3D GRAPHICS
Mastering the Art of 3D Development

With the onset of pervasive and powerful mobile devices there's been an explosion of applications across every field. As a result, companies are finding that a stellar user experience – one that incorporates 3D – is the best way to make their app stand out. Virtual reality, augmented reality, immersive experiences, and 3D data visualizations are therefore finding their way into UI designs and product roadmaps everywhere, and GPUs are doing double-duty as powerful parallel co-processors in a diversity of fields. 3D isn't just for games anymore.

However, creating 3D applications requires specialized expertise with a different design mindset, new programming tools, complicated math, and a detailed understanding of the GPU pipeline and framework APIs. Regardless of whether your application is virtual surgery, CNC control, or geospatial mapping, your team may need expert training and consulting services to shorten the very long 3D learning curve. KDAB is ready to help with comprehensive experience in a wide range of 3D frameworks and tools.

Qt 3D

For developers already familiar with Qt for cross-platform development, Qt 3D is a natural choice. With its high-level APIs in C++ and QML, Qt 3D makes it relatively easy to add 3D content to Qt applications.

Getting beyond the basics however can be a bit tricky. Figuring out how to render something more advanced than the out-of-the-box basics or how to maximize performance in a real-time simulation can be very challenging. There's a right way and a wrong way to extend an application with physics, collision detection, audio, and AI; redoing lots of poorly designed code is never an easy option.

How we can help:

- Training courses on the basics of Qt 3D and 3D development
- Customized Qt 3D onsite workshops
- Qt 3D configuration, optimization, and debugging services on an assortment of desktop and embedded systems
- Integration services for adding Qt 3D into existing applications
- Custom classes, libraries, and frameworks that extend or interface with Qt 3D

OpenGL and OpenGL ES

As the first widely available cross-platform graphics standard, OpenGL and its smaller cousin OpenGL ES are still the best choice for displaying 3D graphics across the widest variety of platforms. With OpenGL, developers can create applications that target a powerful desktop as well as a tightly constrained embedded system. Thankfully there are hundreds of developer resources for OpenGL readily available.

Where we typically help customers is on the super hard problems – like optimizing the balance between CPU and GPU utilization, removing bottlenecks in fragment processing, or creating intricately detailed shaders. Whether you need to build your OpenGL app for X11 and Wayland, double your app's FPS, or build a uniquely architected graphics platform, we live for the challenge.

How we can help:

- Specialized GPU profiling tools for pipeline performance optimization
- Debugging services for difficult tasks like eliminating rendering issues and fixing problematic shaders
- Development services for new rendering frameworks, cross-platform libraries, and shaders using high-powered OpenGL techniques
- Hardware evaluations to validate your platform strategy and select roadmap devices
- Training on best practices for developing beautiful and efficient OpenGL applications
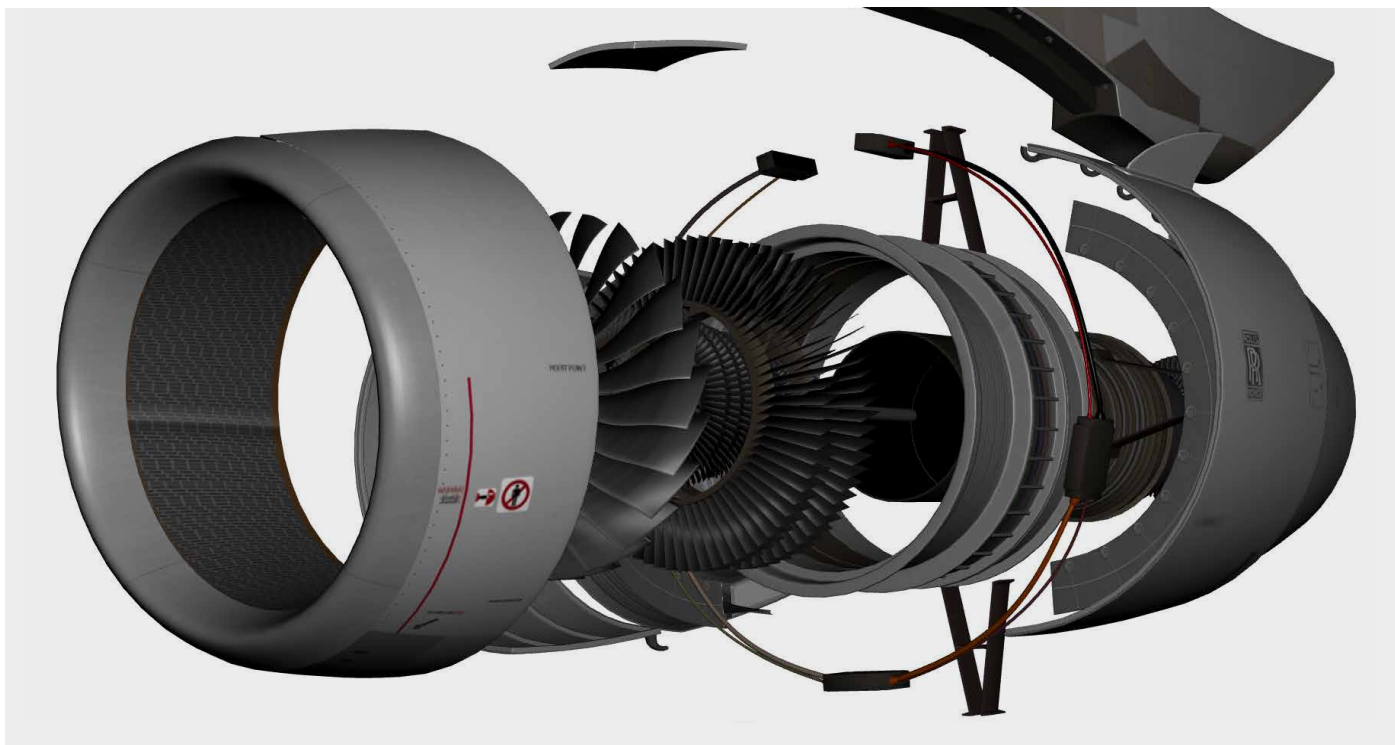
Creating an exploded view is straightforward with object-oriented model manipulation

