



Ferramenta de Depuração para a Linguagem GLSL

Bárbara B. Gonçalves

Manuel M. Oliveira

{bbgoncalves, oliveira}@inf.ufrgs.br



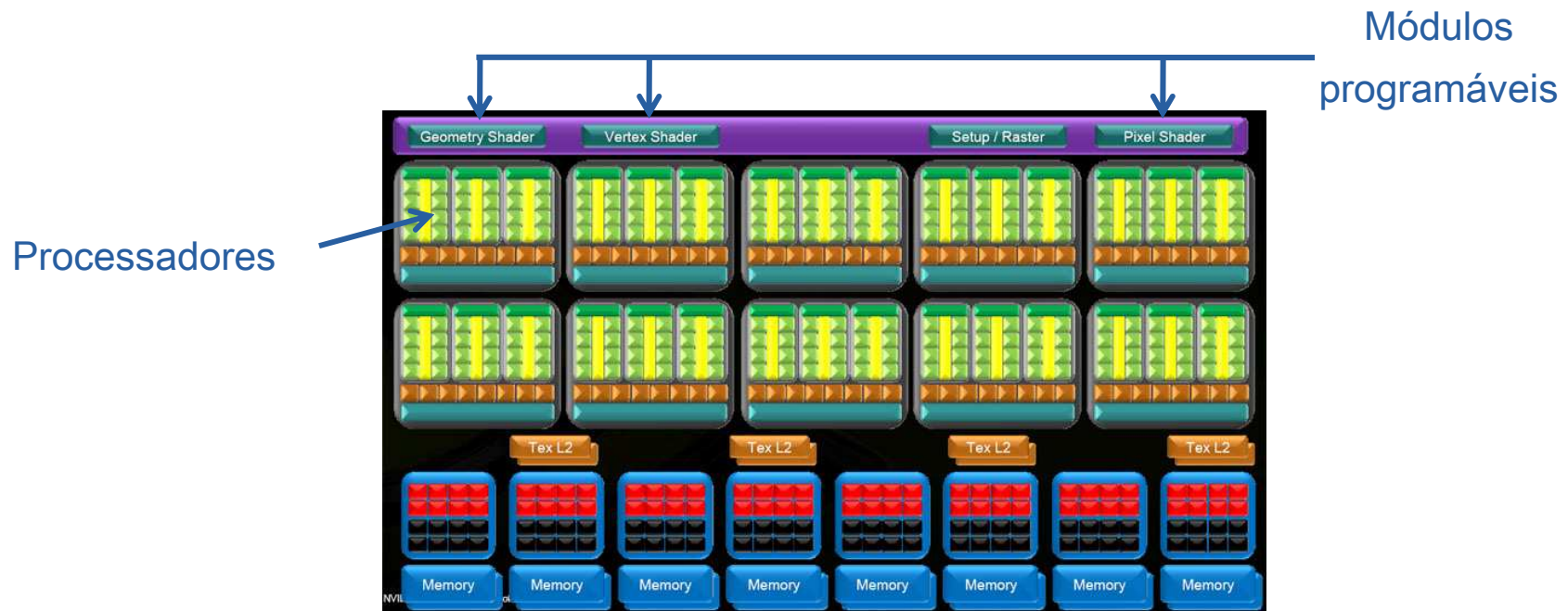
Informática
UFRGS



Microsoft®

Introdução

- Evolução das placas gráficas (GPU)
 - Alto grau de paralelismo
 - Flexibilidade de programação



Introdução (cont.)



Shaders

- Programas desenvolvidos para GPUs.



Shaders



```
void main()
{
    vec3 rr = vec3(0.5,0.5,0.5);
    vec3 rl = vec3(0.3,0.6,0.7);

    vec3 color = mix(rr, rl);

    gl_FragColor = vec4(color, 1.0);
}
```

Programação de Shaders

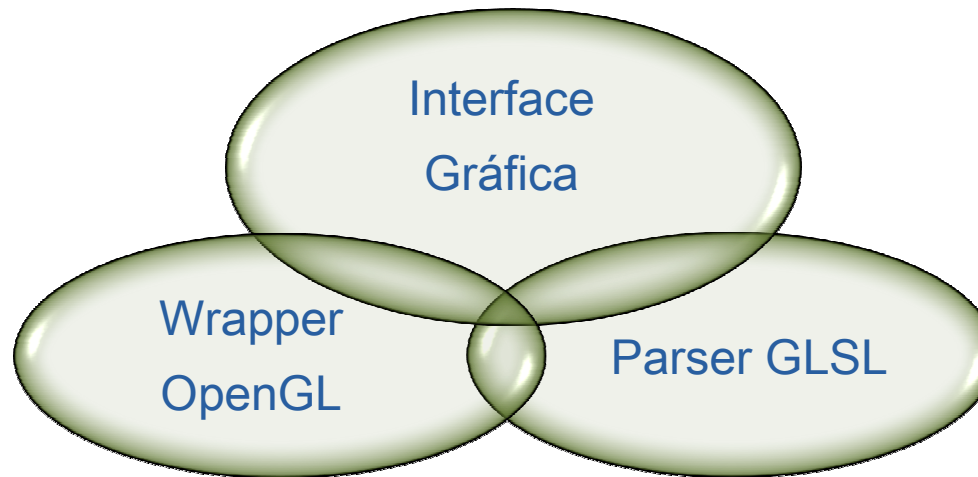
- Evolução similar à programação de CPU
 - Assembly → Alto Nível
- Linguagens
 - CG, HLSL e GLSL
- Dificuldades atuais:
 - GPUs não são de propósitos gerais
 - Ferramentas de desenvolvimento deficitárias
 - Editores e depuradores separados, por exemplo

