Aether3D Internal TODO/Roadmap, subject to change.
Project page: https://github.com/bioglaze/aether3d

Aetherin markkerit ei näy oikein PIX:ssä NVIDIA:llakaan
Aether: siivoa USE_RGP_MARKERS pois

Relnotes 0.8.x

- Features:
  - Update nuklear.h
  - Updated stb_image.c to v2.30
- Bug fixes:
  - Fixed PIX events to render properly in PIX.
- Editor
  - Fix Min2()

BUGS (Major)

- D3D12: GPU hang if too many draw calls
- 2023-04-05: D3D12: v0.8.7 Misc3D GPU crash on Intel Arc A770. Vulkan ei kaadu (testattu main branch, eikä 0.8.7). 2024-05-30 kaatuu edelleen
- D3D12: v0.8.7 Misc3D GPU crash on Intel UHD 630, driver 30.0.101.1994
- Metal: If standard material is used but forward+ is not defined, system hangs.
- (Debug with Pix, DRED
- https://docs.microsoft.com/en-us/windows/win32/direct3d12/use-dred
- or https://community.khronos.org/t/after-a-vk-device-lost-what-options-are-there/103923/4
- GPU hang if using Forward+ in OpenVR. (debug with NVIDIA checkpoint extension or DRED) https://feresignum.com/debugging-vk_error_device_lost-with-nvidia-device-diagnostics/

BUGS (Normal)

- D3D12: NuklearTest-projektissa validaatiovirhe scissorista.
- Vulkan: Depth ja Primary Pass GPU counter näyttää liian isoja lukuja exrmd-koneella.
- Vulkan: LightCuller CPU/GPU -time ei ole profiloitu, mutta D3D12:lla on.
- Misc2D: Glider position varies between Vulkan/D3D12
- Misc3D: viewport varies between Vulkan/D3D12
- OBJ-converter has special case for meshes named "unnamed". Could cause bugs.
- Cube mesh's normal point to wrong direction? Spotlight doesn't light it properly
- D3D12: Primary pass time ei ole profiloitu, mutta Vulkanilla on.
- Particle draw does z / w. Not fast and not sure if the values are right.
- Metal: When using external display, Editor and HUD font too large and UI unusable. Partial fix by modifying InitMetal().
- Metal, Vulkan: spot shadow shows through wall.
- Metal: Texture memory usage doesn't match frame capture Summary.

- Vulkan: light count debug visualization transforms wrongly on Windows, not on Linux. Works on D3D12.
- Tangent orthonormalization maybe should be done in the shader: https://learnopengl.com/Advanced-Lighting/Normal-Mapping
- Bitangent calculation looks correct with cross(tangent, normal) instead of cross(normal, tangent), but mikktspace uses the latter (see tile wall in Misc3D).
- Vulkan: Directional light specular seems to point to opposite direction when compared to shadow casting direction.
- Metal: point shadow binding differs in unlit and standard.
- Vulkan, Metal, D3D12: SSAO and particles look weird if rotating camera, it lags behind.
- Converter Mesh interleave doesn't check bone index and weights?
- SSAO: D3D12 and Vulkan shading don't match. Take screenshot of each. Maybe because of projection matrix?
- Metal: Bloom + SSAO looks broken.
- Metal: Tiling pattern when rotating the view on Metal and Forward+ on iOS
- Bloom/blur texture is 8-bit?! (at least on Vulkan)
- Translucency: thick objects need to be drawn separately for back and front faces.
- Editor: SSAO + Bloom looks broken.
- Editor: if adding point light in the middle of many cubes, some get lighting in the wrong faces. Broken normals?
- **Editor: If light casts shadow, gizmo is dark**
- Editor: slider doesn't work on macOS.
- Editor: gizmo drag doesn't work on macOS
- **Editor: If there is a shadow casting light and you hover over gizmo, the hovered mesh disappears.**
- Skinned mesh is not culled properly
- Rotating mesh is not culled properly
- D3D12: Mipmap generation slightly moves the texture in smaller levels.
- **D3D12, Vulkan: Tile culling is defined out. Possible fix: depthnormals texture to store linear depth, see Sascha Willems SSAO gbuffer shader.**
- D3D12: Skinned mesh casts directional shadow under ground
- D3D12: texCube uses same sampler as texture2d
- D3D12: bounding box drawing breaks cube map binding
- Vulkan: Shadows are not visible in VR
- Vulkan: Forward+ crashes on Intel on Win7, Skylake 6700k, driver 15.45.27.5068

