CGDC

中国游戏开发者大会

CHINA GAME DEVELOPERS CONFERENCE

Developing and Porting PC Level Graphics Games for Android TV

Rev Lebaredian, NVIDIA Senior Director of Engineer

This Talk

- What is necessary to port a AAA PC/Console title to SHIELD?

- Functionality
- Performance

- We want SHIELD/Android to be your easiest porting platform!

Borderlands 2 and Pre-Sequel

Images from borderlands2.com and borderlandsthegame.com

Borderlands 2 and Pre-Sequel

- Developed by Gearbox, 2K Games, 2K Australia

- Borderlands 2 (BL2), released September 2012
- Borderlands: The Pre-Sequel! (TPS), released October 2014

- Based on modified Unreal Engine 3 (June 2011 build)
- Originally released on Windows, X360, PS3, Linux, Mac

- PC renderer based on D3D9
  - OpenGL support added later for Linux port (TPS had it at launch)

NVIDIA SHIELD

Unreal Engine 3 and Android

- Existing UE3 support for Android…
  - Base engine, kind of
  - Libraries
  - ES2 mobile renderer
  - Some sample apps
- What we can re-use…
- What we want to add/replace…

Libraries

- The newer your engine build, the better
- For us (June 2011)
  - Google protobuf
  - AkAudio/Wwise
  - PhysX
  - Bink
  - FaceFX
  - Scaleform!!!
- Integration varies in difficulty

Renderer

- Our building blocks
  - D3D9 Renderer
  - OpenGL 3.2 PC Renderer
  - OpenGL ES2 Renderer

- Addition: OpenGL ES3.1 backend

	PC	SHIELD	Other Android
D3D9	YES	NO	NO
GL 3.2	YES	YES	NO
ES 2.0	YES*	YES	YES
ES 3.1	YES*	YES	YES

PC OpenGL Functionality

- Already have OpenGL support? Good!

- Otherwise…

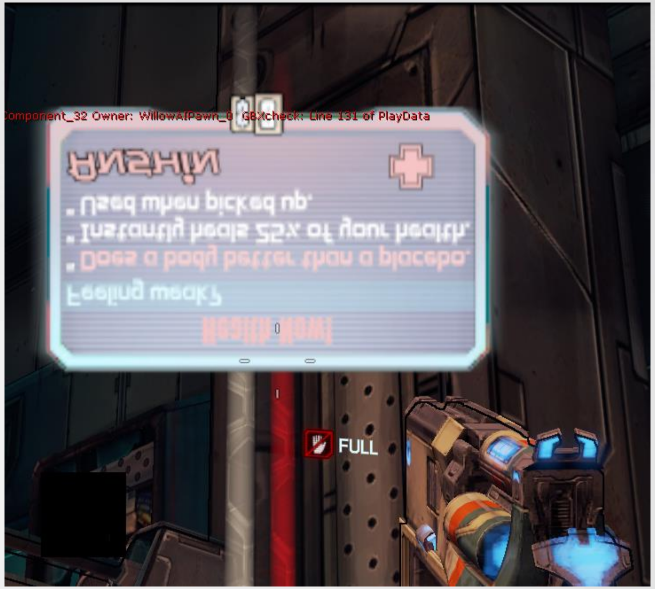
OpenGL Correctness

- Inversions! Artifacts!
- Or worst of all…nothing…

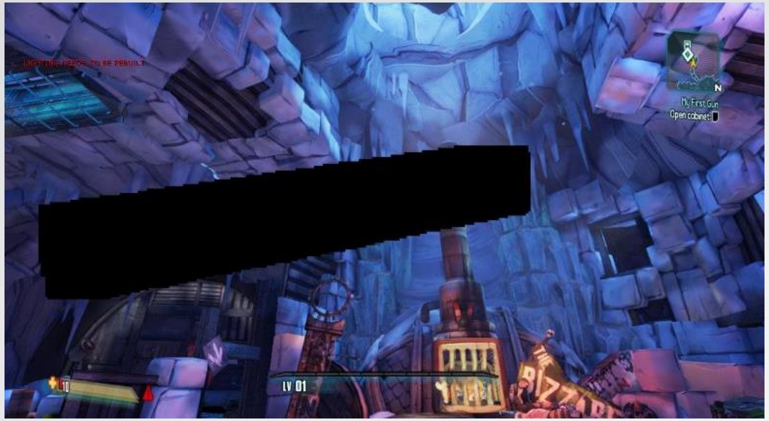
- Stock UE3 effects
- Custom effects

- Common fixes
  - Texture space inversions (t = 1.0 – t)
    - Static versus dynamically generated versus render buffers
  - Clip space Z difference (z = 2z – 1.0)
  - NaN/inf behavior (abs/epsilon)

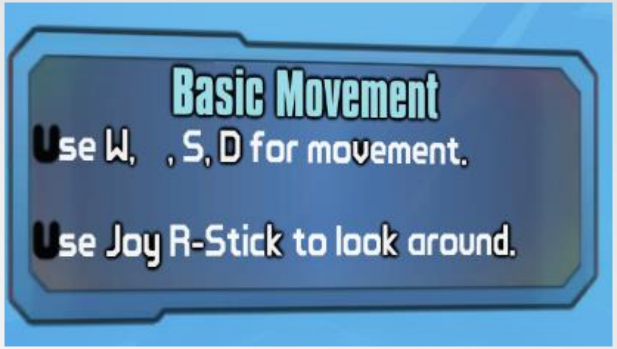
- NSight OpenGL will help you out here


omponent_32 Owner: WillowAIPawn_0 GBXcheck: Line 131 of PlayData
FULL

More Bug Examples



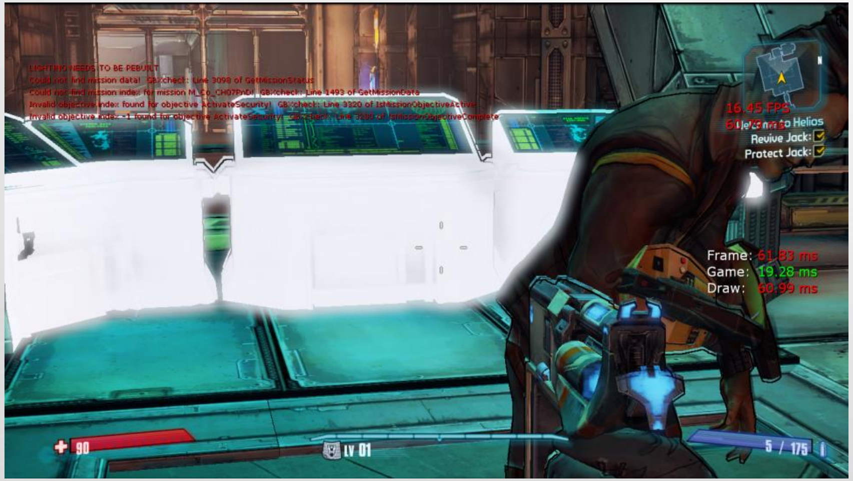
LIGHTING NEEDS TO BE REBUILT



Basic Movement

Use W, , S, D for movement.

Use Joy R-Stick to look around.

