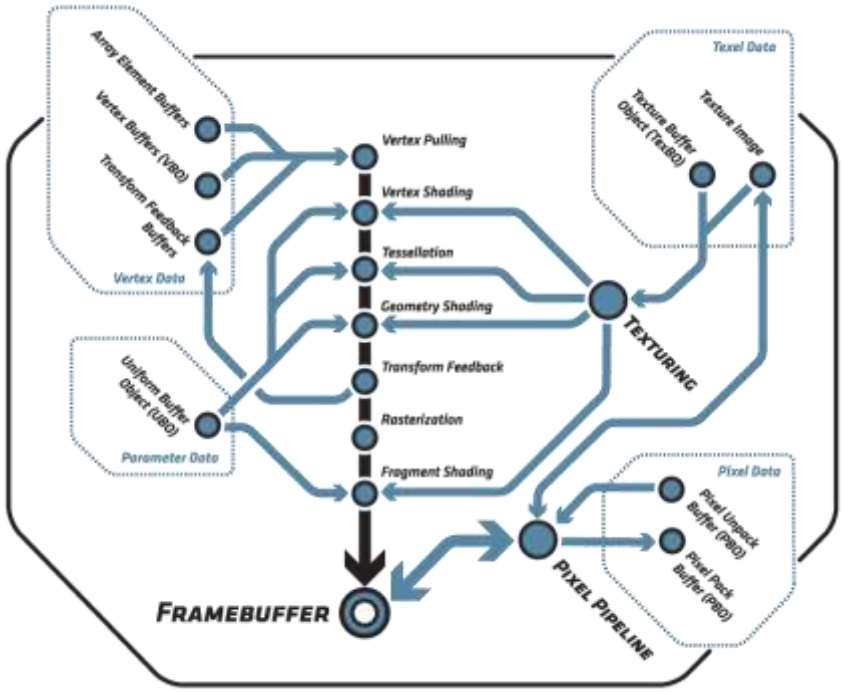


KHRONOS GROUP

From DirectX® to OpenGL

Siggraph 2013

OpenGL®

Karl Hillesland

AMD

KHRONOS GROUP

Overview

- Our use of DirectX and OpenGL
- How to help moving

Sushi DX

Ruby (DX9, DX10)

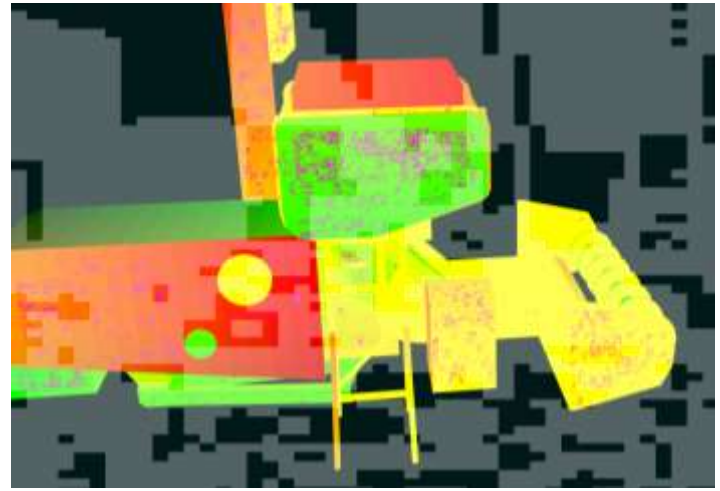
Leo (DX11)

Sushi GL

Ptex
AMD_SparseTexture

TressFX w/Mutex

Shader Demand Paging



Why Another API?

- Access to new features
- Platform support
- New perspectives

```
// Tessellation Control
layout (vertices = 4) out;
void TCS(void)
{
   if (gl_InvocationID == 0)
   {
      gl_TessLevelInner[0] = 2.0;
      …
```

```
// Hull Shader
[outputcontrolpoints(4)]
[patchconstantfunc("ConstantsHS")]

[domain("quad")]
[partitioning("integer")]
[outputtopology("triangle_cw")]

HS_OUTPUT HullShader(…)
```

```
// Tessellation Evaluation
layout (quads, cw, equal_spacing) in
void TES(void)
{
   …
```

```
// Domain Shader
DS_OUTPUT DomainShader(…)
```

OpenGL 4.0 | D3D11

DirectX Programmers

- **Who are they?**
  - Mostly game developers
  - There are a lot of them
- **What do they know?**
  - Real-time rendering
  - GPUs and drivers
  - OpenGL (ES) 1 or 2?
- **Similar crowds**
  - PS3™ developers
  - OpenGL -> DX

All of these skills are easily transferable

OpenGL for DirectX Programmers

- **They don't need basic concepts**
- **Don't even need advanced *concepts***
- **What do they need?**
  - Translations
  - Concept differences

- **Course at Siggraph Asia 2012**
  - Short (1:45)
- **With tweaks for Siggraph Asia 2013**
- **GDC?**

