CUDAGL

CUDA Based Graphics Library

Chenshan Shari Yuan and Muhammad Osama
(github.com/neoblizz/cudagl)

Motivation: Optimizing Graphics Pipeline w/ Compute
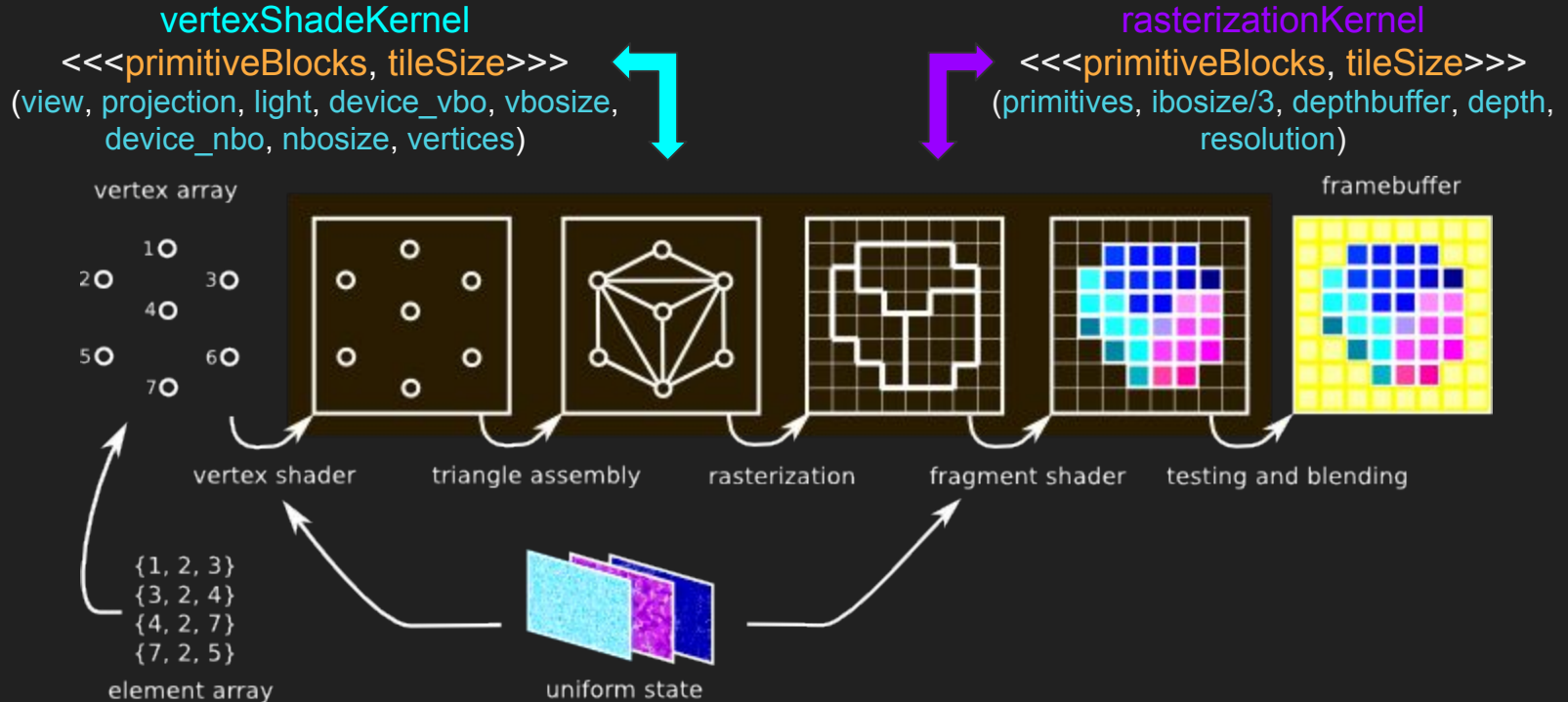
Designing a console:

- Limited space
- Cost restrictions
- Leads to limited hardware capabilities

System	Playstation 4	Discrete Card (AMD)
ALU Operations/Cycle	2,304	8,192
Shader Engines/Cycle	2	4
ALU Operations/Triangle	1,017	2,048

Approach: CUDA Kernels <<<Graphics Pipeline>>>

vertexShadeKernel
<<<primitiveBlocks, tileSize>>>
(view, projection, light, device_vbo, vbosize, device_nbo, nbosize, vertices)

rasterizationKernel
<<<primitiveBlocks, tileSize>>>
(primitives, ibosize/3, depthbuffer, depth, resolution)

vertex array

1
2
3
4
5   6
7

vertex shader   triangle assembly   rasterization   fragment shader   testing and blending

framebuffer

{1, 2, 3}
{3, 2, 4}
{4, 2, 7}
{7, 2, 5}

element array

uniform state

Results: Drawing cool looking models!

- Loading OBJ File: ../objs/cow.obj

- Loaded 5804 faces, 4583 vertices from ../objs/cow.obj