0 A.D.: Vulkan and its obstacles in open-source game

Vladislav Belov

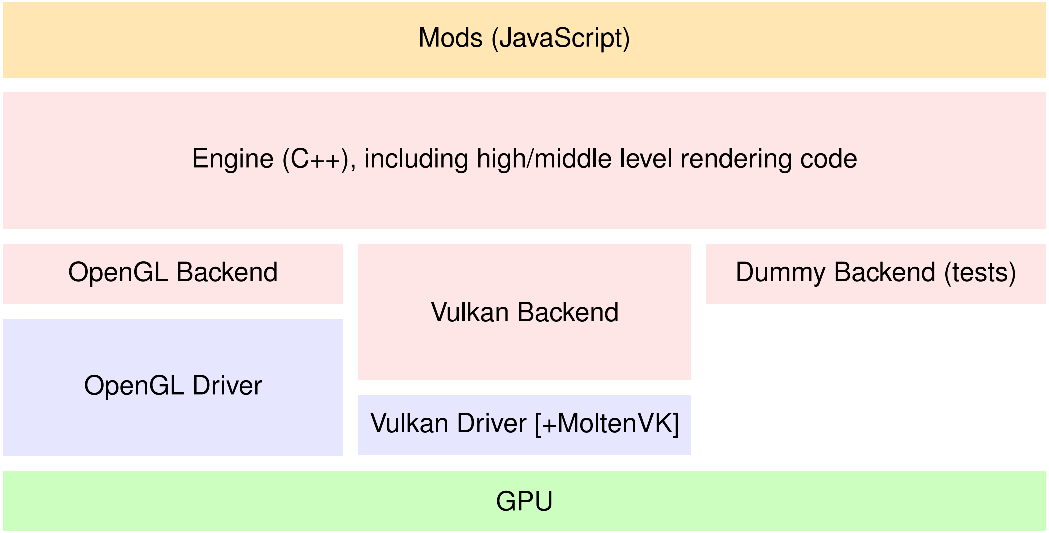
0 A.D. Empires Ascendant

FOSDEM'26

0 A.D.

FOSDEM'26 wildfire GAMES

Pyrogenesis Structure



Renderer Backend Interface

```
class IDevice
{
  // ...
  virtual bool AcquireNextBackbuffer() = 0;
  virtual bool Present() = 0;

  virtual std::unique_ptr<ITexture> CreateTexture(
    ITexture::Type type, uint32_t usage, Format format, ...) = 0;
  virtual std::unique_ptr<IBuffer> CreateBuffer(
    IBuffer::Type type, uint32_t size, uint32_t usage) = 0;

  virtual bool IsTextureFormatSupported(Format format) const = 0;
  // ...
};
```

Renderer Backend Interface

FOSDEM'26 wildfire GAMES

```
class IDeviceCommandContext
{
  // ...
  virtual void UploadTexture(ITexture* texture, void* data, ...) = 0;
  virtual void UploadBuffer(IBuffer* buffer, void* data, ...) = 0;

  virtual void BeginFramebufferPass(IFramebuffer* framebuffer) = 0;
  virtual void EndFramebufferPass() = 0;

  virtual void SetTexture(int32_t bindingSlot, ITexture* texture) = 0;
  virtual void SetUniform(int32_t bindingSlot, std::span<const float> values) =

  virtual void DrawIndexed(
    uint32_t firstIndex, uint32_t indexCount, int32_t vertexOffset) = 0;
  virtual void Dispatch(
    uint32_t groupCountX, uint32_t groupCountY, uint32_t groupCountZ) = 0;
  // ...
};
```

