Chapter 22- Textures in the Game Engine

Since games need to be able to process operations as fast as possible, traditional rendering techniques (specularity, ray tracing reflections and refractions) cannot typically be processed fast enough for a game. For this reason, textures need to be mapped differently. There are also times when you may want to use mapped textures in an actual render. Blender does this through traditional UV Texture Mapping and a new system called GLSL Shading. There is a lot that can be done through both of these methods “beyond the basics” that will be discussed here. For more details, check out the Blender wiki.

UV Texture Mapping

Think of UV Mapping like taking a box and cutting it to lay flat. The texture needs to match each side of the box. It is difficult to do that with traditional materials and textures. Let's say you want to model a dinosaur. The texture changes on various parts of his body and need to be mapped correctly. This is where UV Mapping come into play. For this example, I am going to map the following texture I made in GIMP on a basic cube:

![Image of a texture](image-url)

This is just a jpeg image that would be impossible to map as a standard material/texture.

To begin, start with a basic scene with a cube and change your viewport shading type from Solid to Textured. This is the shading used during game play. You will notice that, by default, textures are effected by the lighting so add some lights to illuminate your scene better and switch to the "UV Editing" window layout. This will give you one 3D viewport and one UV Editor viewport. At the bottom of the UV Editor viewport, hit the Image-Open Image menu option and find the texture you wish to use.

Now, enter Edit Mode for the cube and switch to selecting Vertices to Faces since this is a face applying process. You can select individual faces and put UV textures on that way, but let's “Unwrap” the cube to match our texture. This can be done for any mesh, but we need to mark the seams where we want a split to occur. If we look at the picture, we can see where seams should go. In order to mark seams, we need to switch from Face select to Edge select.

Select the following edges (Shift-RMB) and click “Mark Seam” in the UV Mapping section in the Tool Shelf. These will be the unfold edges. It should match the box layout.
Now go back to Face select mode, hit “A” for All twice to select all faces. Type “U” to bring up the UV Mapping options in the 3D window. You have several option. We want “Unwrap”. You will now see the unwrapped faces in the UV Mapping window. You can select these verticies as you would for any other Blender object and move, scale or rotate them. You will also see the texture on the cube. By pressing “P” you will see the texture in game play. (in object mode) Adjust the verticies so it looks good on the cube.

By using this technique, you can select single or groups of faces on an object to assign textures. By switching back to the Default screen layout, you will find a panel in the Object Data buttons, in Edit Mode, that effect the texture faces. In the current version of Blender, you can only select one face at a time to change these options.

Some useful options here are:

- **Light**: face is effected by light hitting it.
- **Invisible**: good for adding planes along a track as guides.
- **Collision**: unchecked and the actor can go through it.
- **Two-Sided**: by default, texture visible from one side only.
- **Transparency**: options for visibility.

**UV Textures in an Animated Movie:**

Just like game physics can be written into an animation curve, UV can be used with materials and textures. After going through the steps above, add a material and texture to the object. Select “Image” for the texture type, select the picture you used, the under the “Mapping” panel, choose “UV” in the Coordinate box. Pressing F12 should give you a rendered image of the map.
**GLSL Shading**

GLSL shading is very new to Blender and is an area seeing a great deal of development. It is an attempt to add many rendering-like features to the game engine, adding to a more realistic environment. As with the UV Mapping section, this unit will only cover the basics to get you started.

Not all video cards are supported for GLSL shading. Some machines will be unable to use this feature. See the Blender wiki for current specification.

To get started, we'll start a new scene and split the viewport, setting one to **UV/Image Editor**. Set the Engine to **"Blender Game"** and viewport shading to **"Textured"**. The last thing you need to do is switch from **"Multitexture"** to **"GLSL"** shading in the Render panel. You are now ready to work with the GLSL features in the game engine. Not all texture and shading features are available in GLSL, but many are and others are being developed constantly. As mentioned before, we will only be looking at some of these features.

**Adding Textures:**

Basically, to add textures for the game in GLSL, you need to add materials and textures the way you do for any movie. For my example, I am working with a plane for the ground and a cube. I've added a material and texture to the floor, using a stone texture. The stones are too large so I need to repeat it a bit in the texture panel. You can't use the X and Y Repeat in the **"Image Sampling"** panel, but can change the size in the **"Mapping"** panel.

The cube was a bit more difficult. Because the texture wants to map as **Flat** by default and GLSL does not currently work with changing the mapping to **Cube**, I had to apply the texture as we did in the previous section using **UV Texture Mapping** and mapping the texture to each face. Remember to also switch the **Mapping** in the Textures panel to **UV**.

Another nice feature with GLSL is the ability to show **Normal Geometry** to give a texture depth. Looks best with a higher **Specular**.
Shadows in GLSL:
Another nice effect allowed in GLSL is the ability to cast shadows in game play. Right now, ray tracing is not supported so your only option is to use a Spotlight with a Buffer Shadow setting (refer to the lighting chapter for more details on setting the buffer shadow).

Currently, there is a lot of development in GLSL with new features constantly being added. There is also a lot of work in Baking settings to improve performance. There is work on support for indirect lighting, fluids, soft bodies and many more. For up-to-date information, follow the Blender wiki, YouTube, and the forums.

World Settings:
While some World settings work in Multi-texture mode and some work on GLSL Shading. Some features do not work in either at the time (stars for example). In Multi-texture shading, you can get a nice effect with the Mist settings to give a “foggy” feel to your game, but works a bit differently in GLSL. World Horizon and Zenith colors work differently as well.
Open your *Motion Maze* file from the last chapter. Your goal in this exercise is to make it look good while in game play. You may use UV Texture Mapping or a combination of UV Mapping and GLSL Shading. Find or create a nice domino texture along with appropriate textures for all other items. Remember that for a UV Mapped texture to work in GLSL, you need to switch to UV in the mapping panel in Textures.

**Call the instructor when finished**