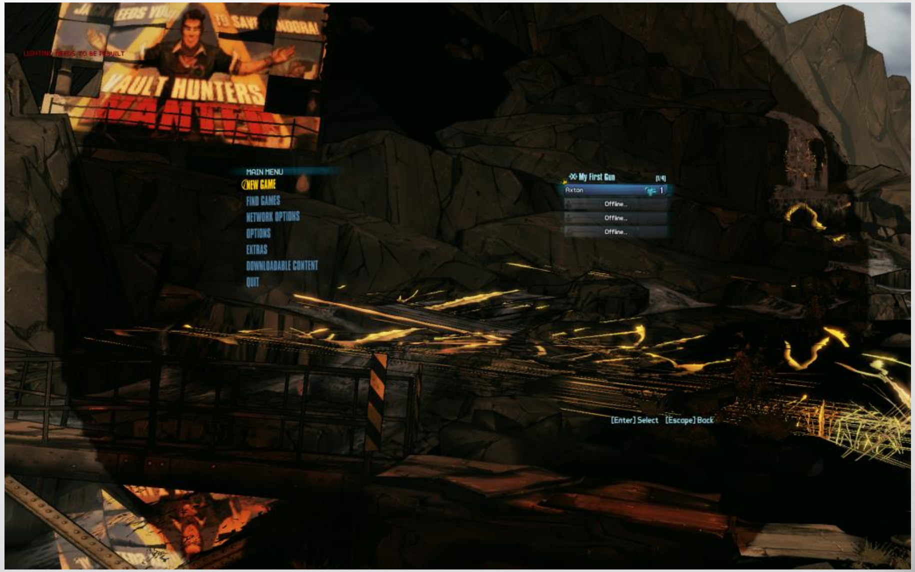


```
LIGHTING NEEDS TO BE REBUILT
Could not find mission data!  GBXcheck: Line 3098 of GetMissionStatus
Could not find mission index for mission M_Co_CH07PnD!  GBXcheck: Line 1493 of GetMissionData
Invalid objective index found for objective ActivateSecurity!  GBXcheck: Line 3320 of IsMissionObjectiveActive
Invalid objective index: -1 found for objective ActivateSecurity!  GBXcheck: Line 3280 of IsMissionObjectiveComplete
```

```
16.45 FPS
60.79 ms
```

Welcome to Helios
Revive Jack: ☑
Protect Jack: ☑

```
Frame:  61.83 ms
Game:   19.28 ms
Draw:   60.99 ms
```



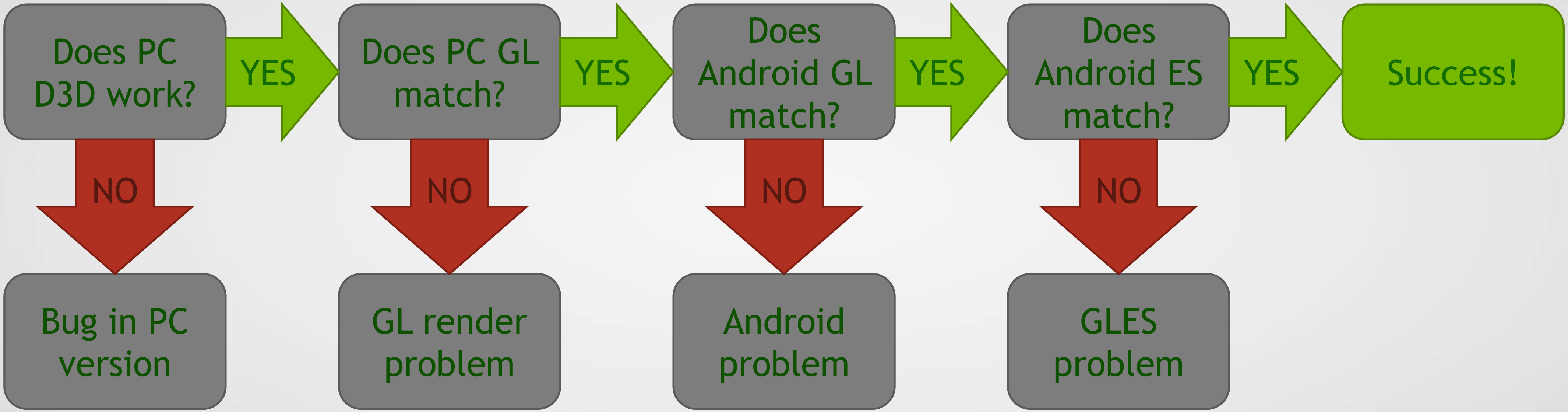
LIGHTING NEEDS TO BE REBUILT

Android OpenGL Functionality

- UE3’s existing Android support is limited
  - Special case ES2 mobile renderer

- Our solution: convert everything over to desktop GL codepath
  - New RHI GL backend using EGL that supports both GL 3.2 and ES 3.1

Debugging Android



Does PC D3D work?

YES

Does PC GL match?

YES

Does Android GL match?

YES

Does Android ES match?

YES

Success!