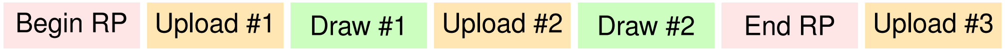
Vulkan

vkCmdResolveImage vkDestroyBuffer vkWaitForFences vkCmdNextSubpass vkWaitForFences vkFreeDescriptorSets vk
ageMemoryRequirements2 vkCmdEndRenderPass vkGetPhysicalDeviceQueueFamilyProperties vkCm
roperties vkCmdDispatch vkWaitSemaphores vkCreateEvent vkCmdCopy vkGetImageMem
rOpaqueCaptureAddress vkCmdBlitImage vkBindBufferMemory vkCmdWriteTimestar
appedMemoryRanges vkQueueSubmit vkGetPhysicalDevice vkCmdCopyBuffer
ceFeatures2 vkCmdEndRenderPass2 vkCreateImage vkCmdEndRenderPas
umerateInstanceExtensionProperties vkCreateEvent vkDestroySampler vk
Reference vkGetQueryPoolResults vkCmdSetEvent vkGetPhysicalDevi
GetPhysicalDeviceFeatures2KHR vkGetPhysicalDeviceMemoryPrope
ateBufferView vkWaitSemaphores vkCreatePipelineCache vkGetPhys
crConversion vkCmdDrawIndexed vkCreateRenderPass2 vkCmdSetL
ments2 vkCmdSetBlendConstants vkCreateEvent vkCmdBeginRende
DestroyInstance vkDestroySampler vkCmdDrawIndirectCount vkQueue
equirements2 vkCreateFramebuffer vkGetRenderAreaGranularity vkWai
PipelineLayout vkCreateRenderPass vkGetQueryPoolResults vkCmdPipel
DescriptorSets vkResetDescriptorPool vkWaitSemaphores vkCmdCopyQuery
StencilCompareMask vkCreateSampler vkDestroyCommandPool vkCreateGraph
DeviceFeatures2 vkFreeCommandBuffers vkFreeDescriptorSets vkCreateDescriptorS
SetWithTemplate vkCmdCopyImageToBuffer vkGetDeviceProcAddr vkCmdWaitEvents vkCr
ormatProperties vkGetImageMemoryRequirements vkTrimCommandPool vkBindImageMemory vkCmd
vkCmdBeginRenderPass vkCmdFillBuffer vkCmdSetViewport vkGetInstanceProcAddr vkCmdSetStencilReference vkGet

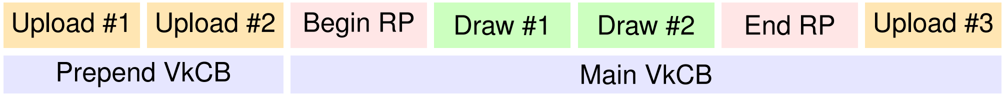
Vulkan®

Vulkan Timeline

DeviceCommandContext Timeline



Submit Timeline

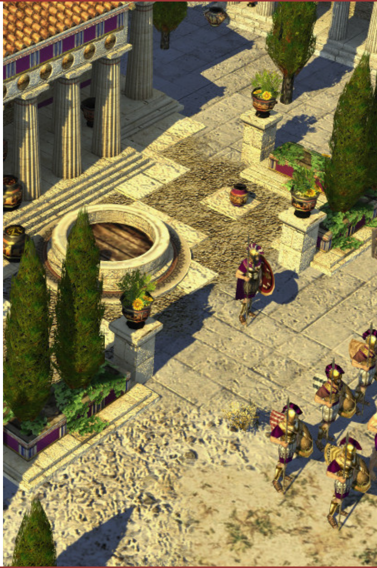


Vulkan support detection

FOSDEM'26

Vulkan support detection

Use SDL to detect Vulkan support presence



Vulkan support detection

The following code was used to detect presence of Vulkan in the player system (checking `vulkan-1.so/vulkan-1.dll` under the hood):

```
if (!SDL_Vulkan_LoadLibrary(nullptr))
{
    void* vkGetInstanceProcAddr = SDL_Vulkan_GetVkGetInstanceProcAddr();
    if (vkGetInstanceProcAddr)
    {
        // Supported.
    }
    SDL_Vulkan_UnloadLibrary();
}
```

Vulkan support detection

On some drivers when were trying to use OpenGL afterwards it was crashing:

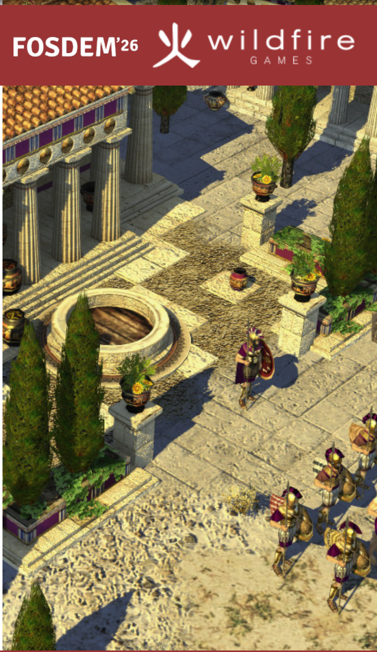
```
SDL_GL_LoadLibrary(nullptr);
// ...
// Draw something...
// ...
SDL_GL_UnloadLibrary();
```

The solution was just stop asking for Vulkan support and enable it only on the player request via options.

