

The background is a collage of various game assets from 0 A.D. It includes a Roman-style building with a tiled roof, a classical temple with columns, a green-tinted architectural model, a white 3D model of a building, a tropical scene with palm trees, and a green-tinted scene with archers. A dark red banner is overlaid in the center.

0 A.D: Graphics Development

"Who said programming is boring?"



A.D.
Empires Ascendant

FOSDEM'21

0 A.D.



0 A.D.







Britons



Iberians



Mauryas



Romans



Athenians



Gauls



Macedonians



Ptolemies



Spartans



Carthaginians



Kushites



Persians



Seleucids

0 A.D.



0 A.D.

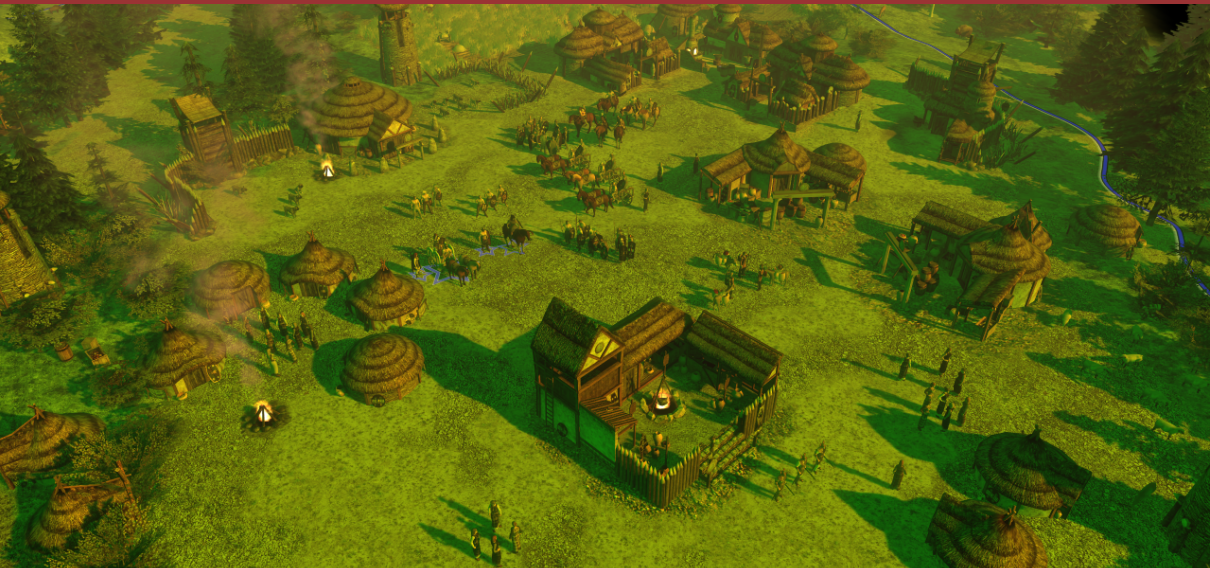












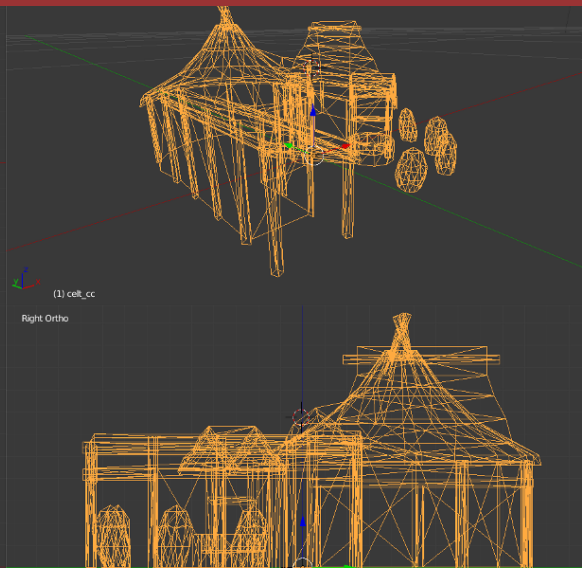
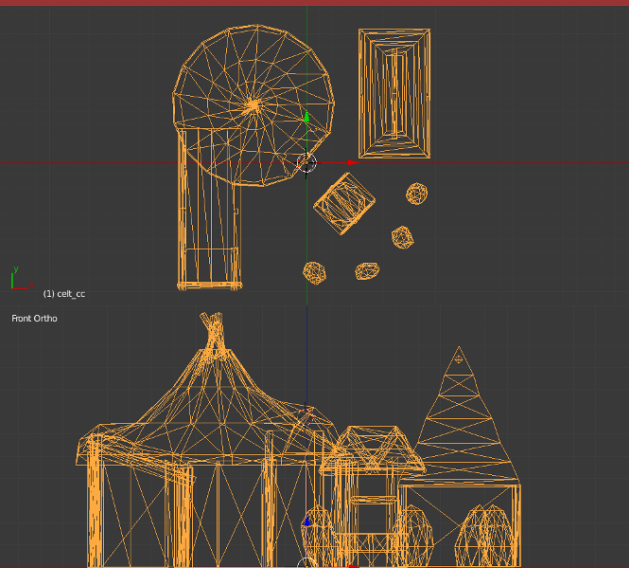




