# OpenGL ES Tools and Techniques

## Get faster, fast!

Session 514

**Seth Sowerby**
Manager, GPU Software Developer Technologies

# Introduction

- iOS is a great platform for games
- OpenGL ES is a great technology for powering games on iOS
- Apple provides OpenGL ES tools to help you make great games

# What You Will Learn

- What OpenGL ES tools are available to you
- Why you need these tools
- What new features are introduced in Xcode 4.5 and iOS 6.0 SDK
- Workflows and methods to get the most from these tools

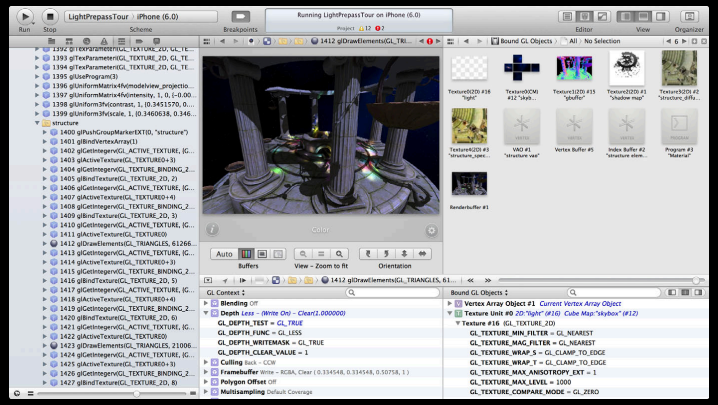
# OpenGL ES Tools Recap

The story so far

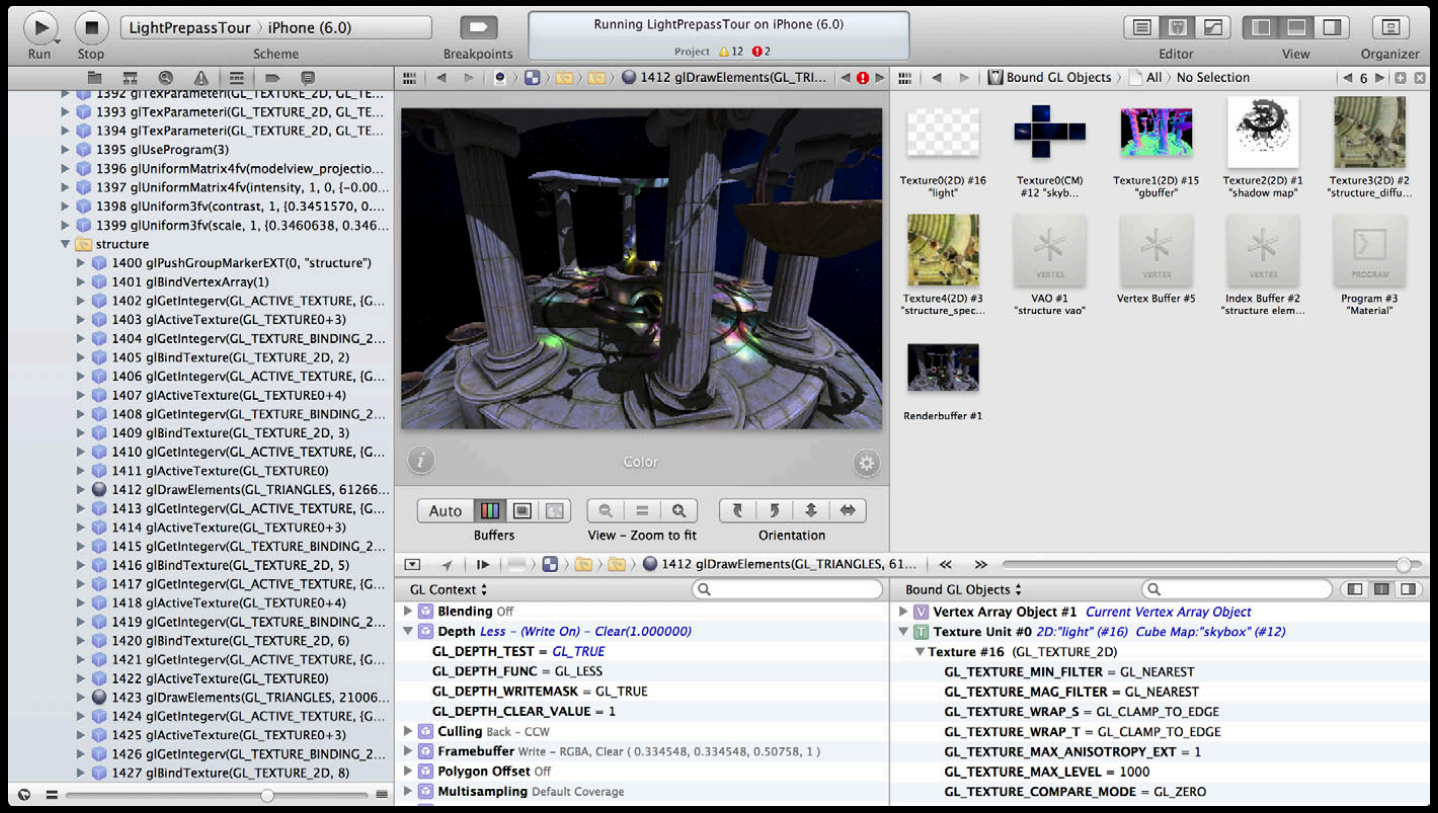
# OpenGL ES Frame Debugger

## Integrated into Xcode

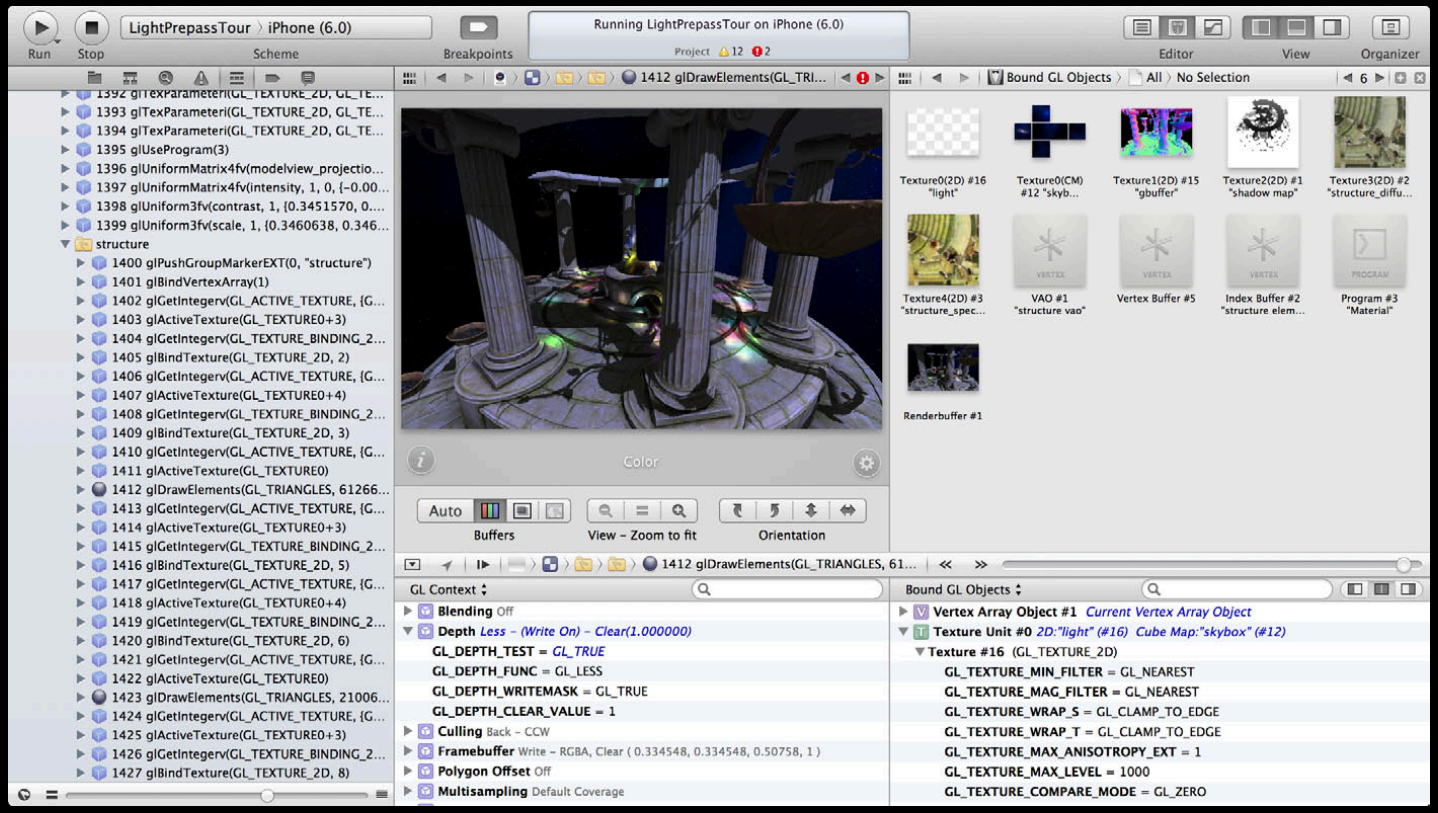
- OpenGL ES debugging built into Xcode
- Launch your app from Xcode
- Capture a frame
- Debug!



# OpenGL ES Frame Debugger



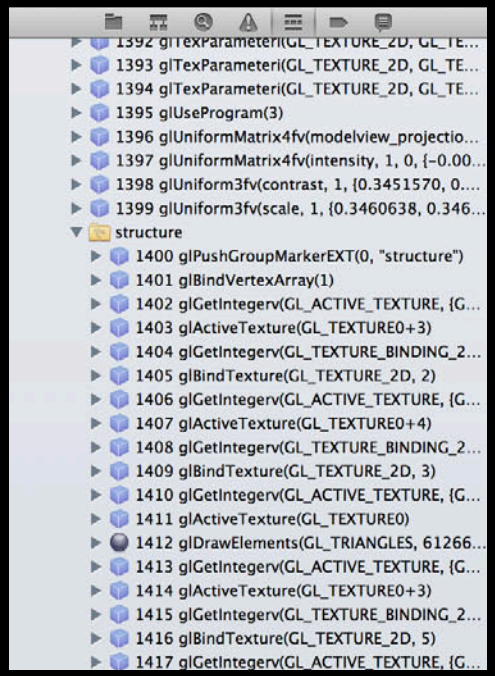
# OpenGL ES Frame Debugger



# OpenGL ES Frame Debugger

## Navigate your frame

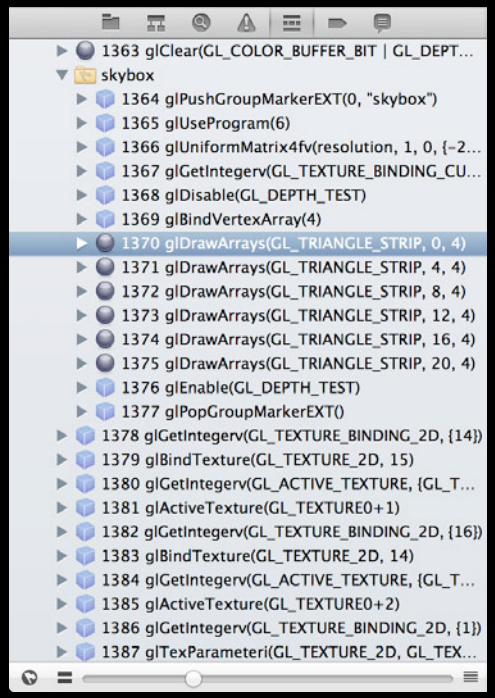
- All the GL calls from your frame displayed in the Debug Navigator
- Select a call to inspect GL at the point
- Annotate your frame with `glPushGroupMarkerEXT` / `glPopGroupMarkerEXT`
- Disclose a call to inspect the call stack



