Deshader

GLSL Shader Debugging Toolkit

Excel@FIT 2024
BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Bc. Ondřej Sabela
Supervisor: Ing. Tomáš Milet, Ph. D.

How to use

1a Inject Deshader into an application

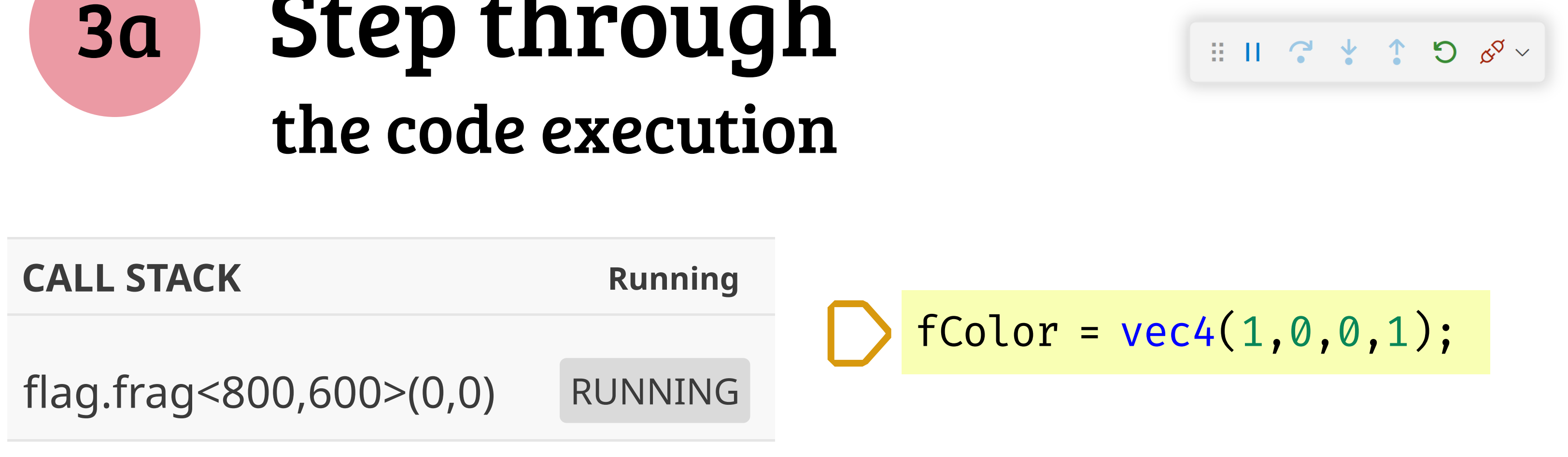
```
$ deshader-run ./the-app
```

→ use `#pragma deshader`, `glDebug`/`Object`, C API, HTTP/WS control interfaces, or VSCode!

2a Manage the shaders

```
const char* label = "flag.frag";
glObjectLabel(GL_SHADER, sh3, strlen(label), label);
glDebugMessageInsert(…,0xDE5ADE,"/>/home/user/shaders",…);

#pragma deshader source flag.frag
#pragma deshader workspace />/home/user/shaders
```

3a Step through the code execution



CALL STACK Running
flag.frag<800,600>(0,0) RUNNING

```
fColor = vec4(1,0,0,1);
```

3c Source code rewriting

```
#version 460
layout(points) in;
layout(TRIANGLE_STRIP, max_vertices = 4) out;
out vec2 flagPos;
float offsetX = 0.3, offsetY = 0.2;
```

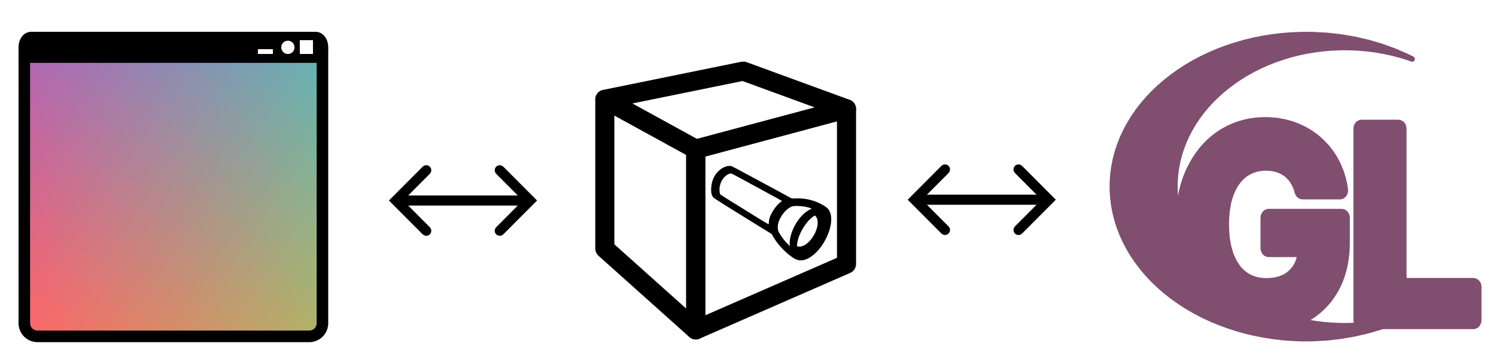
— Added variables

```
uniform float _targetSteps, _bpOffset;
float _step = 0, _bpI = 0, _bpHit = 0;
layout(binding = 0) buffer _logBuffer {uint _cursor; uint _logs[];};
//  … similar buffers for variable watches, stack trace
```

