



# Mighty Meets Mini: Intel® NUC Skull Canyon

*Now generally available, the Intel® NUC Kit NUC6i7KYK (code-named Skull Canyon) is a tiny, powerful PC that is great for gaming, living rooms, tradeshows, festivals, and anywhere where space is at a premium.*

A new Mini PC has emerged as a compact gaming system, powered by Intel® Core™ processors and Iris™ Pro graphics. This form factor draws momentum from technology brands that are among the most popular on the planet. According to the [April 2016 Steam Hardware & Software Survey](#), Intel® processors power about 77 percent of PCs, and Intel's GPUs hold approximately 18 percent of their market segment share.

The most recent addition to Intel's Mini PC lineup for gaming is the Intel® NUC Kit NUC6i7KYK (code-named Skull Canyon). A premium Next Unit of Computer ([NUC](#)) powered by an Intel® Core™ i7 processor and Iris Pro graphics, this worthy gaming system comes in a sleek, compact form factor (roughly 8.3 inches × 4.6 inches × 1.1 inches).



**Intel® NUC Kit NUC6i7KYK (code-named Skull Canyon)**

Under the hood, the NUC is powered by the 6th generation Intel® Core™ i7-6770HQ processor, with four cores running at 2.6 to 3.5 GHz (Turbo) and a 6 MB cache. Iris Pro graphics 580 support vivid 3D gameplay with 72 GPU execution units, in 1080p. It also supports 4K resolution on up to three monitors for those who needs premium screen real estate.

Thunderbolt™ 3 technology gives single-port connectivity to high-speed external storage, 4K displays, and high-end add-ons such as the [Razer Core External Graphics Dock](#) for discrete graphics cards and extra ports.

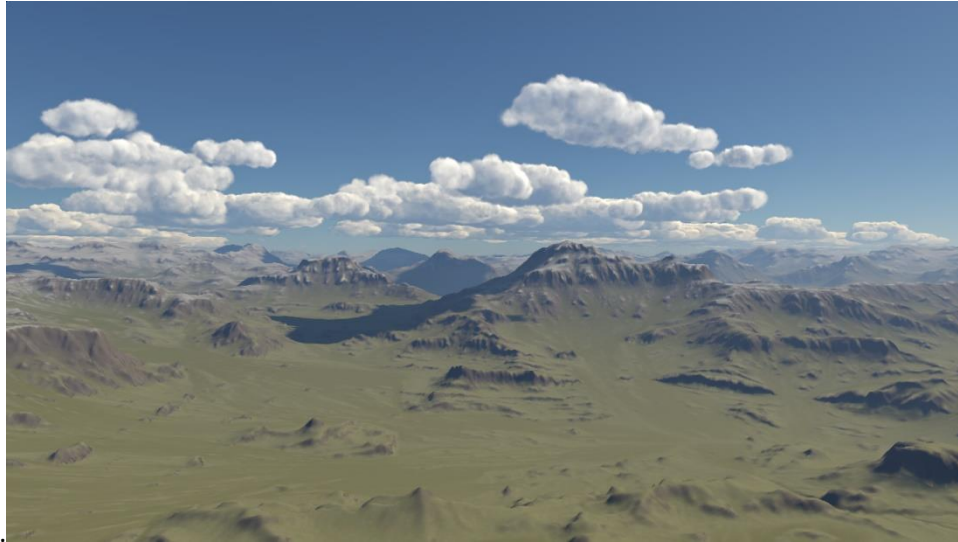
## Graphics Techniques of Interest

Intel pushes the latest graphics APIs by providing support and extensions for advanced features in DirectX\* 12 and Vulkan\*. So, what will you do with that extra CPU and GPU headroom in a system with a 6th gen Intel® Core™ i7

processor and Iris Pro graphics? The potential can take you in lots of different directions, but here are a couple of interesting graphics techniques for game developers to consider.

### Cloud Rendering with Pixel Shader Ordering

Employing natural-looking clouds by shading them realistically in real time is a prime example of how ambient effects can make game worlds more immersive. Puffy cumulus clouds are the most popular type for rendering in games, and as any impressionist painter could have told you, capturing the movement of light through them is the key to success. We demonstrate this approach using Pixel Shader Ordering, a DirectX extension supported on Iris Pro graphics and other Intel® HD Graphics technologies.



**Screenshot generated by cloud-rendering sample code.**

The technique uses randomly rotated and scaled copies of a reference particle, with photon scattering based on pre-computed values that are accessed from a lookup table at runtime. Elements such as a highly efficient particle algorithm and outdoor lighting model allow your games to render high-quality, realistic clouds in real time. This is a particularly great addition to any game that includes flying or large swaths of open terrain.