# OpenGL ES Frame Debugger

## Navigate your frame

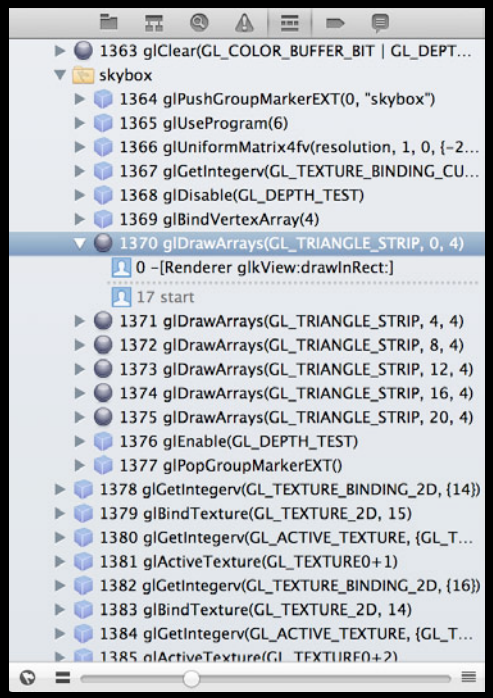
- All the GL calls from your frame displayed in the Debug Navigator
- Select a call to inspect GL at the point
- Annotate your frame with `glPushGroupMarkerEXT / glPopGroupMarkerEXT`
- Disclose a call to inspect the call stack



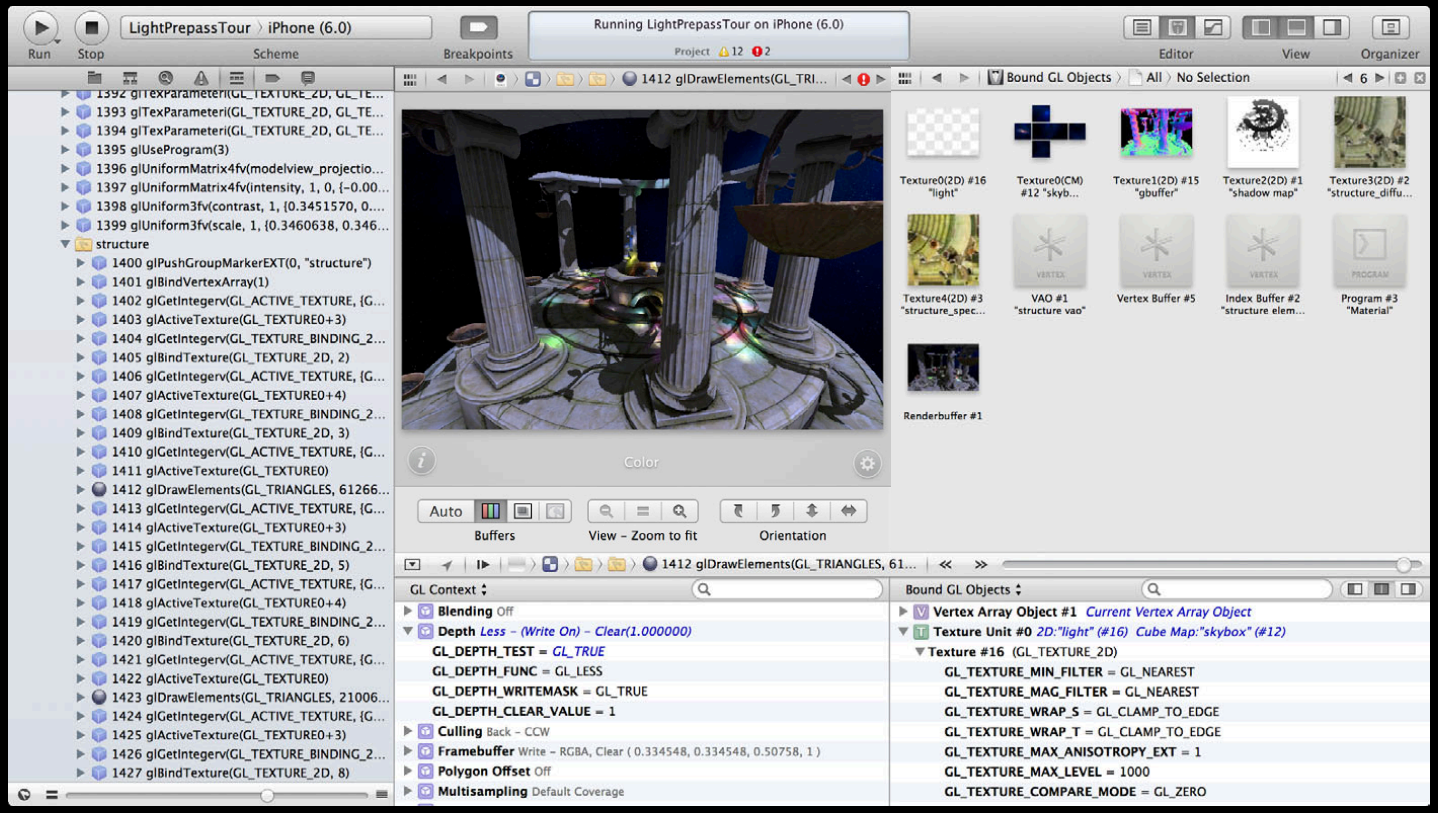
# OpenGL ES Frame Debugger

## Navigate your frame

- All the GL calls from your frame displayed in the Debug Navigator
- Select a call to inspect GL at the point
- Annotate your frame with `glPushGroupMarkerEXT` / `glPopGroupMarkerEXT`
- Disclose a call to inspect the call stack



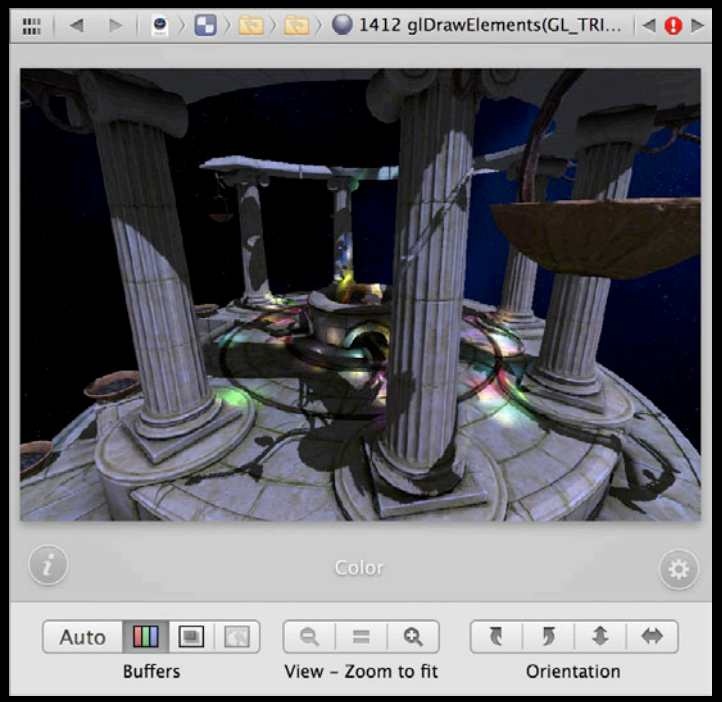
# OpenGL ES Frame Debugger



# OpenGL ES Frame Debugger

## Inspect your framebuffer

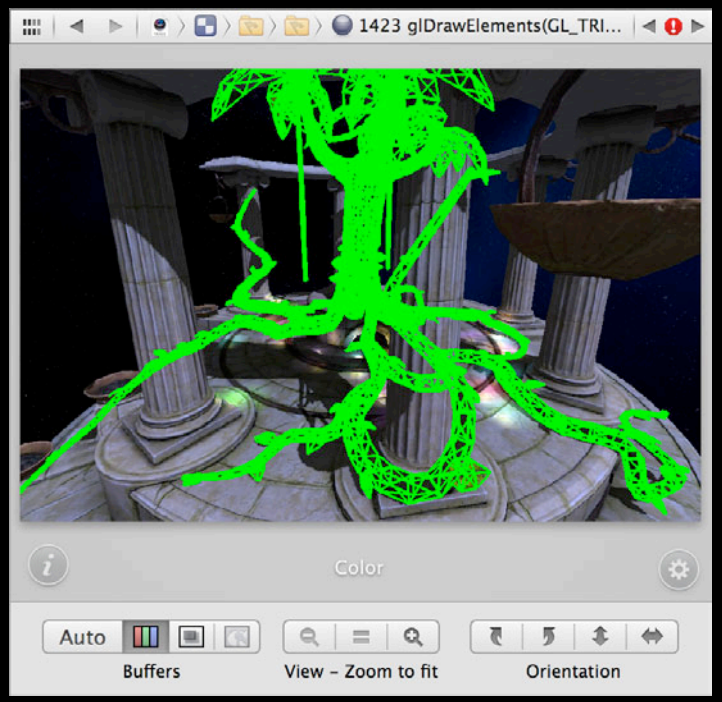
- Inspect the contents of the color, depth, and stencil buffers
- Current draw call highlighted as wireframe



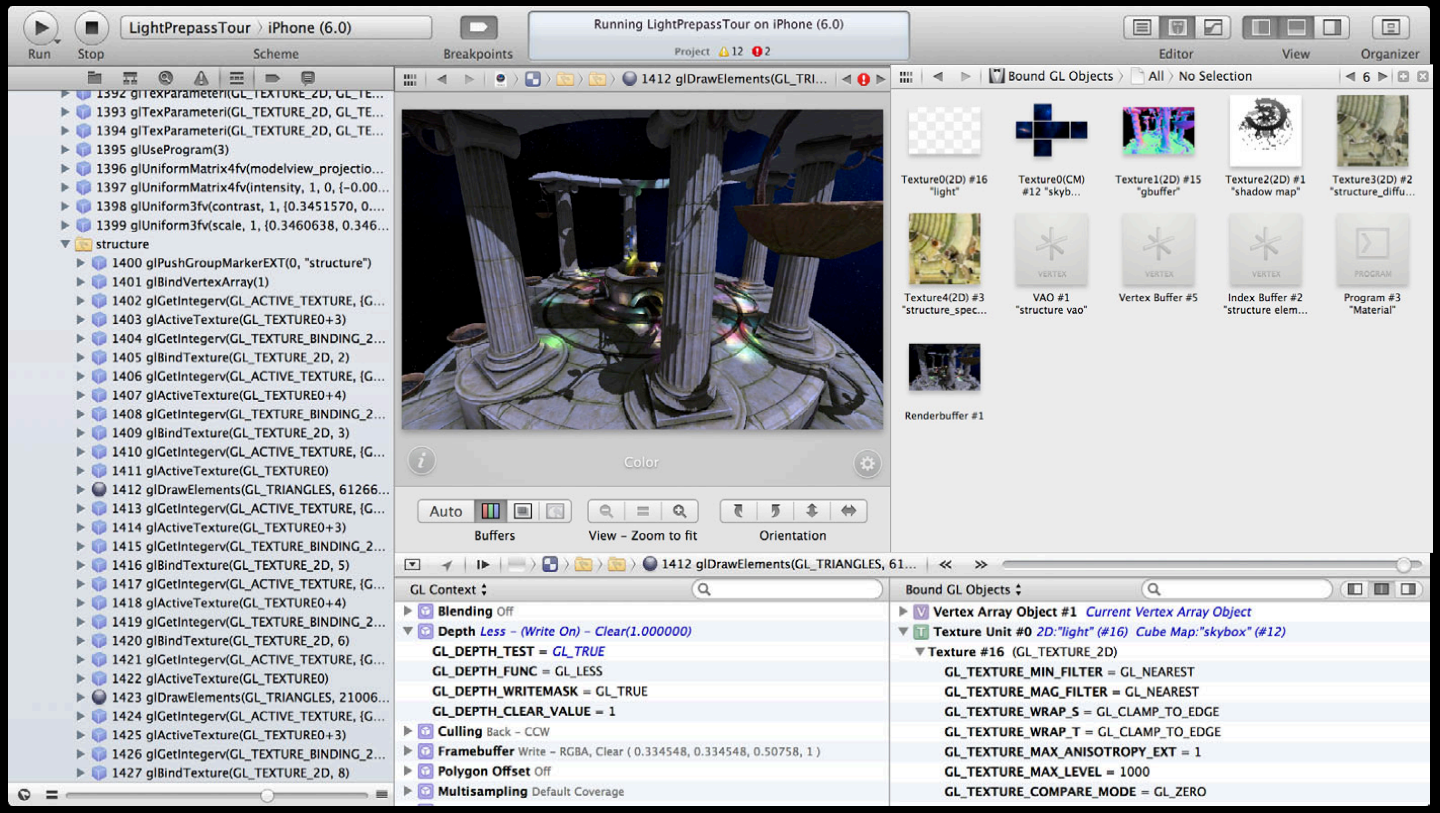
# OpenGL ES Frame Debugger

## Inspect your framebuffer

- Inspect the contents of the color, depth, and stencil buffers
- Current draw call highlighted as wireframe



