

RENDERING ROBLOX

YULKAN OPTIMISATIONS ON POWERYR

PRESENTED BY: ARSENY KAPOULKINE DAVID HAMILL

What is Roblox?

- Online multiplayer game creation platform
- All content is user generated
- Windows, macOS, iOS, Android, Xbox One
- 100M+ MAU, 2.5M+ CCU











Graphics scaling

- Substantial differences between devices
 - PowerVR SGX543: Apple iPad 2 (2011), Samsung Galaxy S4 (2013)
 - PowerVR GM 9446: Oppo Reno Z (2019)
- Developers make the content once
- It's our job to run it well*



Full spectrum design

- How do you implement a graphics feature?
 - Build for the lowest common denominator and upscale?
 - Build for the highest end GPU and optimize?
- Build a feature for the lowest common denominator and upscale?
 - Pro: Consistent behavior
 - Con: Poor quality
- Build a feature for the highest end GPU and optimize?
 - Pro: Great quality on high-end
 - Con: Poor performance on low-end, inconsistent behavior, battery/thermal limits



Full spectrum design

- How do you implement a graphics feature?
 - Build for the lowest common denominator and upscale?
 - Build for the highest end GPU and optimize?
- Sometimes we pick one and iterate
- Prefer to design for the entire spectrum from the beginning



Advanced Settings	Unlocked
Graphics API	Vulkar
Near Volumetric Resolution	High
Far Volumetric Resolution	High
Volumetric Lighting Quality	High
Unlocked Volumetric Raymarch Resolu	ntion Of
Particle Lighting Quality	High
Soft Shadows	High
Grass Shadows	Higt
Long Shadows	Or
Full Resolution Screen Space Ambient	Occlusion Of
Water Refraction Quality	High
Water Reflection Quality	Higt
Water Physics Quality	
Resolution Scale	Of



- Advanced settings give users lots of control! However...
- Suboptimal performance for users who can't tune these well
 - ... or know about them! Especially on mobile.
- Performance is often counter-intuitive
 - CPU vs GPU bottlenecks
 - Example: denser grass is faster
- Different games have different performance characteristics



- We have a single quality slider
 - ... arguably we made it too simple and need a separate "draw distance" slider
- By default, we automatically tune it based on observed performance
 - Do we have headroom on CPU vs GPU?
- All graphics effects are using this to balance quality vs performance
 - NB: Internally the slider is more granular than it is in UI
- Less control for the user, but easier for developers and graphics engineers





- Depending on the effect, scaling strategy can be different!
- Disable effect completely
 - Possible if not gameplay critical
 - Example: SSAO
- Geometry / shading LOD
 - Reduce computational cost while reducing quality
 - Important to keep the overall color balance!
- Temporal throttling
 - Visual pops due to late updates
 - Examples: lighting, terrain LOD



Lighting system: goals

- No baking. Every object can move at any point in time.
- Can't disable lighting altogether large visual and gameplay implications
- Want minimal "performance" annotations from content
- Want to run on low-end laptop/mobile (D3D9 / GL2 / GLES2 class)
- Want good looking lighting on high-end laptop/mobile



Lighting system: high level overview

- Features
 - Light sources: sun/moon, sky, local lights (point, spot, surface)
 - Geometry and light sources are dynamic
 - All light sources can cast shadows
- Today
 - Coarse voxel lighting: runs everywhere; efficient, soft lighting
 - Shadow maps: runs on mid-end to high-end; high quality sun shadows
- Future
 - Forward+: high-end, high quality local lights
 - Better voxel lighting: better skylight



Lighting system: phase 0, voxels

- Most of the work is done on CPU
- Voxelize all geometry dynamically
- Compute light impact per voxel
 - Sunlight
 - Skylight
 - Accumulated local light RGB
- Upload the information to GPU
 - RGB: light + skylight * skycolor
 - A: sunlight
- Final lighting in fragment shader



Lighting system: phase 0, voxels

- Managing CPU performance
 - Voxel grid is split into chunks
 - Only a handful of chunks update each frame
 - Fixed quality, but updates can be "stale"
 - Update kernels use hand-optimized SSE2 / NEON
- Managing GPU performance
 - A single 3D texture lookup with bilinear* filtering
 - GLES2: emulate 3D texture lookup with 2 2D texture atlas lookups
- Fully decoupled geometry vs light complexity