NO

Bug in PC version

NO

GL render problem

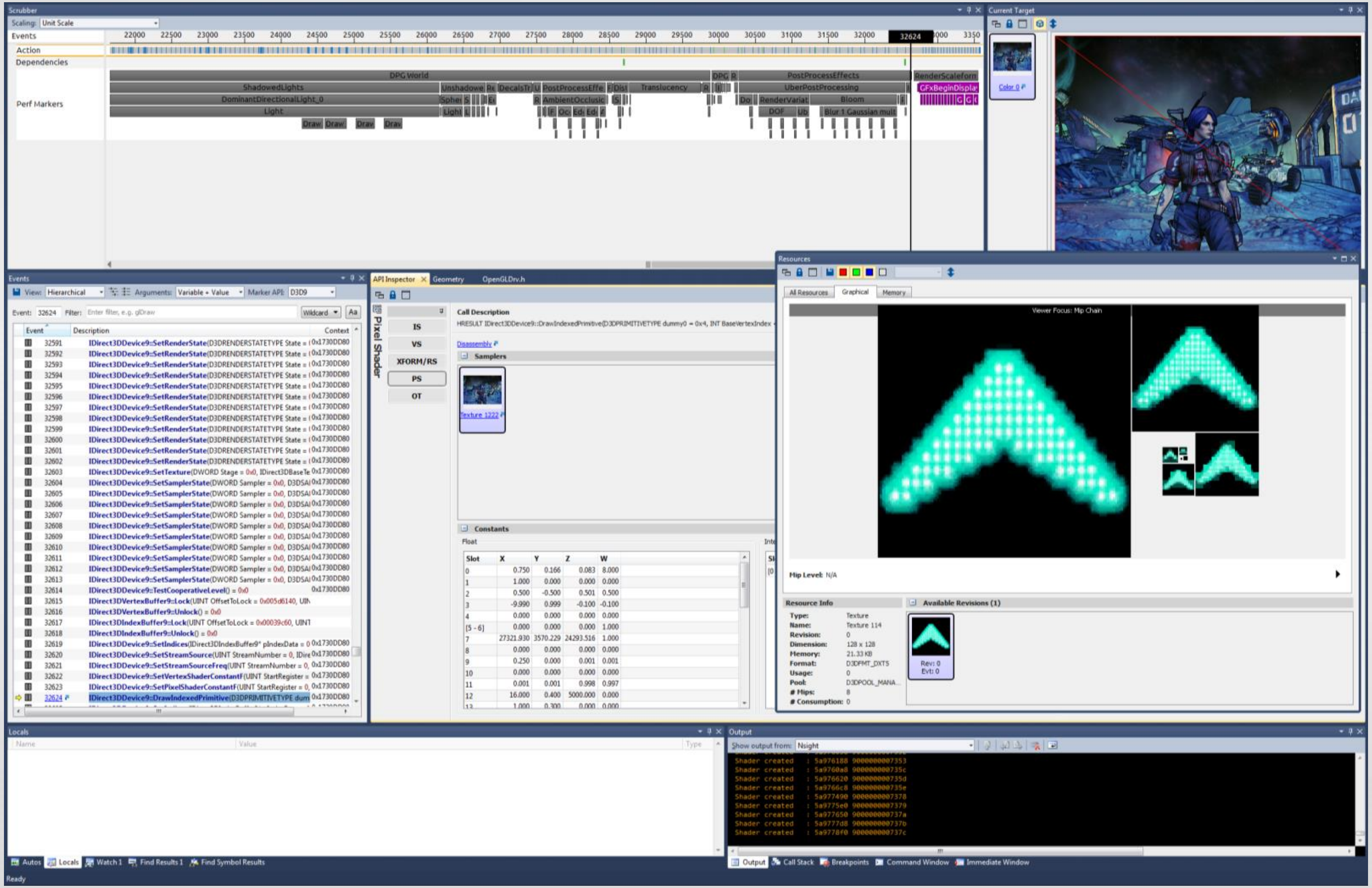
NO

Android problem

NO

GLES problem

Debug View (D3D)



Events

View: Hierarchical Arguments: Variable + Value Marker API: D3D9

Event: 32624 Filter: Enter filter, e.g. glDraw Wildcard Aa

Event	Description	Context
32591	IDirect3DDevice9::SetRenderState(D3DRENDERSTATETYPE State = (	0x1730DD80
32592	IDirect3DDevice9::SetRenderState(D3DRENDERSTATETYPE State = (	0x1730DD80
32593	IDirect3DDevice9::SetRenderState(D3DRENDERSTATETYPE State = (	0x1730DD80
32594	IDirect3DDevice9::SetRenderState(D3DRENDERSTATETYPE State = (	0x1730DD80
32595	IDirect3DDevice9::SetRenderState(D3DRENDERSTATETYPE State = (	0x1730DD80
32596	IDirect3DDevice9::SetRenderState(D3DRENDERSTATETYPE State = (	0x1730DD80
32597	IDirect3DDevice9::SetRenderState(D3DRENDERSTATETYPE State = (	0x1730DD80
32598	IDirect3DDevice9::SetRenderState(D3DRENDERSTATETYPE State = (	0x1730DD80
32599	IDirect3DDevice9::SetRenderState(D3DRENDERSTATETYPE State = (	0x1730DD80
32600	IDirect3DDevice9::SetRenderState(D3DRENDERSTATETYPE State = (	0x1730DD80
32601	IDirect3DDevice9::SetRenderState(D3DRENDERSTATETYPE State = (	0x1730DD80
32602	IDirect3DDevice9::SetRenderState(D3DRENDERSTATETYPE State = (	0x1730DD80
32603	IDirect3DDevice9::SetTexture(DWORD Stage = 0x0, IDirect3DBaseTe	0x1730DD80
32604	IDirect3DDevice9::SetSamplerState(DWORD Sampler = 0x0, D3DSAI	0x1730DD80
32605	IDirect3DDevice9::SetSamplerState(DWORD Sampler = 0x0, D3DSAI	0x1730DD80
32606	IDirect3DDevice9::SetSamplerState(DWORD Sampler = 0x0, D3DSAI	0x1730DD80
32607	IDirect3DDevice9::SetSamplerState(DWORD Sampler = 0x0, D3DSAI	0x1730DD80
32608	IDirect3DDevice9::SetSamplerState(DWORD Sampler = 0x0, D3DSAI	0x1730DD80
32609	IDirect3DDevice9::SetSamplerState(DWORD Sampler = 0x0, D3DSAI	0x1730DD80
32610	IDirect3DDevice9::SetSamplerState(DWORD Sampler = 0x0, D3DSAI	0x1730DD80
32611	IDirect3DDevice9::SetSamplerState(DWORD Sampler = 0x0, D3DSAI	0x1730DD80
32612	IDirect3DDevice9::SetSamplerState(DWORD Sampler = 0x0, D3DSAI	0x1730DD80
32613	IDirect3DDevice9::SetSamplerState(DWORD Sampler = 0x0, D3DSAI	0x1730DD80
32614	IDirect3DDevice9::TestCooperativeLevel() = 0x0	0x1730DD80
32615	IDirect3DVertexBuffer9::Lock(UINT OffsetToLock = 0x005d6140, UIN	
32616	IDirect3DVertexBuffer9::Unlock() = 0x0	
32617	IDirect3DIndexBuffer9::Lock(UINT OffsetToLock = 0x00039c60, UINT	
32618	IDirect3DIndexBuffer9::Unlock() = 0x0	
32619	IDirect3DDevice9::SetIndices(IDirect3DIndexBuffer9* pIndexData = 0	0x1730DD80
32620	IDirect3DDevice9::SetStreamSource(UINT StreamNumber = 0, IDire	0x1730DD80
32621	IDirect3DDevice9::SetStreamSourceFreq(UINT StreamNumber = 0,	0x1730DD80
32622	IDirect3DDevice9::SetVertexShaderConstantF(UINT StartRegister =	0x1730DD80
32623	IDirect3DDevice9::SetPixelShaderConstantF(UINT StartRegister = 0,	0x1730DD80
32624	IDirect3DDevice9::DrawIndexedPrimitive(D3DPRIMITIVETYPE dum	0x1730DD80

Call Description

HRESULT IDirect3DDevice9::DrawIndexedPrimitive(D3DPRIMITIVETYPE dummy0 = 0x4, INT BaseVertexIndex

Disassembly

Samplers

Constants

Float

Slot	X	Y	Z	W
0	0.750	0.166	0.083	8.000
1	1.000	0.000	0.000	0.000
2	0.500	-0.500	0.501	0.500
3	-9.990	0.999	-0.100	-0.100
4	0.000	0.000	0.000	0.000
[5 - 6]	0.000	0.000	0.000	1.000
7	27321.930	3570.229	24293.516	1.000
8	0.000	0.000	0.000	0.000
9	0.250	0.000	0.001	0.001
10	0.000	0.000	0.000	0.000
11	0.001	0.001	0.998	0.997
12	16.000	0.400	5000.000	0.000
13	1.000	0.300	0.000	0.000

Mip Level: N/A

Resource Info

Type:	Texture
Name:	Texture 114
Revision:	0
Dimension:	128 x 128
Memory:	21.33 KB
Format:	D3DFMT_DXT5
Usage:	0
Pool:	D3DPOOL_MANA...
# Mips:	8
# Consumption:	0

