CascadedShadowMaps11

<https://github.com/walbourn/directx-sdk-samples>

This is the DirectX SDK's Direct3D 11 sample updated to use Visual Studio 2012 and the Windows SDK 8.0 without any dependencies on legacy DirectX SDK content. This sample is a Win32 desktop DirectX 11.0 application for Windows 10, Windows 8.1, Windows 8, Windows 7, and Windows Vista Service Pack 2 with the DirectX 11.0 runtime.

**This is based on the legacy DirectX SDK (June 2010) Win32 desktop sample. This is not intended for use with Windows Store apps, Windows RT, or universal Windows apps.**

# Description

****

This sample demonstrates the cascaded shadow map (CSM) algorithm. This algorithm also demonstrates several techniques that can be used to make efficient use of the shadow map.

* **Visualize Cascades** Checking this box highlights the different layers of the cascade in the same color as the slider. This option defaults to off because it is for debug visualization.
* **32-bit Buffer Dropdown**: This drop-down toggles between DXGI\_FORMAT\_D32\_FLOAT, DXGI\_FORMAT\_D24\_UNORM\_S8\_UINT, and DXGI\_FORMAT\_D16\_UNORM. Higher precision depth buffers cost more in terms of performance and size but limit precision issues.
* **Texture Size**: This slider modifies the size of the depth-buffer texture. The size does not change until the mouse button is released in order to make the application more interactive.
* **PCF Blur**: This slider determines the size of the PCF blur kernel. "1" is a 1×1 and there for only uses hardware PCF, "3" uses a 3×3 on top of hardware PCF and so forth.
* **Offset**: The offset value represented by this slider is subtracted from the pixel in light-space to mitigate surface acne. Too much offset leads to objects losing their shadows.
* **Cascade Blur**: The slider determines the size of the area between cascades that is blurred when this option is checked. Blurring between the cascade layers can hide the seam that occurs between different cascades. Even with dynamic branching enabled blurring between layers can be expensive so it is defaulted to off.
* **DDX, DDY Offset**: This technique calculates a unique offset for neighboring texels in the shadow map by mapping the shadow map’s surface to a plane. The white paper "Cascaded Shadow Maps" covers this technique in detail. This technique is expensive and should be used as a last resort for large PCF kernels where an offset is not sufficient.
* **Power Plant Dropdown**: This drop-down toggles between difference scenes.
* **Eye Camera Dropdown**: This drop-down toggles between the eye camera and the light camera. The user can also select any one of the orthographic cameras that correspond to the different cascades.
* **Fit Light to Texels**: This technique moves the light in texel-sized increments. This keeps the shadows edges from swimming when the eye-camera moves.
* **Fit Scene Dropdown**: This drop-down allows the user to switch between Fit Secne and Fit Cascade. These two options determine how the orthographic projection is calculated.
* **AABB/Scene NearFar**: This drop-down allows the user to switch between AABB/Scene NearFar, 0:1 NearFar, and AABB NearFar.
* **Map Selection Dropdown:** This drop-down allows the user to switch between Map Selection and Interval Selection. Map selection makes more efficient use of the cascade and is therefore set to default. Interval selection is slightly faster.
* **Levels Dropdown:** This drop-down determines the number of cascades.
* **Colored sliders:** For each cascade, there appears a slider that determines the partitioning of the view frustum into cascades.

# Dependencies

DXUT-based samples typically make use of runtime HLSL compilation. Build-time compilation is recommended for all production Direct3D applications, but for experimentation and samples development runtime HLSL compilation is preferred. Therefore, the D3DCompile\*.DLL must be available in the search path when these programs are executed.

* When using the Windows 8.x SDK and targeting Windows Vista or later, you can include the D3DCompile\_46 or D3DCompile\_47 DLL side-by-side with your application copying the file from the REDIST folder.

%ProgramFiles(x86)%\Windows kits\8.0\Redist\D3D\arm, x86 or x64

%ProgramFiles(x86)%\Windows kits\8.1\Redist\D3D\arm, x86 or x64

# More Information

[Where is the DirectX SDK (2015 Edition)?](https://walbourn.github.io/where-is-the-directx-sdk-2015-edition/)

[DXUT for Win32 Desktop Update](https://walbourn.github.io/dxut-for-win32-desktop-update/)

[Games for Windows and DirectX SDK blog](https://walbourn.github.io/)