

Creating a Unit for Spring

Creating a new unit for Spring is fairly straightforward, but will require a number of things to get started with.

- A working Spring install
- A map to test on
- A mod to implement your unit within
- A copy of UpSpring – this is a model editor
- A copy of Scriptor – this is used for editing unit behavior scripts

At this point, you should have a working Spring install and map already. You should find the rest on the assignments page in a starter zip file.

Step 1: Model your unit

Once you've determined what you're going to make, open up Max and start building. Use what you've learned. There are a few differences to keep in mind between this model and the ones we've made previously.

First, you'll be creating a custom texture for this model. It doesn't need to be award-winning but it does need to make sense.

Secondly, this means you'll need to create a correct UV layout for your texture. This means unwrapping your model and laying the components out on a sheet to minimize stretching, and allow the entire model surface to be textured.

Third, it will be necessary for this model to be built in a few different pieces to allow in-game animation to happen. At a bare minimum, you'll need a main **body**, a **turret**, and a **barrel** for that turret. Make sure the pivot points for these objects make sense – the pivot point for the turret should be located at the point it rotates around, for example.

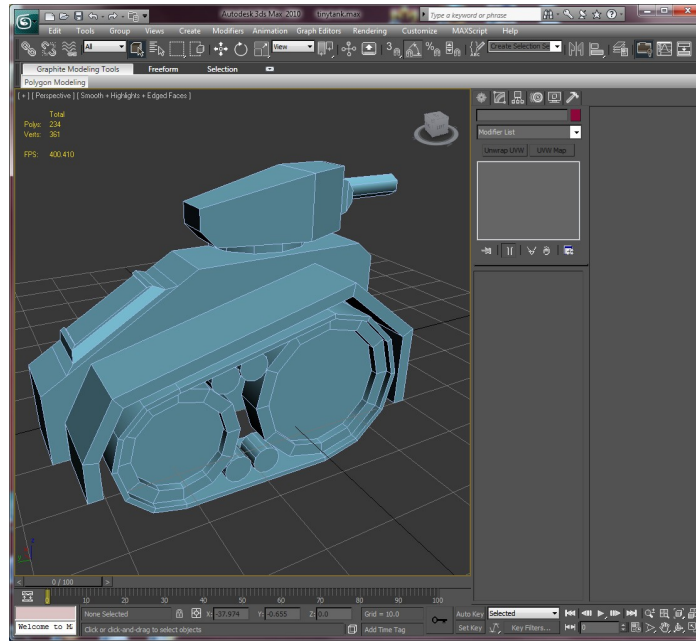
Remember that the primary view in the game is top-down, so make sure your model works acceptably from that vantage point. Make sure your smoothing groups are correct. The model does not need to be perfectly sealed or on a grid, but keep an eye on the size. The default Max grid in the perspective view has the footprint of a large building in-game.

Once all this is done, you'll need to export your model and bring it into UpSpring for setup.

Polygon Budget: No hard limit, but keep it around or under 1,000 in Max. You can view this with the 'Show Statistics' function, located in the upper-left corner of the viewport.

