OpenGL® ES*
Safety-Critical Profile Philosophy

Claude Knaus

July 5th, 2004

*OpenGL is a registered trademark, and OpenGL ES is a trademark, of Silicon Graphics, Inc.

1 Overview

The Safety-Critical profile of OpenGL is being defined to meet the unique needs of the Safety-Critical market which include:

- minimizing code size and complexity to ease certification
- to enable the porting of legacy applications that themselves have already been safety certified

This goal will be achieved by starting with the Common profile of OpenGL ES 1.0 and making minimal changes. The changes might include features that are introduced by future versions of the Common profile. No changes will be made to prevent implementations of the Safety-Critical profile over a hardware accelerated Common profile implementation.

Along with the specification, conformance tests will be provided to protect the integrity of the trademark.

2 Justification and Motivation

The avionic space has a strict certification process, described by the document RTCA DO-178B. It defines the guidelines for development of aviation software.

The automotive industry has so far no standard for certification of Safety-Critical software. It is expected that if implementations of an API and applications using them are certifiable by DO-178B, the API will be adopted by the automotive industry as well.

Safety-Critical applications are otherwise not expected to be as resource (memory and performance) constrained as much as other embedded systems such as mobile information devices.

The types of applications used in the avionic and automotive industry are visualization of:

- instruments (2D)
- maps (2D)
- terrain (3D)

- augmented reality (3D)

Since desktop OpenGL was not designed with Safety-Critical certification in mind, many avionic companies created a subset of OpenGL themselves, only including the features which they needed. As a result, many applications and OpenGL variants were created which have been Safety-Critical certified.

There is a need of creating an open standard OpenGL ES profile for Safety-Critical applications to enhance portability of Safety-Critical applications and to simplify Safety-Critical certification.

The OpenGL ES Common profile which has been created with mobile handsets in mind cover the visualization needs of applications used in the Safety-Critical space. However, the Common profile is partially still too complex to allow Safety-Critical certification. Another reason to have a separate profile is that existing applications which have already been certified, would otherwise require re-certification which is costly and may impact adoption of the profile.

The Safety-Critical working group will use the OpenGL ES 1.0 Common profile as a starting point to leverage the initial investment for the specification of the Common profile. The Safety-Critical profile will be kept as similar as possible to the Common profile to allow implementations of the Safety-Critical profile on top of a hardware accelerated Common profile implementation.

If the above stated requirements are met by the profile, the avionics companies are expected to adopt it as the single flavor of OpenGL for Safety-Critical applications. The automotive industry has been using OpenGL for non-safety-critical "back-seat" applications. They are expected to use this profile for safety-critical "front-seat" instrumentation when it becomes available.

3 Requirements in Detail

The requirements of the Safety-Critical profile in decreasing priority are as follows:

3.1 Minimize Legacy Code Change

Every application in the avionic space must be certified. The certification test is required to test every possible code path of a software. To ease certification and to reduce errors, large parts of applications are code generated. The code generators are certified (includes careful review), which may relax the certification process for the applications.

Code generators are very sensitive to any change. They would require re-certification, which in turn requires re-certification of the applications generated with the code generators. The company providing the code generator is not necessarily the same company writing the application.

3.2 Meet Functional Requirements of Avionic and Automotive Applications

Input on required features are provided by the Safety-Critical working group of The Khronos Group. Some of them may act as proxy to several avionic companies with experience in implementing OpenGL variants and using them.

Additionally, a review process with ISVs and OEMs of the avionic and automotive industry will provide feedback to assure validity of the included features.

3.3 Simplify Certification

3.3.1 Reduce Complexity

Certification tests have to be written in a way that they test every possible code path of an application or library they test.

The API must be carefully chosen, such that a combinatorial explosion of possible code paths is avoided to make writing and execution of certification tests possible at all.

To reduce the burden of implementation and certification testing, redundant features should be reduced to a minimum. Exceptions are cases where legacy code would be widely impacted.

Data format conversions in the implementation often require a lot of code paths to be tested. The goal of the API is to make data format conversion the responsibility of the application where possible.

Unused functionality is by definition redundant but would require testing as well. Careful selection of accepted parameter types to functions is required.

3.3.2 Guarantee Repeatability

System certification tests are typically executed multiple times to ensure repeatability. The API should therefore avoid undefined behavior where possible. The outcome of a legal operation should always be the same.

Decisions made by the implementation should be minimized. A feature is either supported or not supported for all implementations. Hints which may or may not be respected by an implementation are acceptable, provided that their exact behaviors are documented and guarantee repeatability.

3.3.3 Allow Compliance with Real-Time Requirements

The design of the specification should not preclude implementations to meet real-time requirements. Specifically, every function of a particular implementation of the API must be able to specify an upper bound limit of the time it takes to execute for a particular input and for any possible state. Certification tests might not (need to) test for it, but consumers of Safety-Critical software may demand real-time compliance.

Complex strategies such as memory management (display lists, texture) will need to be carefully implemented.

3.4 Relationship to OpenGL ES and Desktop OpenGL

The Safety-Critical profile will be kept as similar as possible to the Common profile to allow implementations of the Safety-Critical profile on top of a Common profile implementation.

The Common-Lite profile targets pure software implementations and systems without a floating point unit. Safety-Critical implementations are expected to be hardware accelerated and offer floating point units.

Safety-Critical applications are typically developed on desktop systems, using OpenGL. To ease development and early testing, the "OpenGL part" of a Safety-Critical application should be able to run with little or no modification on a desktop system. That is, the common function signatures, types and enums of the OpenGL ES Safety-Critical profile and desktop OpenGL (and its extensions) must have the same semantics. Extensions introduced by the Safety-Critical profile which do not overlap with desktop OpenGL and its extensions are excluded from this requirement.

4 Deliverables

The OpenGL ES Safety-Critical Profile 1.0 specification will comprise the following deliverables:

- specification document
- conformance test suite and process
- header files

The specification document is written against the OpenGL 1.3 specification. Annotated reasoning is written against the Common profile to emphasize their close relationship.

A suite of conformance tests is delivered to protect the integrity of the OpenGL trademark. Additionally, a conformance test process will be set up, similar to the existing one for the Common and Common-Lite profiles.

Header files are delivered for the convenience of the implementers.