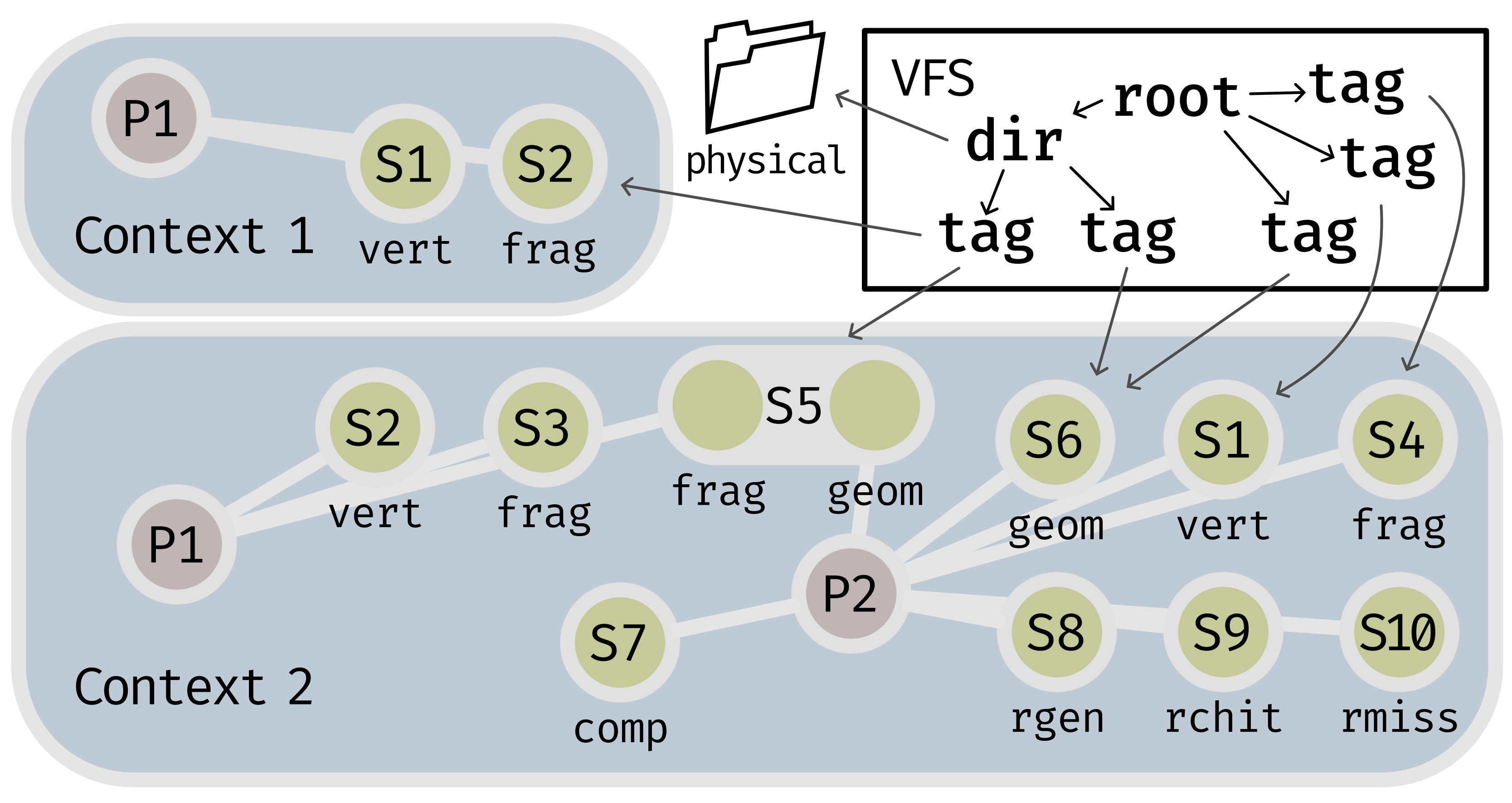
```
void quadVertex(vec2 offset) {
gl_Position = gl_in[0].gl_Position + vec4(offset, 0, 0);
EmitVertex();
}

void main() {
flagPos = vec2(0, 0);
quadVertex(vec2(-offsetX, -offsetY));
#pragma deshader print "%f" flagPos.x
flagPos = vec2(1, 0);
quadVertex(vec2(offsetX, -offsetY));
…
```