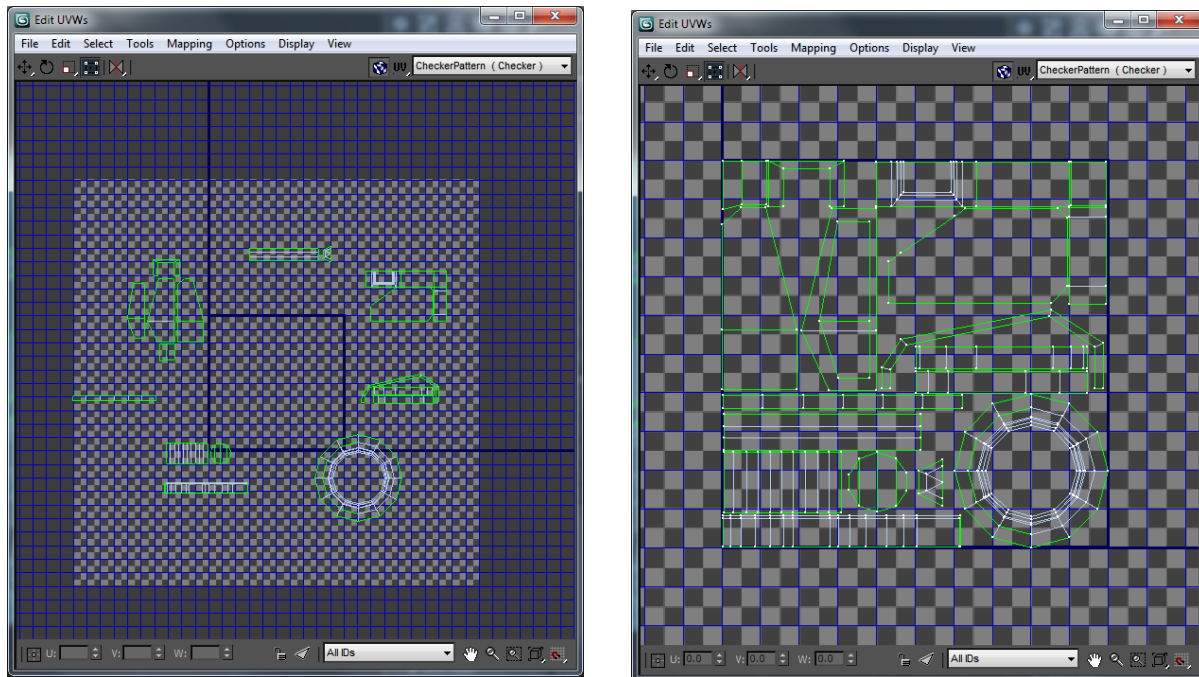
Texture Resolution: A single 512x512 diffuse map.

I decided to build a cartoony tank for my model. Since the unit is symmetrical, I started out by building half of it.



This first half is the thing that I unwrapped and readied for texturing – the texture will be mirrored on the other side, and is designed to meet up cleanly in the middle. Note that the texture doesn't actually exist yet, what I'm doing is unwrapping all the surfaces and placing them in a layout that avoids overlap and doesn't go off the edges of the texture at all. This means I can access everything and paint over it later.

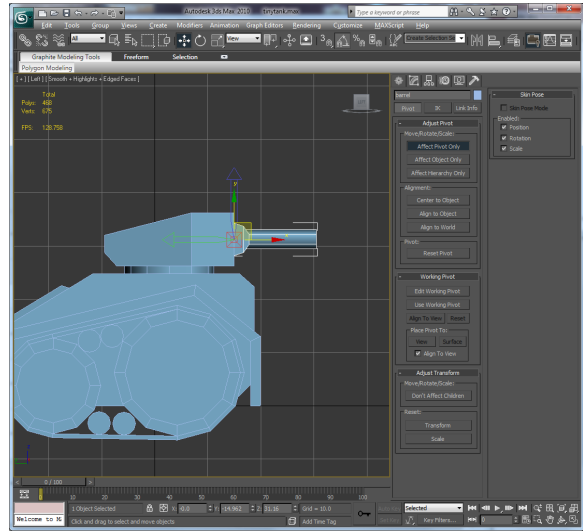
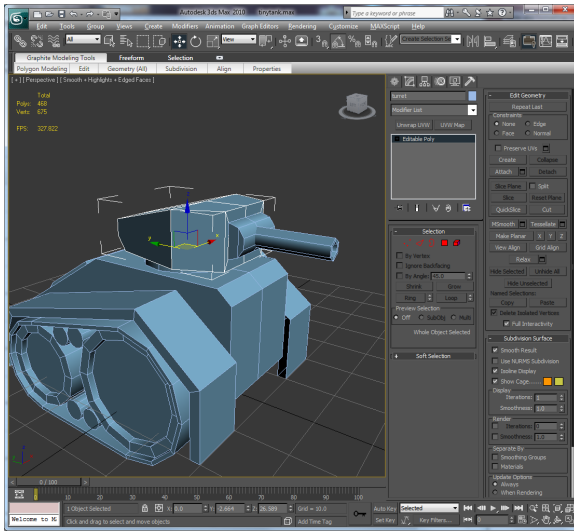
Actually mapping the individual pieces is the same as always – use UVW map and the various unwrapping tools and methodically map each piece of your model. As you finish each section, move its UVs off to the side somewhere so they're out of the way.