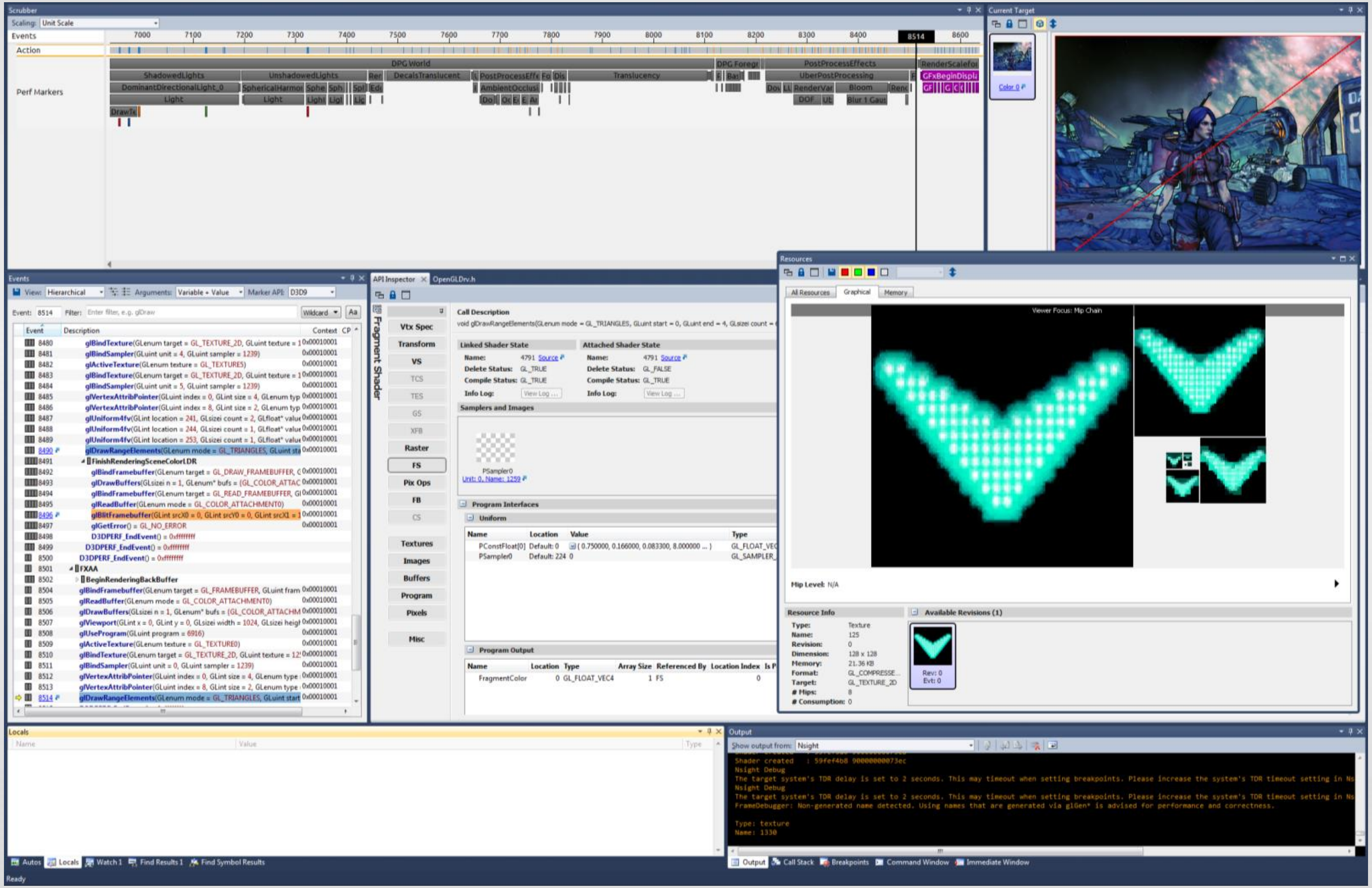
Available Revisions (1)

Rev: 0
Evt: 0

Show output from: Nsight

```
Shader created    : 5a976188 9000000007353
Shader created    : 5a9760a8 900000000735c
Shader created    : 5a976620 900000000735d
Shader created    : 5a9766c8 900000000735e
Shader created    : 5a977490 9000000007378
Shader created    : 5a9775e0 9000000007379
Shader created    : 5a977650 900000000737a
Shader created    : 5a9777d8 900000000737b
Shader created    : 5a9778f0 900000000737c
```

Debug View (PC GL)



Events

View: Hierarchical Arguments: Variable + Value Marker API: D3D9

Event: 8514 Filter: Enter filter, e.g. glDraw Wildcard Aa

```
Event  Description                                                                  Context
8480   glBindTexture(GLenum target = GL_TEXTURE_2D, GLuint texture = 1              0x00010001
8481   glBindSampler(GLuint unit = 4, GLuint sampler = 1239)                        0x00010001
8482   glActiveTexture(GLenum texture = GL_TEXTURE5)                                0x00010001
8483   glBindTexture(GLenum target = GL_TEXTURE_2D, GLuint texture = 1              0x00010001
8484   glBindSampler(GLuint unit = 5, GLuint sampler = 1239)                        0x00010001
8485   glVertexAttribPointer(GLuint index = 0, GLint size = 4, GLenum typ           0x00010001
8486   glVertexAttribPointer(GLuint index = 8, GLint size = 2, GLenum typ           0x00010001
8487   glUniform4fv(GLint location = 241, GLsizei count = 2, GLfloat* valu          0x00010001
8488   glUniform4fv(GLint location = 244, GLsizei count = 1, GLfloat* valu          0x00010001
8489   glUniform4fv(GLint location = 253, GLsizei count = 1, GLfloat* valu          0x00010001
8490   glDrawRangeElements(GLenum mode = GL_TRIANGLES, GLuint sta                   0x00010001
8491   FinishRenderingSceneColorLDR
8492     glBindFramebuffer(GLenum target = GL_DRAW_FRAMEBUFFER, G                   0x00010001
8493     glDrawBuffers(GLsizei n = 1, GLenum* bufs = {GL_COLOR_ATTAC                0x00010001
8494     glBindFramebuffer(GLenum target = GL_READ_FRAMEBUFFER, Gl                  0x00010001
8495     glReadBuffer(GLenum mode = GL_COLOR_ATTACHMENT0)                           0x00010001
8496     glBlitFramebuffer(GLint srcX0 = 0, GLint srcY0 = 0, GLint srcX1 = 1        0x00010001
8497     glGetError() = GL_NO_ERROR                                                 0x00010001
8498     D3DPERF_EndEvent() = 0xffffffff
8499   D3DPERF_EndEvent() = 0xffffffff
8500   D3DPERF_EndEvent() = 0xffffffff
8501   FXAA
8502     BeginRenderingBackBuffer
8504   glBindFramebuffer(GLenum target = GL_FRAMEBUFFER, GLuint fram                0x00010001
8505   glReadBuffer(GLenum mode = GL_COLOR_ATTACHMENT0)                             0x00010001
8506   glDrawBuffers(GLsizei n = 1, GLenum* bufs = {GL_COLOR_ATTACHM                0x00010001
8507   glViewport(GLint x = 0, GLint y = 0, GLsizei width = 1024, GLsizei heigl     0x00010001
8508   glUseProgram(GLuint program = 6916)                                          0x00010001
8509   glActiveTexture(GLenum texture = GL_TEXTURE0)                                0x00010001
8510   glBindTexture(GLenum target = GL_TEXTURE_2D, GLuint texture = 12             0x00010001
8511   glBindSampler(GLuint unit = 0, GLuint sampler = 1239)                        0x00010001
8512   glVertexAttribPointer(GLuint index = 0, GLint size = 4, GLenum type          0x00010001
8513   glVertexAttribPointer(GLuint index = 8, GLint size = 2, GLenum type          0x00010001
8514   glDrawRangeElements(GLenum mode = GL_TRIANGLES, GLuint start                 0x00010001
```

Call Description

void glDrawRangeElements(GLenum mode = GL_TRIANGLES, GLuint start = 0, GLuint end = 4, GLsizei count =

Linked Shader State

Name: 4791 Source
Delete Status: GL_TRUE
Compile Status: GL_TRUE
Info Log: View Log ...