Lighting system: phase 1, better voxels

- Keep the overall system design
- HDR light color
 - Encoded using RGBM to save space
- Store skylight separately
 - Better integration of sky into BRDF
- Anisotropic occupancy
 - Voxelizer keeps 3 axial values per voxel
 - Critical for content compatibility
- Results are closer to shadow maps/Forward+





Lighting system: phase 1, performance

- CPU cost is larger but manageable
 - Anisotropic occupancy rasterization is more expensive
 - RGBM encoding is almost free
- GPU cost is still relatively small
 - Fetch two textures
 - RGBM decode (future: RGB9E5 / R11G11B10F?)
- CPU-GPU upload
 - We upload 1-4 chunks each frame
 - Each chunk is 32x32x16 volume (16K voxels) = 128KB per chunk
 - vkCmdCopyBufferToImage copies 128KB-512KB of data from a VkBuffer each frame
 - ... is this fast?







ANALYSING ROBLOX ON POWERYR GPUS

O Imagination

What is Imagination Technologies?

PowerVR mobile GPUs, *Rogue* architecture and new *A-Series* architecture

VideoLogic discrete graphics

Arcade machines,

Sega Dreamcast,

PS Vita

iPhones and iPads

Sony B B PSVITA SSVITA SSVITA SSVITA SSVITA SSVITA SSVITA SSVITA SSVITA

Now, neural network accelerators,

networking chips, and in the near future hardware ray-tracing...







PowerVR Tile-Based Deferred Rendering



O Imagination

PVRTune



Disconnected 7.3280x 204/9288/x 0.1%



PVRCarbon

Re Edit View Heb						
reade	6 X	(Reference)		W	Render	6
D		Million carry and			111208: egiSwapBuffers	
7548		UED Ende Res	ug Name	Angumenta	والمتحد والمحدولة فالتنا التحديقات	
7550		100206 01/	Seligies contening	(United and Descently)	the second s	
7396		103207		(program=0)	PLOT A	A DESCRIPTION OF TAXABLE PARTY.
7705		109208		(Instance of Controller)		
7714		100209	place Texture	(target- the strike strik	and the second division of the second divisio	
		100010	DATE: CONTRACT OF CONTRACT.	And Agarenesis And Agarenesis According to proper-ig. (1990) And Agarenesis According to the second sec		
		109011		(Larget = 2, that the city, even of some to be, without any, heapter of, heapter of, heapter of, heapter of, h		
		109212	pland restare	(carpet==2<2000.carpet=1)	CONTRACTOR DOCUMENTS	
		109213		(toxure-o_transmis)		
		109214	phone Teaming	(target - 3, 10x1040, 20, texture - 30)		
		109215		(prame+4L (meach Addamat, prom+4))	Attraction Attraction Attraction A	
		109216	ter data for ter data			
		100217		(target=40_ffffffffffff, textura=1)		
		109238	gildening Feature -	Image: Angunetal State Successing (ngunetal: (nguneta	A	
		109219		(target=32_111/1000_20) texture=()		
		109220	prestore	(prame+ULUNINCK_NINDORN, param+1)	CONTRACTOR OF STREET,	- and a second
		109221		Target-9, THTTHE IN, level-9, softest-140, voftest-131, exith=12, height-12, format-94,	and the state of t	THE R R. L. S. S. S. S. S.
		109222	philand Sexturns	(target-D(CTUBC), texture=0)	SALE L'L'L'L'L'L'L'L'L	n i na mana na
		109223	different Timmeri	(texture+G_)		
		100224		(target=lik_like(these_lik, texture=14)	and the second se	And the second second
	A X	109225		(pname-GL_UNINGC_1_UNIVERT, parmet)	S R S G S A	1920 × 109
		109226	pitesSabbrage20	(target=92_10X1040_20, level=1, aufiset=100, vofiset=044, with=10, hep/t=32, formit=60,	Recruited Formehutter	F
low cb-170220.pvrcbn'		109227	Differed Transaction	(target=0,_TEXTURE_10, texture=1)	111208: edimoladers	
inacted to renderer		100228	pikanya Luman	(texture-(increase))		
ceaning has interied to entris, o warrings	M.	109229		(target=30_1000.00, testare=1)	and the second se	
		109230		(prome=GL_UNITACE_ALIBRATIET, paramet)	Chart Weight	
		100231		(target=0), is lies in, leaded, adfast=00, yofset=00, wath=12, height=32, format=0,		
		109232	pillent Texture	(target-0. Thilling_10, texture-1)		
		109223		(texture-stagmentatio)		A DECEMBER OF THE OWNER
		109234	plant Texture	(target=Re_SECTION_III, texture=(0)		
		109235		(prame=UUUDACK_U_UUUUUU), param=1)		
		109236	pressibilitinge20	(target=5,_10x1040/_20, level=0, xoffiet=300, yoffiet=304, with=32, heght=32, formit=60.		
		109237		(target=0_1ECTUR_20, texture=()		
		Static Analysis		0 >		
548 550 550 706 706 714 set 37220_entbri set 40 a nodesi keding his finished (0 entro, 0 warrangs)		Fittor:				4
		Level Frame UD		Description	A CONTRACTOR OF	
		A Warring1 0 328	gDrawRangeElements	No appropriate hamebuffer cleans have been used between gBindFramebuffer and the first gDraw" command. The		and the second second
		A Warringt 0 372	gDrawArrays pDrawBangaElements	No appropriate framebuffer clean have been used between gBindFramebuffer and the first gDraw* command. This is No appropriate framebuffer clean have been used between gBindFramebuffer and the fast official#" command. This		
					HISTORY CONTRACTORY OF THE REAL PROPERTY OF	
		Static Analysis Current Cal	Find Results		R G B B A 125, 662 px 0.065, 0.	55 RGBA(64, 61, 64, 255) 1920 x 1080