The left-hand image here is what my UV layout looked like after I was done mapping the individual pieces.

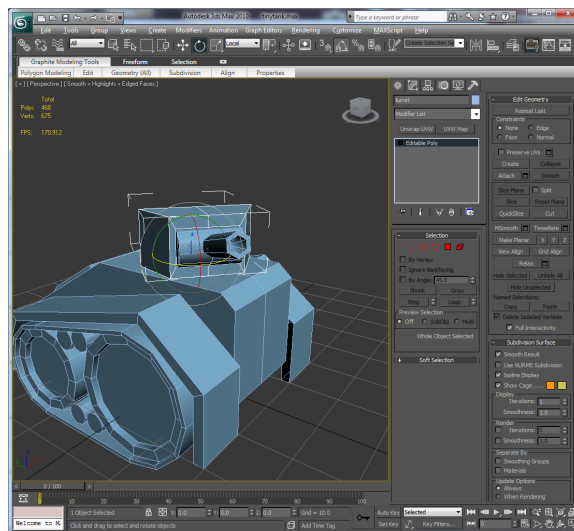
The right-hand image is my final UV layout, once all the pieces were adjusted and fit together as efficiently as I could. From here, I go to the tools menu and select “Render UVW Template”. This creates an image that I can use as my guideline when making the texture. Note that a few things were mirrored or rotated to fit better.

Once I had the first half mapped and ready, I made a mirrored duplicate of it, then attached and welded those two halves together to create a full model. I then separated out the body, turret, and barrel into separate objects by using Detach.



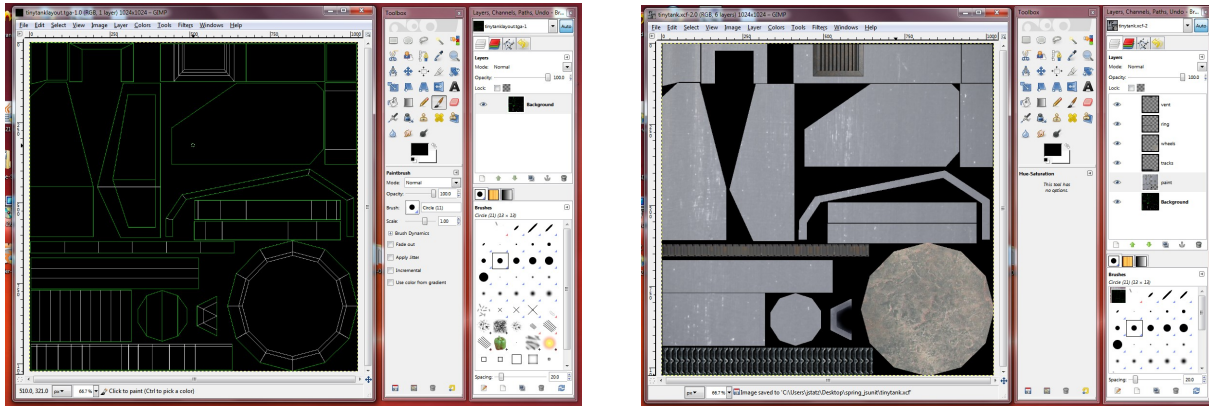
The left-hand image shows my separated and welded model. The right-hand image is me using Affect Pivot Only to move the pivot points of the pieces (body, turret, and barrel) to their appropriate locations. We'll want these to behave correctly when we're ready to export.