Objetivos

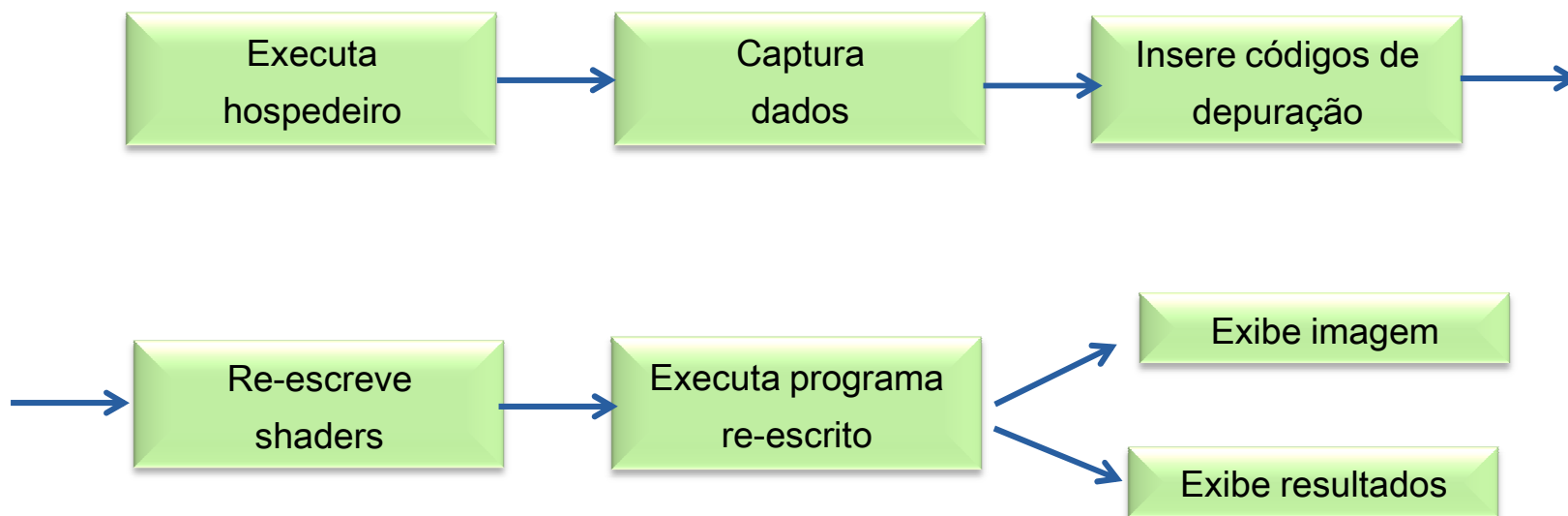
- Desenvolver um depurador de shaders GLSL
 - GLSL é um padrão aberto
 - Poucas ferramentas existentes
 - Multiplataforma