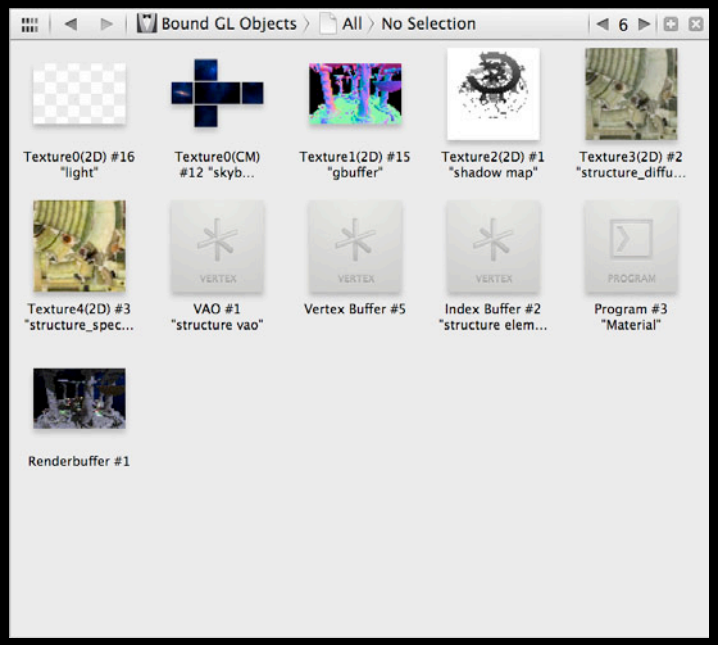
# OpenGL ES Frame Debugger



# OpenGL ES Frame Debugger

## Inspect GL objects

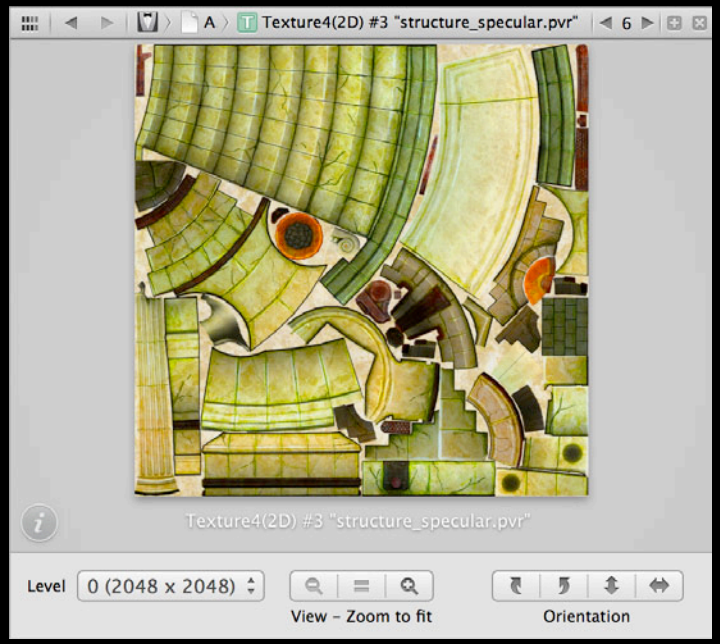
- Inspect all OpenGL ES object types
- Filter by bound objects or all
- Select an object to view in detail
- Label your resources with `glLabelObjectEXT`



# OpenGL ES Frame Debugger

## Inspect GL objects

- Inspect all OpenGL ES object types
- Filter by bound objects or all
- Select an object to view in detail
- Label your resources with `glLabelObjectEXT`



# OpenGL ES Frame Debugger

## Inspect GL objects

- Inspect all OpenGL ES object types
- Filter by bound objects or all
- Select an object to view in detail
- Label your resources with `glLabelObjectEXT`

Bound GL Obj… > All > VAO #1 "structure vao" 6

| | position | | | Attrib[1] | | |
|---|---|---|---|---|---|---|
| elem | x | y | z | x | y | |
| 0 | 9.988630e+01 | 0.000000e+00 | 5.021620e+02 | 1.399000e-01 | 6.969000e-01 | 7 |
| 1 | 9.989340e+01 | -5.403400e+00 | 5.021709e+02 | 1.951000e-01 | 1.900000e-03 | 9 |
| 2 | 1.486330e+02 | -5.403400e+00 | 4.899623e+02 | 2.903000e-01 | 1.900000e-03 | 9 |
| 2 | 1.486330e+02 | -5.403400e+00 | 4.899623e+02 | 2.903000e-01 | 1.900000e-03 | 9 |
| 3 | 1.486259e+02 | 0.000000e+00 | 4.899534e+02 | 2.082000e-01 | 6.969000e-01 | 6 |
| 0 | 9.988630e+01 | 0.000000e+00 | 5.021620e+02 | 1.399000e-01 | 6.969000e-01 | 7 |
| 3 | 1.486259e+02 | 0.000000e+00 | 4.899534e+02 | 2.082000e-01 | 6.969000e-01 | 6 |
| 2 | 1.486330e+02 | -5.403400e+00 | 4.899623e+02 | 2.903000e-01 | 1.900000e-03 | 9 |
| 4 | 1.959411e+02 | -5.403400e+00 | 4.730351e+02 | 3.827000e-01 | 2.000000e-03 | 9 |
| 4 | 1.959411e+02 | -5.403400e+00 | 4.730351e+02 | 3.827000e-01 | 2.000000e-03 | 9 |
| 5 | 1.959341e+02 | 0.000000e+00 | 4.730263e+02 | 2.744000e-01 | 6.969000e-01 | 6 |
| 3 | 1.486259e+02 | 0.000000e+00 | 4.899534e+02 | 2.082000e-01 | 6.969000e-01 | 6 |
| 5 | 1.959341e+02 | 0.000000e+00 | 4.730263e+02 | 2.744000e-01 | 6.969000e-01 | 6 |
| 4 | 1.959411e+02 | -5.403400e+00 | 4.730351e+02 | 3.827000e-01 | 2.000000e-03 | 9 |
| 6 | 2.413624e+02 | -5.403400e+00 | 4.515525e+02 | 4.716000e-01 | 2.100000e-03 | 8 |
| 6 | 2.413624e+02 | -5.403400e+00 | 4.515525e+02 | 4.716000e-01 | 2.100000e-03 | 8 |
| 7 | 2.413553e+02 | 0.000000e+00 | 4.515436e+02 | 3.381000e-01 | 6.969000e-01 | 6 |
| 5 | 1.959341e+02 | 0.000000e+00 | 4.730263e+02 | 2.744000e-01 | 6.969000e-01 | 6 |
| 7 | 2.413553e+02 | 0.000000e+00 | 4.515436e+02 | 3.381000e-01 | 6.969000e-01 | 6 |
| 8 | 2.560410e+02 | -5.403400e+00 | 4.427479e+02 | 5.141000e-01 | 1.200000e-03 | 8 |
| 9 | 2.844520e+02 | 0.000000e+00 | 4.257124e+02 | 3.984000e-01 | 6.970000e-01 | 5 |
| 7 | 2.413553e+02 | 0.000000e+00 | 4.515436e+02 | 3.381000e-01 | 6.969000e-01 | 6 |

# OpenGL ES Frame Debugger

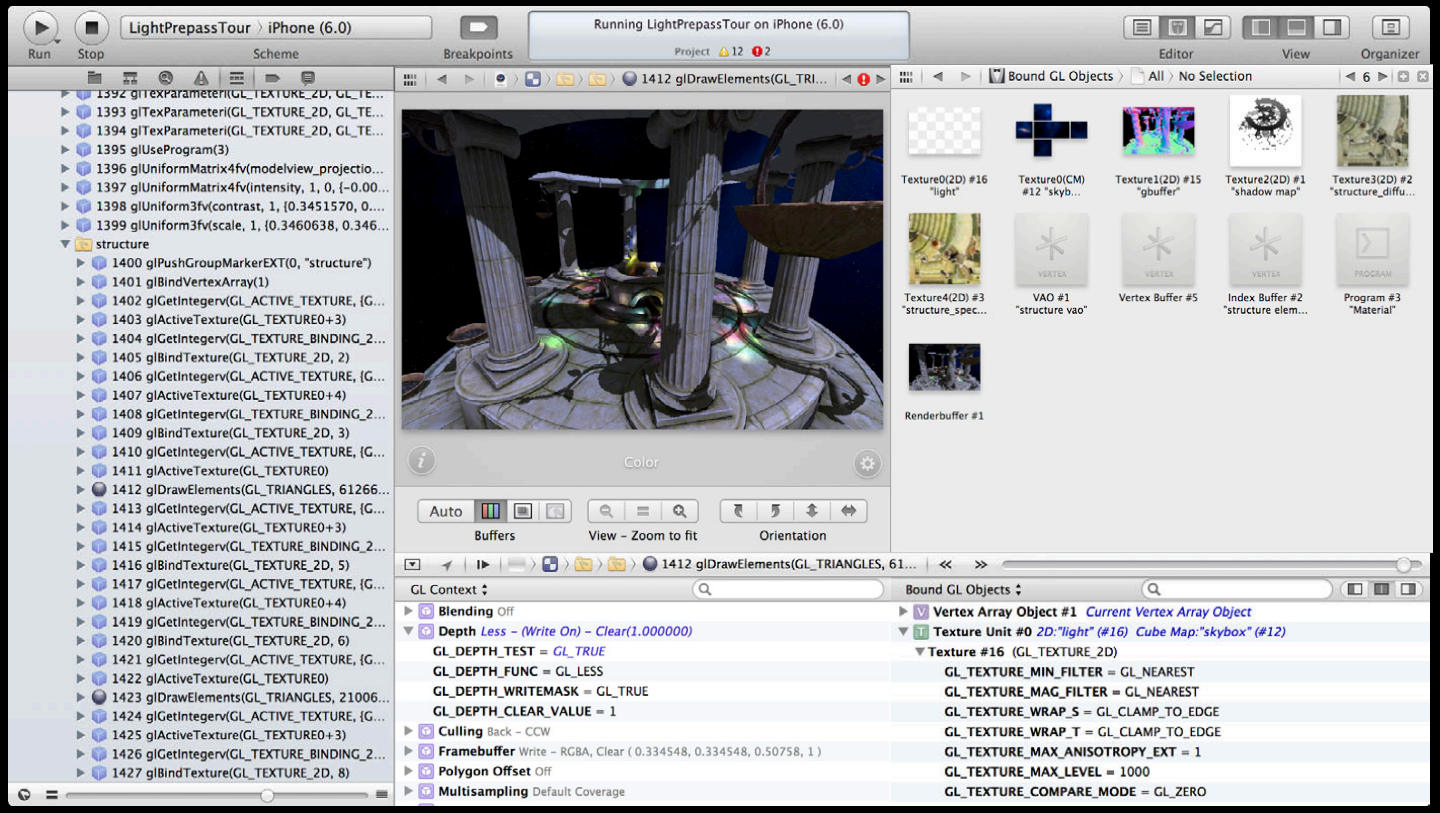
## Inspect GL objects

- Inspect all OpenGL ES object types
- Filter by bound objects or all
- Select an object to view in detail
- Label your resources with `glLabelObjectEXT`