Device selection

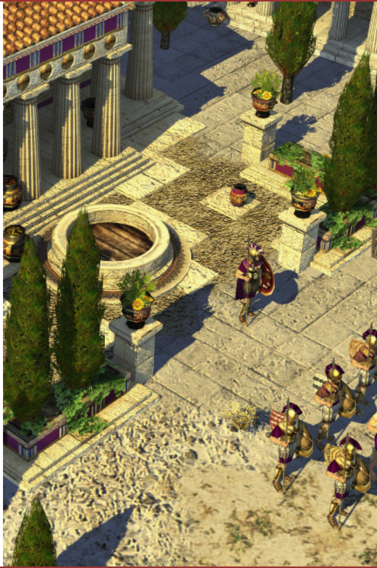
Device selection

Choosing which VkPhysicalDevice to use



Device selection

- Initially we were sorting `VkPhysicalDevice` to select the best
- First by device type, then by device total memory, then by initial index
- It's failed for some drivers which were reporting wrappers as fully capable devices and even more Example:
  `Microsoft Direct3D12 (Radeon Graphics)`
  $\approx$ 7.6GB
  `AMD Radeon (TM) Graphics`
  $\approx$ 7.4GB

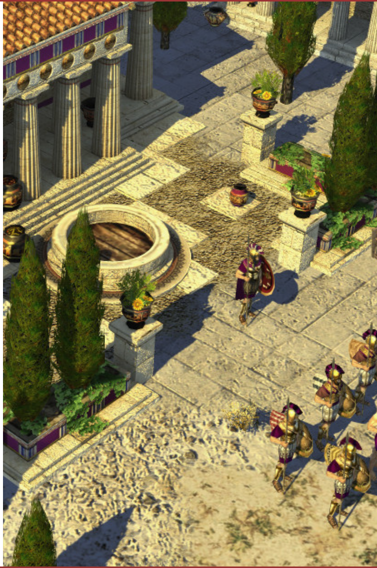


Device selection

```
[0] Intel(R) UHD Graphics 620 (WHL GT2) (Integrated)
[1] NVIDIA GeForce MX250 (Discrete)
[2] llvmpipe (LLVM 20.1.8, 256 bits) (CPU)
```

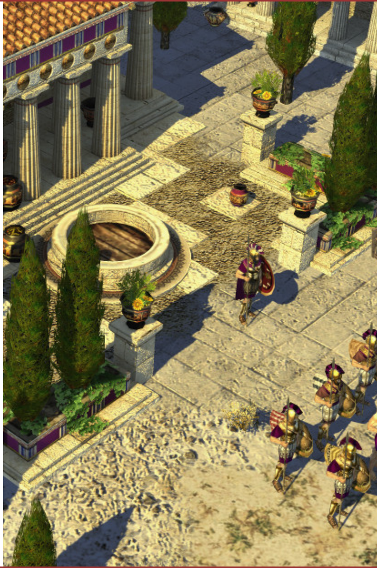
Device selection

- We're failed to create a swapchain for the NVIDIA one `VK_ERROR_INITIALIZATION_FAILED` on `vkCreateSwapchainKHR`
- `vkPhysicalDevice` might be listed in `vkEnumeratePhysicalDevices` even if it's not going to work



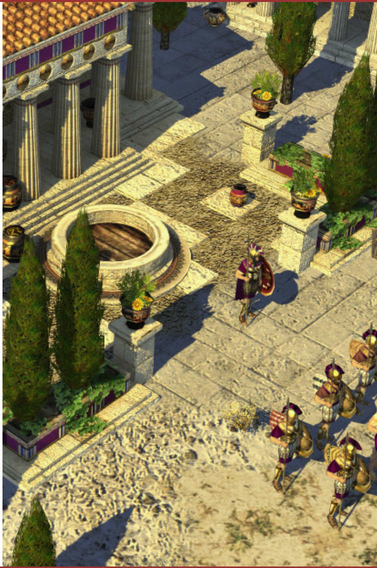
Device selection

- `textureCompressionBC` might be false if all needed BC formats are supported
- Check all needed BC formats individually instead



VK_ERROR_OUT_OF_DEVICE_MEMORY

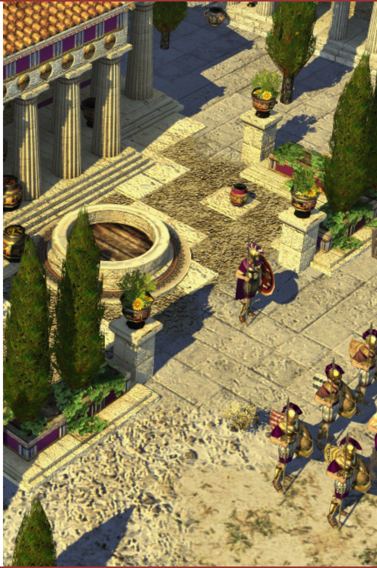
VK_ERROR_OUT_OF_DEVICE_MEMORY



Device selection

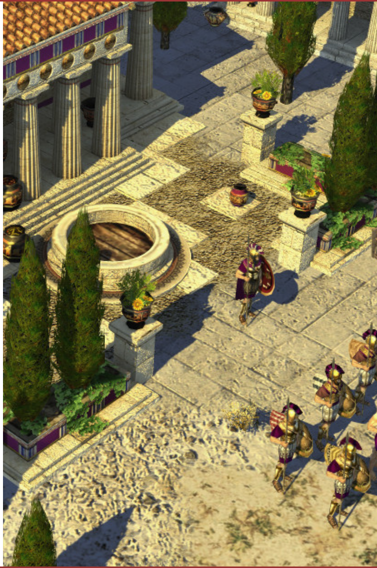
FOSDEM'26 wildfire GAMES

- `VK_ERROR_OUT_OF_DEVICE_MEMORY` might be returned when we're allocating memory (`vkAllocateMemory`)
- All creation functions (`vkCreate*`) might return it
- `vkAcquireNextImageKHR`/`vkQueuePresentKHR` might return it
- `vkQueueSubmit` might return it
- `vkWaitForFences` might return it
- Several others