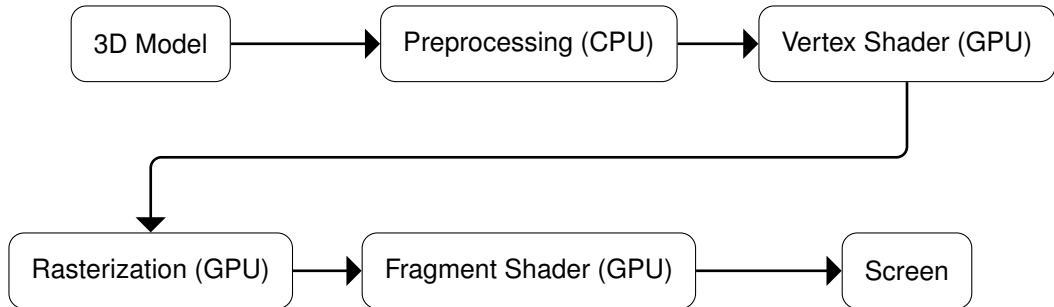










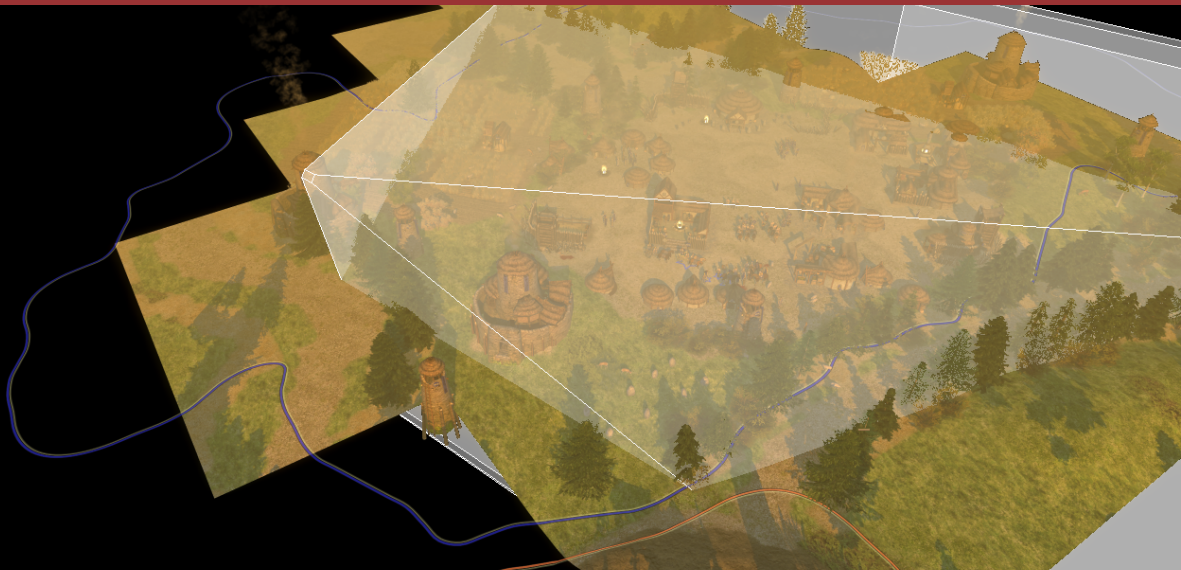


$$L_o(\mathbf{x}, \omega_o, \lambda, t) = L_e(\mathbf{x}, \omega_o, \lambda, t) + \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_o, \lambda, t) L_i(\mathbf{x}, \omega_i, \lambda, t) (\omega_i \cdot \mathbf{n}) d\omega_i$$