BUGS (Minor)

- Vulkan: if bloom is enabled and stuff is on top, it bleeds to bottom.
- Linux: keeping a movement key down doesn’t move indefinitely (Xorg)

PERF

- Misc3D Sponza Forward+: FPS on Metalilla about 53, jos piirtokutsuja 470 ja DEBUG_FAST Standard-shaderissa.

- https://developer.nvidia.com/blog/identifying-shader-limiters-with-the-shader-profiler-in-nvidia-nsight-graphics/
- City
  - oma kone Linux: 17.3 ms
  - 60 FPS: Kakkoskone Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz GTX 780
  - Kakkoskoneella City 55 FPS -O3 Bloomin kanssa, 60 FPS ilman.
- Testaa, olisiko Vulkan-tileculleri nopeampi, jos puskurit ovat GPU-muistissa.
- AMD:llä 32-bit depth buffer on ok, NVIDIA:lla hitaampi.
- Move constructorin kannattaisi olla noexcept
- Jos render texturessa on neljä kanavaa, on nopeampaa kirjoittaa ne kaikki eikä .rgb
- Compilation time on Linux: make -f Makefile_Vulkan: 11.0 s 2018-12-28, clang: 9.7 s
- Kakkospöytäkoneen aether-kääntö 15 s. (HDD), 13 s (SSD)
- Vanhan MBP:n aether-kääntö 37 s.
- MBP 2013:n aether-kääntö 15 s.
- Työkoneella VS2019 aether-kääntö alle 2.5 s (Vulkan, Debug).
- Omalla koneella VS2019 kääntö 4.2 s (Vulkan Debug)
- Omalla uudella koneella VS2022 kääntö 2.4 s (Vulkan Debug)
- Näytä profilointistatseissa scenen aabb ja matriisien transformointiaika
- GTX 980 should handle at least 1 M particles.
- MBP:llä directorial shadow map 55 FPS
- Työkone: Vulkan Bloom Release build: 35 kpl vkQueueWaitIdle, 3.24 ms
- Työkone: Vulkan SSAO: Koko framessa 16 kpl vkQueueWaitIdle, 3.22 ms.
- Työkone Vulkan SSAO shader dispatch: 0.46 ms GPU
- Vulkan: Query waiting takes 1.8 ms CPU time on work PC/2080Ti, SSAO enabled.
- Frustum culling result from depthnormals pass is probably not used in primary pass.
- Check that important functions are inlined.
- D3D12 waits after present.
- D3D12: all vertex buffers are D3D12_HEAP_TYPE_UPLOAD, which is slower than DEFAULT. NSight Graphics PCI usage 46 %. Vastaava arvo Pixissä Cityssä PCIe read.
- D3D12: vertex buffer state is D3D12_RESOURCE_STATE_GENERIC_READ instead of vertex of index buffer.
- D3D12: UploadPerObjectUBO is slow.
- D3D12: GetDescriptorHandleIncrementSize is called each draw. Could be called during device init. Uses 0.05 ms/frame on RTX 3090, Release build. Call count: 1589, draw calls 396. GetGPUVirtualAddress also slow, 0.04 ms
- D3D12: Cache GetXXXDescriptorHandleForHeapStart() and GetGPUVirtualAddress
- LightCuller can be interleaved with shadow pass.
- Buffer<float4> voi olla hitaampi kuin StructuredBuffer<float4>
- Linux: Converting McGuire Sponza .obj or .fbx is very slow even with -O3
- Consider using FMA in CPU code
- Prefer raw buffers over typed buffers
- pow() can be replaced with a custom function if the exponent is always the same
- Test how much time frustum culling takes in Sponza scene

