# 02562 - w10p4

*Emil Wraae Carlsen s204458 and Kristoffer Overgaard s194110*

Cycles and WebGPU serve two very different purposes when it comes to rendering. There are pros and cons for both choices.

**Cycles:**
Pros:

- Ease of Use: Cycles is integrated seamlessly into Blender, providing a simple interface for users. Its integration with Blender makes it accessible and convenient. A user will also reach a visible product much faster using Cycles.

- Online assistance: Blender has a large and active community, which means there are plenty of tutorials and support available for users of Cycles. This community support is incredibly valuable for users (such as ourselves).

Cons:

- Limited Customization: While Cycles is powerful and flexible, it does not offer the same level of customization as a low-level rendering API like WebGPU. It can be very difficult to obtain specific visual effects not yet supported by Cycles.

- Performance: Cycles is not as optimized for performance as some specialized renderers or low-level APIs. For large-scale, real-time applications, Cycles will not provide the speed and efficiency that WebGPU offers.

**WebGPU:**
Pros:

- Low-Level Control: WebGPU, being a low-level API, allows users to have more control over the rendering process. This is beneficial for implementing custom rendering pipelines and achieving specific visual effects.

- Performance: WebGPU provides high-performance rendering. Which makes it suitable for real-time applications, making it well-suited for games, simulations, and other interactive experiences.

- Scalability: WebGPU takes advantage of modern GPU architectures and efficiently utilizes multiple cores and threads. This scalability is crucial for handling complex scenes and achieving high frame rates in real-time uses.

Cons:

- Steep Learning Curve: Working with low-level APIs like WebGPU requires a much deeper understanding of graphics programming, making it way more challenging for users who are not familiar with coding or shader development.

- Integration: Unlike Cycles, which is seamlessly integrated into Blender, working with WebGPU may involve additional steps and considerations for integration into a specific software or pipeline.

We miss knowing exactly how our changes to specific parameters, code and shaders affect the result of the render. With Cycles there is no requirement to understand everything you do entirely. We aren’t always sure what actually happens, when we change parameters and sliders.

We like the fact that when rendering in Cycles it is not possible to get a compile error. You wil always be able to see how your changes affect the rendering. There is also live rendering, which makes setting up light sources much simpler. These benefits are simply impossible to achieve using WebGPU.