

Quality Settings

Unity allows you to set the level of graphical quality it will attempt to render. Generally speaking, quality comes at the expense of framerate and so it may be best not to aim for the highest quality on mobile devices or older hardware since it will have a detrimental effect on gameplay. The **Quality Settings** inspector (menu: **Edit->Project Settings->Quality**) is split into two main areas. At the top, there is the following matrix:-



Unity lets you assign a name to a given combination of quality options for easy reference. The rows of the matrix let you choose which of the different platforms each quality level will apply to. The Default row at the bottom of the matrix is not a quality level in itself but rather sets the default quality level used for each platform (a green checkbox in a column denotes the level currently chosen for that platform).

Unity comes with six quality levels pre-enabled but you can add your own levels using the button below the matrix. You can use the trashcan icon (the rightmost column) to delete an unwanted quality level.

You can click on the name of a quality level to select it for editing, which is done in the panel below the settings matrix:-

The quality options you can choose for a quality level are as follows:-

Name	The name that will be used to refer to this quality level
Pixel Light Count	The maximum number of pixel lights when Forward Rendering is used.
Texture Quality	This lets you choose whether to display textures at maximum resolution or at a fraction of this (lower resolution has less processing overhead). The options are Full Res , Half Res , Quarter Res and Eighth Res .
Anisotropic Textures	This enables if and how anisotropic textures will be used.
Disabled	Anisotropic textures are not used.
Per Texture	Anisotropic rendering will be enabled separately for each Texture.
Forced On	Anisotropic textures are always used.
AntiAliasing	This sets the level of antialiasing that will be used. The options are 2x , 4x and 8x multi-sampling.
Soft Particles	Should soft blending be used for particles?

Shadows	This determines which type of shadows should be used
Hard and Soft Shadows	Both hard and soft shadows will be rendered.
Pixel Light Count	2
Texture Quality	Full Res
Hard Shadows Only	Only hard shadows will be rendered.
Anisotropic Textures	Per Texture
Anti Aliasing	Disabled
Disable Soft Particles	No shadows will be rendered.
Shadow Resolution	Shadows can be rendered at several different resolutions: Low , Medium , High and Very High . The higher the resolution, the greater the processing overhead.
Shadow Projection	There are two different methods for projecting shadows from a directional light. Close Fit renders higher resolution shadows but they can sometimes wobble slightly if the camera moves. Stable Fit renders lower resolution shadows but they don't wobble with camera movements.
Shadow Cascades	The number of shadow cascades can be set to zero, two or four. A higher number of cascades gives better quality but at the expense of processing overhead (see the Directional Shadows page for further details).
Shadow Distance	The maximum distance from camera at which shadows will be visible. Shadows that fall beyond this distance will not be rendered.
Blend Weights	The number of bones that can affect a given vertex during an animation. The available options are one, two or four bones.
VSync Count	Rendering can be synchronised with the refresh rate of the display device to avoid "tearing" artifacts (see below). You can choose to synchronise with every vertical blank (VBlank), every second vertical blank or not to synchronise at all.
LOD Bias	LOD levels are chosen based on the onscreen size of an object. When the size is between two LOD levels, the choice can be biased toward the less detailed or more detailed of the two models available. This is set as a fraction from 0 to 1 - the closer it is to zero, the more the bias is toward the less detailed model.
Maximum LOD Level	The highest LOD that will be used by the game. Models which have a LOD above this level will not be used and omitted from the build (which will save storage and memory space).

Tearing

The picture on the display device is not continuously updated but rather the updates happen at regular intervals much like frame updates in Unity. However, Unity's updates are not necessarily synchronised with those of the display, so it is possible for Unity to issue a new frame while the display is still rendering the previous one. This will result in a visual artifact called "tearing" at the position onscreen where the frame change occurs.



