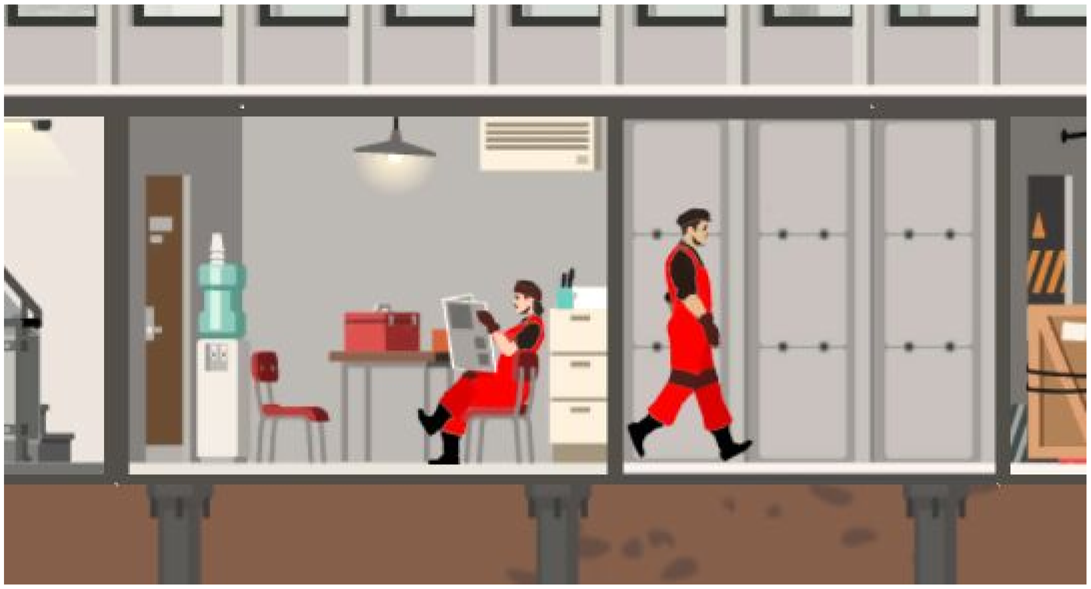
PROJECT HIGHRISE™ by SomaSim

Project Highrise Modding, Part 2: Layered Decorations

Welcome back! Now that you've gone through Part 1 and created your own basic decoration, we'll learn how to make more complicated ones that contain several different images in different layers.

Step 1. Introduction to Layers

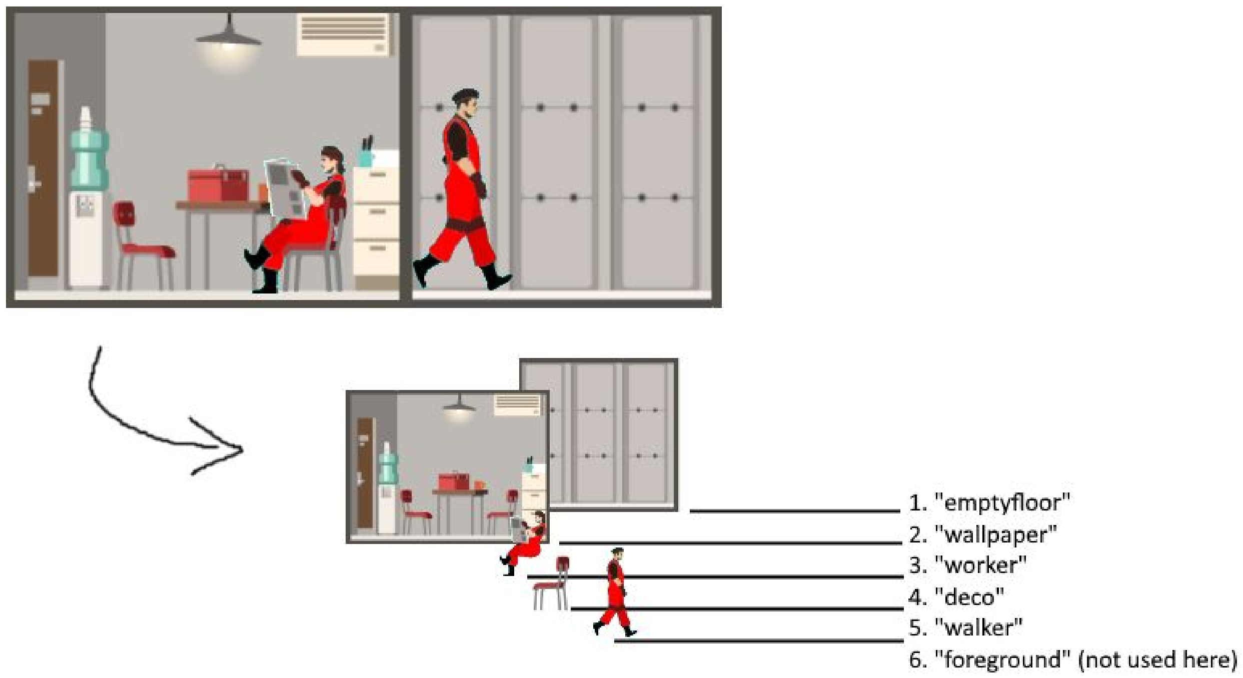
First, let's talk about layers. The building view you see in the game is composed of many layers, for example if we look at this screenshot:



… we can see underground dirt in the background, then on the right is some empty floor space, then a construction office to the left, finally people sitting or walking inside this space.

All of these elements are separate visual objects, and they can live on different layers.

Here's an illustration of how that image comes together:



For the purpose of modding, we're going to ignore the dirt and pipes underground, and only focus on six layers used inside the building. They are, in order:

1. **emptyfloor** - this is the layer for empty floor tiles, furthest in the background
2. **wallpaper** - this is the layer for the wall backdrop of an office, restaurant, etc; typically this image will include both the wall coloring and background decorations (pictures on the wall, filing cabinets, etc) already baked in to form a single image
3. **worker** - this is the layer where people are placed while they're working, for example when they're sitting behind a desk, or standing behind the counter (or in this illustration, just reading a newspaper)
4. **deco** - this is the layer for decorations that separate workers from people just walking by in front of them; if you have desks or counters, they go into this layer
5. **walker** - this is the layer for people walking from one place to another, or standing in line at a restaurant or retail shop
6. **foreground** - this layer is in front of everything else, it's useful for elements like door frames, but we're going to skip it in this example for the sake of simplicity :)

As you can see, these images are layered on top of each other, from the **emptyfloor** layer furthest in the back, to the **foreground** one up front.

Ok, now that we know about layers, let's make ourselves a deco object that uses them to create an illusion of depth.

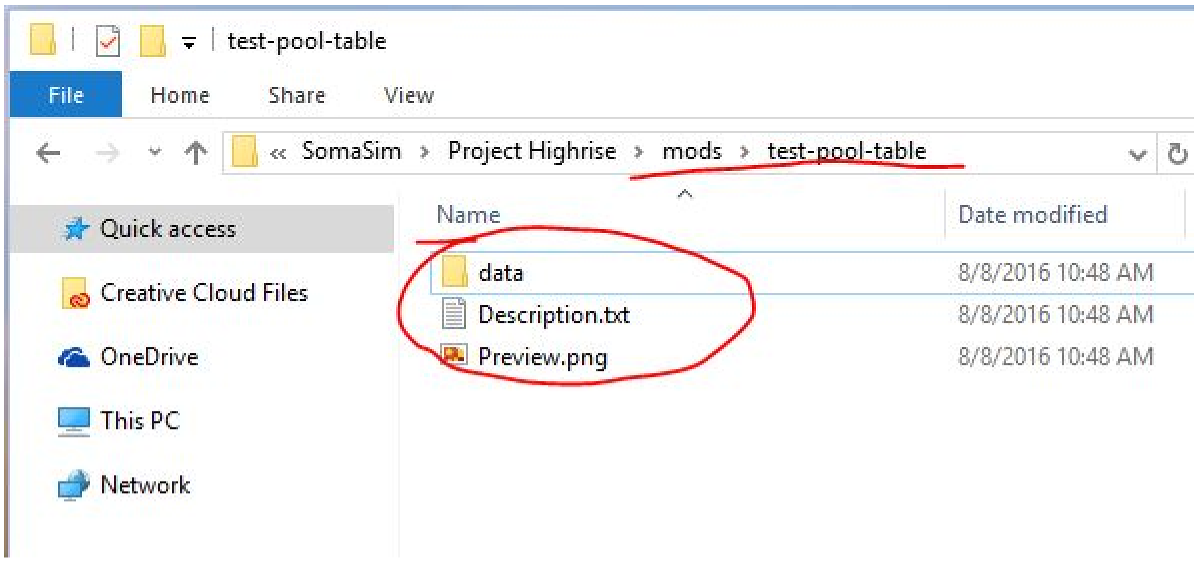
<u>Step 2. Definition</u>

We're going to create a pool table decoration composed of two objects:

1. A pool table that stands in the foreground
2. Also, a cue rack behind the table on the wall

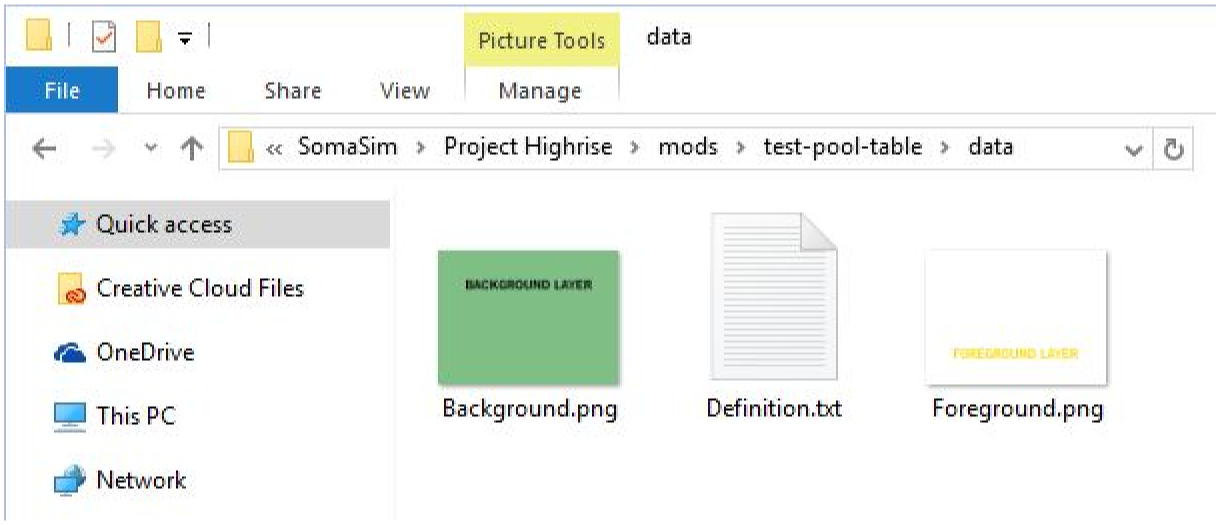
Because of this layering, we'll make it look like people walking back and forth will walk *behind* the pool table but *in front* of the cue rack. This deco is going to be larger than the plant - it's going to be 4 tiles wide.

First, let's create a directory for this mod as we did before. We'll call it **test-pool-table**. Second, download this template for layered decos:
https://storage.googleapis.com/highrise-modding/templates/deco-layered-4-tiles-wide.zip
... and unzip it inside your new folder. In the end, you should have a mod folder that looks like this:



Name	Date modified
data	8/8/2016 10:48 AM
Description.txt	8/8/2016 10:48 AM
Preview.png	8/8/2016 10:48 AM

Inside the **data** folder you will notice something is different. Instead of a single **Deco.png**, we have two images, **Background.png** and **Foreground.png**:



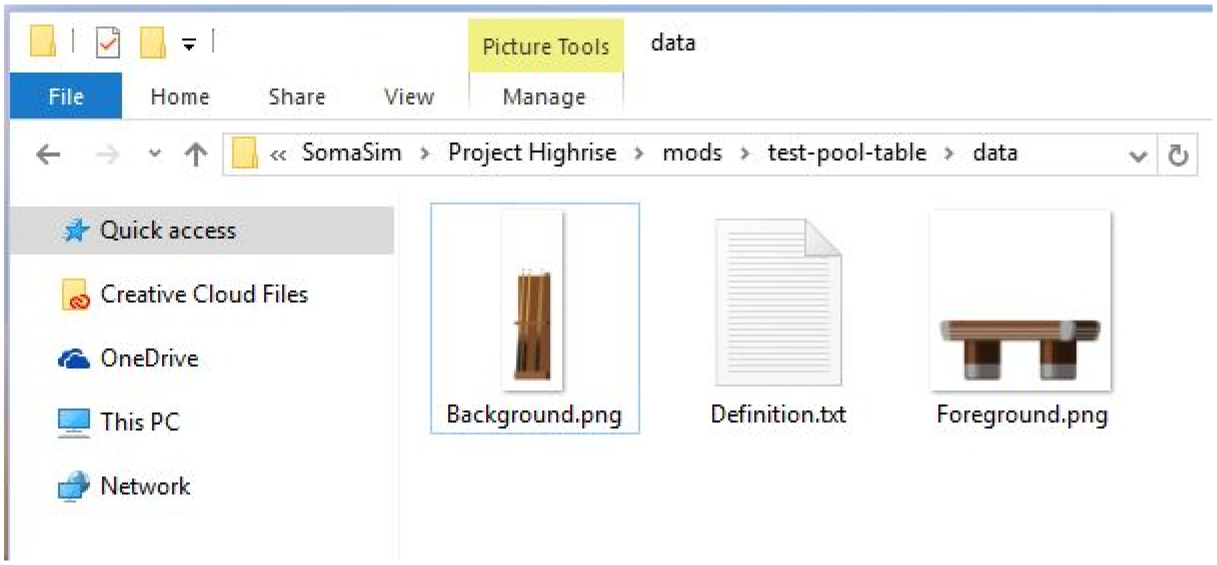
These are *placeholder* images. Since the deco is 4 tiles wide and 1 tile tall, the images are both 400px wide and 300px tall (because each tile is 100 x 300 px).