- Metal: Reduce redundant SetDepthStencilState, SetViewport, SetCullMode, SetFrontFacing, SetTriangleFillMode etc.
- Check idle GPU in AMD's profiler
- Render skybox after other objects
- light tiler: 2048-kokoinen (pointLightCenterAndRadius ja spotLightCenterAndRadius) SRV -> CBV auttaa NVIDIA:lla.
- Testaa, montako kertaa kutakin pso:ta käytetään framen aikana, sorttaa jos on hyötyä.
- CBV RST:hen, auttaa sekä AMD:llä että NVIDIA:lla. Tsekkaa detailit DX12 Do's And Don'ts
- Equal depth test
- VR: Sponza misses frames
- Instancing for depth passes
- Generate mips using CS
- Vulkan: Only call vkBindPipeline if it's changed
- Vulkan: resolveAttachment instead of vmCmdResolve
- Vulkan: Combine vertex and index buffer memory
- Vulkan: custom memory allocator
- http://gpuopen.com/vulkan-and-doom/
- *"it is possible to optimize by factoring out binding of Resources to just the start of each Command Buffer: "Bind Everything" as one Descriptor Set, use Push Constants to supply indexes when resource indexes need to be specialized per draw or dispatch"*, see Vulkan Fast Paths.
- *Getting multiple Command Buffers recording in parallel is the first step in ensuring a high-draw-count application stays GPU bound*
- GPUView: Test with VSync off and see if there are bubbles
- Submit some command buffers right after present to keep GPU busy

Design issues/improvements for the next engine

- vkQueueWaitIdle in Texture2D::SetLayout (fix? remove compute command list, do everything in graphics cmdlist)
- Timestampit handlataan eri tavoin D3D12/Vulkan. Koodi ei ole portattavaa.
- OpenAL32.dll:ää ja WinPixEventRuntime.dll:ää ei kopioida automaattisesti.
- Ennen submitointia featureita pitäisi pystyä testaamaan usealla koneella, eli varmaan git:n brancheissa?
- AMD recommends that render targets and UAVs use CreateCommittedResource (d3d12) or VK_KHR_dedicated_allocation, but otherwise CreatePlacedResource should be used.
- Metal: muistinkäytön kysely: https://developer.apple.com/videos/play/wwdc2022/10106/
- std::from_chars is faster than sscanf
- https://github.com/KhronosGroup/Vulkan-Guide/blob/master/chapters/hlsl.adoc
- Compute shader dispatch call takes different parameters on Metal.
- Texture2D and RenderTexture have different SetLayout

- Scene.cpp should not use renderer backend defines.
- Low-level code in ParticleSystemComponent.cpp
- Low-level #ifdefs in Scene.cpp
- Low-level #ifdefs in Misc3D
- Transparent rendering should use premultiplied alpha.
- Particle buffer should be a member of ParticleSystemComponent.
- ComputeShader.hpp: joissain metodeissa parametrit väärinpäin.
- Vulkan: RenderTexture::MakeCpuReadable is a hack.
- Vulkan: Every shader uses the same pipelineLayout, so adding eg. a buffer to one shader needs a lot of code changes.
- D3D12 ClearScreen sets render target
- D3D12 WaitForPreviousFrame, fix https://github.com/Microsoft/DirectX-Graphics-Samples/blob/master/Samples/UWP/D3D12HelloWorld/src/HelloFrameBuffering/D3D12HelloFrameBuffering.cpp
- Material.cpp contains Metal specific code paths
- Backbuffer can be smaller than wanted because of window border.
- Shader compile script can be combined for windows/linux. Windows runs .sh files universally.
- D3D12 fence stuff: https://anteru.net/blog/2015/debugging-d3d12-fences-queues/
- D3D12 memory stuff: https://therealmjp.github.io/posts/gpu-memory-pool/
- Cameras should never render into backbuffer, it complicates the internal engine regarding MSAA etc.
- D3D12: Verify that next engine uses GetCurrentBackBufferIndex instead of frame +1 % framecount
- Linux executable names should be in lower case, easier to type.
- UBO struct in shader code is duplicated for Vulkan and D3D12. A fix would be to use ConstantBuffer< UBO > instead of cbuffer, but would require changing each access.
- Does not use D3D12 Enhanced Barriers: https://microsoft.github.io/DirectX-Specs/d3d/D3D12EnhancedBarriers.html