Program #3 "Material" Vertex Shader 6

```
//*****************************************************************
// Edits to captured shaders are not saved to the source shader file
// Copy them back to your shader files when done or retrieve them fr
// the GPU Debug Log.
//*****************************************************************
//
//  Material.vsh
//  Light Prepass
//
//  Copyright 2011-2012 Apple, Inc. All rights reserved.
//

uniform mat4 modelview_projection_matrix;
uniform mat4 shadow_texture_matrix;

varying vec4 v_lightcoord;
varying vec4 v_shadowcoord;
varying vec4 v_texcoord;

attribute vec4 position;
attribute vec4 texcoord;

void main()
{
    gl_Position = modelview_projection_matrix * vec4(position.xyz, 1
    v_lightcoord = gl_Position * 0.5 + 0.5 * gl_Position.w;
    v_shadowcoord = shadow_texture_matrix * vec4(position.xyz, 1.0);
    v_texcoord = texcoord;
}
```

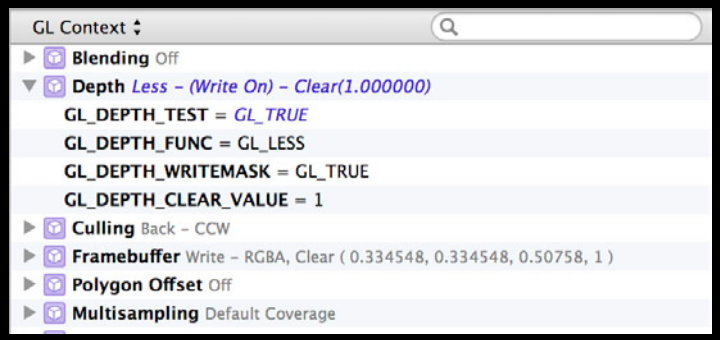
# OpenGL ES Frame Debugger

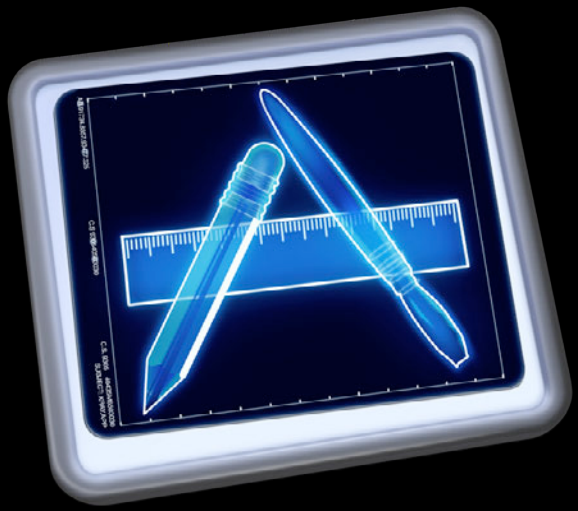


# OpenGL ES Frame Debugger

## Inspect GL state

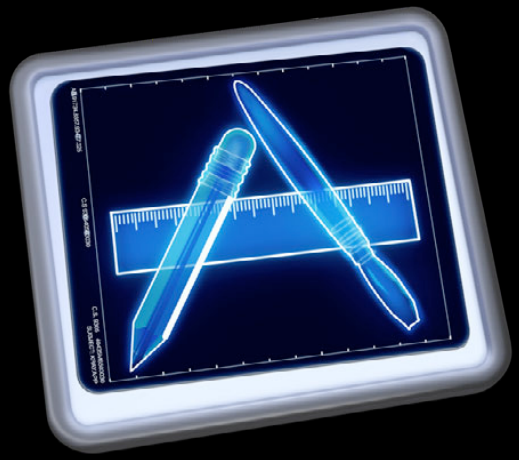
- Inspect all GL state
- Context state
- Object state
- Renderer capabilities
- Summarizes related states
- Highlights changes states



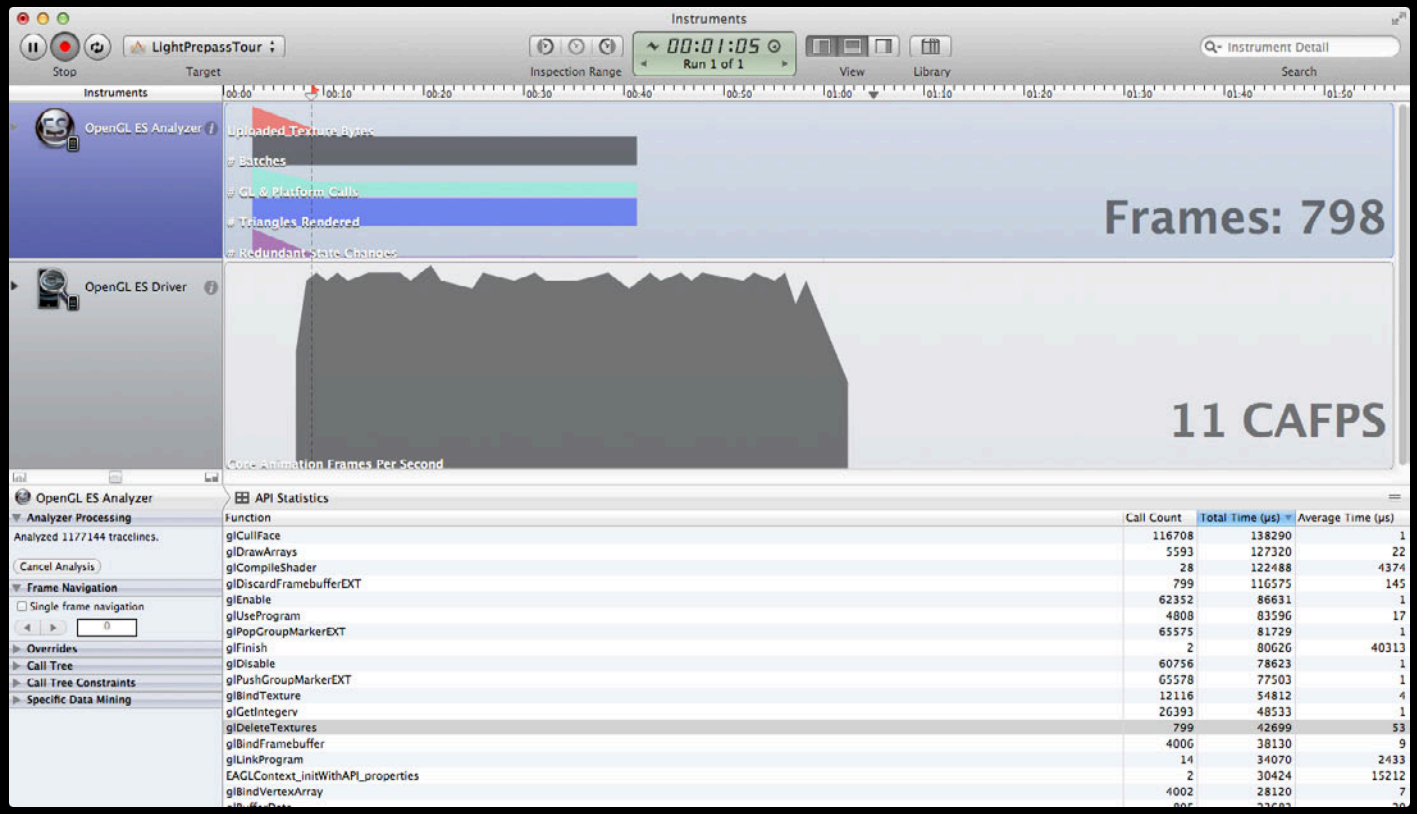


# OpenGL ES Analyzer Instrument

- Part of Instruments
- Graph representation of GL performance data
- Use alongside CPU profiling etc.



# OpenGL ES Analyzer Instrument



# OpenGL ES Analyzer Instrument

## OpenGL ES API Statistics

API Statistics

| Function | Call Count | Total Time (µs) | Average Time (µs) |
|---|---|---|---|
| glDrawElements | 254583 | 6519618 | 25 |
| glGenerateMipmap | 1 | 2275145 | 2275145 |
| EAGLContext_presentRenderBuffer | 1663 | 1559831 | 937 |
| glUniform3fv | 242936 | 1079557 | 4 |
| glUniformMatrix4fv | 251253 | 914282 | 3 |
| glCompressedTexImage2D | 105 | 489800 | 4664 |
| glClear | 6655 | 405659 | 60 |
| glDiscardFramebufferEXT | 1663 | 346054 | 208 |
| glStencilOp | 239610 | 325154 | 1 |
| glStencilFunc | 239610 | 318970 | 1 |
| glColorMask | 239610 | 317806 | 1 |
| glCullFace | 242938 | 306760 | 1 |
| glDrawArrays | 11641 | 259941 | 22 |
| glTexImage2D | 1679 | 219948 | 130 |
| glUseProgram | 9992 | 184302 | 18 |
| glEnable | 129787 | 178038 | 1 |
| glPopGroupMarkerEXT | 136460 | 172754 | 1 |

# OpenGL ES Analyzer Instrument

## OpenGL ES Expert

Expert > Categories

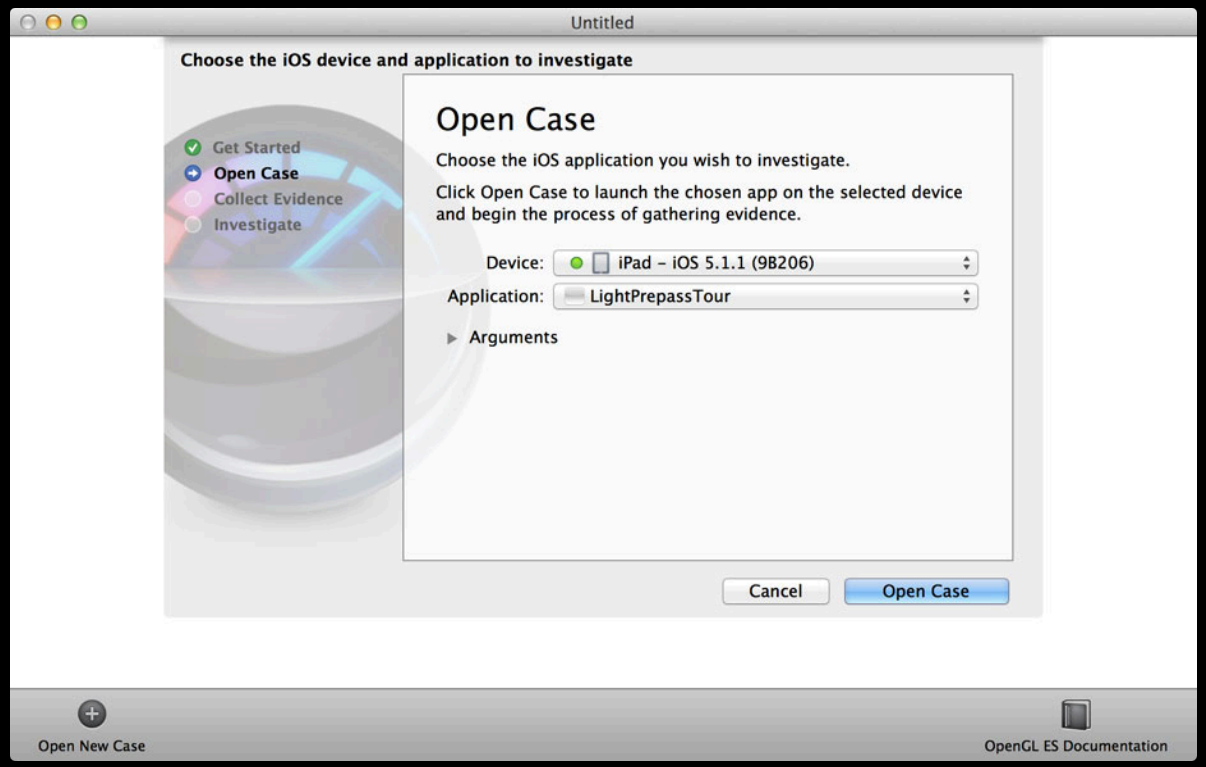
| Severity | Total... | Unique... | Category | Summary |
|---|---|---|---|---|
| ■ | 1862 | 1 | Validation Error | Invalid OpenGL ES API usage detected |
| ■ | 1862 | 1 | GL Error: Invalid Operation | GL Error: Invalid Operation |
| ▲ | 2 | 2 | CPU wait on GPU for Finish | Finish caused CPU wait for GPU |
| ▲ | 2 | 2 | Unnecessary Logical Buffer Store | Unnecessary logical buffer store operation |
| ▲ | 32002 | 19 | Redundant Call | Redundant state call |
| ▲ | 1862 | 1 | Report Opaque After Transparent | Opaque geometry rendered after transparent geometry |
| ▲ | 1861 | 1 | Fragment Shader Dynamic Branching + Memory Access | Dynamic branching and main memory access in fragment shader |
| ▲ | 1 | 1 | GenerateMipmap Texture Synchronization | GenerateMipmap caused texture synchronization |
| ▲ | 7444 | 2 | Dependent Texture Sampling | Dependent texture samples in fragment shader |
| ▲ | 1862 | 1 | Inefficient State Update | Inefficient state update |
| ▲ | 109 | 3 | Mip-Linear Filtering Might Not Be Needed | Mip-Linear filtering might not be needed |
| ▲ | 1863 | 2 | Logical Buffer Load | Slow framebuffer load |
| ▲ | 1867 | 4 | Texture Format Compactness | 8-bit per channel texture format |
| ▲ | 1859 | 1 | State Query Call Count | Numerous state query calls in current frame |
| ▲ | 1 | 1 | Recommend TexParameter Setup Before Upload | Mipmap usage changed after uploading texture data |
| ▲ | 1862 | 2 | Mipmapping Usage | Possible mipmapping usage scenario |

