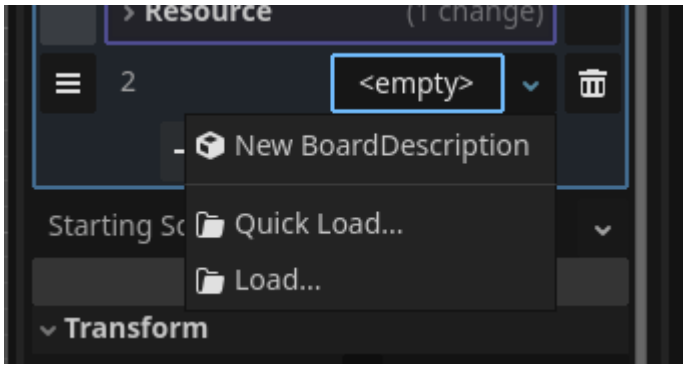
You can open it and see every board available in the saves menu:

Let’s add our own board here!

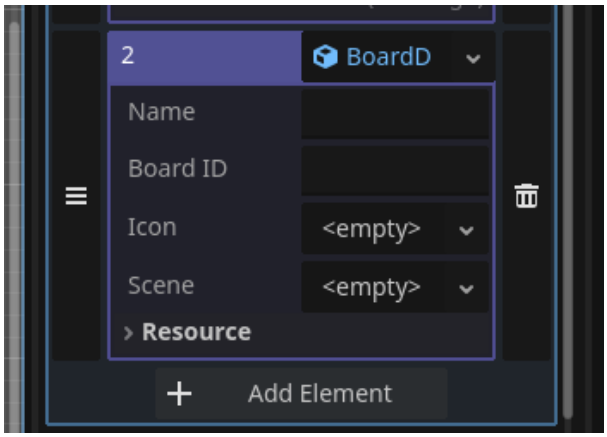


Okay, now what?

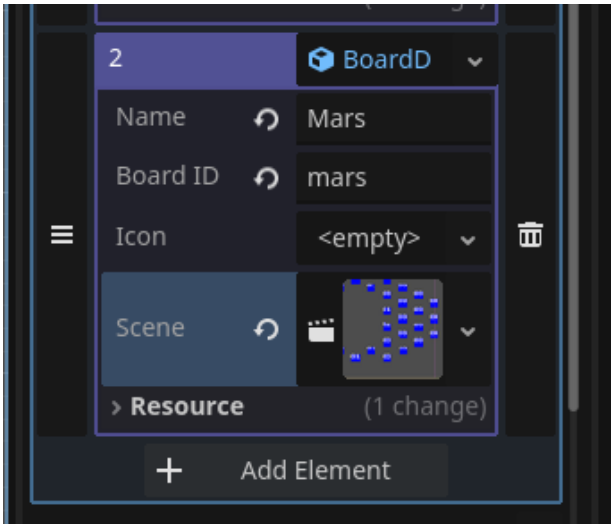
Click on “<empty>”.



Now on “New BoardDescription”.

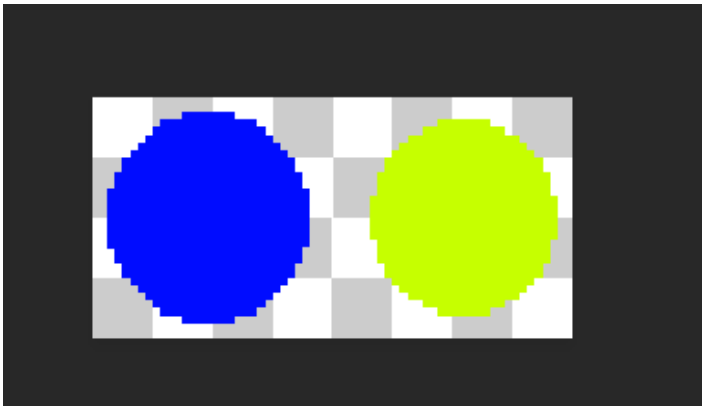


Now we can change the properties.