Renderer: Host

Current Frame: 60/100



PVRTune timeline



Vulkan – use double/triple buffering please



Analysing Roblox on PowerVR





PVRCarbon analysis

🏟 D/Discurrents/convert/initias/01	M_api/prication/ile	icidil7, can, soft	Lovertes - PVH	Carbon set.5						- 0 X
File Edit View Help										
Threads	ø×	Filten					A	Render 64723: skDueuePresentKHR		٥×
20055		UD	Error Res	ut	Name	Arguments		Comparison of the South of the		
20007		64071	WC.	SUCCESS	Waterness	(device=locatelinit), fenceCount=1, pFences=lobitin7bel), watAl=ut_TRUE, tmeout=10000704077				
20052		64072	. WK.	SUCCESS		(device=list21005clit, fenceCount=1; pFences=list210 /buil)		and the second second		
20084		64073				(device=inclanateli, buffer=inclination, pAlocator=Huit)				
20164		64074				(device-live.http:///.buffer-thir25/s546, pAlocator-live.)		and the second se		
20170		64075			willwatzubathilter	(device-lizeballow), buffer-bib350e031, pAlocator-buil.)		A DESCRIPTION OF THE OWNER OWNER OF THE OWNER OWNER OF THE OWNER OWNE		V
		04076				(device=0x2500x20, buffer=0x2556770, pAlocator=100.0)		And I wanted to be a second second	A CONTRACTOR OF A	
		64077				(device-inclimital), buffer-del i 16008, pAlocator-NULL)		Contraction of the	Constant of the local division of the local	
		64078				(device=lint(hill), buffer=lini() 9.007, pAlecator=400.0)			and the second s	And and a state of the state of
		64079			widestroyoutfor	(device=(willhill()), buffer=3h(25Hill()), pAlocator=(0.01)			-	
		64000				(device-metrimonit, buffer-betchess), pAkocater-Statia)			and the second second	and the second division of the second divisio
		04085			•KDentrosBuffer	(device=Withmild(), buffer=Uni2564007, pAlocator=UUL)				and the second second
		64082			- Administrative Limbs	(device-inclusion, buffer-intrision, endocator-risio)			A Los and	

97371 vkCr

vkCmdPipelineBarrier

97372

ge (commandBuffer=0xc2e01f80, srcBuffer=0xb92870d0, dstImage=0xc0b10140, dstImageLayout=VK_IMAGE_LAYOUT_TRANSFER_DST_OPTIMAL, regionCount=1, pRegik (commandBuffer=0xc2e01f80, srcStageMask=VK_PIPELINE_STAGE_TRANSFER_BIT, dstStageMask=VK_PIPELINE_STAGE_ALL_COMMANDS_BIT, dependencyFlags=0x00