Implementação

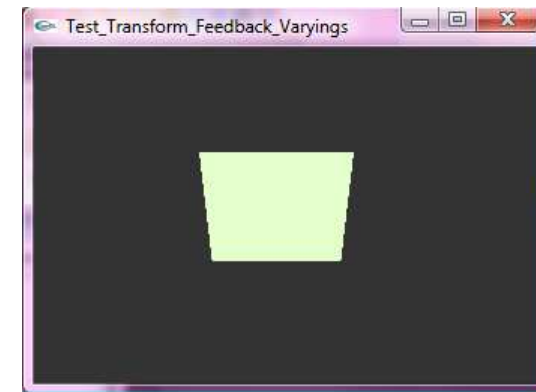
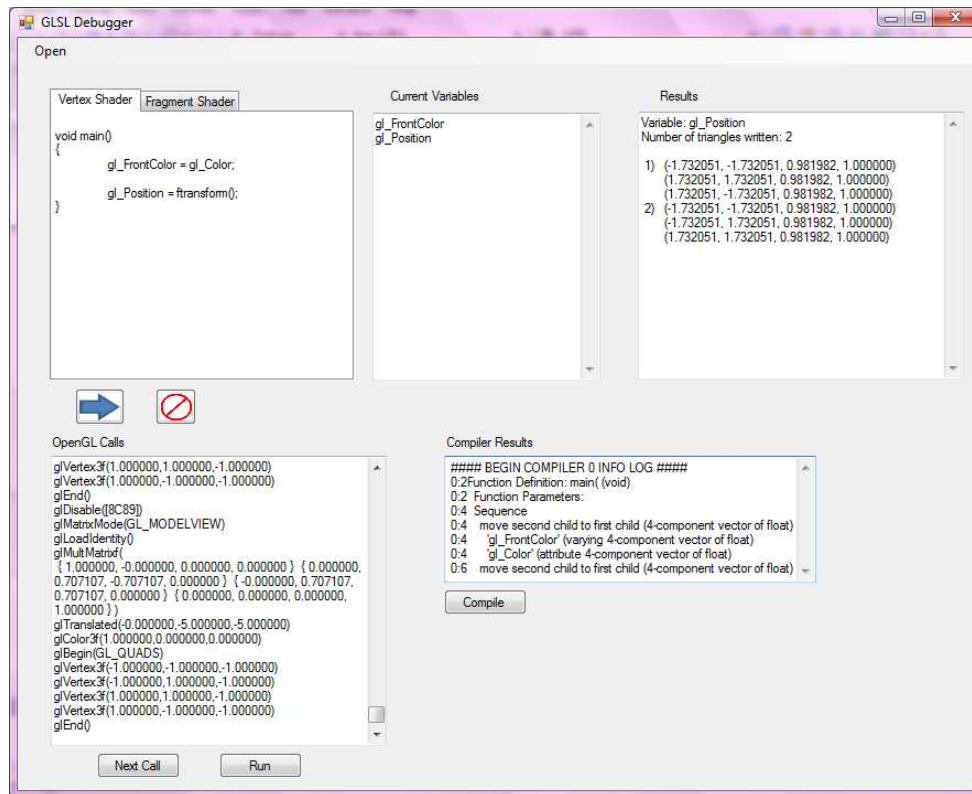
- Baseado na ferramenta GLSLDevil
- É um projeto de código aberto



Implementação



Resultado

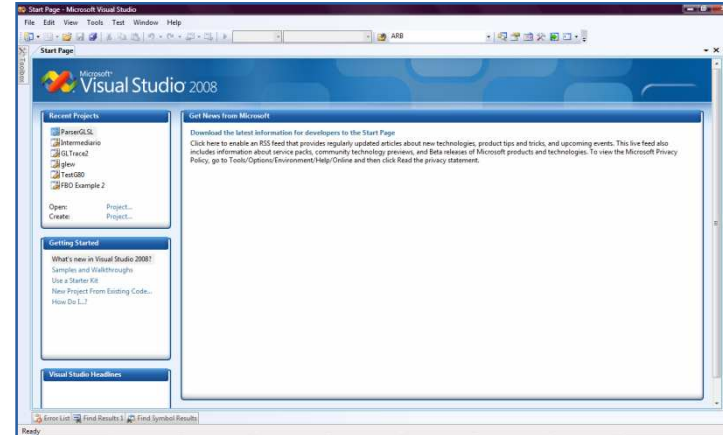
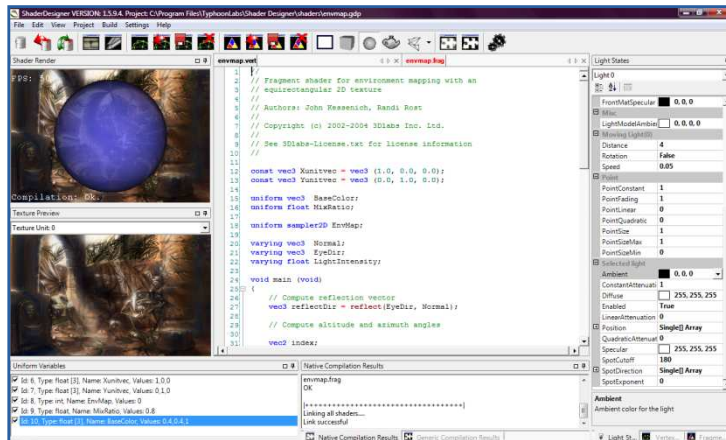


Conclusão

- Ferramenta de depuração
 - Programa OpenGL hospedeiro
 - Shader GLSL
- Desenvolvimento integrado da aplicação

Trabalhos Futuros

- Integração com ferramentas existentes
 - Shader Designer
 - Microsoft Visual Studio
- Ambiente único de desenvolvimento



Perguntas?