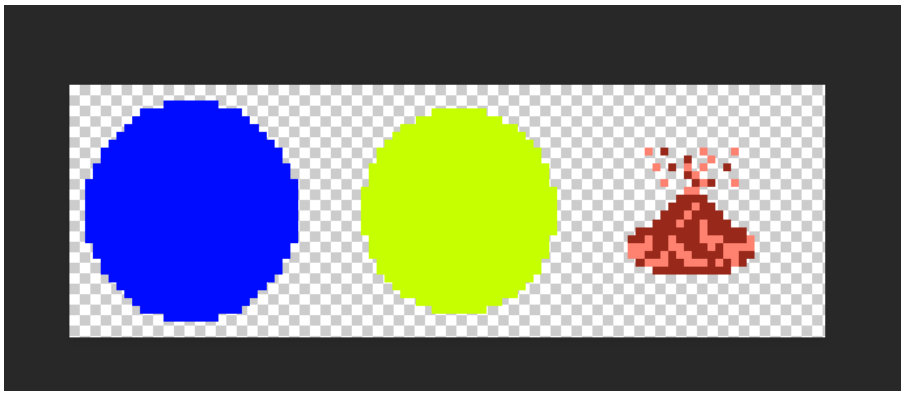


But I left the icon property empty, that’s because we haven’t created an icon yet.

Open “Sprites/Levels/PlanetIcons.png”.



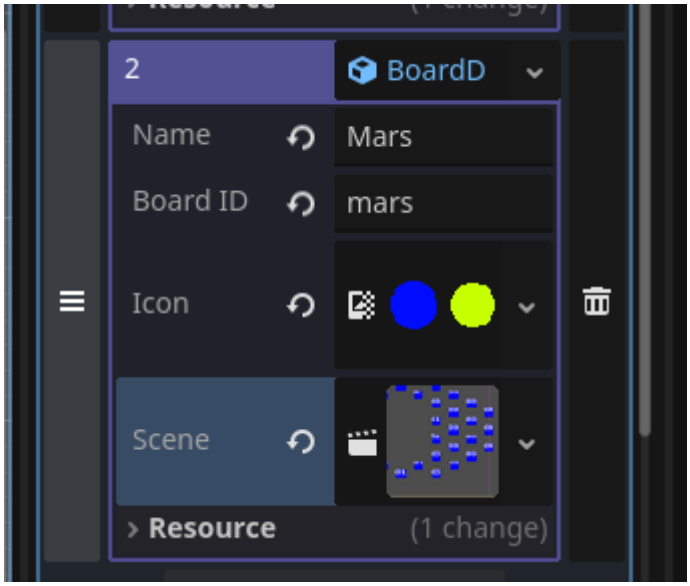
Now you can extend it and add your own icon.



Good enough for a tutorial.

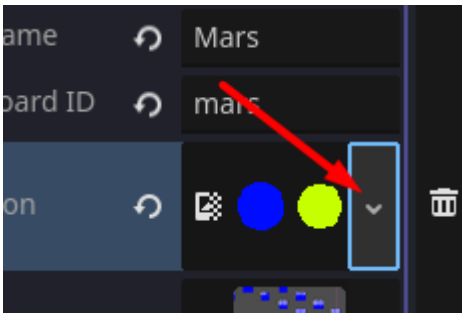
Now let’s add it to the game.

You can notice that it doesn’t look well if you just drag and drop the image in:

