Dragonfly: A High Level Low Overhead OpenGL Framework

Csaba Bálint[a], **Róbert Bán**[b]

[a]Eötvös Loránd University
csabix.balint@gmail.com

[b]Eötvös Loránd University
bundas@inf.elte.hu

Abstract

Computer graphics developers and researchers usually have to choose between graphics APIs that are high performing or frameworks that are high level. In the former case, built in debugging and developing tools are usually lacking. Current high level frameworks, however, are complicated to use or suffer performance overhead and sometimes are even harder to debug.

Dragonfly combines high performance and short client code while providing built-in debug tools. The framework achieves this via C++17 templates trading compilation time for the near-zero runtime overhead. Additionally, our framework can generate GUIs for most classes allowing runtime monitoring and debugging.

The abstraction encapsulates most OpenGL operations in four classes: Shader Program, Framebuffer, Texture, and Buffer. With Dragonfly, a single line of code can represent a pipeline of rendering commands. Furthermore, Dragonfly provides a variety of utility classes resulting in expressive code.

An application implementing deferred shading being two dozen lines of code demonstrates the productivity increase gained from using our framework. Prototype and production graphics applications may both be implemented in Dragonfly.

Keywords: Computer graphics, OpenGL, Framework