Cross-Platform Vulkan with GLFW

http://web.engr.oregonstate.edu/~royl/vulkan-glfw.odp

Lance Roy
Vulkan is Headless

- Vulkan API provides no way of drawing to the screen
- Need to use extensions
  - VK_KHR_surface
  - VK_KHR_swapchain
- Need to get a surface
  - This requires platform dependent extension.
- Creating a window is also platform dependent.
XCB

- VK_KHR_surface, VK_KHR xcb_surface
- Need to give Vulkan a window
- Similar for XLib

```c
typedef struct VkXcbSurfaceCreateInfoKHR {
    VkStructureType             sType;
    const void*                 pNext;
    VkXcbSurfaceCreateFlagsKHR  flags;
    xcb_connection_t*           connection;
    xcb_window_t                window;
} VkXcbSurfaceCreateInfoKHR;

VkResult vkCreateXcbSurfaceKHR(
    VkInstance                                  instance,
    const VkXcbSurfaceCreateInfoKHR*            pCreateInfo,
    const VkAllocationCallbacks*                pAllocator,
    VkSurfaceKHR*                               pSurface);
```
Win32

- VK_KHR_surface, VK_KHR_win32_surface
- Same idea, but need a Win32 window.

```c
typedef struct VkWin32SurfaceCreateInfoKHR {
    VkStructureType                 sType;
    const void*                     pNext;
    VkWin32SurfaceCreateFlagsKHR    flags;
    HINSTANCE                       hinstance;
    HWND                            hwnd;
} VkWin32SurfaceCreateInfoKHR;
```

```c
VkResult vkCreateWin32SurfaceKHR(
    VkInstance                                  instance,
    const VkWin32SurfaceCreateInfoKHR*          pCreateInfo,
    const VkAllocationCallbacks*                pAllocator,
    VkSurfaceKHR*                               pSurface);
```
Window Creation Library

- We need to do the same thing on all platforms
- We can do it in the same way on all platforms
- Seems like a library would be useful
GLFW

• Cross-platform
  – Windows
  – Linux
    • X11
    • Wayland
    • Mir
  – Mac (through MoltenVK)

• Cross-API
  – Vulkan
  – OpenGL
GLFW: Initialization

- Initialize GLFW
- Create the window.

```c++
glfwInit();
// ...
MainWindow = glfwCreateWindow( Width, Height, "Vulkan Sample", NULL, NULL );
```

(from sample.cpp)
GLFW: Vulkan

- Get the required extensions
  - Pass them into `vkCreateInstance`

- Create the Vulkan surface.
  - Can then be used to create the Swapchain.

```c
int glfwVulkanSupported();

const char**
glfwGetRequiredInstanceExtensions(uint32_t* count);

VkResult glfwCreateWindowSurface(VkInstance instance,
GLFWwindow *window,
const VkAllocationCallbacks *allocator,
VkSurfaceKHR *surface);
```
Mac

Glossary

**adjacency primitive** One of the primitive topologies that includes, for each primitive, additional vertex data representing adjacent primitives in the original geometry. Examples include triangles and lines with adjacency.

**aliasing** Technically, the loss of signal information in an image reproduced at some finite resolution. It is most often characterized by the appearance of sharp, jagged edges along points, lines, or polygons due to the nature of having a limited number of fixed-size pixels.

**alpha** A fourth color value added to provide a degree of transparency to the color of an object. An alpha value of 0.0 means complete transparency; 1.0 denotes no transparency (opaque).

**ambient light** Light in a scene that doesn’t come from any specific point source or direction. Ambient light illuminates all surfaces evenly and on all sides.

**anti-aliasing** A rendering method used to smooth lines, curves, and polygon edges. This technique averages the color of pixels adjacent to the line. It has the visual effect of softening the transition from the pixels on the line and those adjacent to the line, thus providing a smoother appearance.

**Apple** An apple is a piece of fruit. Fruit does not support Vulkan.

**ARB** Acronym for the Architecture Review Board, the committee body consisting of 3D graphics hardware vendors, previously charged with maintaining the OpenGL Specification. This function has since been assumed by the Khronos Group. See Khronos Group.
Metal

- Apple wants developers to use their own low-level graphics API
- No Official Vulkan support
- Third party, commercial compatibility layer called MoltenVK
  - GLFW support
Mobile

- Vulkan works on Android
  - Not supported by GLFW (yet).
Making sample.cpp work on Linux

- http://web.engr.oregonstate.edu/~royl/sample-vulkan-linux.patch
- I will quickly run through the changes
- Compilation Command:

  ```
g++ -ggdb sample.cpp -fpermissive --std=c++11 -lvulkan -lglfw -o sample
  ```