Once your pivot points are correct, you can test the behavior by using the Link tool in the upper-left. Link the barrel to the turret, and the turret to the body. If you set your rotation mode to Local (the center of the top toolbar), you should be able to rotate the body, turret, and barrel and see everything stay aligned and make sense.



Now, we've got a sensible model, but haven't worked out a texture yet. You can make use of whatever source material you find online for this, and can use your exported UV layout as a guideline to paint on top of. I strongly recommend saving this in a native format (PSD for Photoshop, XCF for Gimp) so that layers are preserved.

To actually use this in-game, you'll want to save it out as a 32-bit TGA file.



On the left, I have my saved UV layout open. On the right, I've filled that in with surface texture, mostly from source material online. Using the magic-wand to select the shapes inside the UV wireframe, then expanding the selection via the selection menu, allowed me to easily mask the images I was filling in to the correct shapes. This gives me a good starting point for later details and shading.

Things to Remember:

- Hopefully you got familiar with the different layer blending techniques. In particular Normal (opaque), Screen (brighter), Overlay (brighter and darker), Multiply (darker).
- Rather than erasing, you can try using layer masks (it's the button with the circle in the rectangle). Black makes it transparent, white opaque. If you need to change things later you can just paint into the layer mask.
- Photos are good places to get textures/decals/vents/etc. Google image search is your friend.
- There are sites with free textures specifically for this sort of thing, such as cgtextures.com or mayang.com/textures
- To create the illusion of plates/vents/etc you can create the shape (with the pen tool, rounded rectangle, texture, etc) then right click on the layer and use blending options. Set the inner/outer glow to black color and set the blending mode to multiply to create the illusion that it's raised (outer glow) or inset (inner glow). Careful about emboss or drop shadow as this has directional lighting that might look wrong if the light isn't from that angle in game.
- To easily darken/lighten stuff, such as for shading, you can create a layer with overlay blending with 50% grey, and then use the burn and dodge tool.
- To create the illusion of dirt/rust/wear you can set your brush to spatter and add in scatter and opacity jitter to make it seem more random.

Team Colors:

You've probably noticed that the units in Spring have coloration based on what team you're on. Spring uses the alpha channel of your image to determine where to place this coloration. Painting white into the alpha channel causes the team color to show up in that location, while black means no team color. You can also use any in-between value to adjust how much coloration shows up.

Unfortunately this is a bit inconvenient to work with, as that means big chunks of your image turn invisible as you're working. You'll probably want to avoid putting the team coloration into the alpha channel until you're largely done.

Step 2: Exporting your data

The first thing you'll need is a mod to implement your unit in. I recommend using XTA for this purpose, and you'll find an unpacked copy available on the class web site.

A quick note about filenames: Remember how everything Spring reads in is a package of some kind, generally a zip file? Well, you can use a directory as well, and for our working projects that's the easiest route to go. If you give a directory the extension `.sdd`, Spring will read from it the same way it will read from a package file, but you get to skip the packaging-up step!

This means you should end up with a directory named `xta_comp162.sdd` inside your mods folder. This is where we'll be placing your exported files.

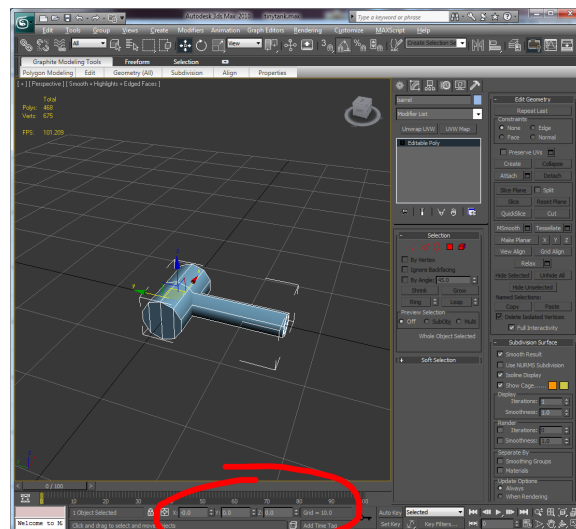
Your Texture:

Save your texture as a 32-bit TGA. Give that file a sensible name and place it in the `unittextures` subfolder. Don't forget the team coloration.