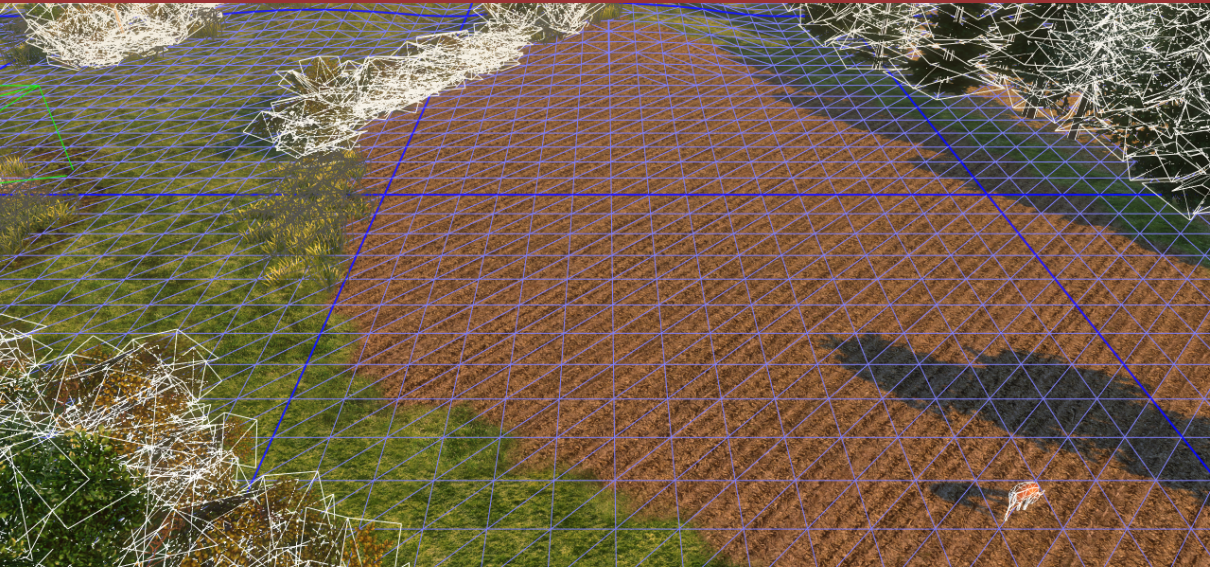
$$\text{Light}_{\text{reflected toward a camera}} = \text{Ambient light}_{\text{of surrounding space}} + \\ \text{Light}_{\text{from a sun}} \cdot (\overrightarrow{\text{Normal}}_{\text{of the position in space}} \cdot \overrightarrow{\text{Direction}}_{\text{toward the sun}})$$

