# Old stuff

GPUs have been traditionally accessed through industry APIs with most under the umbrella of the Kronos Group a non-profit organisation with multiple world leading companies advising and signing off on industry standards. OpenCL and OpenGL are two of their more well-known standards. Although there are other very popular APIs such as CUDA and DirectX that are not maintained by the Kronos Group they are for specific hardware and software. DirectX for example is only accessed through the Windows operating system.

OpenCL stands for open compute language. It is an API designed to allow programmers to process data in parallel across multicore devices such as GPUs. OpenCL is best used for computer vision and image processing where traditional algorithms can be paralysed to great effect. OpenCL also has a Web variant called WebCL which provides much of the same functionality to Web based applications. OpenGL on the other hand stands for open graphics library which aims to provide a pipeline with programmable elements. It’s used for high detailed, high resolution 3D rendering but can also be used for advanced 2D work and general graphics applications.

The section people learn and experiment the most with are shaders in OpenGL and kernels in OpenCL. These are custom built programs built to take data provided by the program and produce an output either on screen or to an output buffer. The platform we build will be providing a high level environment to mutate and prepare data sent to these programs, but will let the user write the custom shader and kernel programs which can then be imported into any application.

Before proceeding it is best to summarise why GPU programming is an important subject to address. In recent years GPU’s have become more mainstream. If you for instance look at the latest 6th generation intel processors, they all ship with integrated GPU chips on the die (Intelcom, 2016). As such most computers which ship with an Intel processor now have a high performing GPU chip waiting to be utilized.