Instead of using these, let's get an actual pool table, and a pool rack, from the sample modding assets archive. (If you remember from the first tutorial, it lives here: https://storage.googleapis.com/highrise-modding/project-highrise-modding-props.zip)

Inside that zip file, find the following files and use them to replace the sample ones:

- **misc / Prop_Recreational_PoolTable.png → Foreground.png**
- **misc / Prop_Recreational_QstickRack.png → Background.png**

… to get something like this:

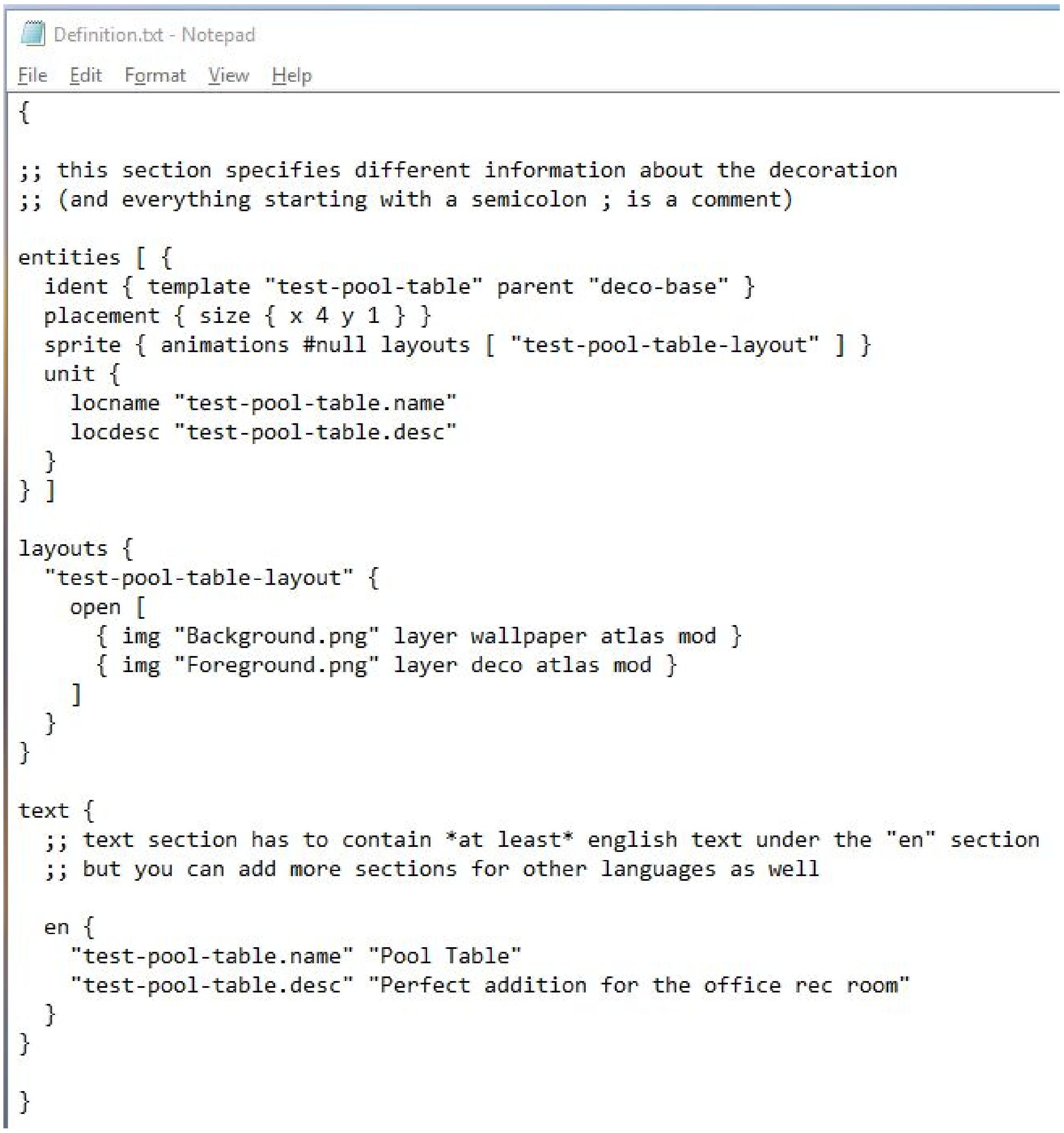


By the way, did you notice that the pool cue rack is not 400 px wide? We'll deal with that a little bit later.