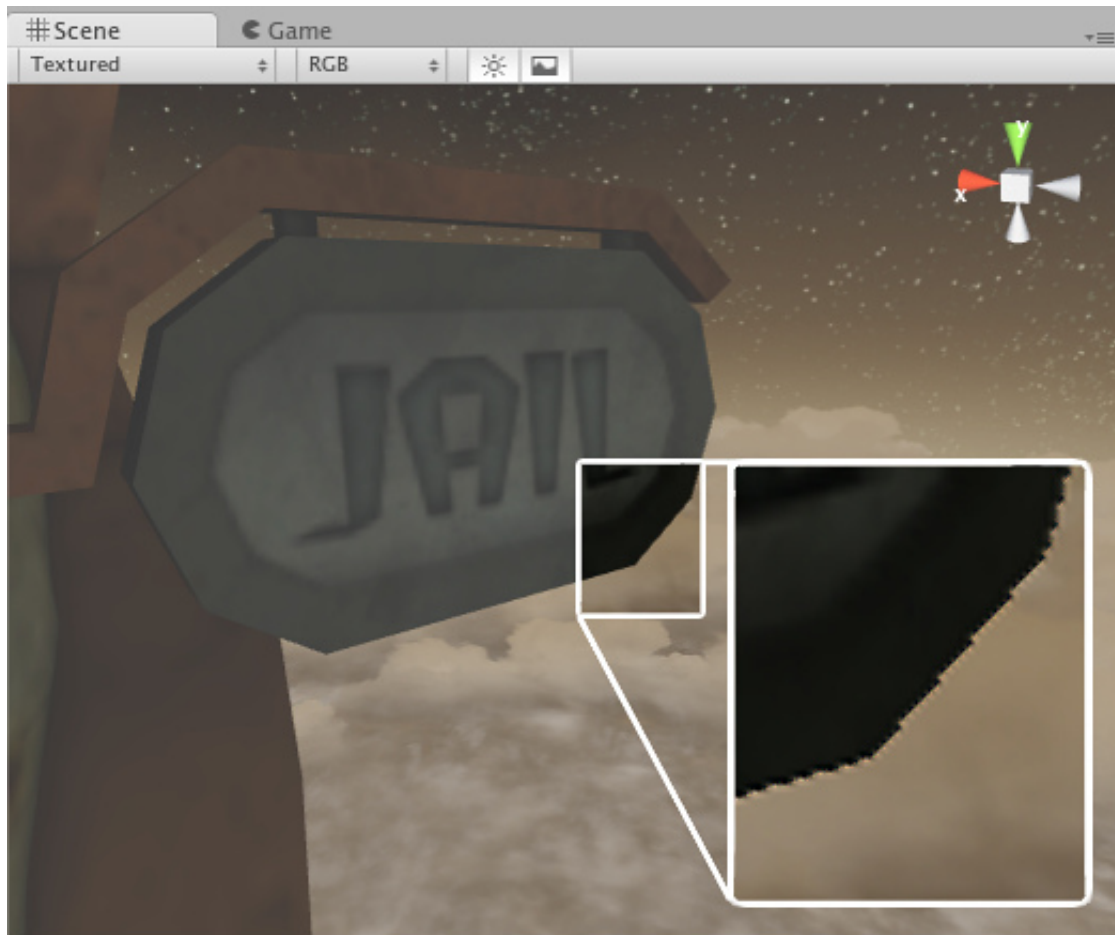
Simulated example of tearing. The shift in the picture is clearly visible in the magnified portion.

It is possible to set Unity to switch frames only during the period where the display device is

not updating, the so-called "vertical blank". The VSync option on the Quality Settings synchronises frame switches with the device's vertical blank or optionally with every other vertical blank. The latter may be useful if the game requires more than one device update to complete the rendering of a frame.

Anti-aliasing

Anti aliasing improves the appearance of polygon edges, so they are not "jagged", but smoothed out on the screen. However, it incurs a performance cost for the graphics card and uses more video memory (there's no cost on the CPU though). The level of anti-aliasing determines how smooth polygon edges are (and how much video memory does it consume).