Attached Shader State

Name: 4791 Source
Delete Status: GL_FALSE
Compile Status: GL_TRUE
Info Log: View Log ...

Samplers and Images

PSampler0
Unit: 0, Name: 1259

Program Interfaces

Uniform

Name	Location	Value	Type
PConstFloat[0]	Default: 0	(0.750000, 0.166000, 0.083300, 8.000000 ...)	GL_FLOAT_VEC
PSampler0	Default: 224	0	GL_SAMPLER_

Program Output

Name	Location	Type	Array Size	Referenced By	Location Index	Is P
FragmentColor	0	GL_FLOAT_VEC4	1	FS	0	

Mip Level: N/A

Resource Info

Type: Texture
Name: 125
Revision: 0
Dimension: 128 x 128
Memory: 21.36 KB
Format: GL_COMPRESSE...
Target: GL_TEXTURE_2D
Mips: 8
Consumption: 0

Available Revisions (1)

Rev: 0
Evt: 0

Locals

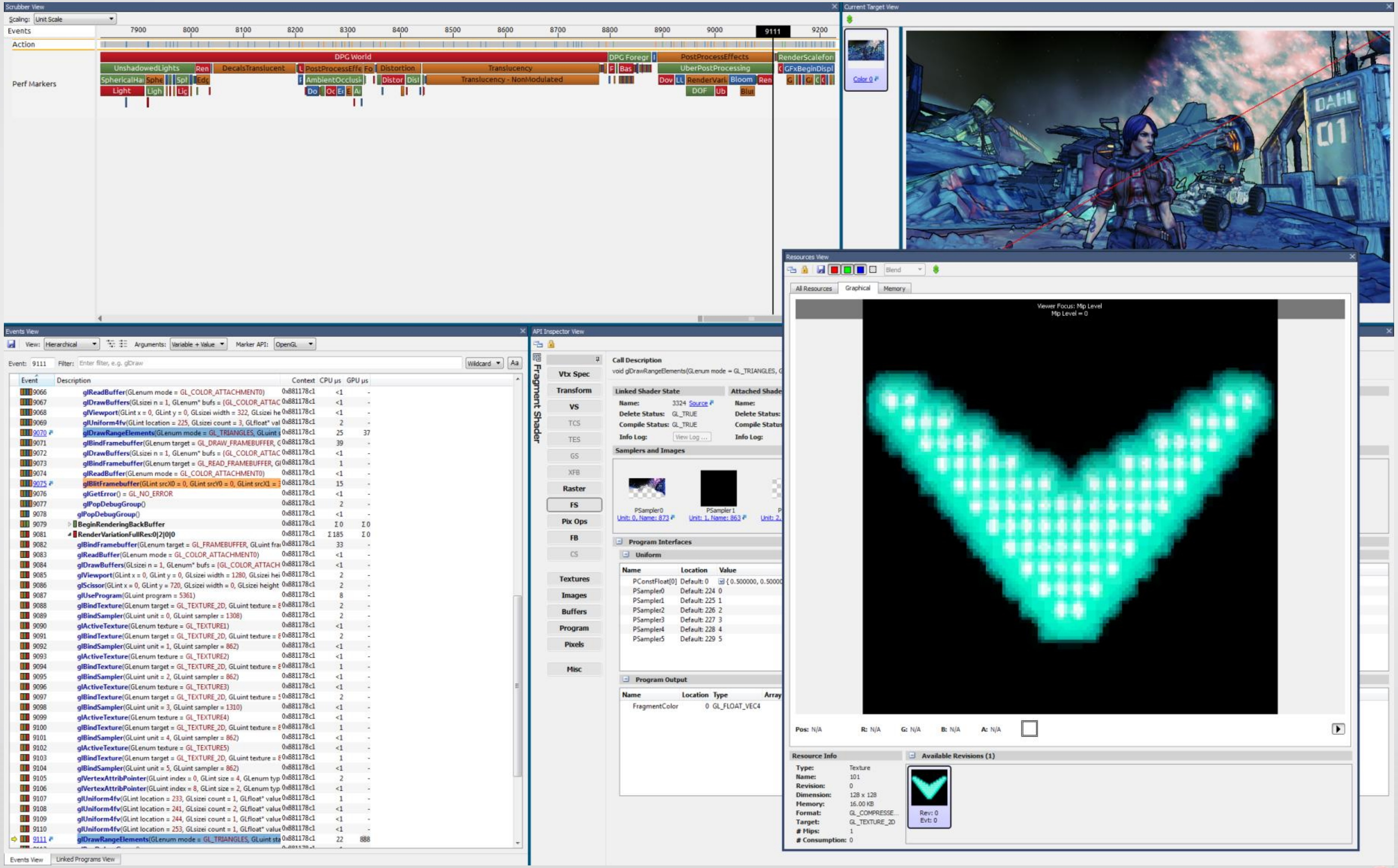
Name	Value	Type

Show output from: Nsight

```
Shader created    : 59fef4b8 90000000073ec
Nsight Debug
The target system's TDR delay is set to 2 seconds. This may timeout when setting breakpoints. Please increase the system's TDR timeout setting in Ns
Nsight Debug
The target system's TDR delay is set to 2 seconds. This may timeout when setting breakpoints. Please increase the system's TDR timeout setting in Ns
FrameDebugger: Non-generated name detected. Using names that are generated via glGen* is advised for performance and correctness.

Type: texture
Name: 1330
```

Debug View (SHIELD)