Shadow research:
http://blog.kiteandlightning.la/variance-shadow-maps/
MJP shadow samples
http://www.gamedev.net/topic/673197-cascaded-shadow-map-shimmering-effect/
VSM should use linear Z.
Clear VSM to (MAX_Z, MAX_Z*MAX_Z)

https://github.com/BennyQBD/3DEngineCpp/blob/master/src/rendering/lighting.cpp
http://roar11.com/2015/05/dealing-with-shadow-map-artifacts/

Neutral normal map: (128, 128, 255)
Helmet reference: https://gkjohnson.github.io/threejs-sandbox/shader-debugger/
https://bartwronski.com/2019/09/22/local-linear-models-guided-filter/)

0.8.x Release (Gameplay)

- Physics
- Add AmdPixEXT to WinPixEventRuntime
- Make sure that OpenAL32.dll is copied to aether3d_build
- Add sphere.ae3d to .zip (PBRSample uses it)
- Decals, ref https://tuket.github.io/posts/2022-08-05-explosion-decals/
- Decal shader ref https://stackoverflow.com/questions/47593690/projected-decal-shader-using-forward-rendering-in-opengl-es
- Profile time to first present(), compare Vulkan and D3D12. Metal: 5.1 s
- Add City assets to distribution and update the comment in City.cpp.
- Multiple particle emitters.
- More transparent stuff in test scene.
- Particle depth testing on D3D12 and Vulkan.
- Particle age, velocity, collisions from depthnormals.
- and https://www.youtube.com/watch?v=tI70-HIc5ro
- Chrome tracing or Tracy: https://github.com/hrydgard/minitrace/blob/master/minitrace.c https://stackoverflow.com/questions/38550396/timestamp-in-chrome-devtool-s-timeline-json
- Audio Gain and Pitch
- Hide mouse pointer
- Mouse pointer wrap: https://github.com/mcpcpc/xwm/blob/main/xwm.c
- Some gameplay functionality (FPS movement, open doors, keypads, operate elevators)
- Editor:
  - Animation controls (play different anims, play frame-by-frame etc.).
  - Multi-selection.
  - Ctrl+backspace in textfields
  - Pixel-perfect picking.
  - Rotation in Inspector
  - Component enabled checkbox
  - When dragging an object, snap to grid if Control is pressed (or some other key)
  - Gizmo ref: https://github.com/TheCherno/ImGuizmo
  - Camera component preview
  - Drag .ae3d from file manager to scene.

0.8.9 Release (VFX)

- Water rendering
- Better Bloom, see http://www.iryoku.com/next-generation-post-processing-in-call-of-duty-advanced-warfare
- DoF
- Material editor (Nuklear)

- Specular or roughness maps. "specular" or "metalness" workflow. Filament seems to be "metalness"?
- IBL: Use Cube map in Standard shader.
- normal vector in world space when sampling cube map
- Tree mesh, eg. from https://casual-effects.com/data/index.html
- MikkTSpace normals, per-pixel (Blender implementation has some bugfixes related to original)
- Chromatic aberration
- VR: Frustum cull only once
- ASTC mipmaps and compatibility
- Mipmap Kaiser filter
- Foliage (alpha test should use Kaiser mipmaps)
- Distortion pass, sniper lens distort
- Film grain
- Filmic Tone mapping
- Editor
    - Drag texture from file manager to object
    - Two-axis transform, also duplicate object when shift is pressed during transform
    - Grid reacts to two-axis transform, orients to the current axis
    -

0.9 Release

- Procedural sky
- BC7 support
- Shadow gaussian blurring (5x5)
- VR: Show stats in HMD
- VR: Camera moving
- VR: Haptics
- VR: Controller rotation
- VR: Remove hmdView from Transform if possible
- Attach nodes into skinned mesh joints
- Premultiplied alpha
- Billboards
- IES Light profiles
- YouTube demo
- Eye adaptation
- Metal: Faster texture loading
- Multiple shadows: https://www.microbasic.net/tutorials/shadow-mapping/Full.html
- Shadow test: spotlight in a directional light's shadow, casting shadow
- Resolution changing at runtime
- Animation blending
- Video textures
- Controller rumble

