

## Shaders and Textures

### Wogrim's Brief Guide to Shaders and Textures

#### Why Is the Shader Important?

The shader is what decides how to draw your item. It does many calculations to decide what color pixel to draw on the screen, based on textures, lighting, shadows, and any other variables you or the game gives it.

#### What Shaders Should Be Used?

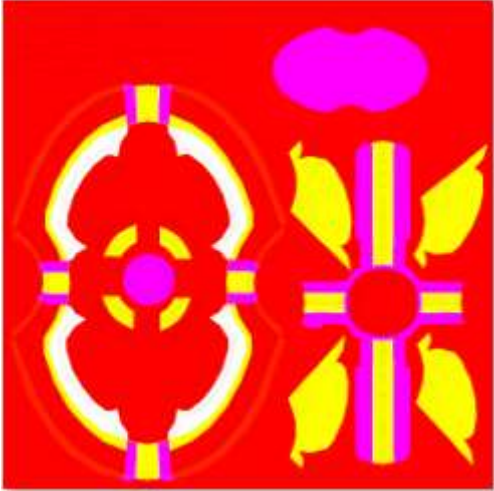
For the vast majority of items, the best shaders are the game shaders because they work with game customization features like color picker, game menu options like outline width, and gameplay features like liquid pattern on clothes; other shaders will not work with these features unless programmed to do so. This may change if someone manages to fully reverse-engineer all the game-specific shader features.


The game uses a few other shaders, but here are the most common ones. Note that a complicated item may use different shaders on different parts.


Item Type	Recommended Shader	Notes
Accessory	Shader Forge/main_item	use for most opaque or cut-out accessory parts
Accessory	Shader Forge/toon_glasses_lod0	use for glass-like parts (uniform transparency and color)
Clothes	Shader Forge/main_opaque	use for opaque or cut-out clothes
Clothes	Shader Forge/main_alpha	use for semi-transparent clothes
Hair	Shader Forge/main_hair_front	including hair accessories, any hair that may cover eyebrows/eyes
Hair	Shader Forge/main_hair	including hair accessories, any back hair pieces
Studio	Shader Forge/main_item_studio	use for most opaque or cut-out studio items
Studio	Shader Forge/main_item_studio_alpha	use for semi-transparent studio items
Face/Body	Shader Forge/main_skin	use for head and body mods

### What Features Do The Game Shaders Have?


To keep the explanations simple and not worry about slight differences between the shaders, I am omitting some details of how exactly these features work. For many of these features, the shaders have variables to tweak, which you can do with Material Editor.


feature	<b>works with game color pickers</b>
availability	most/all
description	the ColorMask texture determines which parts of the item are which colors
picture	


feature	<b>normal map</b>
availability	most/all
description	the NormalMap texture alters the direction the mesh is considered facing for lighting calculations
picture	


feature	<b>shadows and outlines which work with game settings</b>
availability	most/all
description	there are some global shader variables that many of the shaders use for a consistent look
picture	 <p>This setting applies to the display of all in-game characters.</p> <p>Shadow Type: All Smooth</p> <p>Shadow Density: 74 (Reset)</p> <p>Outline Density: 100 (Reset)</p> <p>Outline Size: 31 (Reset)</p>


feature	<b>texture highlights</b>
availability	most/all
description	red on the DetailMask texture gives a highlight on that part of the item, even when the light is not on that part of the item
picture	

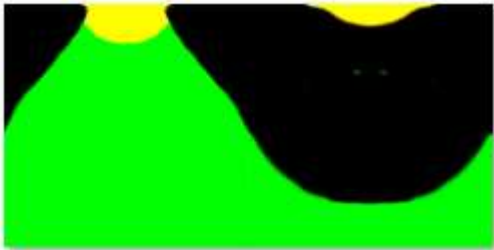
feature	<b>ramped texture highlights</b>
availability	most/all
description	the AnotherRamp texture affects texture highlights based on the viewing angle in a way that gives it a fake reflective look, frequently used on metal items
picture	


feature	<b>texture shadows</b>
availability	most/all
description	green on the DetailMask texture forces shadows on that part of the item, which are affected by the game's shadow settings
picture	

feature	<b>texture lines</b>
availability	most/all
description	blue on the DetailMask texture draws lines on that part of the item, which are affected by the game's outline settings
picture	


feature	<b>specular lighting</b>
availability	most/all
description	the surface shines where the light would reflect off to the camera
picture	


feature	<b>rim lighting</b>
availability	most/all
description	fake lighting appears on surfaces that are at a sharp angle relative to the camera
picture	

feature	<b>hide part of the item based on item in layer above</b>
availability	Clothes (in bra or inner top slot) and Body
description	item in layer above puts a texture in this item's AlphaMask slot; if all is right, appropriate parts of this item are made invisible to make sure they don't clip through the above layer
picture	

feature	<b>liquid pattern from sex actions</b>
availability	Clothes and Face/Body
description	textures named liquidmask, Texture2, and Texture3 determine how the juice shows up
picture	 A 3D render of a person's back, showing a liquid-like pattern (juice) flowing down the spine and spreading across the skin. The background is a bright blue sky with green foliage.

feature	<b>works with hair gloss picker</b>
availability	Hair
description	hair gloss gets put in the HairGloss texture; hair gloss works likes a texture highlight
picture	 A close-up 3D render of a person's hair, showing a glossy finish with highlights. The hair is dark brown and styled in a bun. The background is a bright blue sky with green foliage.

feature	<b>draws eyes/eyebrow in front of hair</b>
availability	Shader Forge/main_hair_front
description	this is an option in game settings; I don't know exactly how it works
picture	

feature	<b>draws additional textures on certain parts</b>
availability	Shader Forge/main_skin
description	nipple and pubic hair are textures given to the body's material to draw; similar for a couple of the face texture items
picture	

Here's some other features that you'll come across on some of the game shaders:

- casts shadows
- receives shadows
- double-sided vs one-sided (backface culling)
- alpha clip (cutout) or alpha blend (transparency) based on alpha channel of MainTex
- emissive (glow in the dark)



### What Cool Things Can Be Done With Custom Shaders?

All kinds of cool things, but not everything. There are limitations based on how the game's graphics work. Also, any new game-controlled shader feature would require a plugin to control it.

We don't have a good way to decompile the game shaders, so you can't just add a feature to them; you need a shader with your cool new feature, and then add in reverse-engineered versions of whichever game shader features you want. If you are interested in making custom shaders, know that the game shader features have been largely figured out; ask around. Here's an image of a custom shader on the backpack which gives a special shading effect based on screen coordinates; the color picker works.