Events View

View: Hierarchical | Arguments: Variable + Value | Marker API: OpenGL

Event: 9111 Filter: Enter filter, e.g. glDraw

Event	Description	Context	CPU µs	GPU µs
9066	glReadBuffer(GLenum mode = GL_COLOR_ATTACHMENT0)	0x881178c1	<1	-
9067	glDrawBuffers(GLsizei n = 1, GLenum* bufs = {GL_COLOR_ATTAC	0x881178c1	<1	-
9068	glViewport(GLint x = 0, GLint y = 0, GLsizei width = 322, GLsizei he	0x881178c1	<1	-
9069	glUniform4fv(GLint location = 225, GLsizei count = 3, GLfloat* val	0x881178c1	2	-
9070	glDrawRangeElements(GLenum mode = GL_TRIANGLES, GLuint s	0x881178c1	25	37
9071	glBindFramebuffer(GLenum target = GL_DRAW_FRAMEBUFFER, G	0x881178c1	39	-
9072	glDrawBuffers(GLsizei n = 1, GLenum* bufs = {GL_COLOR_ATTAC	0x881178c1	<1	-
9073	glBindFramebuffer(GLenum target = GL_READ_FRAMEBUFFER, GI	0x881178c1	1	-
9074	glReadBuffer(GLenum mode = GL_COLOR_ATTACHMENT0)	0x881178c1	<1	-
9075	glBlitFramebuffer(GLint srcX0 = 0, GLint srcY0 = 0, GLint srcX1 = 3	0x881178c1	15	-
9076	glGetError() = GL_NO_ERROR	0x881178c1	<1	-
9077	glPopDebugGroup()	0x881178c1	2	-
9078	glPopDebugGroup()	0x881178c1	<1	-
9079	BeginRenderingBackBuffer	0x881178c1	Σ 0	Σ 0
9081	RenderVariationFullRes:0|2|0|0	0x881178c1	Σ 185	Σ 0
9082	glBindFramebuffer(GLenum target = GL_FRAMEBUFFER, GLuint fra	0x881178c1	33	-
9083	glReadBuffer(GLenum mode = GL_COLOR_ATTACHMENT0)	0x881178c1	<1	-
9084	glDrawBuffers(GLsizei n = 1, GLenum* bufs = {GL_COLOR_ATTACH	0x881178c1	<1	-
9085	glViewport(GLint x = 0, GLint y = 0, GLsizei width = 1280, GLsizei hei	0x881178c1	2	-
9086	glScissor(GLint x = 0, GLint y = 720, GLsizei width = 0, GLsizei height	0x881178c1	2	-
9087	glUseProgram(GLuint program = 5361)	0x881178c1	8	-
9088	glBindTexture(GLenum target = GL_TEXTURE_2D, GLuint texture = 8	0x881178c1	2	-
9089	glBindSampler(GLuint unit = 0, GLuint sampler = 1308)	0x881178c1	2	-
9090	glActiveTexture(GLenum texture = GL_TEXTURE1)	0x881178c1	<1	-
9091	glBindTexture(GLenum target = GL_TEXTURE_2D, GLuint texture = 8	0x881178c1	2	-
9092	glBindSampler(GLuint unit = 1, GLuint sampler = 862)	0x881178c1	<1	-
9093	glActiveTexture(GLenum texture = GL_TEXTURE2)	0x881178c1	<1	-
9094	glBindTexture(GLenum target = GL_TEXTURE_2D, GLuint texture = 8	0x881178c1	1	-
9095	glBindSampler(GLuint unit = 2, GLuint sampler = 862)	0x881178c1	<1	-
9096	glActiveTexture(GLenum texture = GL_TEXTURE3)	0x881178c1	<1	-
9097	glBindTexture(GLenum target = GL_TEXTURE_2D, GLuint texture = 5	0x881178c1	2	-
9098	glBindSampler(GLuint unit = 3, GLuint sampler = 1310)	0x881178c1	<1	-
9099	glActiveTexture(GLenum texture = GL_TEXTURE4)	0x881178c1	<1	-
9100	glBindTexture(GLenum target = GL_TEXTURE_2D, GLuint texture = 8	0x881178c1	1	-
9101	glBindSampler(GLuint unit = 4, GLuint sampler = 862)	0x881178c1	<1	-
9102	glActiveTexture(GLenum texture = GL_TEXTURE5)	0x881178c1	<1	-
9103	glBindTexture(GLenum target = GL_TEXTURE_2D, GLuint texture = 8	0x881178c1	1	-
9104	glBindSampler(GLuint unit = 5, GLuint sampler = 862)	0x881178c1	<1	-
9105	glVertexAttribPointer(GLuint index = 0, GLint size = 4, GLenum typ	0x881178c1	2	-
9106	glVertexAttribPointer(GLuint index = 8, GLint size = 2, GLenum typ	0x881178c1	<1	-
9107	glUniform4fv(GLint location = 233, GLsizei count = 1, GLfloat* value	0x881178c1	1	-
9108	glUniform4fv(GLint location = 241, GLsizei count = 2, GLfloat* value	0x881178c1	<1	-
9109	glUniform4fv(GLint location = 244, GLsizei count = 1, GLfloat* value	0x881178c1	<1	-
9110	glUniform4fv(GLint location = 253, GLsizei count = 1, GLfloat* value	0x881178c1	<1	-
9111	glDrawRangeElements(GLenum mode = GL_TRIANGLES, GLuint sta	0x881178c1	22	888

API Inspector View

Call Description

void glDrawRangeElements(GLenum mode = GL_TRIANGLES, G

Linked Shader State

Name: 3324 Source

Delete Status: GL_TRUE

Compile Status: GL_TRUE

Info Log: View Log ...

Attached Shade

Name:

Delete Status:

Compile Status

Info Log:

Samplers and Images

PSampler0 Unit: 0, Name: 873

PSampler1 Unit: 1, Name: 863

Unit: 2,

Program Interfaces

Uniform

Name	Location	Value
PConstFloat[0]	Default: 0	{ 0.500000, 0.50000
PSampler0	Default: 224	0
PSampler1	Default: 225	1
PSampler2	Default: 226	2
PSampler3	Default: 227	3
PSampler4	Default: 228	4
PSampler5	Default: 229	5

Program Output

Name	Location	Type	Array
FragmentColor	0	GL_FLOAT_VEC4	

Pos: N/A R: N/A G: N/A B: N/A A: N/A

Resource Info

Type: Texture
Name: 101
Revision: 0
Dimension: 128 x 128
Memory: 16.00 KB
Format: GL_COMPRESSE...
Target: GL_TEXTURE_2D
Mips: 1
Consumption: 0

Available Revisions (1)

Rev: 0
Evt: 0

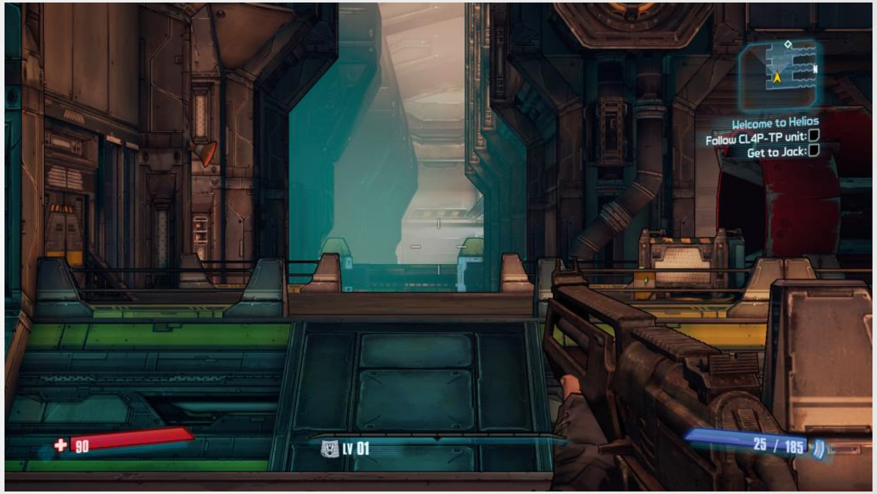
Checkpoint

- At this point, we have a (mostly) functionally complete, playable game
- Framerate may vary
- Next step is to boost performance to make it more enjoyable

