OpenGL and Vulkan Interoperability

Eleni Maria Stea <estea@igalia.com>

igalia

XDC 2020

Outline

1. OpenGL interoperability extensions

2. Additions to Mesa and Piglit

3. Questions

4. References

Outline

1. OpenGL interoperability extensions
2. Additions to Mesa and Piglit
3. Questions
4. References

Interoperability Concept

INTEROPERABILITY

EXT_external_objects(_fd) extensions enable:

- **Allocation** of resources (textures, buffers) from Vulkan and **sharing** of those resources from both APIs
- **Synchronization** of the access to shared resources between APIs.

Example: a VR compositor where the left and right eye images are allocated and displayed by Vulkan but their contents are filled by OpenGL processes.

OpenGL extensions for interoperability on Linux

OpenGL Extensions (Linux)

EXT_external_objects_fd (EXT_memory_object_fd, EXT_semaphore_fd):

- enables an OpenGL application to import a memory object or semaphore from POSIX file descriptor external handles

EXT_external_objects (EXT_memory_object, EXT_semaphore):

- enables an OpenGL application to create OpenGL objects corresponding to Vulkan objects (textures, buffers, semaphores) and synchronize access to shared objects across APIs
- extends existing GL calls that set and query properties

Outline

1 OpenGL interoperability extensions

2 Additions to Mesa and Piglit

3 Questions

4 References

Igalia’s contributions

Work on the feature

Mesa

Igalia contributed to the implementation of the interoperability extensions for different **Mesa drivers**.

Piglit (drivers testing framework)

As there were **no conformance tests** available for the extensions, we’ve written a Vulkan framework for **Piglit** and Vulkan/GL tests that cover some common EXT_external_objects(_fd) use cases.

Additions to Mesa (the details)

Contributions to Mesa drivers

Driver	Igalia's Contributions	Status
iris	**implemented** EXT_external_objects, EXT_external_objects_fd	under review (MR !4337)
freedreno	**implemented** EXT_external_objects, EXT_external_objects_fd	upstream
radeonsi	**fixes** to the already implemented feature	work in progress (WIP MR !6364)
i965	helped with the **review**, patches	under review (Draft MR !5594)

Contributors, reviewers, people who helped!

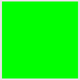
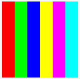
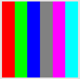
Driver	Patches	Reviews, Comments, Advice
iris	Juan A. Suárez Romero, Eleni Maria Stea, Tapani Pälli, Eduardo Lima Mitev	Tapani Pälli, Rohan Garg, Eric Engestrom, Kenneth Graunke, Jason Ekstrand, Oscar Barenys, Daniel Stone, Matt Turner, Tomeu Vizoso
freedreno	Eduardo Lima Mitev, Hyunjun Ko	Rob Clark, Marek Olšák, Jonathan Marek, Rohan Garg
radeonsi	The feature was implemented by AMD a few years ago and we are mostly working on fixes (WIP).	feedback from: Bas Nieuwenhuizen, Marek Olšák
i965	Rohan Garg, Juan A. Suárez Romero, Tapani Pälli	Tomeu Vizoso, Eleni Maria Stea, Oscar Barenys

- The information above was accurate at the beginning of September when I pre-recorded this presentation! Apologies if I've not included someone or something!

Additions to Piglit (the details) I

Contributions to Piglit I

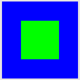
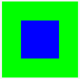
Reusing color images:

Test name	Description	Output	Status
vk-image-overwrite	• **Vulkan:** allocates images of different formats and tiling modes • **OpenGL:** fills them with green • **Expectation:** green screen		upstream
vk-image-display	• **Vulkan:** creates an image and fills it with colored bars • **OpenGL:** draws the image on screen using semaphores to synchronize access • **Expectation:** a stripped image on screen		upstream
vk-image-display-overwrite	• **Vulkan:** creates an image and fills it • **OpenGL:** overwrites that image with another stripped pattern • **Vulkan:** reads back the contents of the image • **Expectation:** the modified contents to be read back		upstream

Additions to Piglit (the details) II

Contributions to Piglit II

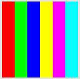
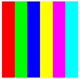
Reusing the depth and stencil buffer:

Test name	Description	Output	Status
vk-depth-display	• **Vulkan:** creates a depth buffer pattern • **OpenGL:** draws a fullscreen quad at a greater depth • **Expectation:** quad partially obscured		under review
vk-stencil-display	• **Vulkan:** creates a pattern in the stencil buffer • **OpenGL:** draws a blue fullscreen quad with stencil testing • **Expectation:** quad masked by stencil pattern		under review

Additions to Piglit (the details) III

Contributions to Piglit III

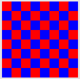
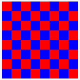
Reusing a pixel buffer:

Test name	Description	Output	Status
vk-buf-exchange	• **Vulkan:** creates and fills an image with color bars and copies its content to a buffer • **OpenGL:** uses the buffer as pixel storage for a texture and displays it • **Expectation:** colored bars on screen		upstream
vk-pix-buf-update-errors	Same as above and: • **OpenGL:** attempts to overwrite the buffer • **Expectation:** error is generated, the buffer remains unchanged		work in progress

Additions to Piglit (the details) IV

Contributions to Piglit IV

Reusing a vertex buffer:

Test name	Description	Output	Status
vk-vert-buf-update-errors	- **Vulkan:** creates a vertex buffer and fills it with only the odd quads of a chess board pattern - **OpenGL:** clears the screen to red and draws the vertex buffer in blue - **OpenGL:** attempts to overwrite the VBO - **Expectation:** a red-blue chess board pattern on screen and an invalid operation error is generated		work in progress
vk-vert-buf-reuse	- **Vulkan:** same as above - **OpenGL:** same as above minus the invalid overwrite test - **Vulkan:** reuses the VBO in a renderpass to draw the chess board - **Expectation:** chessboard on screen		work in progress

Additions to Piglit (the details) V

Contributions to Piglit V

Many people helped us with the work on the tests:

Contributors	Contributions
Eleni Maria Stea	tests, Vulkan framework code
Juan A. Suárez Romero	contributions to the framework, and vk-image-overwrite, testing
Tapani Pälli (*many thanks!*)	reviews, testing and patches for the framework
Bas Nieuwenhuizen	framework patch to support dedicated memory
Topi Pohjolainen	we’ve used some old patches of him as an example at the very beginning

- Also thanks to Eduardo Lima Mitev and to Hyunjun Ko for the reviews and to Germán Poo Caamaño who helped tidying up the initial mess of the Vulkan framework commits.
- Again, this information was accurate when I prepared and pre-recorded the slides. Sorry if I am missing someone or something!

Piglit Results

Conclusions from running the tests:

- Most of the new Piglit tests are passing in the mesa drivers mentioned before (iris, freedreno, radeonsi, i965).
- Unfortunately Mesa does not yet support the depth and stencil buffer exchange. The only implementation I ’ve found that currently supports this feature is the Nvidia proprietary driver.
- radeonsi needs some fixes to better support buffers and optimal tiling (WIP).

Outline

1 OpenGL interoperability extensions

2 Additions to Mesa and Piglit

3 Questions

4 References

Questions?

E-mail: estea@igalia.com
IRC/Freenode: hikiko

Outline

1 OpenGL interoperability extensions

2 Additions to Mesa and Piglit

3 Questions

4 References

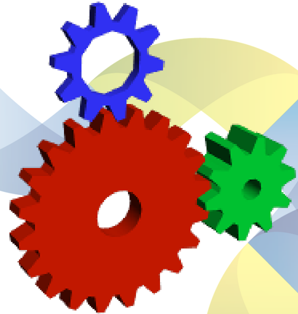
Links I

- *OpenGL and Vulkan Interoperability on Linux (Blog posts on the Piglit tests use cases)*.
  https://eleni.mutantstargoat.com/hikiko/category/igalia/graphics_drivers/opengl-and-vulkan-interoperability/.

- Carsten Rohde, Dave Airlie, James Jones, Jan-Harald Fredriksen, Jeff Juliano, and Michael Worcester.
  EXT_external_objects (GL_EXT_memory_object, GL_EXT_semaphore), July 18, 2018.

  https://www.khronos.org/registry/OpenGL/extensions/EXT/EXT_external_objects.txt.

- Carsten Rhode, James Jones, Jan Harald Fredriksen, and Jeff Juliano.
  EXT_external_objects_fd (GL_EXT_memory_object_fd, GL_EXT_semaphore_fd), June 2, 2017.
  https://www.khronos.org/registry/OpenGL/extensions/EXT/EXT_external_objects_fd.txt.



Thank you!