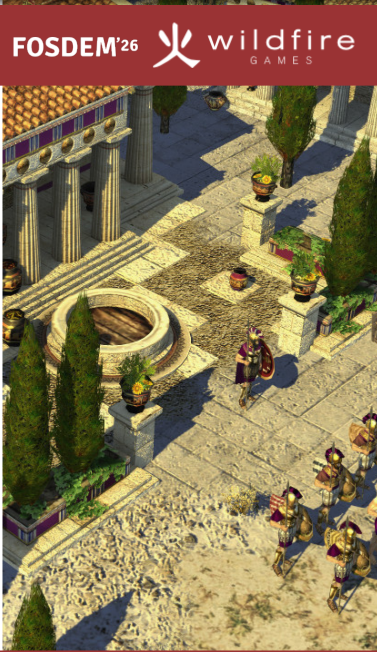
Device selection

- The problems is that non-allocating cases are separated from places where we can free some memory
- We're using VMA (Vulkan Memory Allocator) so freed memory won't go immediately to GPU
- Prefer to use `VK_EXT_memory_budget` when possible

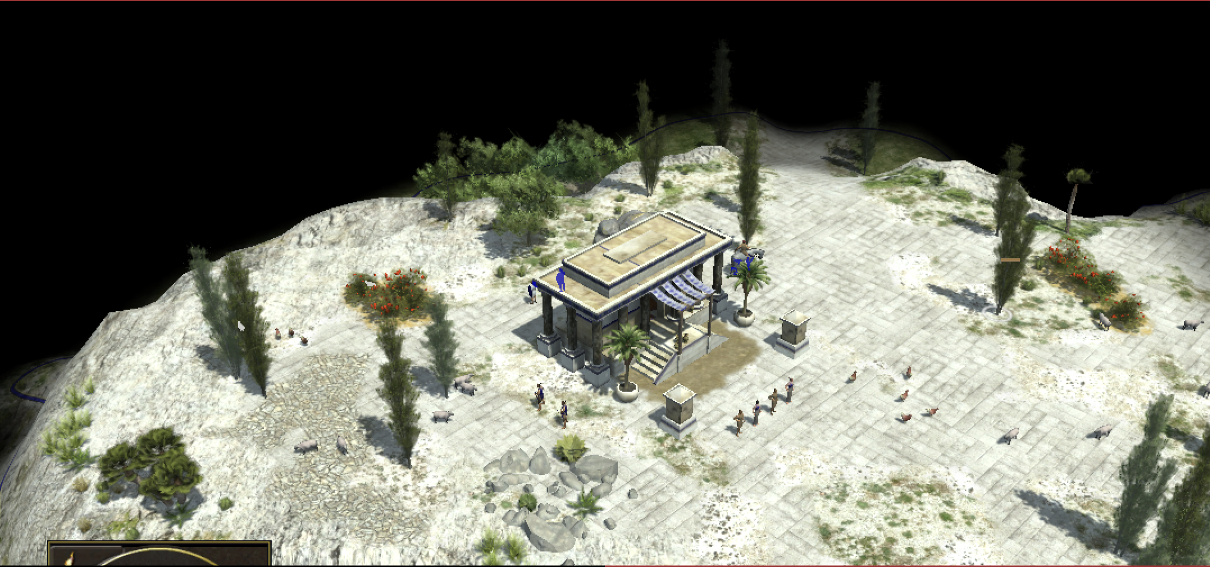


GPU skinning artifacts

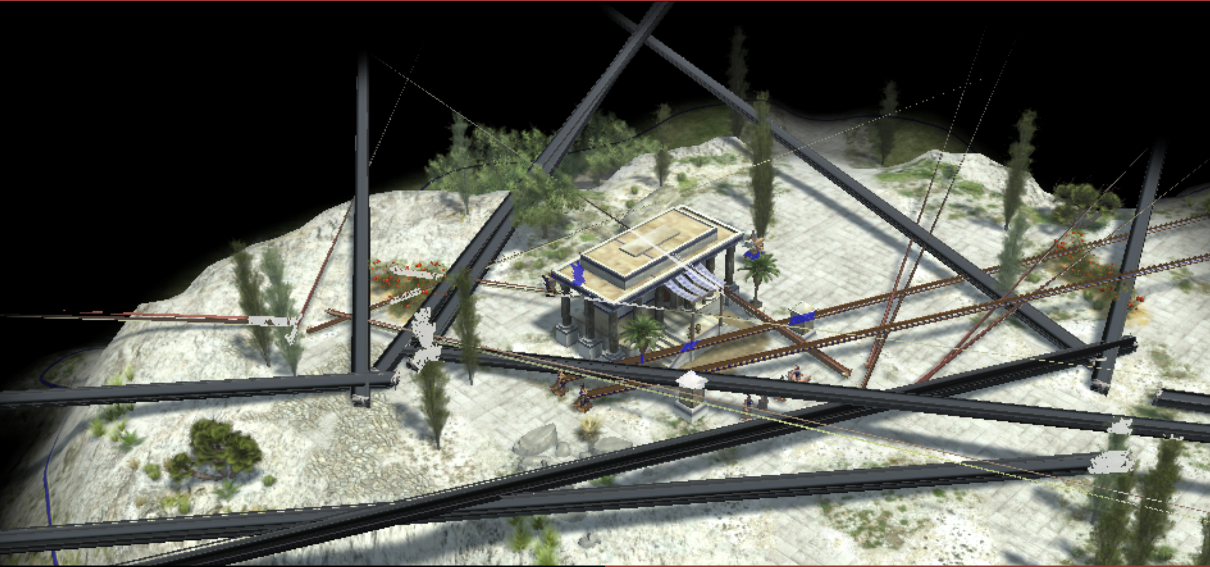
GPU skinning artifacts



GPU skinning artifacts

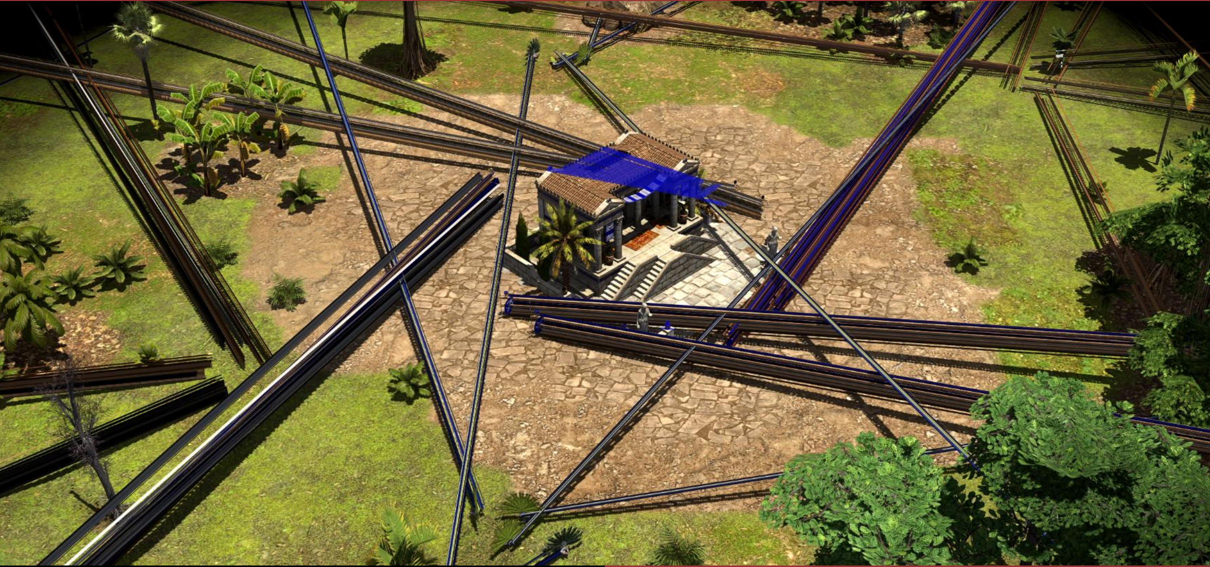


GPU skinning artifacts



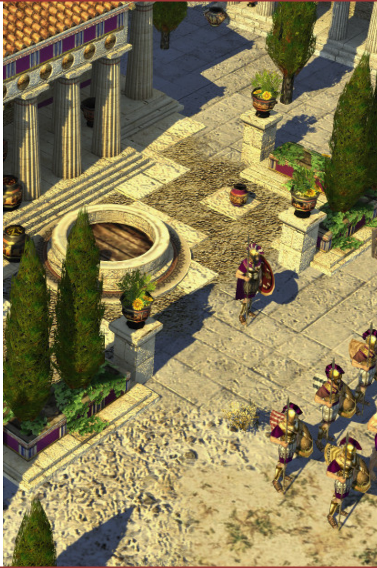
GPU skinning artifacts

