KUESA™
Bridging the Gap Between Design and Development

The software behind 3D interfaces isn't easy to create. Much of that difficulty comes from the perennial disconnect between UX designers, 3D artists, and software developers. While UX designers and 3D artists may collaborate to create visually impressive 3D models, the tools they use 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 3D assets and workflow diagrams into a smoothly running and bug-free UI—typically without desktop hardware and processing power at their disposal.

KDAB specifically created KUESA™ to eliminate this problem. KUESA makes for an easy, integrated 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.

Workflow for creating and using 3D assets

KUESA is a solution that provides a unified workflow for designers and developers to create, optimize, and integrate real-time 3D content from design tools into a 3D or hybrid 2D/3D software application. It is a workflow solution that consists of:

- **KUESA digital content creation (DCC) plugins** – plugins that augment 3D design tools, allowing them to create and export 3D scenes
- **KUESA tools** – utilities for designers and developers to preview, condition, sanity check, and optimize assets
- **KUESA runtime** – libraries on top of Qt and Qt3D for developers to integrate 3D scenes into their applications

How KUESA helps designers

With KUESA, designers can use familiar, standard professional 3D design tools such as Autodesk 3ds Max and Blender to create 3D scenes and animations. Because KUESA plugins seamlessly integrate into popular design tools, designers can also benefit from new capabilities:

- KUESA-supported physically based rendering (PBR), non-PBR, and node-based materials
- Support for building objects (geometries, cameras, materials, animations, lights, objects, particle engines, etc) that can be accessed and manipulated by developers
- Ability to save 3D scenes in glTF

How KUESA helps developers

Developers can use KUESA tools to inspect, preview, condition, sanity check, and optimize a project's 3D assets. Because the KUESA runtime is built on top of Qt 3D, developers can also benefit from seamless integration to Qt applications:

- Support for importing, managing and rendering glTF 3D assets as well as integrating 2D assets
- Utility to compress textures, meshes, and images for faster load times, reduced application size, and optimal use of GPU resources
- Optimized frame graphs for high-level descriptions of the OpenGL pipeline configuration
- Full access to scene items via collections
- PBR support for photo-realistic results consistent across design tools and applications
- Special effects like bloom or depth-of-field
- Tools that can be incorporated into a continuous integration system for consistent 3D asset verification and conditioning
- Runtime for both desktop and embedded platforms

KUESA is easy to integrate into a 3D or hybrid 2D / 3D Qt application with either a C++ or QML API. Developers gain access to the internals of the 3D scenes from code, allowing them to create user interactions, special effects, and customize for specific hardware.

KUESA versus other approaches

There are traditionally two main ways to integrate real-time 3D content into an application:

Workflow	Description	Drawbacks
Manual	Designers create videos of scenes and developers recreate them in code	• Creates unnecessary team overhead, communication, and work • Introduces translation errors and provides suboptimal results • Makes editing assets extremely difficult, which inhibits UX refinement
External tool	Designers and developers work on a common tool, editing scenes for integrating into an application	• Limits the creative process and application functionality to the constraints of a less-powerful tool • Designers and developers require retraining on unfamiliar tools • Restricts knowledge, debugging, and adaptation of system through black box software which cannot be modified

Physically Based Rendering (PBR)

PBR is able to approximate the appearance of real-world objects by considering the scientific properties of materials such as albedo, conductivity, and microfacets, while still rendering objects in real-time. KUESA uses PBR so that materials used in the 3D design tool and engine are displayed identically with photo-realistic results.
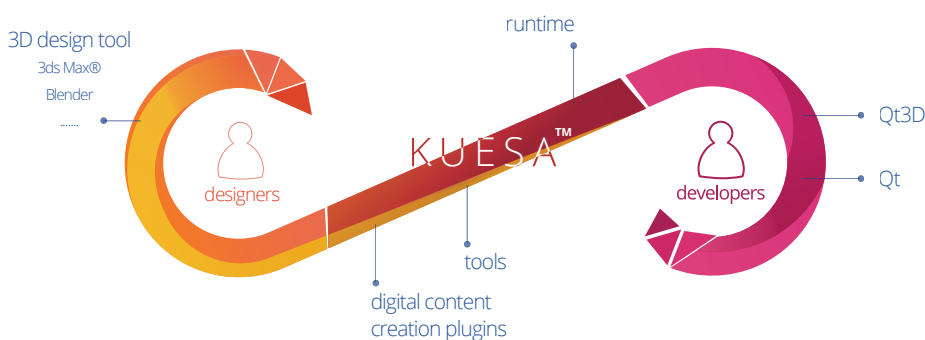
In contrast, KUESA provides an easy, integrated and unified workflow without any compromises for designers and developers giving:

· Great performance on both desktop and embedded boards
· High-quality real-time 3D scenes
· Full expressiveness for designers, using professional 3D design tools
· Full control of integration for developers
· Reduced time to market

K U E S A
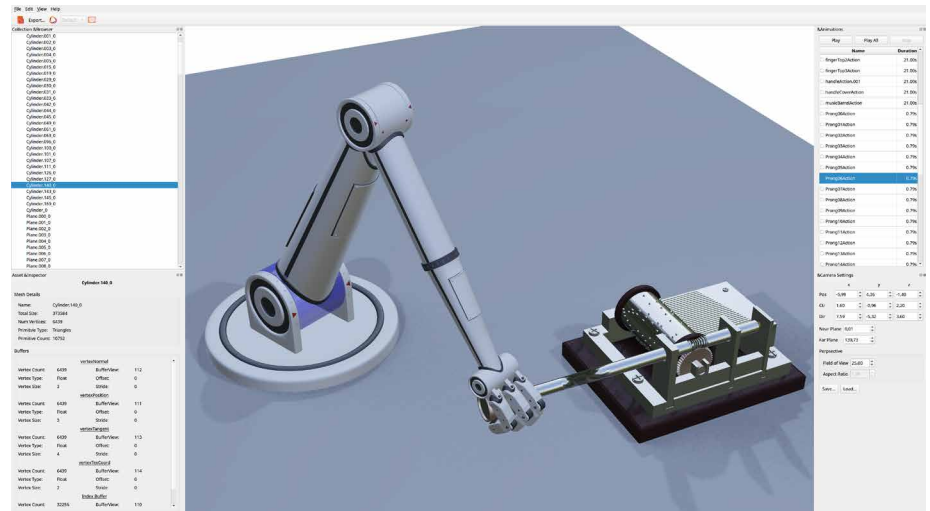
www.kdab.com

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.