Vulkan

At the time OpenGL was created, its designers sacrificed some GPU performance to achieve cross-platform compatibility. Vulkan was created to address this – it's a graphics standard that provides the bare-metal performance of Direct3D 12 and Metal with the multi-platform consistency of OpenGL. Vulkan was also designed to directly incorporate OpenCL-like features, opening up many new possibilities for parallel data processing.

Vulkan is still a relatively new beast with a good learning curve, however, it is definitely Open GL's heir apparent for cross-platform 3D graphics APIs, especially for performance demanding applications. With the significant performance benefits it can bring, Vulkan is worth considering. We're at the leading edge when it comes

to integrating Vulkan into desktop and mobile applications and are here to help.

How we can help:

- Vulkan training, workshops, and mentoring
- Configuration, optimization, and debugging services for Vulkan apps on desktop, mobile and embedded systems

- Migration of your OpenGL legacy code to Vulkan or embedding Vulkan code in your application
- Development services for creating non-graphical GPU-accelerated apps that need big data processing and machine learning
- Consultations on how to increase battery life and control thermal profiles
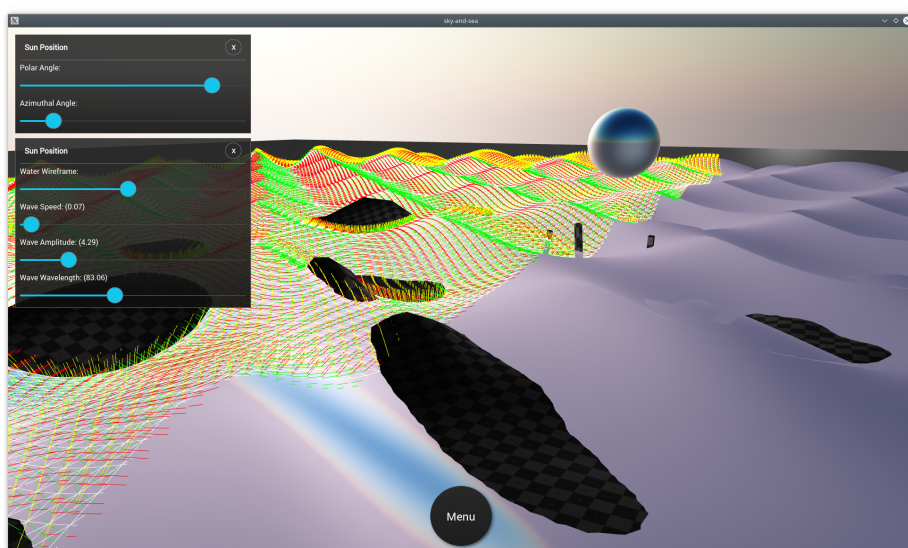
KDAB is an expert at creating 3D models that mimic natural phenomenon

Kuesa

The software behind 3D interfaces isn't easy to create. Much of that difficulty comes from the perennial disconnect between UX designers and software developers. While UX designers create UI visuals and mock-ups with tools that can finely control appearances, these tools produce graphical assets that are not directly usable by production software. That leaves the development team tasked with the non-trivial effort of turning a pile of pictures into a smoothly running and bug-free UX – typically without desktop hardware and processing power at their disposal.

KDAB specifically created Kuesa to eliminate this problem. Kuesa makes for an easy, integrated, and unified workflow, bridging the gap between designers and developers. Designers can continue to use their preferred tools to create 3D assets while developers can use Kuesa for previewing, conditioning, optimizing, manipulating, and displaying those assets with files directly from design tools.

How Kuesa can help

- Continuous integration workflow between design and development
- Support for professional 3D design tools like Autodesk 3ds Max and Blender
- Ability for designers to structure complex scenes with properties, layers, and animations that can be separately accessed and manipulated by software
- Direct integration of designed assets into C++ or Qt applications
- Photorealistic results that look identical in both design and development environments using Physics Based Rendering

PBR Expertise

Physical-based rendering (PBR) accurately simulates the behavior of light through the methodical application of reflection, scattering, and absorption physics. It provides enhanced realism with foolproof materials that look correct regardless of the lighting environment. This gives artists the ability to develop tantalizingly realistic 3D models and to define consistent properties for easier collaboration and sharing of object libraries. KDAB is at the forefront of embedded PBR rendering, providing the PBR rendering capability for Qt Automotive Suite and Qt 3D, and has the expertise to bring your projects into the physically rendered realm.

www.kdab.com

About the KDAB Group

The KDAB Group is the world's leading software consultancy for architecture, development and design of Qt, C++ and OpenGL applications across desktop, embedded and mobile platforms and is one of the biggest independent contributors to Qt. Our experts build run-times, mix native and web technologies, and solve hardware stack performance issues and porting problems for hundreds of customers, many among the Fortune 500. KDAB's tools and extensive experience in creating, debugging, profiling and porting complex applications help developers worldwide to deliver successful projects. KDAB's trainers, all full-time developers, provide market leading, hands-on, training for Qt, OpenGL and modern C++ in multiple languages. Founded in 1999, KDAB has offices throughout North America and Europe.