FOSDEM'26 wildfire GAMES

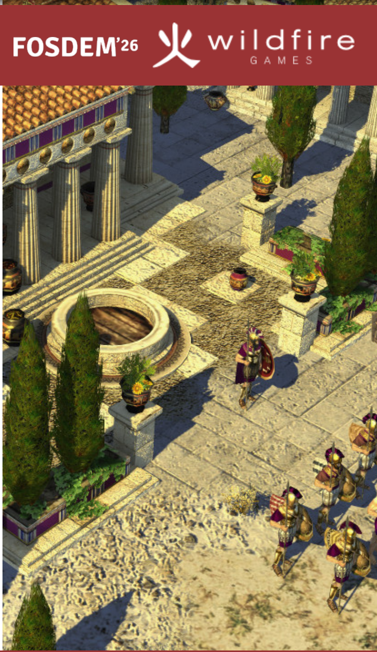


GPU skinning artifacts

- When GPU skinning was enabled in options the game was using data prepared for CPU skinning
- We just forgot to invalidate the skinning flag when player enables it in options



Collected GPUs statistics

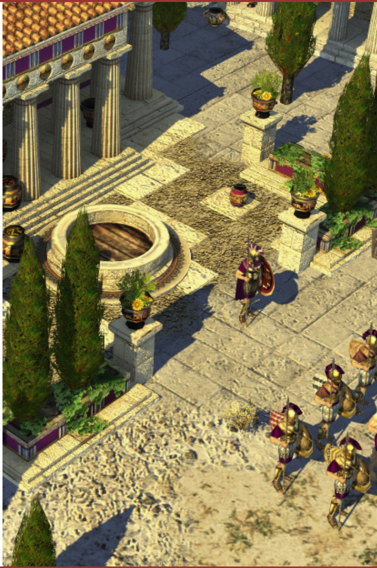


Collected GPUs statistics

Data reported voluntarily by players

OpenGL: Different names the same GPU

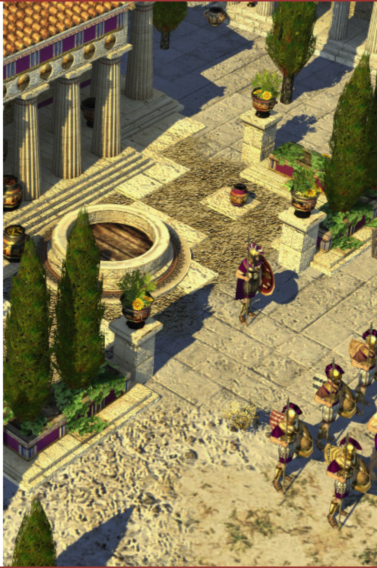
- Radeon RX 550
- Radeon (TM) RX 550
- Radeon(TM) RX 550
- Radeon RX550/550 Series
- AMD Radeon RX 550 / 550 Series