Initial Performance Notes

- A mix of CPU and GPU bound cases in the render thread
- More hitching on Android versus PC

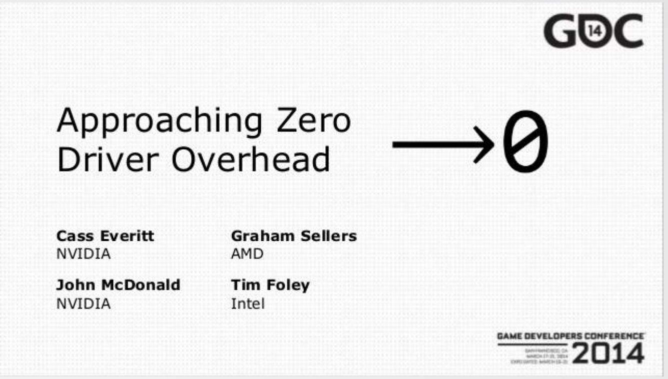
- Example scene: main menu
  - 15-30FPS, GPU bound on TX1 1080p
- Example scene: first level
  - 26-28FPS, CPU/GPU bound on TX1 810p
  - Null GPU runs at 40FPS



- Game thread 10-12ms on TX1 during play

State of the UE3 GL RHI

- OpenGL 3.2, mostly core
- Pros
  - Feature parity with D3D9
  - Ease of use
  - API idiosyncrasies
- Cons
  - Performance and efficiency



GDC 14

Approaching Zero
Driver Overhead ⟶0

Cass Everitt
NVIDIA

Graham Sellers
AMD

John McDonald
NVIDIA

Tim Foley
Intel

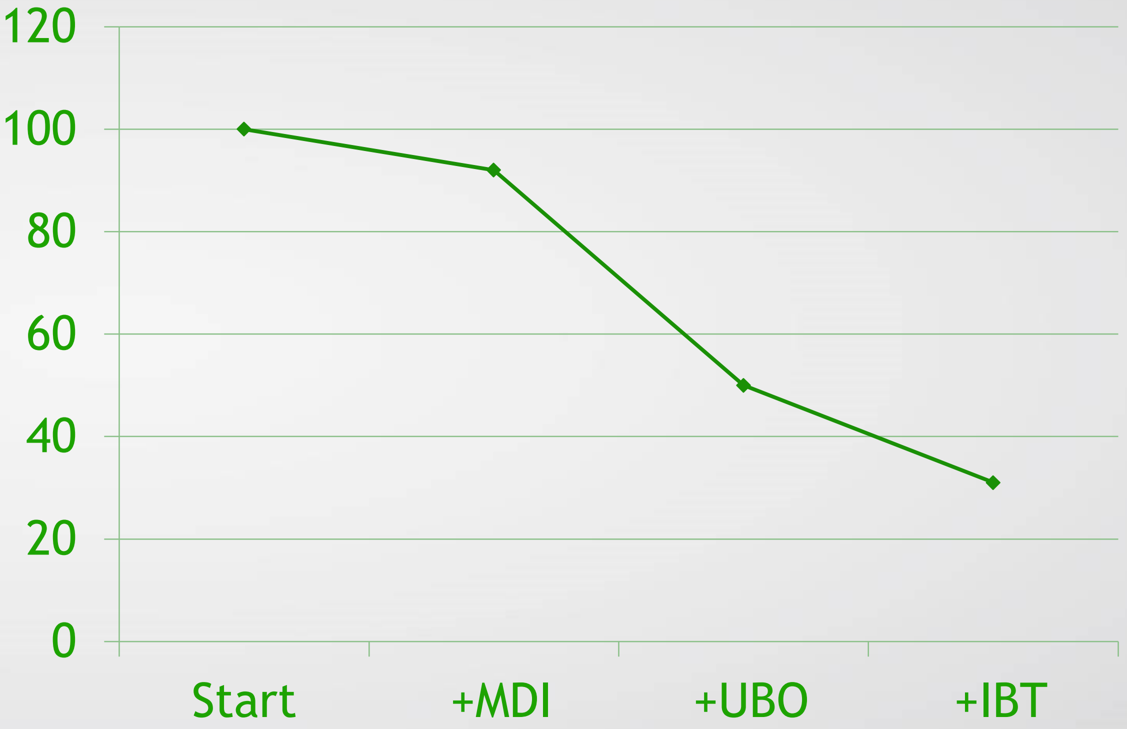
GAME DEVELOPERS CONFERENCE 2014

- There are many known ways to increase GL API efficiency
  - See “Approaching Zero Driver Overhead” from GDC 2014

OpenGL API Optimizations

- First level test scene
  - 10K+ events reported by debugger
  - 1636 draw calls
- MultiDrawIndirect
  - 1494 draw calls (-8% from start)

- Ongoing work…
- One large UBO indexed by vertex
  - 809 draw calls (-50%)
  - Fewer uniform-related binds
- Indexed bindless textures
  - 502 draw calls (-69%)

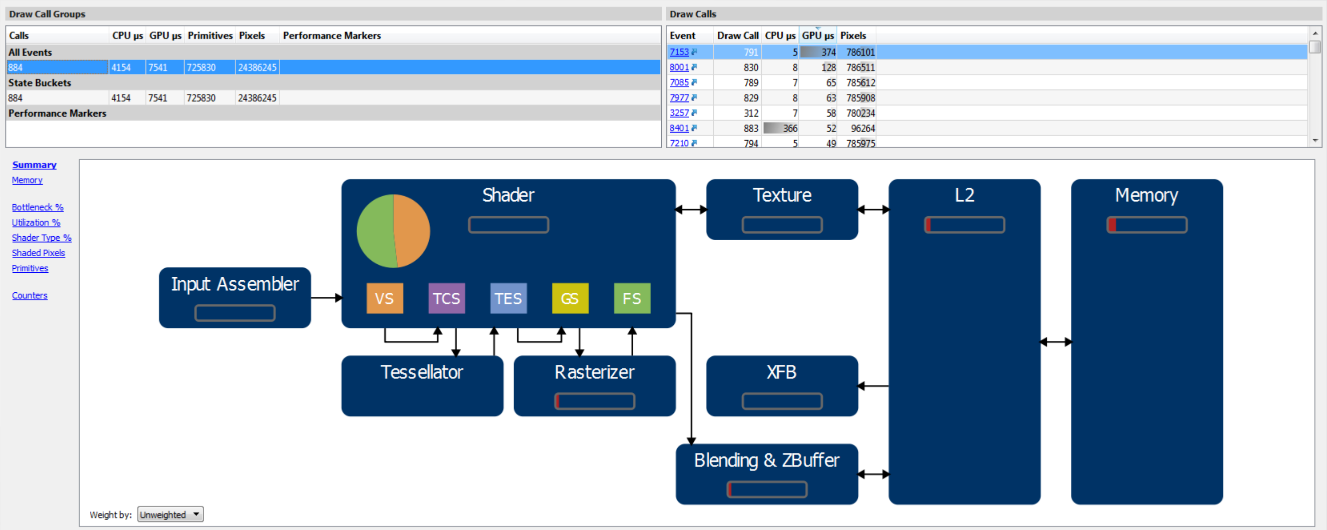


GPU Performance

- Ongoing work on GPU bottlenecks

- Possible techniques
  - Z-prepass
  - Post-process quality
  - Anti-aliasing options
  - Resolutions
  - Shader optimization

