Post OpenGL

Casual graphics development

COLLABORA

Erik Faye-Lund

XDC 2022

Open First

COLLABORA

OpenGL vs Vulkan

OpenGL is fairly easy

```
$ cat minimal-opengl-example.c
#include <GL/glut.h>
void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_TRIANGLES);
    glColor3f(1, 0, 0);
    glVertex3f(-1, -0.75, 0);
    glColor3f(0, 1, 0);
    glVertex3f(0, 0.75, 0);
    glColor3f(0, 0, 1);
    glVertex3f(1, -0.75, 0);
    glEnd();
    glutSwapBuffers();
}
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutCreateWindow("Hello, world!");
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
$ wc -l minimal-opengl-example.c
22 minimal-opengl-example.c
```

Vulkan… not so much

…I'm not even going to show any code here, because it'll be pointless.

```
$ wc -l minimal-vulkan-example.c
546234 minimal-vulkan-example.c
```

…ok, maybe not **that** bad, more realistically around 1000 lines of code.

OpenGL vs Vulkan

OpenGL

- Easy to get started
- Relatively easy to get right
- Strong ecosystem
- Straight-forward-ish API
- No need to worry about barriers
- **Doesn't expose modern features**

Vulkan

- Hard to get started
- Difficult to get right
- Ecosystem still has a long way to go
- Feels like filling out custom-forms
- Manual barrier placement is tedious
- **Has all the new GPU features!**

COLLABORA

How can we bring back the FUN?

Some alternatives

- Use pre-existing middleware
- Create new, hopefully better middleware
- Expose new features in OpenGL using Zink

Existing middleware: bgfx

Pros:

- Provides both C and C++ API
- Actively maintained

Cons:

- More of a graphics API abstraction

Existing middleware: V-EZ

Pros:

- Targets Vulkan directly
- Deals with annoying things like barrier placement

Cons:

- Practically speaking abandoned
- C++ API

Create something new?

Here's what's needed to make things less tedious:

- Higher level abstractions to create pipelines, render-targets, textures, command buffers, etc
- Helpers for “automatic” barrier placement
- Automatic / easy memory allocations
- Fixed function shaders!
- Things like vertex streamer helpers

Add new features to OpenGL with Zink

- Leverages the existing ecosystem
- Probably needs to link entire GL stack into the application
- Requires creating extensions that the rest of the community doesn't want or need?
  - Would probably be highly Zink-specific...

COLLABORA

Thouhgts? Let’s discuss!

Thank you!

We are hiring

col.la/careers