- AMD Radeon RX 550 Series
- AMD Radeon RX550 series



Vulkan: Different names the same GPU

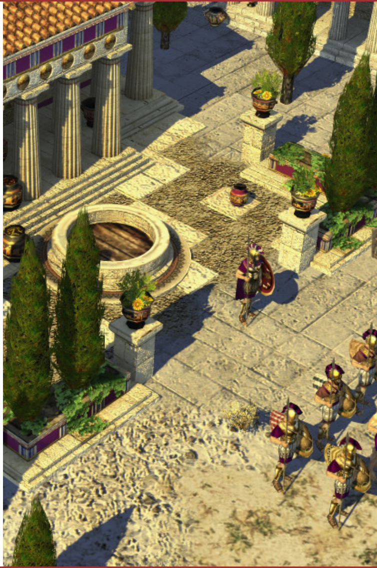
FOSDEM'26 wildfire GAMES

- AMD Radeon RX 550 / 550 Series (POLARIS12)
- AMD Radeon RX 550 / 550 Series
- AMD Radeon RX 550 Series (POLARIS11)
- AMD Radeon RX 550
- Radeon RX 550 Series
- Radeon (TM) RX 550



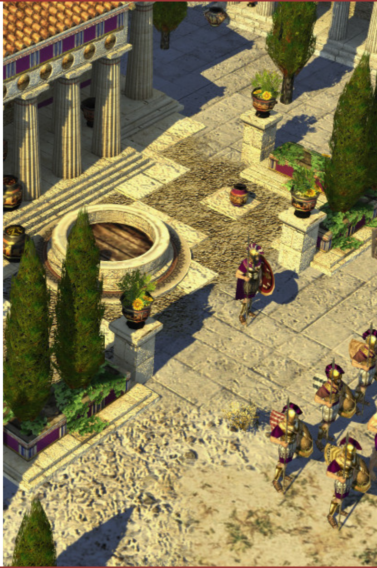
Vulkan: GPUs differentiating

- deviceID isn't enough to differentiate GPUs
- deviceUUID might work if presents
- deviceUUID might be 0-10000000-0-0
- We still parse deviceName but it's much simpler