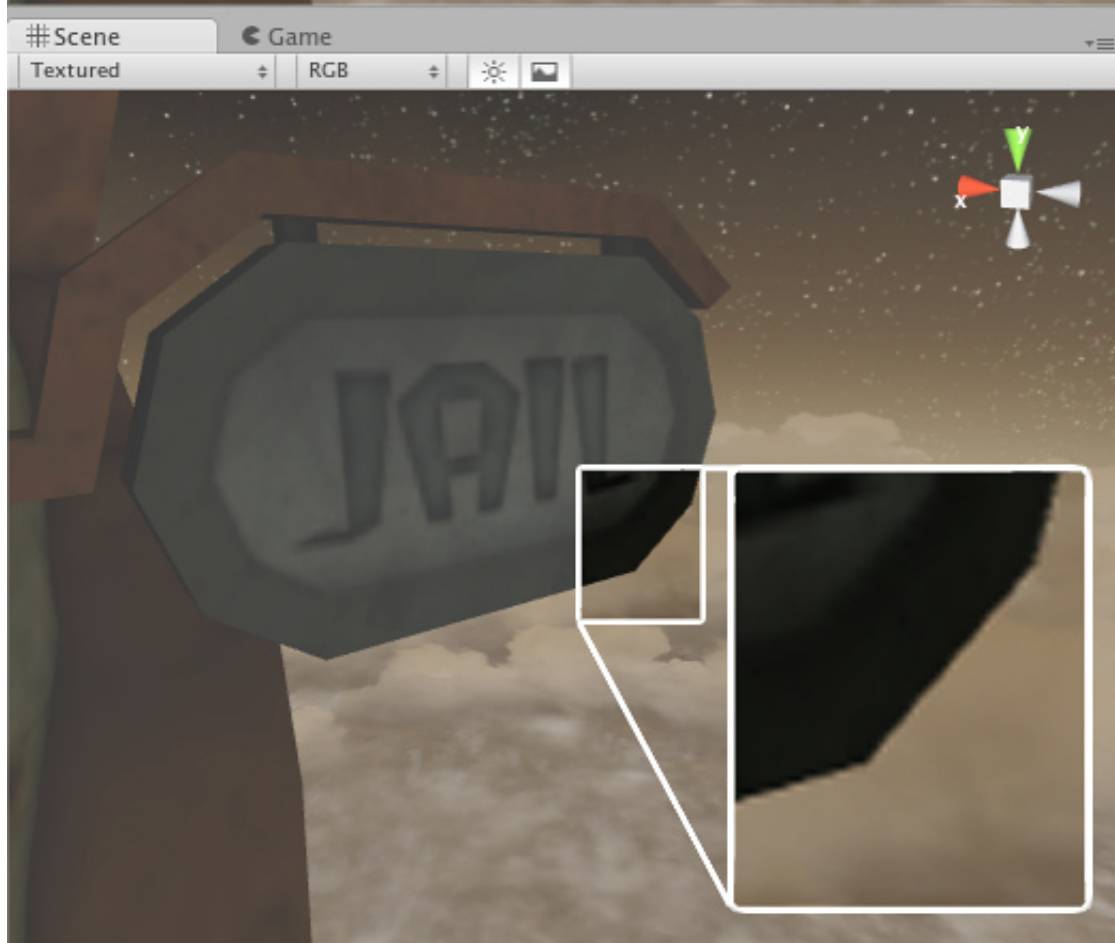


Without anti-aliasing, polygon edges are "jagged".

With 6x anti-aliasing, polygon edges are smoothed out.

Soft Particles

Soft Particles fade out near intersections with other scene geometry. This looks much nicer, however it's more expensive to compute (more complex pixel shaders), and only works on platforms that support [depth textures](#). Furthermore, you have to use [Deferred Lighting](#) rendering path, or make the camera render [depth textures](#) from scripts.



sections with the scene.

Without Soft Particles - visible inter-

*With Soft Particles
- intersections fade
out smoothly.*

Page last updated:
2012-09-26