Now open the **Definition.txt** file and do the following, just like in the first tutorial:

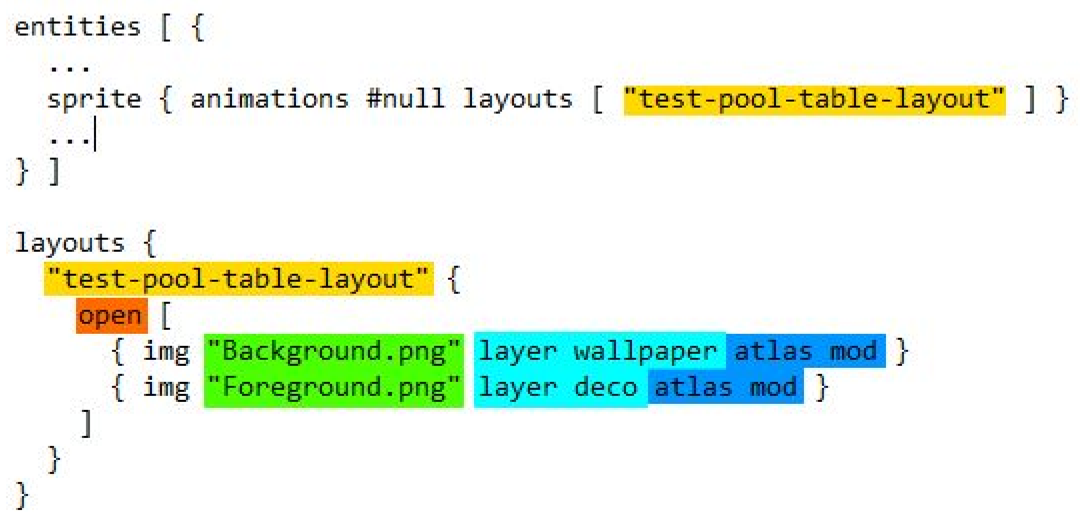
1. Replace all occurrences of "**MY_MOD_ID**" with the new ID, "**test-pool-table**"
2. Type in your own deco item name and description inside the "text" section

In my case, it looks like this:



```
{

;; this section specifies different information about the decoration
;; (and everything starting with a semicolon ; is a comment)

entities [ {
  ident { template "test-pool-table" parent "deco-base" }
  placement { size { x 4 y 1 } }
  sprite { animations #null layouts [ "test-pool-table-layout" ] }
  unit {
    locname "test-pool-table.name"
    locdesc "test-pool-table.desc"
  }
} ]

layouts {
  "test-pool-table-layout" {
    open [
      { img "Background.png" layer wallpaper atlas mod }
      { img "Foreground.png" layer deco atlas mod }
    ]
  }
}

text {
  ;; text section has to contain *at least* english text under the "en" section
  ;; but you can add more sections for other languages as well

  en {
    "test-pool-table.name" "Pool Table"
    "test-pool-table.desc" "Perfect addition for the office rec room"
  }
}

}
```

You might have noticed one new section in this definition file: the “layouts” section. Here’s how it works:



```
entities [ {
  ...
  sprite { animations #null layouts [ "test-pool-table-layout" ] }
  ...
} ]

layouts {
  "test-pool-table-layout" {
    open [
      { img "Background.png" layer wallpaper atlas mod }
      { img "Foreground.png" layer deco atlas mod }
    ]
  }
}
```

Here instead of specifying a single image, we link up the entity to a layout entry named “test-pool-table-layout”. Inside there, we have just one section called “open” which is the default section (in future tutorials we’ll see more section types).

The open section contains two images which will be displayed together. They’re defined like this:

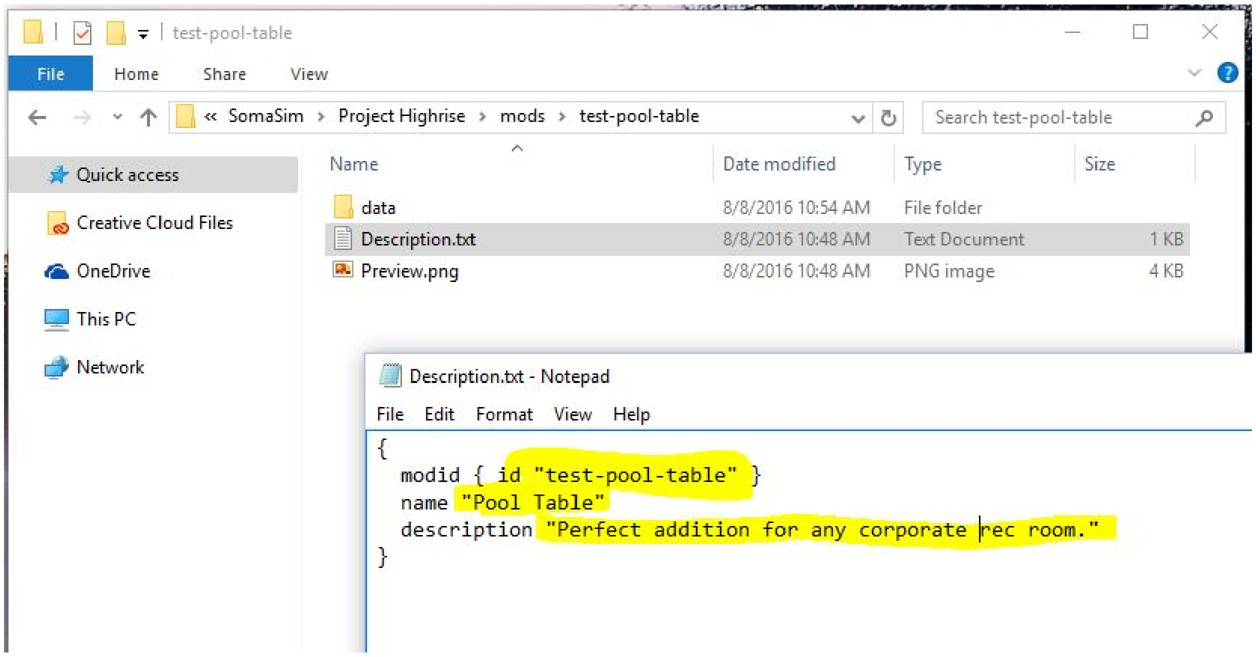
- “Background.png” and “Foreground.png” are the filenames that we’ve seen before
- layer wallpaper and layer deco specify which named layers each of them belongs to
- atlas mod means that they should be loaded up from user-specified images.
    - (This is because we can mix-and-match user-supplied images with built-in images inside a single entity. We’ll see this in the next tutorial.)

So in short, this means that Foreground.png (which contains the pool table) will show up in the “deco” layer up front, while Background.png (which contains the cue rack) will show up in the “wallpaper” layer in the back.

Let’s see it in action!

Step 3. Testing

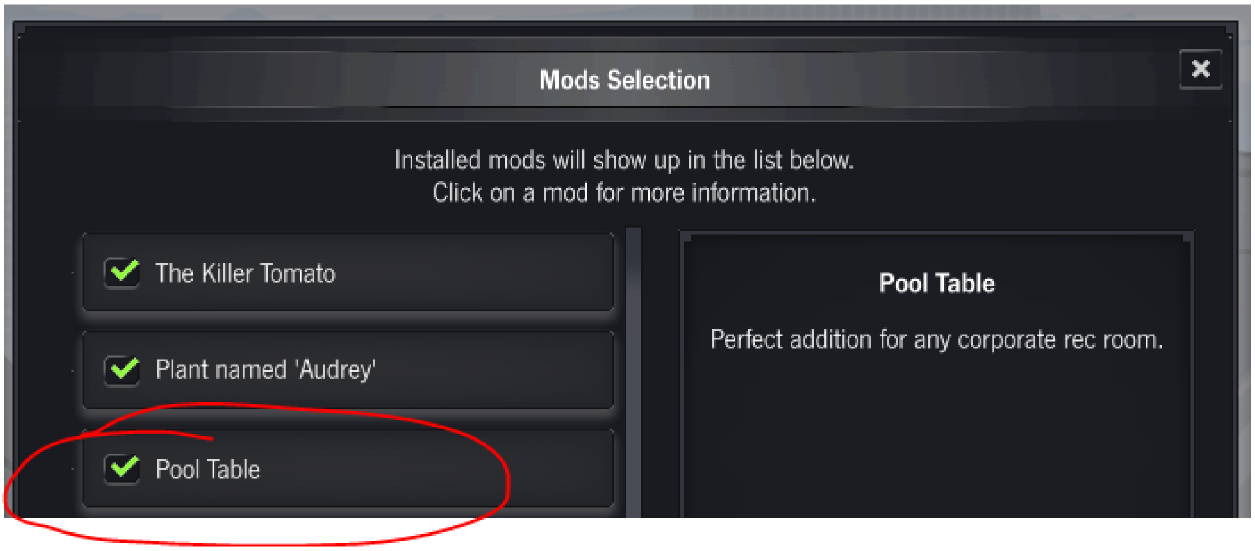
As before you will also need to edit **Description.txt** and **Preview.png** files for your new pool table. You can use whatever strings and images you want, but don't forget to *at least* replace the MY_MOD_ID inside Description.txt with your mod id:

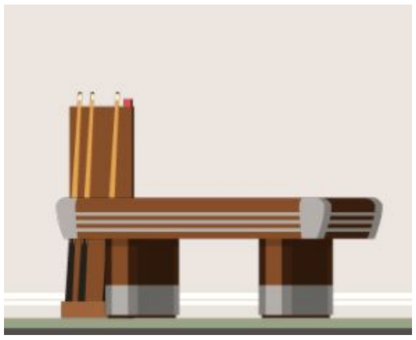


Name	Date modified	Type	Size
data	8/8/2016 10:54 AM	File folder	
Description.txt	8/8/2016 10:48 AM	Text Document	1 KB
Preview.png	8/8/2016 10:48 AM	PNG image	4 KB

```
{
  modid { id "test-pool-table" }
  name "Pool Table"
  description "Perfect addition for any corporate rec room."
}
```

Now start the game and enable your new mod in the Mods dialog:

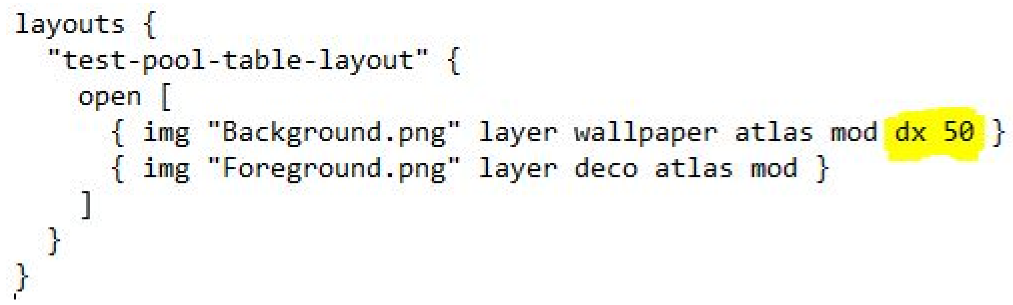

Mods Selection
Installed mods will show up in the list below.
Click on a mod for more information.
The Killer Tomato
Plant named 'Audrey'
Pool Table
Pool Table
Perfect addition for any corporate rec room.



Hmm, both of the images are left-aligned. Remember when we said that the cue rack is smaller than 400px wide, and we’ll come back to it later? Well, later is now. :)

When the images are not of the same size, they will get aligned to their bottom-left corner by default. In order to change that, we’ll add a new “**dx**” offset to the layout definition, to push the cue rack one tile to the right.

Inside **Definition.txt**, change your layouts section to read as follows:

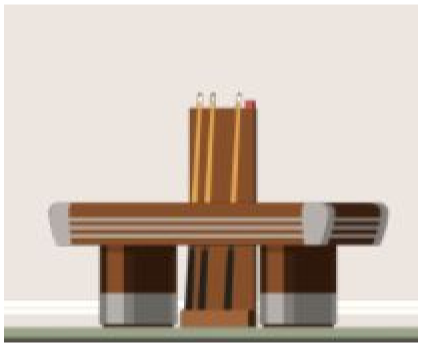


```
layouts {
  "test-pool-table-layout" {
    open [
      { img "Background.png" layer wallpaper atlas mod dx 50 }
      { img "Foreground.png" layer deco atlas mod }
    ]
  }
}
```

There are two modifiers you can apply to layout images: **dy** and **dx** will move the image up and to the right, because otherwise they will be left- and bottom-aligned together.

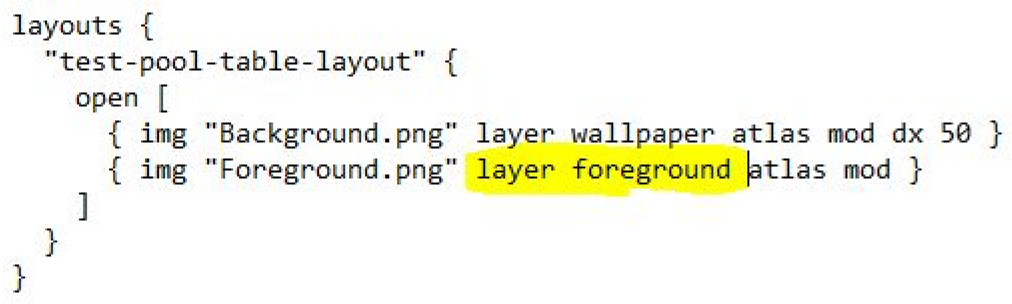
*(Due to boring historical reasons, the units are pixels divided by two, so **dx 50** will actually move the image 100px to the right, similarly **dy 150** would move it 300 pixels up...)*

Save and restart the game, place it down again, and let’s see it in action!