- Velocity blur
- Make sure that UI authoring is easy
- Skin shading
- Fog
- Blendshape animation
- LineRendererComponent
- Option to freeze culling but allowing to move to see what's culled
- Performance test scenes
- Editor
  - Asset browser
  - Test file open dialogs etc. on KDE.
  - Prefabs
  - When dragging a game object, pressing Shift leaves a copy in current position.
  - Pan velocity dependent on distance
  - Scale gizmo
  - Scene statistics
  - Selection groups
  - Invert selection
  - Game object hierarchy
  - Particle inspector
  - Texture inspector
  - Rotate gizmo
  - Material inspector
  - Physics inspectors
  - If multiple objects are selected, make transformation inspector change update all of them
  - Duplicate entire selection
  - Gizmo to not receive shadow
  - Camera layer mask and game object layer
  - Camera frustum visualization
  - Inspector scrolling
  - Grid snapping

1.0 Release

- Linux installation directory, .dpkg etc.
- Gnome alt-tab to show app name and icon
- Occlusion culling
- Wiki: Comparison table with other engines
- Fur (geometry shaders? OpenGL Programming Guide)
- Proof-of-concept game
- Light shafts
- HDR texture support
- DOF

- GI
- macOS properly packaged build
- Document how to create a new project that uses Aether3D
- glTF mesh converter
- Editor
    - TBD

X. Later/undecided

- Test changing MSAA sampling patterns per-frame
- Wireframe rendering mode
- Check that triangle count matches Pix, Xcode or Radeon profiler.
- Make ORCA scenes load: https://developer.nvidia.com/orca
- Test D3D12 over RDP, also closing RDP connection when game is running.
- Metal: Fix MSAA when not rendering to backbuffer.
- MetalSampleiOS: SSAO
- ASTC cube map faces
- Fix rotating mesh culling (PBR-cube)
- Point light shadows https://learnopengl.com/Advanced-Lighting/Shadows/Point-Shadows
- Blender: Export skin and animation, ref https://github.com/MomentsInGraphics/vulkan_renderer/blob/main/tools/io_export_vulkan_blender28.py ja https://momentsingraphics.de/ToyRenderer5Animations.html
- Blender: Export materials.
- Tangent test: reflective cube
- VR: MSAA
- Support mipmap generation for NPOT textures, if their dimension is multiple of 4
- Shadow map MSAA https://github.com/MauriceGit/Variance_Shadow_Maps
- Decals
- Light cookie
- win32 input to work with multiple monitors
- Optimize frustum culling
- Cloth rendering (eg. verlet)
- WM_DPICHANGED etc. http://ourmachinery.com/post/dpi-aware-imgui/
- Detail maps
- Nice-to-have: Camera in (0, 0, -10), cube in (0, 0, 0), must see it.
- Hair rendering
- Area light: Rectangle
- Area light: Sphere
- Webcam texture
- .obj converter to triangulate all meshes
- Stream scene from server

Ongoing:

- Fix TODOs and FIXMEs

- High framerate can reveal synchronization issues.
- Reduce storing/using pointers
- Reduce globals
- Reduce branching
- Try to calculate test coverage
- Don’t make internal methods public in Include folder
- Avoid unneeded dynamic memory allocation
- Cache misses: "perf stat"
- Better demo scene, also make it so heavy that it can't reach 60 Hz on R9 Nano. City scene, include a mecha

Misc Vulkan:

- own command pool for staging
- memory allocator
- Verify hardware limits when creating/binding resources
- combine vertex and index buffer storage, helps cache

Release manual testing checklist (TODO: automate as much as possible):

- Compile with all configurations (reminder: also Release on macOS)
- Disable Sponza in Misc3D
- Update Github Wiki.
- Copy the release and assets into external hard drive
- Soak test
- Visualize BRDF (like in MERL), compare shading against Unity, Unreal and Disney BRDF Explorer
- Test on Windows with MinGW: https://nuwen.net/mingw.html
- Check DONE items and Features listed in the project’s GitHub page
- Test previous releases’ change lists
- Test that disabled game object doesn’t render or cast shadow
- Check build steps
- Git clone into a clean directory
- Check for memory leaks
- Test translucency
- Test transformation hierarchy, also culling
- DONE Test all tools, including Blender exporter
- Test renderers on NVIDIA (DONE), AMD (DONE) and Intel
- DONE Test 2D and Cube render textures
- Test fullscreen
- DONE Test MSAA, especially when the scene contains render textures
- DONE Test Editor on multiple computers, especially cameras
- Test hot-reloading
- Test gamepad
- Process Explorer can show GPU mem usage