- Depends on specific needs



Draw Call Groups

Calls	CPU µs	GPU µs	Primitives	Pixels	Performance Markers
All Events					
884	4154	7541	725830	24386245	
State Buckets					
884	4154	7541	725830	24386245	
Performance Markers					

Draw Calls

Event	Draw Call	CPU µs	GPU µs	Pixels
7153	791	5	374	786101
8001	830	8	128	786511
7085	789	7	65	785612
7977	829	8	63	785908
3257	312	7	58	780234
8401	883	366	52	96264
7210	794	5	49	785975

Weight by: Unweighted

Miscellaneous Notes and Pitfalls

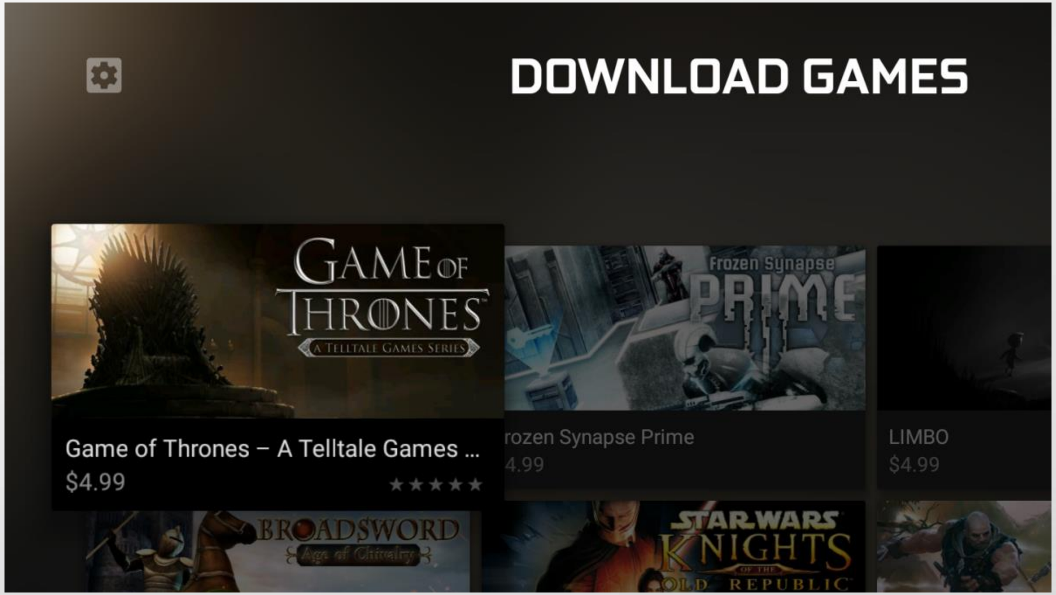
- Debug context
- BGRA support on OpenGL ES
- PF_V8U8 bump maps
- ARM NEON support
- Array cookie size
- Memory alignment

Online Functionality

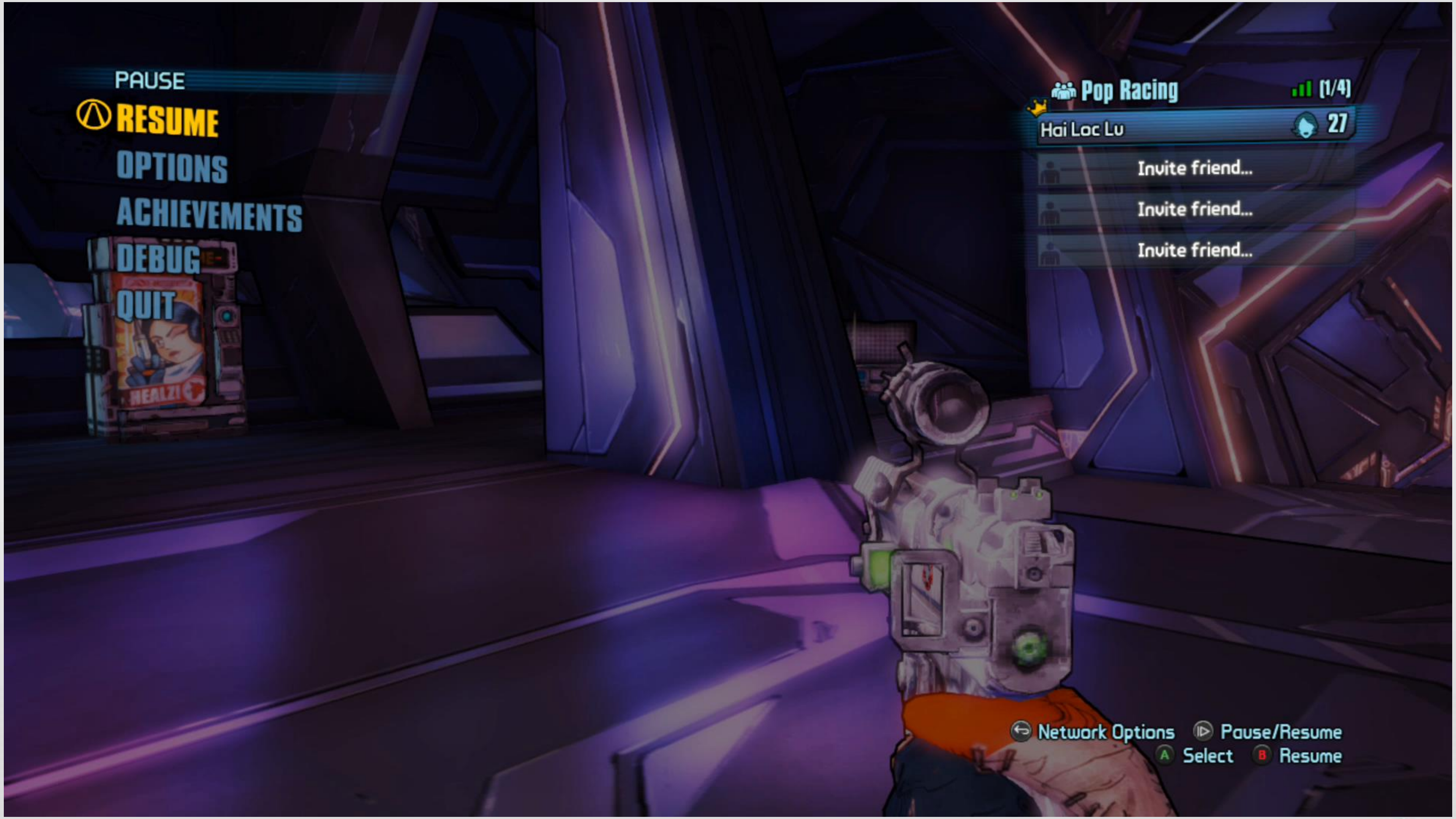
- Adjusted for Android

- Multiplayer
- Cloud saves, achievements
- DLC/add-on content
- Storefront integration

GET IT ON Google play



Borderlands: TPS Footage



End