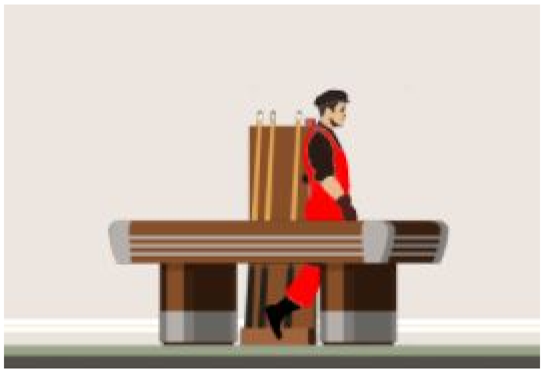
Now here's another exercise. Since the pool table is in the "deco" layer, that means people will be walking in front of it all the time. But what if we wanted the pool table to be in the very front of everything, and have people walking behind it?

If you look at the list of layers in Step 1, the frontmost layer is called "foreground". Let's change the pool table's definition to use that layer:



```
layouts {
  "test-pool-table-layout" {
    open [
      { img "Background.png" layer wallpaper atlas mod dx 50 }
      { img "Foreground.png" layer foreground atlas mod }
    ]
  }
}
```

Let's run it again! Now when somebody is walking by, it should look like this:

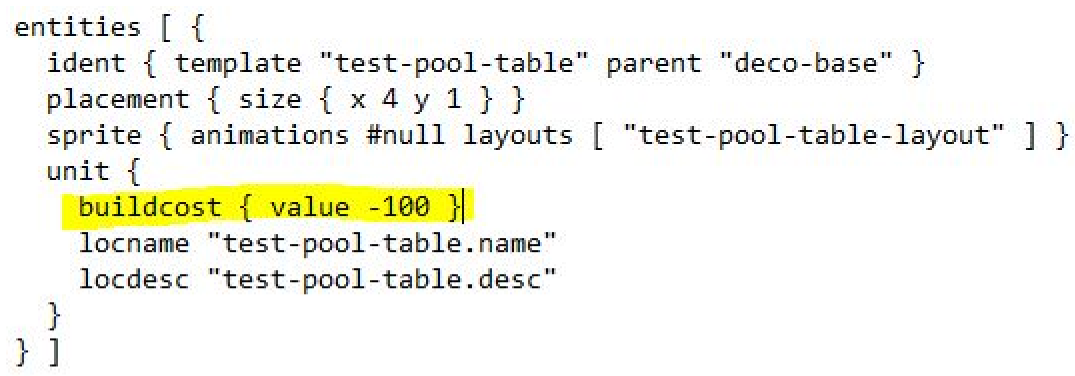


That looks correct, the table is layered in the foreground, while the rack sorts behind other layers and in front of the wall.

<u>Step 4. Make Them Pay</u>

So far our decorations were free. But what if we wanted the player to pay for them?

Here's an easy addition to our **Definition.txt** file:

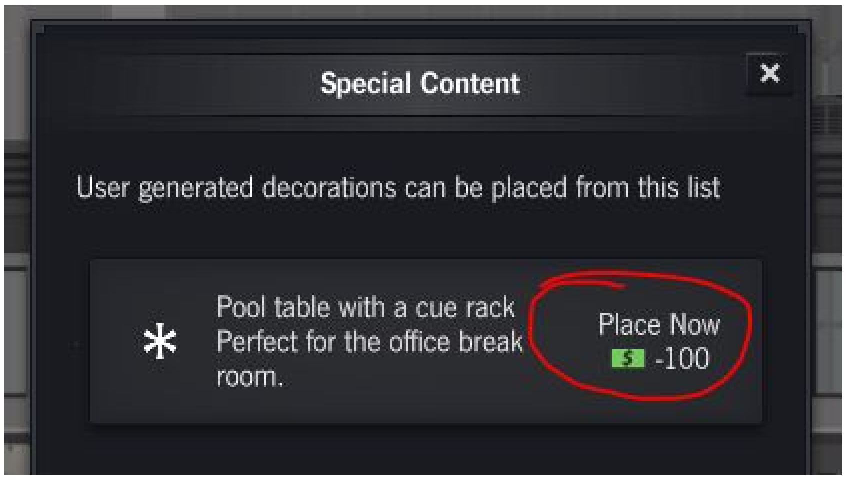


```
entities [ {
  ident { template "test-pool-table" parent "deco-base" }
  placement { size { x 4 y 1 } }
  sprite { animations #null layouts [ "test-pool-table-layout" ] }
  unit {
    buildcost { value -100 }
    locname "test-pool-table.name"
    locdesc "test-pool-table.desc"
  }
} ]
```

This new **buildcost** element represents how much it will cost the player to place this item. It's negative, to indicate we're subtracting the player's money by placing it.

(NOTE: build cost must always be negative or zero, otherwise an error will be reported!)

Try it in action:



Special Content

User generated decorations can be placed from this list

Pool table with a cue rack
Perfect for the office break room.

<u>HOMEWORK</u>

Now build your own multi-layered deco! :)