SIGGRAPH ASIA 2012

KHRONOS GROUP

Translation Table: Draw Calls

OpenGL	D3D
glDrawArrays	Draw
glDrawArraysInstanced	DrawInstanced(…,0)
glDrawArraysInstancedBaseInstance	DrawInstanced
glDrawArraysIndirect	DrawInstancedIndirect
glMultiDrawArrays	for(int i=0; i<n; ++i) Draw(count[i], start[i]);
glMultiDrawArraysIndirect	for(int i=0; i<n; ++i) DrawInstancedIndirect(…)
glDrawElements	DrawIndexed
…And so forth	

Translation and Concept

```
// Tessellation Control
layout (vertices = 4) out;
void TCS(void)
{
   if (gl_InvocationID == 0)
   {
      gl_TessLevelInner[0] = 2.0;
      …
```

```
// Hull Shader
[outputcontrolpoints(4)]
[patchconstantfunc("ConstantsHS")]

[domain("quad")]
[partitioning("integer")]
[outputtopology("triangle_cw")]

HS_OUTPUT HullShader(…)
```

```
// Tessellation Evaluation
layout (quads, cw, equal_spacing) in
void TES(void)
{
   …
```

```
// Domain Shader
DS_OUTPUT DomainShader(…)
```

OpenGL 4.0

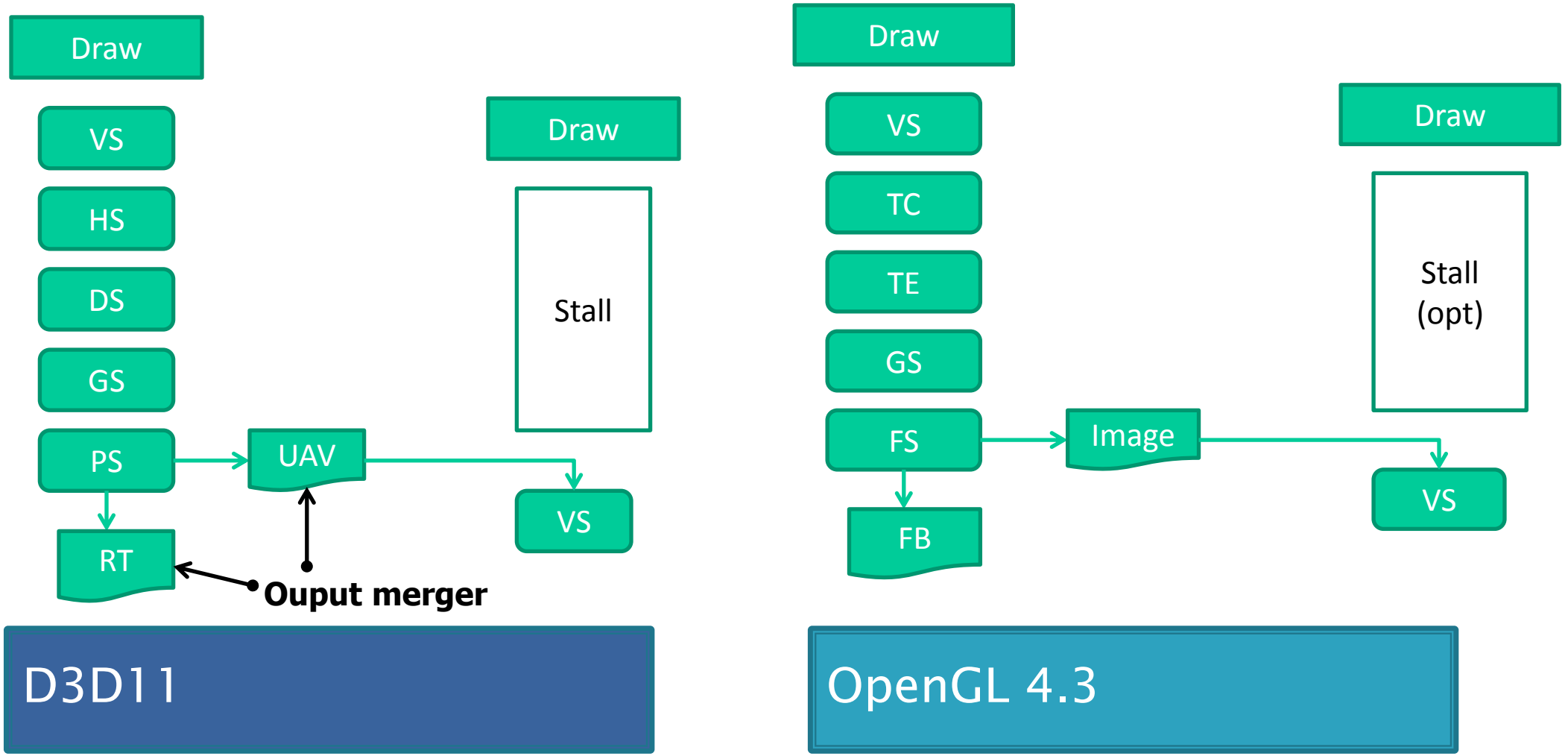
D3D11

Concepts

- Bind Semantics
- Program Objects
- "Views"
- UAVs ≠ Render Target

- TexStorage
- ~~TexImage~~

UAV vs Image



Wrap Up

It's worth targeting the specific needs of DirectX programmers interested in OpenGL

Thank You!