- python "/path/to/metrix++.py" collect --std.code.complexity.cyclomatic
- cd Engine; cloc . --exclude-dir=ThirdParty

Misc links:
WAVE: www-mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/WAVE.html

iOS motion filtering and reference position:
https://github.com/stfnjstn/MotionManagerDemo/blob/master/MotionManagerDemo/MotionManagerSingleton.m

Triangulate polygons in .obj:
https://www.opengl.org/discussion_boards/showthread.php/198728-loading-obj-file-how-to-triangularize-polygons
YouTube demo video ref: https://www.youtube.com/watch?v=MsOC8ntXVqM
Joku engine (kiva editori): https://www.youtube.com/watch?v=HYjhopUmD_k
Mipmap generation:
https://github.com/michal-z/pbr-test/blob/main/src/shaders/generate_mipmaps_cs.hlsl

Parameter doc example: Units: 1/m, range: [0, inf]: plain English here

https://github.com/Reedbeta/reed-framework/blob/master/asset-mesh.cpp
Peter Panning fix: http://www.gamedev.net/topic/675034-directional-shadow-map-problem/

Use srcStageMask = ALL_GRAPHICS_BIT and dstStageMask = FRAGMENT_SHADING_BIT when synchronizing render passes with each other.

linear depth should be 0 near light, 1 at far plane. Metalissa on nyt toisin päin, mutta toimii. (reverse depth?)
vulkan shadow rendering clip matrix:
https://blogs.igalia.com/itoral/2017/07/30/working-with-lights-and-shadows-part-ii-the-shadow-map/
testaa z * w vaihtoehtona z / w

```
float linearizeDepth( float z )
{
      float f = 400.0f;
      float n = 0.1f;
      float linearZ = (2 * n) / (f + n - z * (f - n));
      return linearZ;
}
```

Sascha Willems SSAO linear depth (inputtina MVP pos.z):

```
float linearDepth(float depth)
{
    float z = depth * 2.0f - 1.0f;
    return (2.0f * ubo.nearPlane * ubo.farPlane) / (ubo.farPlane +
ubo.nearPlane - z * (ubo.farPlane - ubo.nearPlane));
}
```

Shadow linear depth http://roar11.com/2015/05/dealing-with-shadow-map-artifacts/

"Yeah GGX explodes if its roughness value reaches zero (and also gets some crazy numerical imprecision bugs if it's close to zero), so I remap my roughness textures into some range like (0.05,1) too"

"Bilateral filter for downsampling instead of bilinear to reduce filtering errors. Bilateral is depth-aware. Incorrect to separate bilateral into horiz/vert but in practice it's neglible"

Do MSAA before post-processing.

	Vulkan	D3D12	Metal
MSAA	4	3	
Point Light	ok	ok	ok
Spot Light	ok	ok	ok
Spot Shadow	ok	ok	ok
Dir Shadow	ok	ok	ok
Render Texture 2D	ok	ok	ok
Render Texture Cube	ok	ok	ok
Bloom	ok	ok	ok
BC4	ok	ok	ok
BC5	ok	ok	ok
16-bit render target	ok	ok	ok

SSAO	ok	ok	ok
Particles	ok	ok	ok

1. Works if camera is rendered straight into backbuffer, otherwise no AA.
2. Flickering if using F+
3. validation error
4. Ei toimi partikkelien kanssa, validaatiovirhe.
5.

Profiloi city-scenen vulkan-kutsut, etenkin descriptorien bindaus ja pipelinejen settaus.

bitsPerPixel D3D12:
https://github.com/GPUOpen-LibrariesAndSDKs/D3D12MemoryAllocator/blob/master/src/D3D12MemAlloc.cpp

```
#if defined(FP_FAST_FMAF)
   float lerp3(float a, float b, float t) noexcept {
      return std::fma(t, b, std::fma(-t, a, a));
}
#endif
```

https://en.cppreference.com/w/c/numeric/math/fma