How is it accomplished

1b shared library Interception

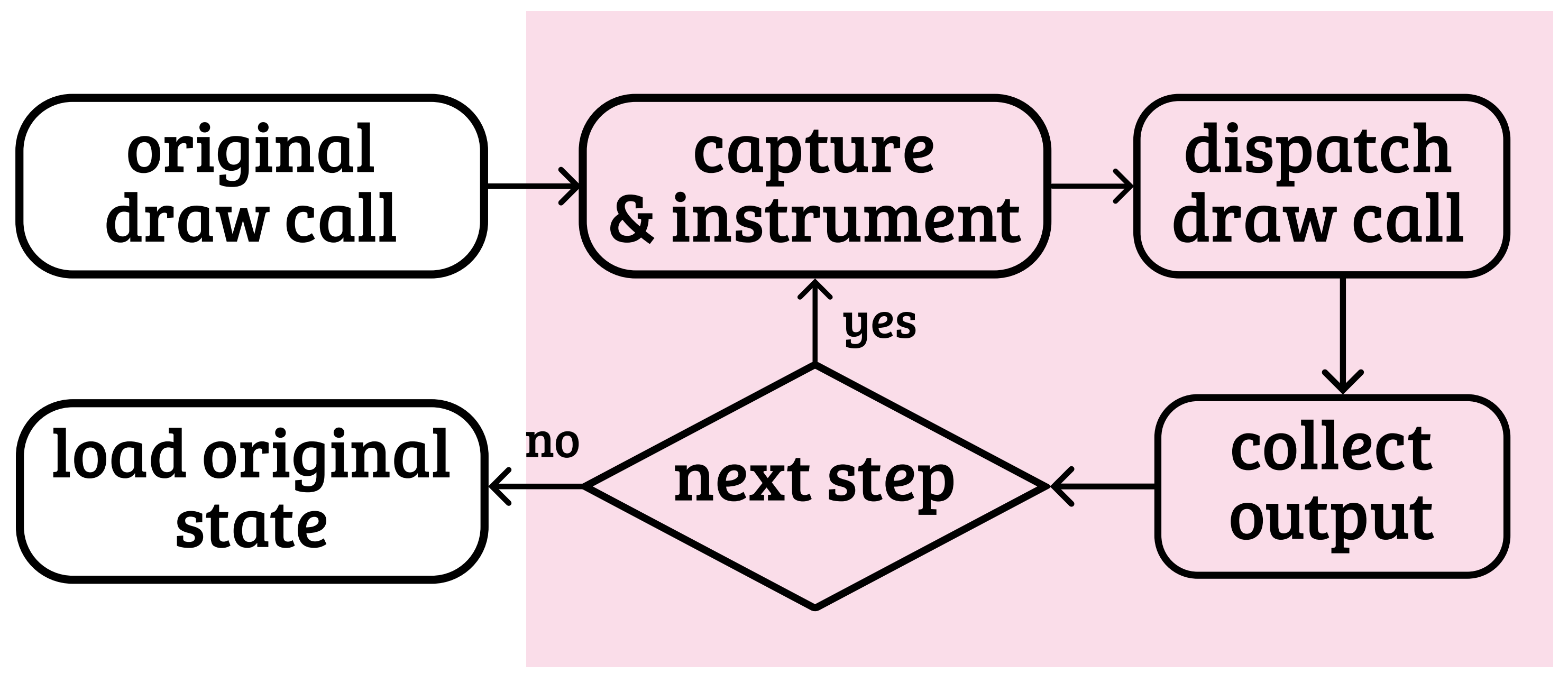


2b Hybrid virtual file system



3b Instrumentation state capture & shader analysis

Debugging loop —



3d Markers

step	`if(_step++ ⩾ _targetSteps) {return;}`
● breakpoint	`if(_bpI++ ⩾ _bpOffset) {_bpHit=[bpID]; return;}`
guard	`if(_bpHit > 0) {return;}`
log	`_logs[atomicAdd(_cursor,1)]=[charCode]`