"Remote" debugging

FOSDEM'26 wildfire GAMES



"Remote" debugging

"Remote" debugging

RenderDoc helpers:

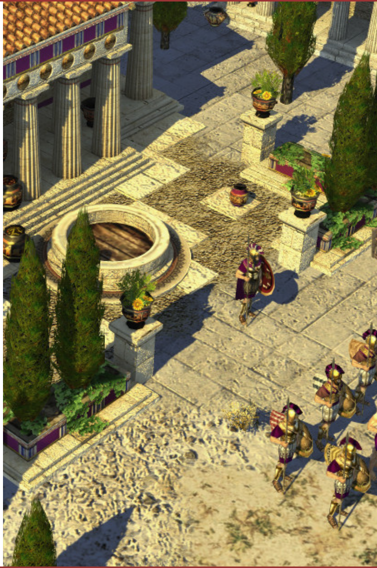
```
renderer.backend.debuglabels = "true"
renderer.backend.debugscopedlabels = "true"
```

Messages (`VK_EXT_debug_utils`):

```
renderer.backend.debugmessages = "true"
```

Validation layers (`VK_LAYER_KHRONOS_validation`):

```
renderer.backend.debugcontext = "true"
```



"Remote" debugging

FOSDEM'26 wildfire GAMES

Disable bindless:

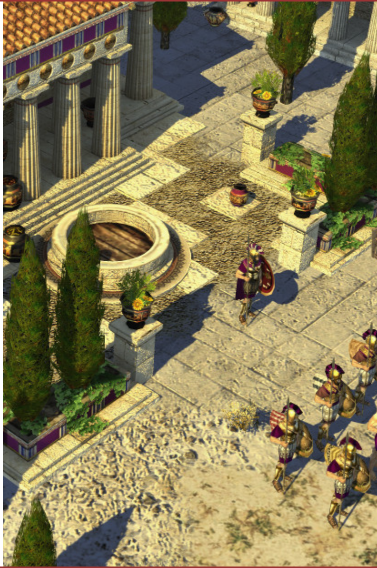
```
[renderer.backend.vulkan]
disabledescriptorindexing = "true"
```

Override used device:

```
[renderer.backend.vulkan]
deviceindexoverride = "0"
```

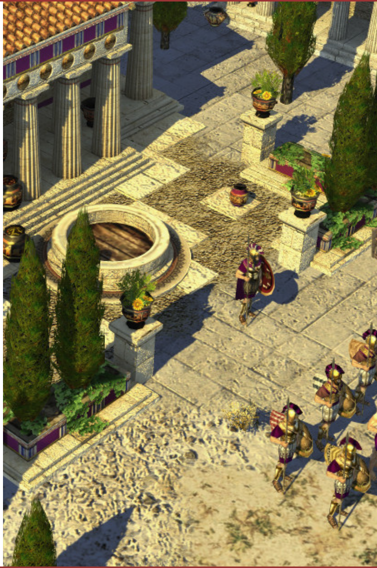
Debug barriers and waits:

```
[renderer.backend.vulkan]
debugbarrierafterframebufferpass = "true"
debugwaitidlebeforeacquire = "true"
debugwaitidlebeforepresent = "true"
debugwaitidleafterpresent = "true"
```



Main menu artifacts on RPI4 (V3DV)

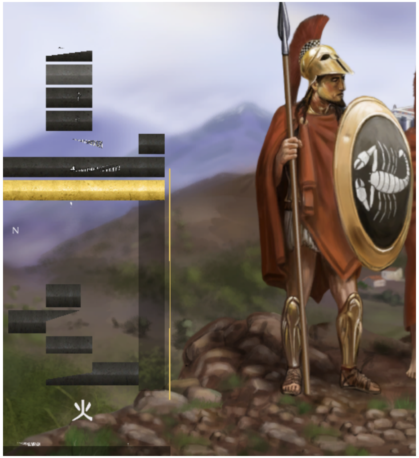
FOSDEM'26 wildfire GAMES

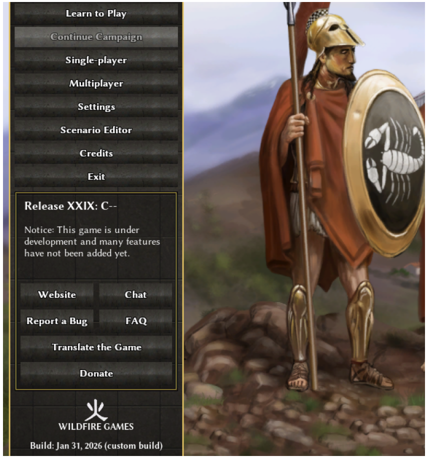


Main menu artifacts on RPI4 (V3DV)