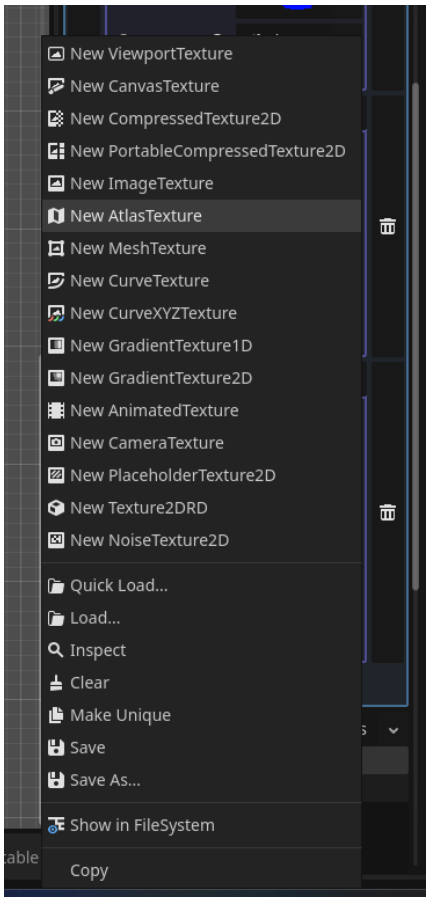


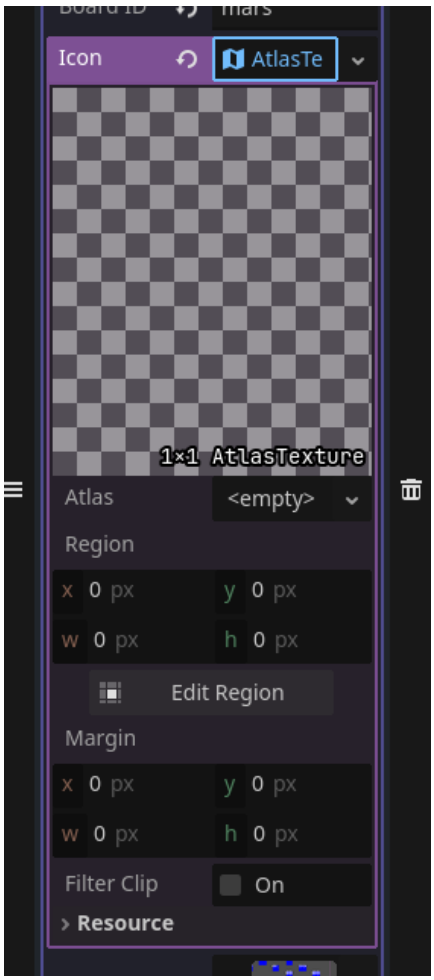
What do we do?

Click on this button:



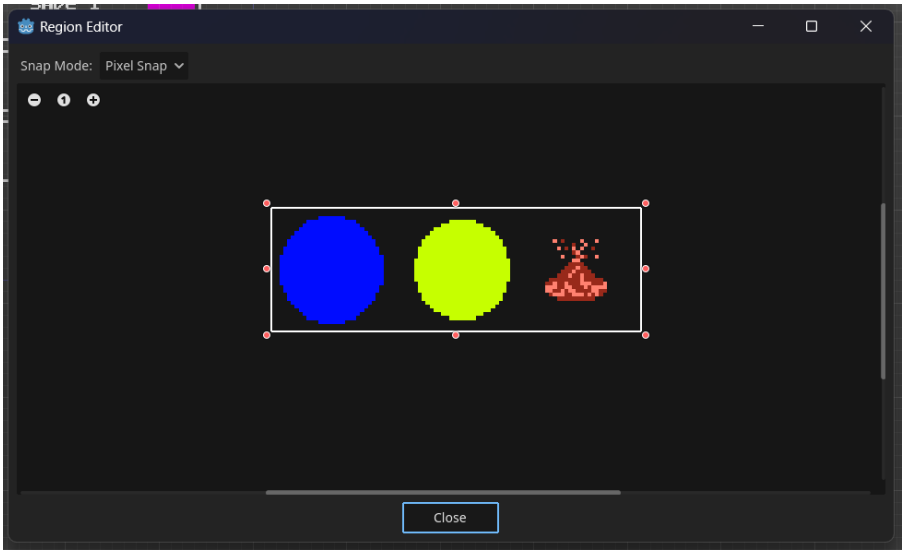
And then “New AtlasTexture”:



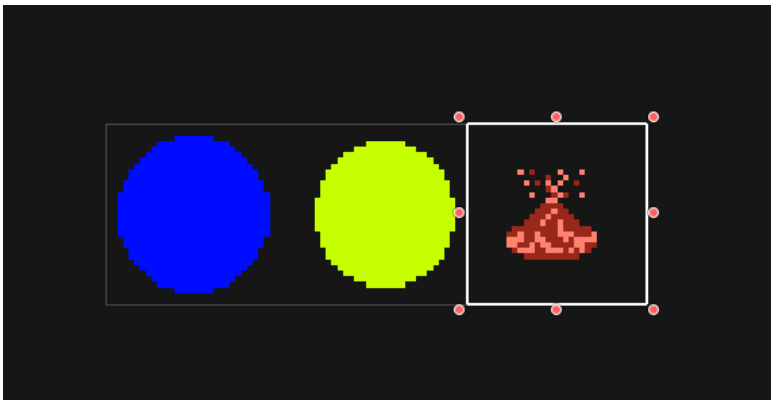


Drag and drop your icons spritesheet in the “Atlas” property:

Now click on “Edit Region” below.



Now you can select the new icon.



Now close it and it’s done!