Download the sample code and supporting documentation at <https://software.intel.com/en-us/blogs/2014/08/31/cloud-rendering-sample-updated>.

### Software Occlusion Culling

Significant efficiency and performance advantage can be gained by optimizing the mechanisms that cull objects from the frame that don't need to be rendered because they're obscured by others. Offloading occlusion culling from the GPU to the CPU can help avoid GPU bottlenecks. It also avoids the cost of submitting occlusion queries to the GPU and transferring occluders between the GPU and CPU.

Intel offers sample code and accompanying documentation for a software occlusion algorithm that runs on the CPU. This algorithm efficiently categorizes screen objects as *occluders* or *occludees* based on a depth comparison and culls the latter set of objects from the rendering pipeline.



Screenshot generated by software occlusion culling sample code.

The code implementation of this algorithm is optimized for multi-threaded Intel Core processors using Intel® Advanced Vector Extensions 2.0 (Intel® AVX2.0). The present code outperforms an earlier version that was itself optimized using Intel® Streaming SIMD Extensions 4.1 (Intel® SSE4.1), reducing cull time by up to 1.32x. Ongoing refinements continue to drive up performance, reducing draw time per frame for game developers who take advantage of it.

Download the sample code and supporting documentation at <https://software.intel.com/en-us/articles/software-occlusion-culling>

## Increased Graphics Settings

Beyond the addition of software features, the GPU and CPU performance headroom afforded by the Intel NUC Kit NUC6i7KYK allows for wholesale benefits in player experience by increasing engine-level graphics settings. For example, in the context of Unreal Engine\* 4, the game *Dreadnought*\*—a space-borne flight simulator developed by Yager Development and published by Grey Box with Six Foot—is an Unreal Engine 4 game that runs great on Iris Pro graphics.



Image credit: Grey Box

### ***Dreadnought\****

Testing with *Dreadnought* on Iris Pro graphics shows that it is possible to increase the quality level from 720p low or medium settings to 1080p medium or high settings. To achieve this in Unreal Engine 4 , call **SetFromSingleQualityLevel** in the **GameUserSettings** class and set the value to 1 for medium and 2 for high.

[Click here to learn more about setting quality settings in Unreal Engine 4.](#)

In a typical game on the Unity\* engine, such as *Kerbal Space Program\**, Iris Pro graphics supports quality settings of 4 or 5 at 1080p resolution. Graphics settings in the Unity engine are made by accessing the **QualitySettings** class and modifying **SetQualityLevel**.



**Kerbal Space Program\***

[Click here to learn more about setting quality settings in the Unity engine.](#) You can also use the code sample below in your game to create a script that sets your quality settings.

```
using UnityEngine;
```

```
using System.Collections;
```

```
public class SillyExample : MonoBehaviour {
```

```
    private int QualityValue;
```

```
    void QualityLevel(QualityValue) {
```

```
        QualitySettings.SetQualityLevel(QualityValue);
```

```
    }
```

```
}
```

Note that the appropriate setting in any engine for a specific circumstance will vary based on factors such as how many particle effects are in use, how many triangles are being rendered, and general scene complexity. To get the most out of your game, see how high you can set your graphics settings when you detect Intel® Iris™ Pro Graphics 580 in your game!

## Conclusion

With the Intel NUC Kit NUC6i7KYK, high-end gaming is now supported on a Mini PC form factor—not as a compromise, but as a full-featured platform. 6th gen Intel Core processors and Iris Pro graphics support advanced techniques for graphics and 1080P gaming, and their tiny footprint make for a mean developer machine.

We look forward to hearing about your adventures with the new Intel NUC and the games you are able to create by taking advantage of 6th gen Intel Core processors and Iris Pro graphics.

## Other Related References

<https://software.intel.com/en-us/blogs/2016/03/25/introduction-to-games-engineering-at-intel>

<https://software.intel.com/en-us/gpa/faq>

<https://software.intel.com/en-us/articles/Unreal-Engine-4-with-x86-Support>

<https://software.intel.com/en-us/articles/unity>

<https://software.intel.com/en-us/articles/unity-software-performance-optimizations-for-games-best-practices>

## About the Author

Landyn Pethrus is an engineer at Intel, avid gamer, and hardware enthusiast. When he is not fountain sniping with the Ancient Apparition in *Dota 2*\*, slaying bosses, or pursuing higher-level education, he can be found fishing on the rivers of Oregon.

## Notices

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at [intel.com](http://intel.com).

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting [www.intel.com/design/literature.htm](http://www.intel.com/design/literature.htm).

This sample source code is released under the [Intel Sample Source Code License Agreement](#).

Intel, the Intel logo, Intel Core, Iris, and Thunderbolt are trademarks of Intel Corporation in the U.S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

© 2016 Intel Corporation.