

**FERRAMENTA DE DEPURACÃO PARA A LINGUAGEM GLSL.** *Barbara Bellaver Goncalves, Manuel Menezes de Oliveira Neto (orient.) (UFRGS).*

Com a recente evolução das placas gráficas, suas arquiteturas passaram a apresentar alto grau de paralelismo e flexibilidade de programação. Os programas desenvolvidos para esta classe de hardware programável são chamados de shaders. As linguagens de programação de shaders também evoluíram, passando de Assembly para linguagens de mais alto nível, como CG, HLSL e GLSL. CG e HLSL são linguagens proprietárias bastante similares. Já GLSL, é uma extensão de OpenGL, que constitui um padrão aberto. Comparando as ferramentas existentes para depuração e análise de código destas linguagens, percebe-se que as ferramentas de GLSL ainda requerem aprimoramento para que a linguagem possua as mesmas facilidades de programação de linguagens de propósitos gerais ou mesmo das demais linguagens de programação de shaders. Isto faz com que desenvolvedores de software ainda gastem uma grande quantidade de tempo localizando e depurando erros de programação. Assim, o objetivo deste trabalho é desenvolver uma ferramenta que permita ao usuário depurar tanto o shader quanto o programa hospedeiro, facilitando o desenvolvimento integrado da aplicação. Para isto, estão sendo desenvolvidos: (i) uma biblioteca de funções para interceptar as chamadas OpenGL durante a execução do programa, (ii) um parser de GLSL para criar uma representação intermediária que permita identificar a ordem de execução do programa, (iii) um mecanismo para determinação do escopo de variáveis e manipulação correta do código do shader e, (iv) uma interface gráfica para controlar a execução da aplicação e do shader alvo. Uma vez que o depurador tenha sido finalizado, pretende-se, como trabalho futuro, integrar esta ferramenta com outras existentes permitindo ao usuário não apenas depurar, mas também editar e visualizar o shader.