Your Model:

Here is where it gets a bit complicated. In the starter package you should have a program called UpSpring. You'll be using this to get your model in-game. Fortunately it's a fairly easy thing to work with. Before that, however, you'll need to export the pieces of your model.

Go back into Max and select one of your model's components. Enable the move tool, and at the bottom of the screen make sure the XYZ coordinates of your current component are all at 0.



Now, go to the Max logo menu at the top of the screen, and select Export → Export Selected. Select the OBJ export format, with the default settings. Save this to your working folder, making sure to give it an appropriate filename.

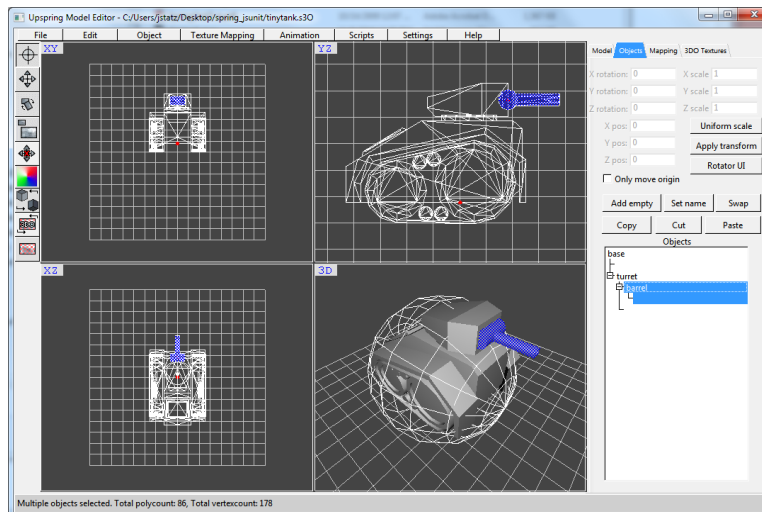
Working with your exported model in UpSpring:

Now that you've exported your model, run UpSpring. You'll see a four-panel view similar to the default 3DS Max layout. What we're going to do here, is assemble your unit's components and set their heirarchy.

The heirarchy is a simple tree of parent/child relationships. When a parent moves, its children move along with it (keeping their relative location in tact). By parenting the barrel to the turret, then the turret to the base, you don't end up with a turret floating in space just because the turret went to face another direction.

Activate the Objects tab. Create an empty and name it **base**. Select it, then go to the top menu and select **object** → **insert from file** to import the main base/center of your object. This will be the top of the hierarchy. Check where the front of your unit is in the top-down view, if it's oriented correctly it should be facing towards the top of the screen.

Make sure 'base' is highlighted in the objects list, and add another empty. Name this one 'turret'. Notice how it shows as indented underneath base. Do the same thing for 'barrel', making sure it is underneath 'turret' in the hierarchy, then insert their geometry appropriately. If you have additional objects, make sure they're parented appropriately as you add them.



At this point, you should have all your geometry in UpSpring and named correctly. Select the move tool on the lefthand side and move your objects to their correct positions in space now.

Once that's done, you'll need to point it at your texture. Select the 'Mapping' tab and browse for Texture 1. Select your saved TGA.

Finally, once everything looks right here, set the dimensions of your unit. This is done via the Model tab. You'll need to set a centerpoint, the radius, and the height. You can push the 'Estimate' buttons to do exactly that, then adjust the numbers from there. On my unit, I had to reduce the radius from its 45 unit estimate to 30, to get the sphere to be a roughly correct size.

Now that all this is done, you're going to save your unit to its game-ready state. Go to **File**→**Save**

Model and save it to .s3o format. This is the file format Spring itself uses. Copy your .s3o file to the **objects3d** subfolder.