# OpenGL ES Performance Detective

- Automated performance analysis of your scene
- Runs rich set of experiments to find OpenGL ES performance bottlenecks
- Gives actionable advice to address performance

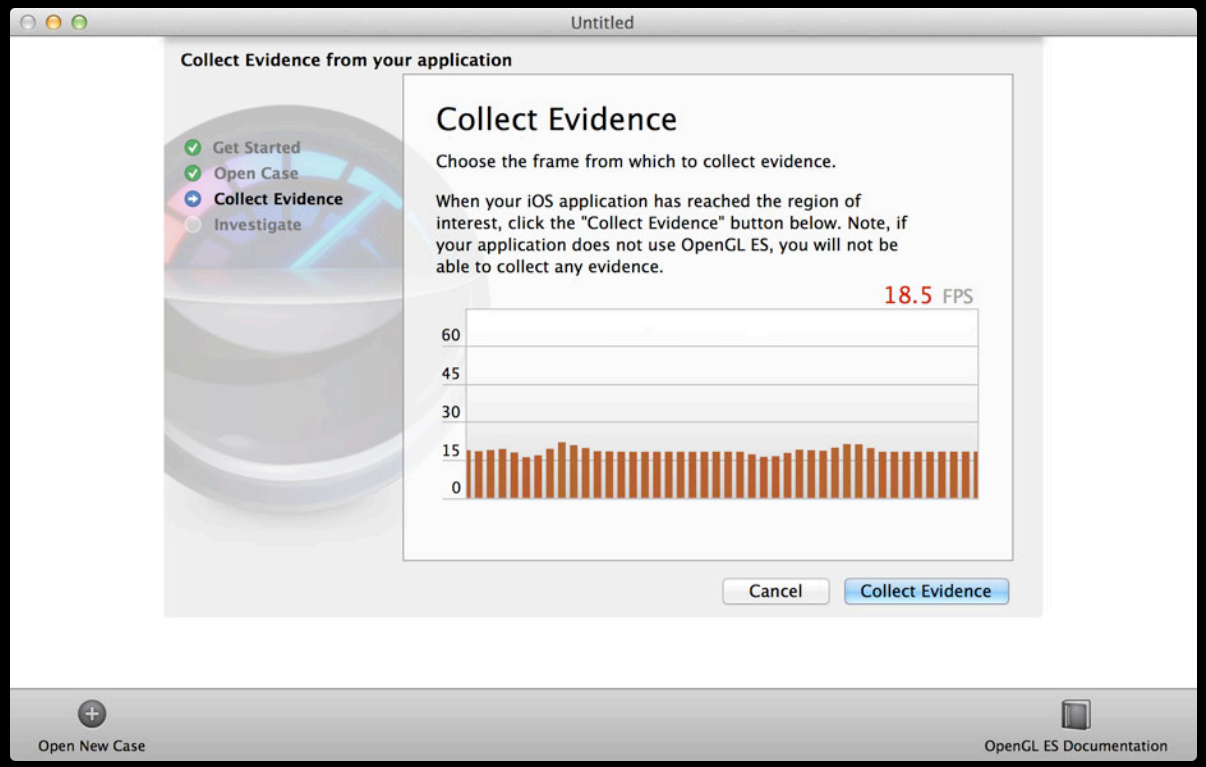
# OpenGL ES Performance Detective

## Select your app



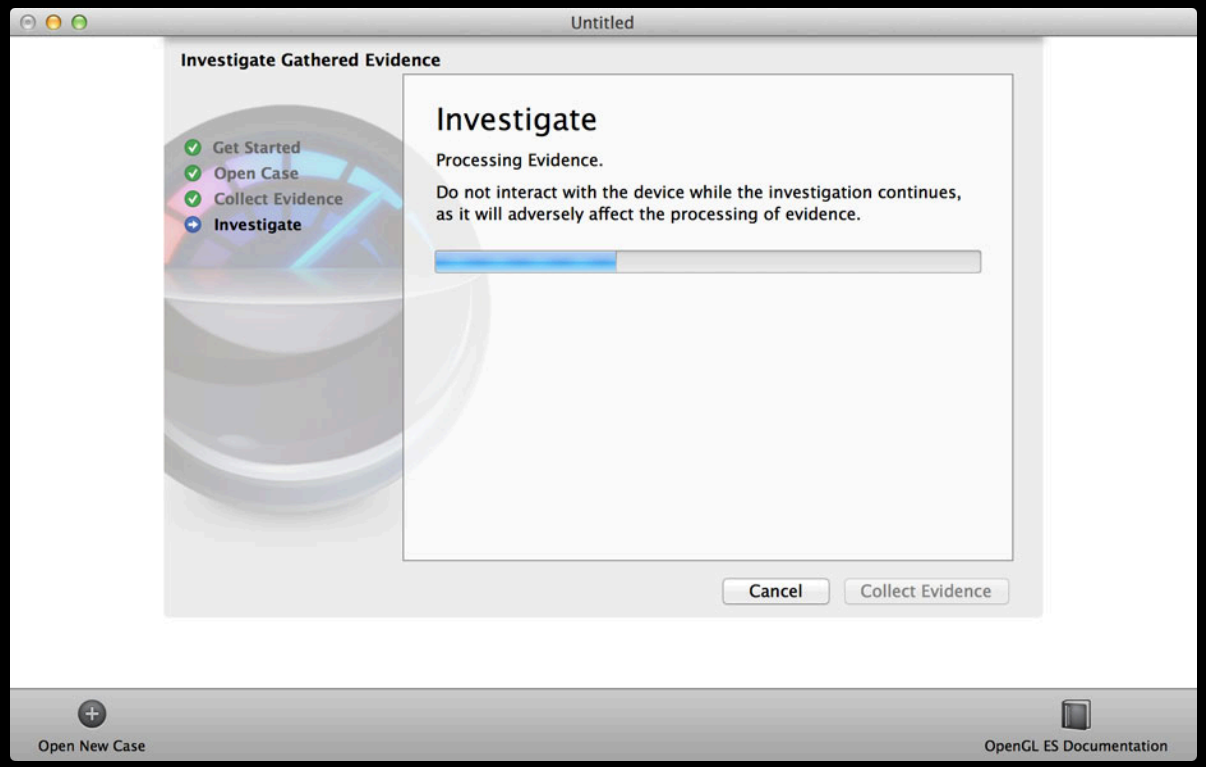
# OpenGL ES Performance Detective

## Trigger when to investigate



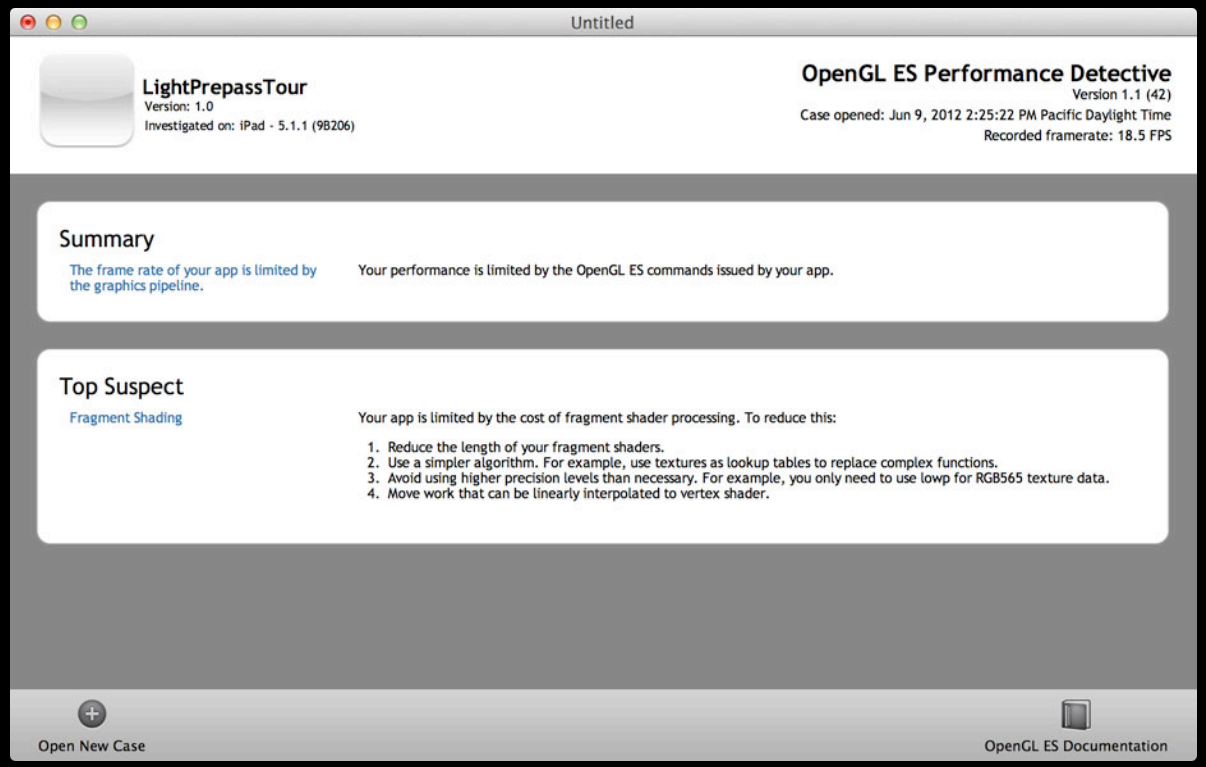
# OpenGL ES Performance Detective

## Let investigations ensue



# OpenGL ES Performance Detective

## Get actionable performance analysis



# What's New in Xcode 4.5

## Improvements for OpenGL ES

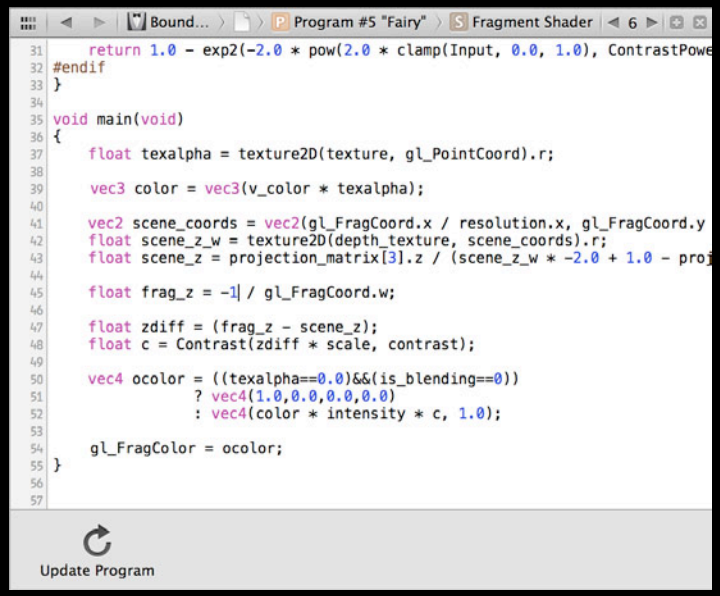
# 8 Great New Features

1. Shader edit and continue
2. Integrated OpenGL ES Expert
3. Multi-context debugging
4. Save and load frame captures
5. Performance monitoring
6. Integrated OpenGL ES Performance Analysis
7. OpenGL ES Performance Analysis 2.0
8. Find your slow shaders