Raspeberry Pi 4 with Mesa 24

Main menu artifacts on RPI4 (V3DV)





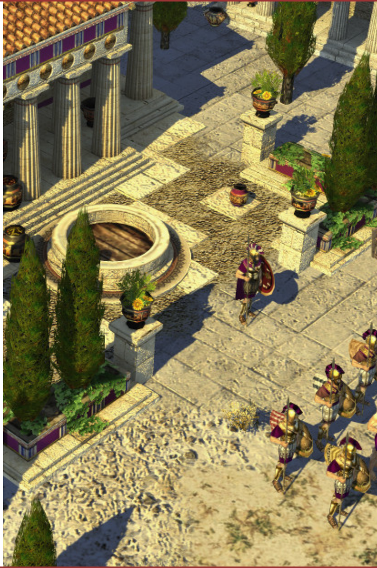
Release XXIX: C--

Notice: This game is under development and many features have not been added yet.

Build: Jan 31, 2026 (custom build)

Main menu artifacts on RPI4 (V3DV)

- Debug barrier with all stages/masks didn't help
- Explicit wait via vkDeviceWaitIdle didn't help
- The only thing helped is split vkQueueSubmit with explicit semaphore



Main menu artifacts on RPI4 (V3DV)

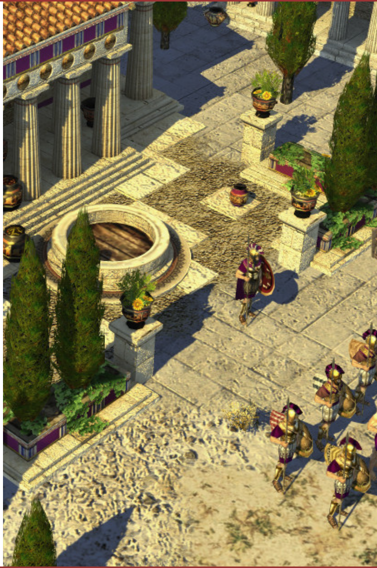
FOSDEM'26 wildfire GAMES

```
...
submitInfo.pSignalSemaphores = &semaphore;
vkQueueSubmit(m_Queue, 1, &submitInfo, ...);

...
submitInfo.pWaitSemaphores = &semaphore;
vkQueueSubmit(m_Queue, 1, &submitInfo, ...);
```

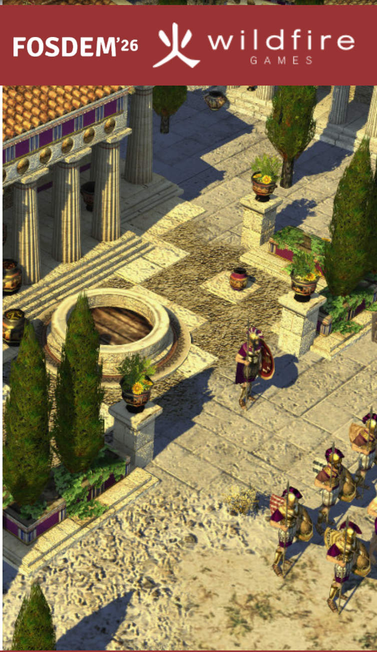
Main menu artifacts on RPI4 (V3DV)

- It was the driver issue which was fixed pretty fast (thanks to José and his colleagues)
- The more unusual an application uses the Vulkan API, the more likely a driver error will occur
- A local reproduce significantly increases a chance to fix the issue



GPU performance measurements

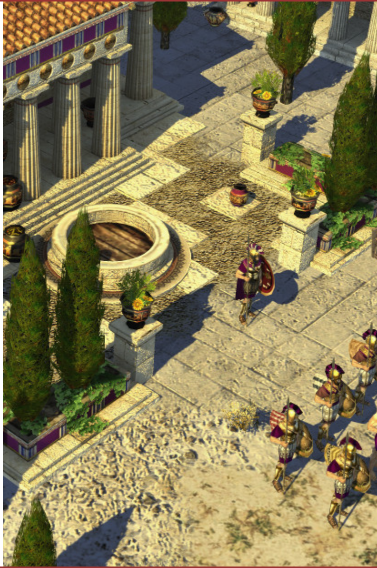
FOSDEM'26 wildfire GAMES



GPU performance measurements

GPU performance measurements

- Prefer using tools from vendors
- In-game measured with `VK_QUERY_TYPE_TIMESTAMP`
- Measurements are affected by other processes
- Measurements might be affected by temperature
- Measurements might show worse results for better code



If you enjoy creating games, you will always be welcome!

play0ad.com
gitea.wildfiregames.com/0ad/0ad/wiki/
vladislavbelov at wildfiregames.com