MoltenVK

Layering Vulkan Over Metal

Vulkanised 2023

Bill Hollings
The Brenwill Workshop Ltd.
February 2023
# A Brief History of MoltenVK

<table>
<thead>
<tr>
<th>Company</th>
<th>Year</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cocos3D</strong></td>
<td>2011</td>
<td>OpenGL ES2 game engine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Objective-C</td>
</tr>
<tr>
<td><strong>MoltenGL</strong></td>
<td>2015</td>
<td>OpenGL ES2 over Metal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Objective-C</td>
</tr>
<tr>
<td><strong>MoltenVK</strong></td>
<td>2016</td>
<td>Vulkan over Metal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C++</td>
</tr>
<tr>
<td><strong>Valve</strong></td>
<td>2018</td>
<td>Open-source MoltenVK</td>
</tr>
<tr>
<td><strong>Core Weavers</strong></td>
<td>2018</td>
<td></td>
</tr>
</tbody>
</table>
Open-Source Layering Projects
Fighting Platform Fragmentation

<table>
<thead>
<tr>
<th>Layers Over</th>
<th>Vulkan</th>
<th>OpenGL</th>
<th>OpenCL</th>
<th>OpenGL ES</th>
<th>DX12</th>
<th>DX9-11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vulkan</td>
<td>Zink</td>
<td>clspv + clvk</td>
<td>Angle</td>
<td>vkd3d-Proton vkd3d</td>
<td>DXVK</td>
<td>WineD3D</td>
</tr>
<tr>
<td>OpenGL</td>
<td>Ashes</td>
<td></td>
<td></td>
<td>Angle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DX12</td>
<td>Microsoft Dozen</td>
<td>Microsoft ‘GLOn12’</td>
<td>Microsoft ‘CLOn12’</td>
<td></td>
<td>Microsoft D3D11On12</td>
<td></td>
</tr>
<tr>
<td>DX11</td>
<td>Ashes</td>
<td></td>
<td></td>
<td>Angle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metal</td>
<td>MoltenVK</td>
<td></td>
<td>clspv + SPIRV-Cross?</td>
<td>MoltenGL Angle</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Vulkan as porting target for multiple APIs**
- **ROWS:** Bring more APIs to Platforms
- **COLUMNS:** Making APIs available across platforms

Vulkan functionality available everywhere
MoltenVK Current Status (Feb 2023)

- Supports Vulkan 1.2 + extensions
- Embedded in the macOS Vulkan SDK
- Key component of the Vulkan Portability Initiative
- Currently undergoing extensive ongoing activities to reduce CTS failures, ideally to zero to accomplish Vulkan conformance.
- Other improvements come from bug reports and enhancement requests posted to the MoltenVK GitHub repository issues list.

https://github.com/KhronosGroup/MoltenVK/issues
MoltenVK Functionality

- Open-source project with a small dedicated team
  - Working hard to minimize the unavoidable delay to integrate new Vulkan functionality
  - Additional contributors/contributions always welcome!

- Supports almost all Vulkan 1.2 + extensions
  - Main Vulkan specification is at 1.3 + more extensions

- Development prioritizes functionality needed for games
  - Most games find all the Vulkan functionality they need

- Some titles do hit functionality limits
  - AAA titles pushing tech and spec limits
  - Binary ports or emulators originating in DirectX
  - Functional limits of the underlying Metal API

- Working towards 100% Vulkan Portability Conformance
  - Vulkan 1.0 CTS: 174K pass, 0.2K fail (0.15%)
  - Vulkan 1.2 + extns CTS: 322K pass, 10K fail (3.2%)
  - Most failures are now edge cases with available workarounds
MoltenVK Performance

- MoltenVK is a layered implementation of Vulkan on Metal, which adds some unavoidable memory and performance overhead

- Metal is a higher-level API than Vulkan
  - Not always a 1:1 mapping between Vulkan & Metal calls and functionality
  - Metal lacks some of the fine-tuning performance controls of Vulkan

- Transpiling shaders from SPIR-V to MSL can sometimes introduce shader inefficiencies (vs hand-tuned shader code)

- We are committed to improving performance when made aware of it
  - Performance issues often only show up in particular game contexts, depending on how a game’s use of Vulkan maps to Metal

- Some titles hit performance limits
  - AAA titles pushing tech and spec limits
  - Binary ports or emulators originating in DirectX
  - Some issues solvable with MoltenVK development focus
  - Some issues hit performance effects described above
ISVs Shipping with MoltenVK on Apple

Commercial Vulkan Developers
From LunarG 2021 Developer Survey
Using MoltenVK

- Games shipping with MoltenVK:
  - DOTA 2
  - Metro Exodus
  - Final Fantasy XIV
  - Dark Souls: Remastered
  - Dark Souls III
  - DOTA Underlords
  - Aerofly Flight Simulator 2
  - Path of Exile
  - Raft
  - The Elder Scrolls Online
  - Celeste
  - Transport Fever 2
  - Shadow Warrior 2
  - Streets of Rage 4
  - Jupiter Hell
  - Wreckfest
  - Victoria 3
  - Artifact
  - GZDOOM
  - vkQuake & vkQuake2

- Games runnable by users via Crossover and MoltenVK:
  - Halo: Combat Evolved
  - God of War (2018)
  - Grand Theft Auto V
  - World Of Tanks
  - Forsaken Remastered
  - Elder Scrolls V Skyrim: SE
  - Guild Wars 2
  - Battlefield 1
  - Battlefield II
  - Age of Empires II: Definitive Edition
  - Witcher 3

- Applications shipping with MoltenVK:
  - Autodesk Fusion 360

- Engines using MoltenVK:
  - Google Filament
  - Defold
  - Ultra Engine
  - Diligent Engine
  - Blender Vulkan (PoC)
  - ncnn
  - Clausewitz Engine (Paradox)

- Platform emulators using MoltenVK:
  - VKD3D (Direct3D 12)
  - DXVK (Direct3D 9/10/11)
  - Google Android Emulator
  - Dolphin (Wii & GameCube)
  - Ryujinx (Switch)
  - Cemu (Wii U)
  - RPCS3 (PS3)
  - PCSX2 (PS2)

Testing 15 games with MoltenVK support
https://youtu.be/xDGQcjqpYql
MoltenVK and Metal 3

- Buffer device address (VK_KHR_buffer_device_address)
- Improved descriptor indexing support on macOS and iOS via significant argument buffer enhancements.
- Mesh shaders
- Geom shaders and faster Tess shaders via Mesh shaders