$$\text{Light}_{\text{reflected toward a camera}} = 0^*$$

*Special case when you're looking into a black hole in its Galactic Center











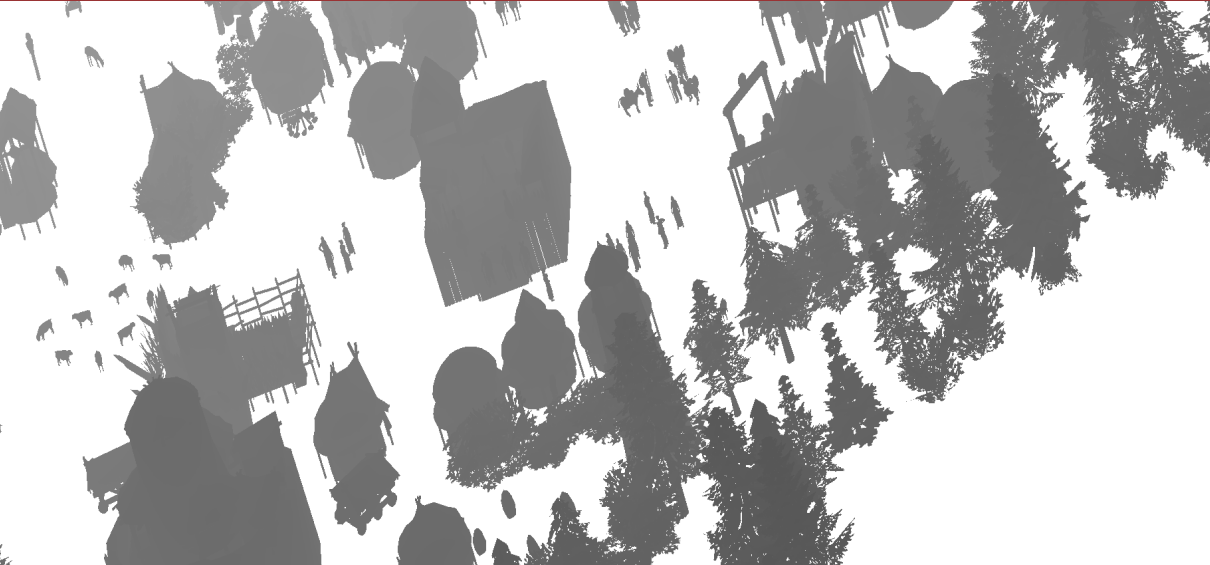
Fragment Shader



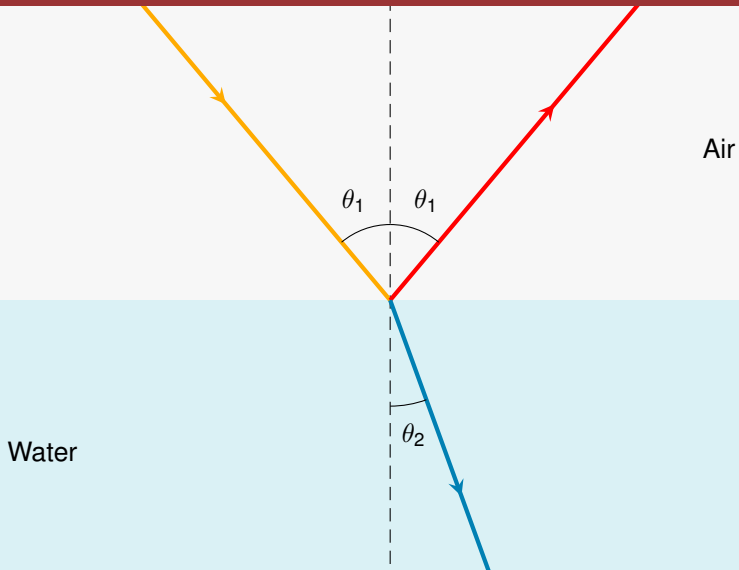
Fragment Shader



Shadow Mapping



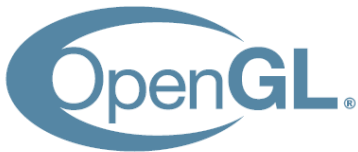


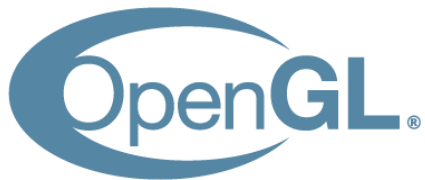


- Enumerate all objects on a map
- Cull invisible objects
- Render shadow map
- Render reflection map
- Render refraction map
- Render main game scene using previous textures
- Apply post-processing