1

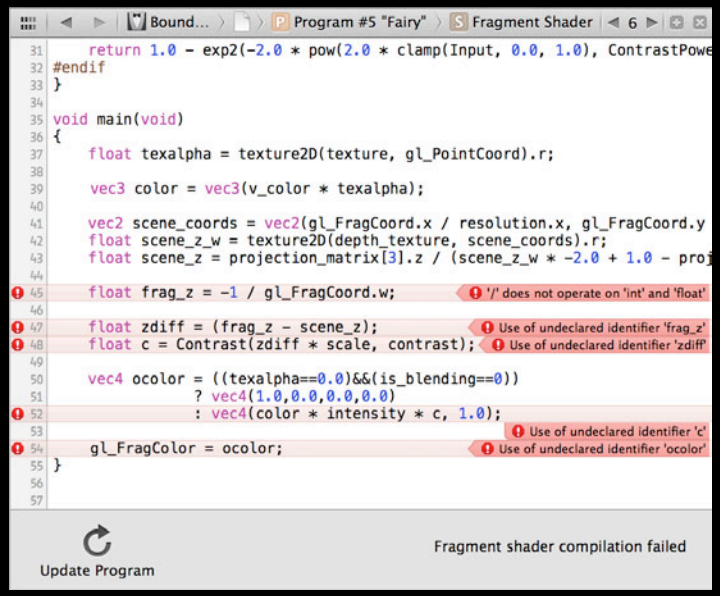
# Shader Edit and Continue

- Edit the GLSL for your programs
- See compilation and link errors inline
- See the effect of your changes instantly
- Edits saved to a shader edit log
- Resume the app to see your changes live



# Shader Edit and Continue

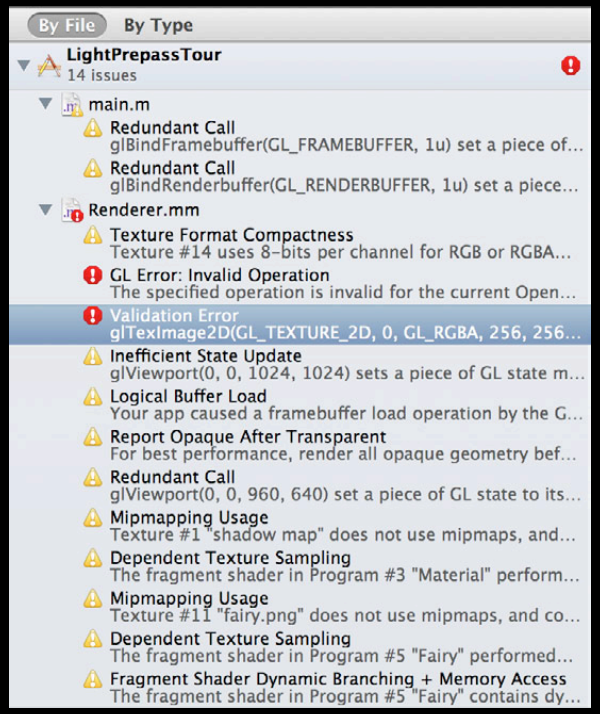
- Edit the GLSL for your programs
- See compilation and link errors inline
- See the effect of your changes instantly
- Edits saved to a shader edit log
- Resume the app to see your changes live



2

# Integrated OpenGL ES Expert

- GL issues for your captured frame in the frame debugger
- Catch bugs early and often
- No need to call glGetError all the time
- Go straight to your code to fix in place



By File | By Type

**LightPrepassTour**
14 issues

- main.m
  - Redundant Call
    glBindFramebuffer(GL_FRAMEBUFFER, 1u) set a piece of...
  - Redundant Call
    glBindRenderbuffer(GL_RENDERBUFFER, 1u) set a piece...
- Renderer.mm
  - Texture Format Compactness
    Texture #14 uses 8-bits per channel for RGB or RGBA...
  - GL Error: Invalid Operation
    The specified operation is invalid for the current Open...
  - Validation Error
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, 256, 256...
  - Inefficient State Update
    glViewport(0, 0, 1024, 1024) sets a piece of GL state m...
  - Logical Buffer Load
    Your app caused a framebuffer load operation by the G...
  - Report Opaque After Transparent
    For best performance, render all opaque geometry bef...
  - Redundant Call
    glViewport(0, 0, 960, 640) set a piece of GL state to its...
  - Mipmapping Usage
    Texture #1 "shadow map" does not use mipmaps, and...
  - Dependent Texture Sampling
    The fragment shader in Program #3 "Material" perform...
  - Mipmapping Usage
    Texture #11 "fairy.png" does not use mipmaps, and co...
  - Dependent Texture Sampling
    The fragment shader in Program #5 "Fairy" performed...
  - Fragment Shader Dynamic Branching + Memory Access
    The fragment shader in Program #5 "Fairy" contains dy...

By File | By Type

**LightPrepassTour**
14 issues

- main.m
  - Redundant Call
    glBindFramebuffer(GL_FRAMEBUFFER, 1u) set a piece of...
  - Redundant Call
    glBindRenderbuffer(GL_RENDERBUFFER, 1u) set a piece...
- Renderer.mm
  - Texture Format Compactness
    Texture #14 uses 8-bits per channel for RGB or RGBA...
  - GL Error: Invalid Operation
    The specified operation is invalid for the current Open...
  - Validation Error
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, 256, 256...
  - Inefficient State Update
    glViewport(0, 0, 1024, 1024) sets a piece of GL state m...
  - Logical Buffer Load
    Your app caused a framebuffer load operation by the G...
  - Report Opaque After Transparent
    For best performance, render all opaque geometry bef...
  - Redundant Call
    glViewport(0, 0, 960, 640) set a piece of GL state to its...
  - Mipmapping Usage
    Texture #1 "shadow map" does not use mipmaps, and...
  - Dependent Texture Sampling
    The fragment shader in Program #3 "Material" perform...
  - Mipmapping Usage
    Texture #11 "fairy.png" does not use mipmaps, and co...
  - Dependent Texture Sampling
    The fragment shader in Program #5 "Fairy" performed...
  - Fragment Shader Dynamic Branching + Memory Access
    The fragment shader in Program #5 "Fairy" contains dy...

# Integrated OpenGL ES Expert

By File | By Type

**LightPrepassTour**
14 issues

- main.m
  - Redundant Call
    glBindFramebuffer(GL_FRAMEBUFFER, 1u) set a piece of...
  - Redundant Call
    glBindRenderbuffer(GL_RENDERBUFFER, 1u) set a piece...
- Renderer.mm
  - Texture Format Compactness
    Texture #14 uses 8-bits per channel for RGB or RGBA...
  - GL Error: Invalid Operation
    The specified operation is invalid for the current Open...
  - Validation Error
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, 256, 256...
  - Inefficient State Update
    glViewport(0, 0, 1024, 1024) sets a piece of GL state m...
  - Logical Buffer Load
    Your app caused a framebuffer load operation by the G...
  - Report Opaque After Transparent
    For best performance, render all opaque geometry bef...
  - Redundant Call
    glViewport(0, 0, 960, 640) set a piece of GL state to its...
  - Mipmapping Usage
    Texture #1 "shadow map" does not use mipmaps, and...
  - Dependent Texture Sampling
    The fragment shader in Program #3 "Material" perform...
  - Mipmapping Usage
    Texture #11 "fairy.png" does not use mipmaps, and co...
  - Dependent Texture Sampling
    The fragment shader in Program #5 "Fairy" performed...
  - Fragment Shader Dynamic Branching + Memory Access
    The fragment shader in Program #5 "Fairy" contains dy...

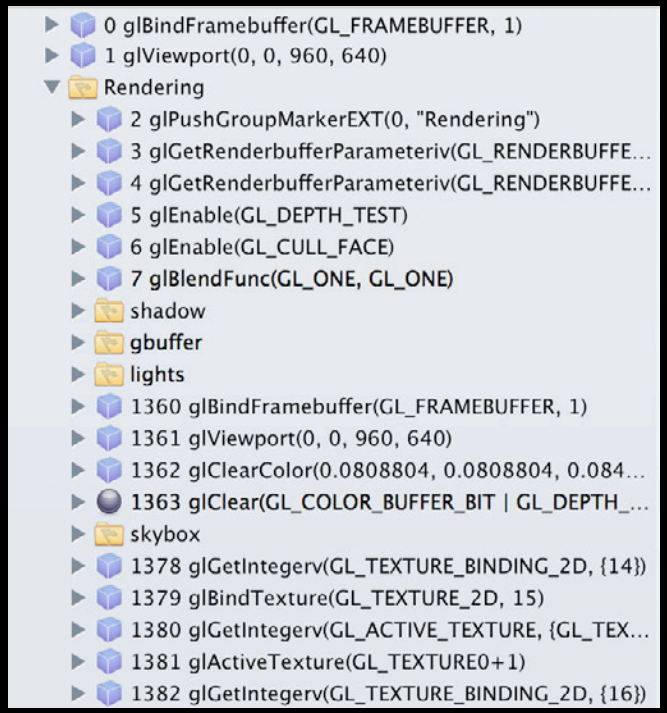
```
                farPlane];
1450            loaded = YES;
1451        }
1452
1453    #ifdef MULTI_CONTEXT
1454        [EAGLContext setCurrentContext:backgroundContext];
1455
1456        if(lastTexture != 0)
1457        {
1458            glDeleteTextures(1, &lastTexture);
1459        }
1460
1461        glGenTextures(1, &lastTexture);
1462
1463        glActiveTexture (GL_TEXTURE0);
1464        glBindTexture(GL_TEXTURE_2D, lastTexture);
1465
1466        glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, 256, 256, 0, GL_RGB,
                GL_UNSIGNED_BYTE, NULL);
1467        The specified operation is invalid for the current OpenGL state.
1468        glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, 256, 256, 0, GL_RGB, GL_UNSIGNED_BYTE, nullptr)...
1469    #endi  Texture #14 uses 8-bits per channel for RGB or RGBA data. If the image quality is acceptable, sa...
1470
1471        [self updateLightsForTime:frameTime];
1472        [lpView resizeRTTBuffers];
1473
1474        glEnable(GL_DEPTH_TEST);
1475        glEnable(GL_CULL_FACE);
1476        glBlendFunc(GL_ONE, GL_ONE);
1477
1478        if (shadowMode != kShadowModeNone)
1479        {
1480            [self lpView:lpView renderShadowBufferForTime:frameTime];
1481            if (debugView == kDebugViewShadow)
1482            {
```

3

# Multi-Context Debugging

- GL context switches shown within the Debug Navigator
- Quickly diagnose multi-context and threading issues
- Meets the needs of the upcoming wave of multi-context apps

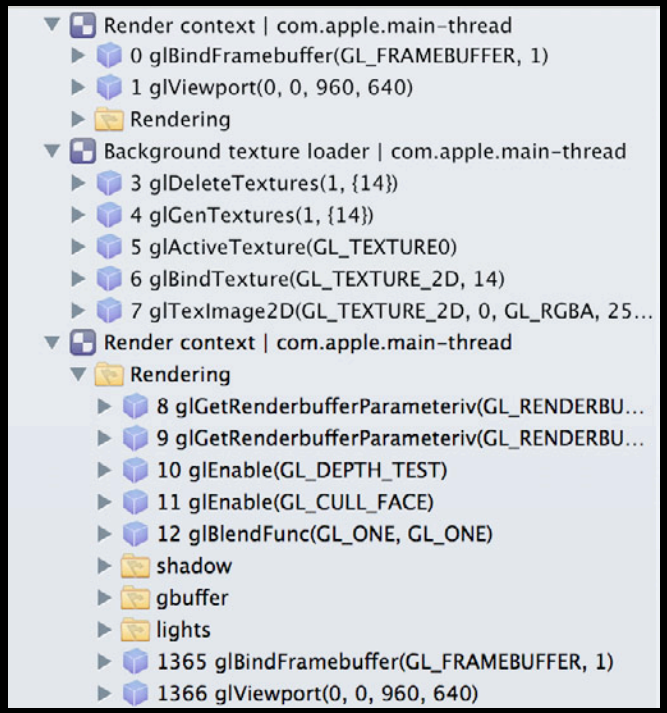
```
[EAGLContext setDebugLabel:]
```



# Multi-Context Debugging

- GL context switches shown within the Debug Navigator
- Quickly diagnose multi-context and threading issues
- Meets the needs of the upcoming wave of multi-context apps