Graphics API

glRotate glUseProgram glGetActiveUniform glGetSeparableFilter glActiveTexture glMultiMatrix glCopyConvolutionFilter2D
glClearDepth glCopyTexSubImage1D glDetachShader glGetShaderSource glGetQueryiv glMultiTexCoord
glCopyTexSubImage1D glGetTexImage glColorPointer glScissor glAreTexturesResident glBlendColor glTexSubImage2D
glDeleteShader glGenTextures glGetError glBufferSubData glPixelTransfer glCopyTexSubImage2D
glEvalMesh glNormal glCallList glDeleteShader glBufferSubData glMultiTexCoord glClear
glTexSubImage1D glDepthMask glGetProgram glColorMask glCompileShader glTexParameter glIndex glTexCoord
glClipPlane glVertexAttrib glPolygonOffset glClipPlane glGetBufferPointerv glReadBuffer glInitNames glPixelTransfer
glPointSize glCopyColorTable glDeleteTextures glCompressedTexSubImage3D glPixelStorei
glTexEnv glCompressedTexImage1D glTexSubImage3D glDepthFunc glPassive glInterleavedArrays glVertexAttribPointerv
glGetActiveUniform glColorTableParameter glSeparableFilter2D glRect glMapGrid glGetBufferPointerv
glGetAttachedShaders glCreateShader glListBase glGetMap glStencilMask





Past

2021

Future



GL4

Past

2010

2021

Future



GL3 GL4

Past

2008

2010

2021

Future



GL2 GL3 GL4

Past

2004

2008

2010

2021

Future



GL2 GL3 GL4

Past

Dinosaurs were
killed by an
asteroid

2004

2008

2010

2021

Future



GL1



GL2 GL3 GL4

Past 1992 Dinosaurs were killed by an asteroid 2004 2008 2010 2021 Future

Future Graphics API



vkCmdResolveImage vkDestroyBuffer vkWaitForFences vkCmdNextSubpass vkWaitForFences vkFreeDescriptorSets vkCmdSetImageMemoryRequirements2 vkCmdEndRenderPass vkGetPhysicalDeviceQueueFamilyProperties vkCmdCreateEvent vkCmdCopy vkGetImageMemoryRequirements vkBindBufferMemory vkCmdWriteTimestamp vkGetPhysicalDevice vkCmdCopyBuffer vkCreateImage vkCmdEndRenderPass2 vkCreateEvent vkDestroySampler vkCmdSetEvent vkGetPhysicalDeviceMemoryProperties vkGetPhysicalDeviceMemoryProperties vkCreatePipelineCache vkGetPhysicalDeviceMemoryProperties vkCreateRenderPass2 vkCmdSetLayout vkCreateEvent vkCmdBeginRender vkCmdDrawIndirectCount vkQueueSubmit vkGetRenderAreaGranularity vkWaitForFences vkGetQueryPoolResults vkCmdPipeline vkWaitSemaphores vkCmdCopyQuery vkDestroyCommandPool vkCreateGraphicsPipeline vkFreeDescriptorSets vkCreateDescriptorSet vkGetDeviceProcAddr vkCmdWaitEvents vkTrimCommandPool vkBindImageMemory vkCmdDraw

Bonus



$$M_x = \begin{pmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{pmatrix}$$

$$M_y = \begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$





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

play0ad.com
trac.wildfiregames.com/wiki/WikiStart