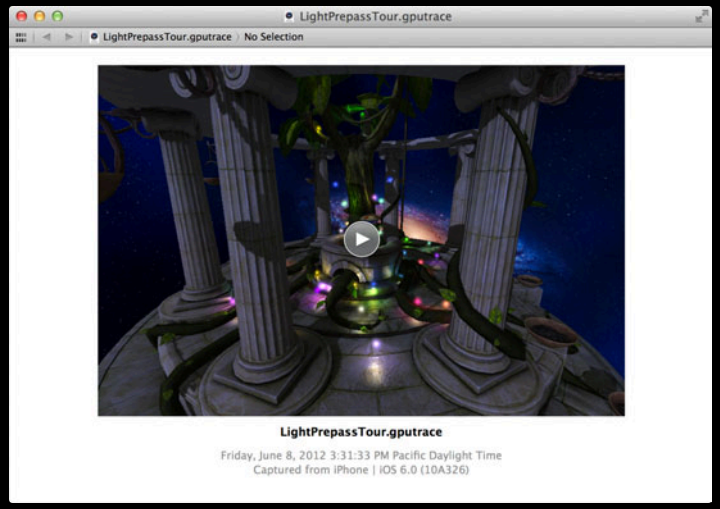
```
[EAGLContext setDebugLabel:]
```



4

# Save and Load Captured Frames

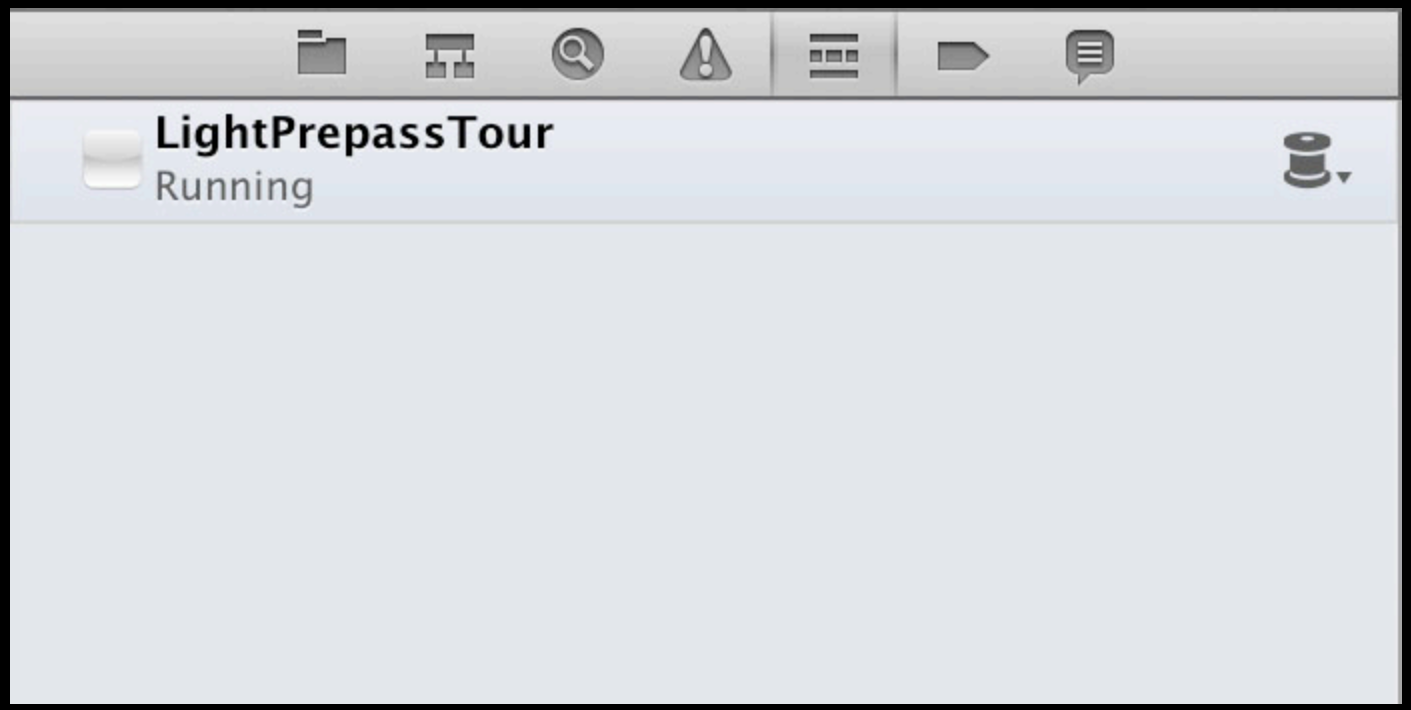
- Export the current frame while in the Frame Debugger
- Reload later to continue your work
- Or to test on another device
- Or send to someone else for their input



5

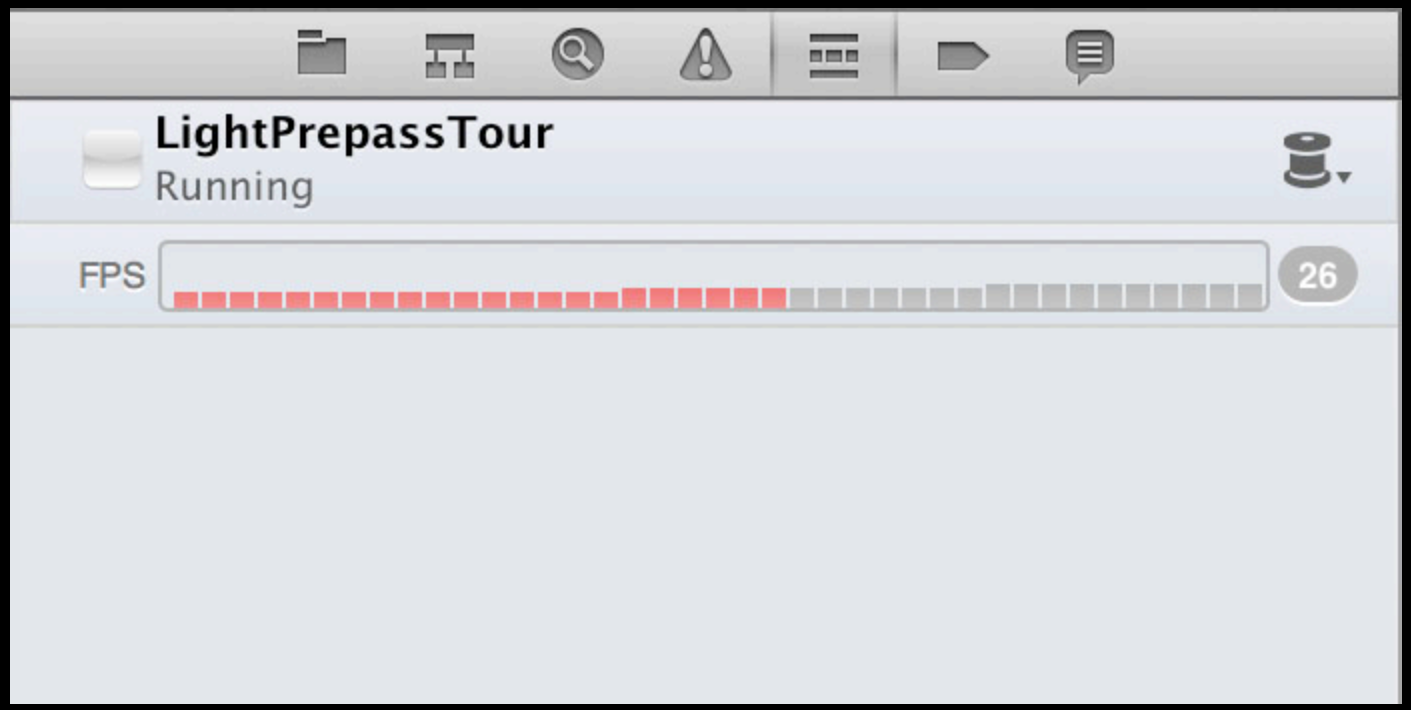
# Performance Monitoring

## Debug Navigator FPS

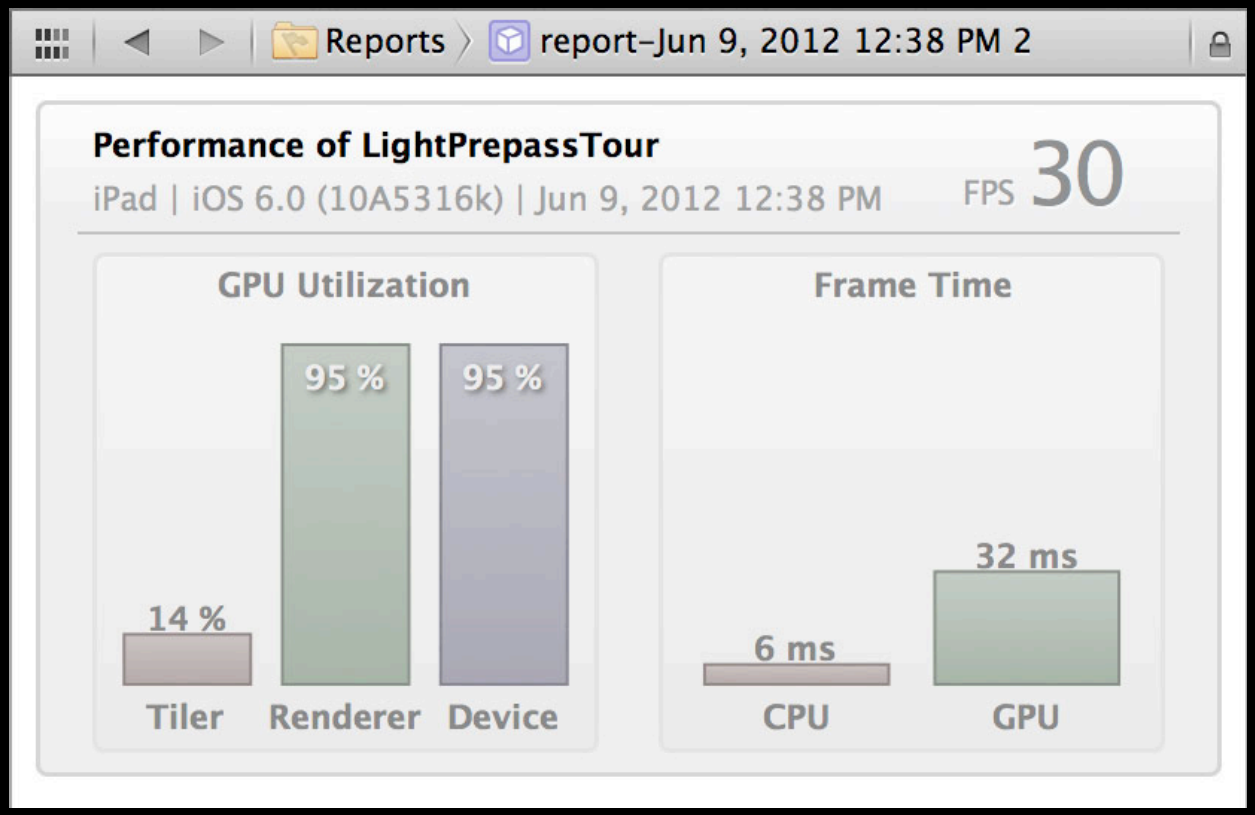


# Performance Monitoring

## Debug Navigator FPS



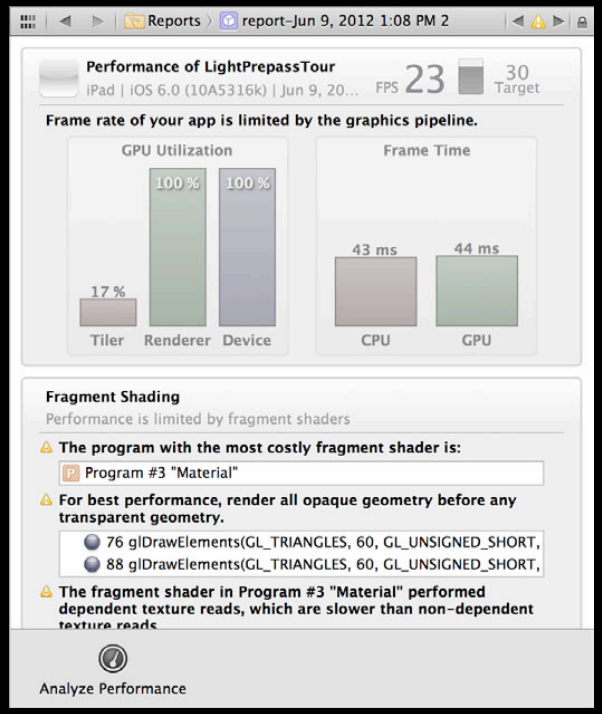
# Performance Monitoring



6

# Integrated OpenGL ES Performance Detective

- Performance analysis in Xcode for a cohesive workflow
- Works with the OpenGL ES Frame Debugger to let you drill down



7

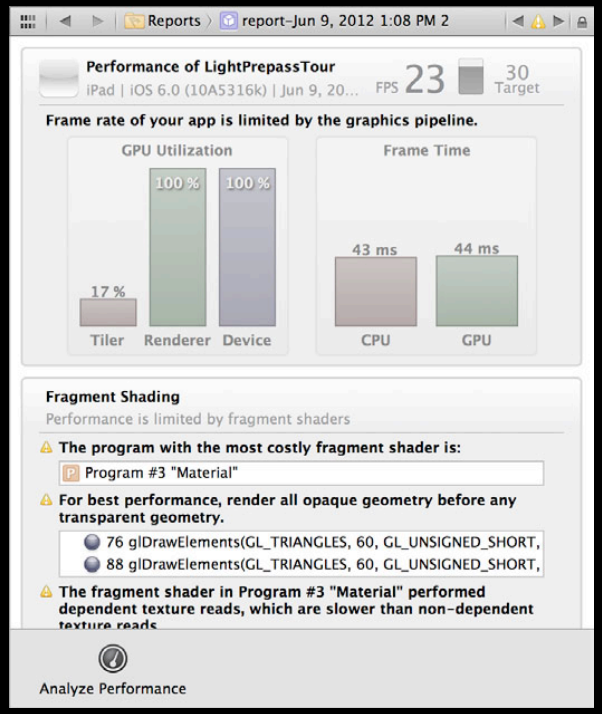
# OpenGL ES Performance Analysis 2.0

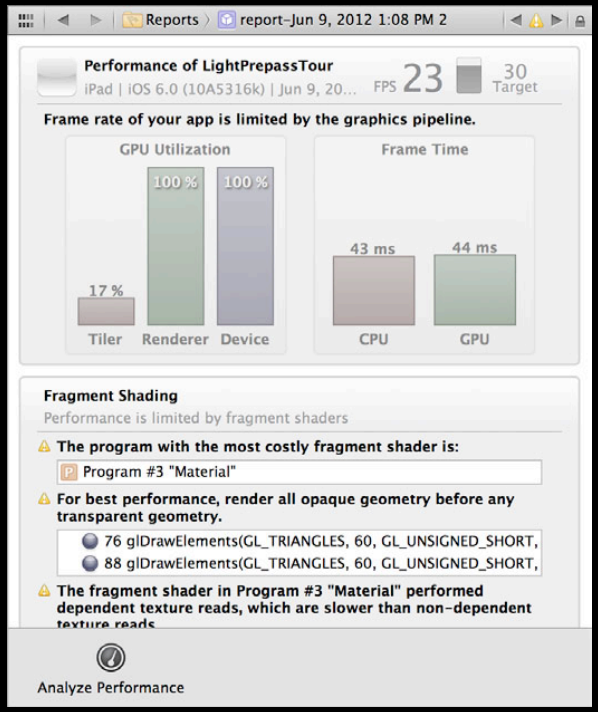
- Now detects CPU bottlenecks in OpenGL ES
  - Texture upload time
  - Vertex upload time
  - State validation time
- Detects more GPU bottlenecks
- Detects your target frame rate to guide its recommendations
- Integrated with the OpenGL Expert to relate its analysis to your GL commands

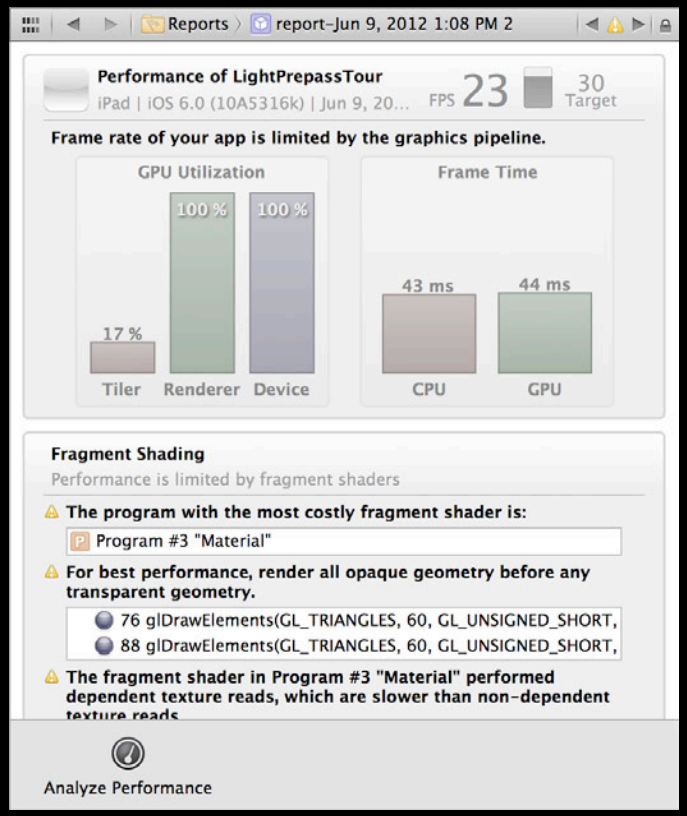
8

# Finds Your Slow Shaders

- Even more analysis for fragment shader bound apps
- Finds the most costly shader
- Finds shaders with potential performance issues
- Takes you direct to the GLSL source code
- Edit your shader there and then




Reports
report-Jun 9, 2012 1:08 PM 2
Performance of LightPrepassTour
iPad | iOS 6.0 (10A5316k) | Jun 9, 20...
FPS 23
30
Target
Frame rate of your app is limited by the graphics pipeline.
GPU Utilization
100 %
100 %
17 %
Tiler
Renderer
Device
Frame Time
43 ms
44 ms
CPU
GPU
Fragment Shading
Performance is limited by fragment shaders
The program with the most costly fragment shader is:
Program #3 "Material"
For best performance, render all opaque geometry before any transparent geometry.
76 glDrawElements(GL_TRIANGLES, 60, GL_UNSIGNED_SHORT,
88 glDrawElements(GL_TRIANGLES, 60, GL_UNSIGNED_SHORT,
The fragment shader in Program #3 "Material" performed dependent texture reads, which are slower than non-dependent
Analyze Performance


Reports
report-Jun 9, 2012 1:08 PM 2
Performance of LightPrepassTour
iPad | iOS 6.0 (10A5316k) | Jun 9, 20...
FPS 23
30
Target
Frame rate of your app is limited by the graphics pipeline.
GPU Utilization
100 %
100 %
17 %
Tiler
Renderer
Device
Frame Time
43 ms
44 ms
CPU
GPU
Fragment Shading
Performance is limited by fragment shaders
The program with the most costly fragment shader is:
Program #3 "Material"
For best performance, render all opaque geometry before any transparent geometry.
76 glDrawElements(GL_TRIANGLES, 60, GL_UNSIGNED_SHORT,
88 glDrawElements(GL_TRIANGLES, 60, GL_UNSIGNED_SHORT,
The fragment shader in Program #3 "Material" performed dependent texture reads, which are slower than non-dependent texture reads
Analyze Performance

# Finds Your Slow Shaders

Reports 〉 report-Jun 9, 2012 1:08 PM 2

**Performance of LightPrepassTour**
iPad | iOS 6.0 (10A5316k) | Jun 9, 20...
FPS 23 30 Target

**Frame rate of your app is limited by the graphics pipeline.**

GPU Utilization: Tiler 17 %, Renderer 100 %, Device 100 %
Frame Time: CPU 43 ms, GPU 44 ms

**Fragment Shading**
Performance is limited by fragment shaders

**The program with the most costly fragment shader is:**
Program #3 "Material"

**For best performance, render all opaque geometry before any transparent geometry.**
76 glDrawElements(GL_TRIANGLES, 60, GL_UNSIGNED_SHORT,
88 glDrawElements(GL_TRIANGLES, 60, GL_UNSIGNED_SHORT,

**The fragment shader in Program #3 "Material" performed dependent texture reads, which are slower than non-dependent texture reads**

Analyze Performance

Automatic 〉 Program #3 "Material" 〉 Fragment Shader

```
1  //*********************************************************************
2  // Edits to captured shaders are not saved to the source shader file.
3  // Copy them back to your shader files when done or retrieve them from
4  // the GPU Debug Log.
5  //*********************************************************************
6  //
7  //  Material.fsh
8  //  Light Prepass
9  //
10 //  Copyright 2011-2012 Apple, Inc. All rights reserved.
11 //
12
13
14 precision highp float;
15
16 #define USE_HARDWARE_PCF 1
17
18 #if USE_HARDWARE_PCF
19 #extension GL_EXT_shadow_samplers : require
20 uniform sampler2DShadow shadow_texture;
21 #else
22 uniform sampler2D shadow_texture;
23 #endif
24
25 uniform sampler2D light_texture;
26 uniform sampler2D gbuffer_texture;
27 uniform sampler2D diffuse_texture;
28 uniform sampler2D specular_texture;
29 uniform vec3 sun_direction;
30 uniform vec3 sun_color;
31
32 varying highp vec4 v_lightcoord;
33 varying highp vec4 v_shadowcoord;
34 varying highp vec4 v_texcoord;
35
36 float GetShadowValue(highp vec4 shadowCoord)
37 {
38 #if USE_HARDWARE_PCF
39     return shadow2DProjEXT(shadow_texture, v_shadowcoord);
```

# *Demo*

**Seth Sowerby**
GPU Software Developer Technologies

# *Demo*

**Michael Mayers**
GPU Software Developer Technologies

# Workflow Recommendations

# Workflow Recommendations

- The Issue Navigator will find GL issues for you
  - Fix them early like you would compiler errors and warnings
  - Remember that GL errors can leave GL in an undefined state
- Annotate your frame with `glPushGroupMarkerEXT / glPopGroupMarkerEXT`
- Label your resources with `glLabelObjectEXT`

# Workflow Recommendations

- Shader edit and continue let's you tweak your shaders in place
  - Great for making performance vs. quality trade-offs
- Use integrated performance analysis to find bottlenecks
  - Don't just assume you know why you're slow
  - Think about performance early—don't leave it till it's too late

# More Information

**Allan Schaffer**
Graphics and Game Technologies Evangelist
aschaffer@apple.com

**Michael Jurewitz**
Developer Tools and Performance Evangelist
jury@apple.com

**Apple Developer Forums**
http://devforums.apple.com

# Labs

| | |
|---|---|
| OpenGL ES Lab | Graphics, Media & Games Lab A<br>Thursday 9:00AM |

# Summary

- Great OpenGL ES developers tools
- Even better in Xcode 4.5
- Harness them to make great games

WWDC2012

The last 3 slides after the logo are intentionally left blank for all presentations.

The last 3 slides after the logo are intentionally left blank for all presentations.

The last 3 slides after the logo are intentionally left blank for all presentations.