	04005		**DestroyBuffer	(device-inclusion	30, buffer=3rl(29+1111), pAlocator=3931.)			
	64089			(device+0x20000	UI, mageView=0v9940000, pAlocator=0011.)			
	64090			(device-inclosing	dit, mage=(set)(ster, pAlocator-(stat.)			1920 x 1080 px
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	04091			(device=manual	<ol><li>mapeView=0.00107/00, pAlocator=0.001.)</li></ol>	and the second second second		
Opening File 'Dc/Documents/current/robios/robios-	64092			(device+monitor	To mage-indification pAlecator-Tell.)	Recorded Framebuller		D X
Connected to renderer	64093		witherstroyBuffer	(device-inclaim)	00, buffer=0(11)fec(10, pAlocator=1001)	CA722 VEQUEDISTICITY		
File loading has finished (0 errors, 0 warnings). File closed	64094	WK_SUCCESS		(device- meridian	til, commandPool=()=(1100000, flags=(=0000000))			
Opening File 'D:/Documents/current/robios/	64095	WC SUCCESS		(physicalDevice=	christoli, suface-(scheffill), pSufaceCapabilites-(schbinnon)			
com.roblax.clent 2019 10 18 11 33 18.pvrcbn'	04096	VK_SUCCESS	-degetCommon@Utilier	(commund@uffer=	bin(1900, peeprenio+(abin(2010)		and the second se	
Disconnected from renderer	64097	VK_SUCCESS		(device=0xc2660x	ili, oCreateInfo-(aeffit/1011, pAlocator-page, oBuffer-(nin; idiatil)	and the second se	and the second	
Connected to renderer	04099			in (device-wicening	36, buffer=340119x000; pNenoryRegurementa=0x040017000)	the second s		1
Re closed Oneming File Tri/Documental current (retriev)	64100	VK_SUCCESS	vieldenfanterinnen	(device+mclambo	Si, buffer=(sh11)sidim, memory=inclicit(217); memoryOffset=(115))	and the second second		
PBIL_apk/pvrcarbon/dasco597_csm_sdft0.pvrcbn'	64101	WK_SUCCESS		(device-incident)	iii, pCreateInfo-Gio/Un7jiiii, pAlocator-UNL2, pduffer-Inini (00100)		and and the second division of the second div	
Deconnected from renderer Connected to renderer	04103		Waintelleffertervfread	device - mediate	11, buffer="https://www.pMemoryRegurements=Doinful?W//		and the second	Street of Street of Street
	64104	VK_SUCCESS		(device - manage	N, buffer=3dy1040.00, memory=342027740, memoryOffset=10000)		A CONTRACTOR	and the second se
And the second sec							and the second second	and the second se
	Static Analysis					θ×	al F	
WE WILL WILL DURING	Filters							and the second second
He loading has maried (0 errors, 0 warnings)	Level Form	un cai	Des	artion				
	A Wammel 0	328 oDrawR	angeElements No a	iorooriate framebuffer	clears have been used between dBirdFramebuffer and the first eDraw" command.	Thi		
	A warmost 0	372 dibrawA	rava No a	promite framebuffer	clean have been used between ullandFramebuffer and the first ofDrave* constand.	Thi	THE STREET	The second second
	A Wampel 0	419 dDrawR	angefilements filo a	propriete framebuffer	clean have been used between dEndFramebuffer and the first oDcaw* command.	Thi		
	A second second second	Saw Average	an and a second	Contraction of the second s				
	Static Analysis Ou	ment Call Find Re-	euto -				41, 573 px   0.073, 0.53   RGBA(30, 19, 13,	255)   1920 x 1000 µx
					Current Frame: 137	240	Benderert Hort	



#### **Updated Performance**



#### Glorious task packing!



### **GLSL** optimisations

vec4 1 = (float 1 * vec4 2) * float 2;  $\longrightarrow$  vec4 1 = (float 1 * float 2) * vec4 2; Group scalar arithmetic abs() negate and clamp(..., 0.0, 1.0)/saturate can be free  $vec 0 = vec 1 * vec 2 + vec 3; \longrightarrow vec 0 = clamp(vec 1 * vec 2 + vec 3, 0.0, 1.0);$ 

PowerVR developer documents: <u>https://docs.imgtec.com/</u>



### **GLSL** optimisations - utilisation

Too many registers being used

- less threads can be processed in a cluster at a time
- reduced utilisation

Mediump allows PowerVR to use 16 bit floats

• reduce register pressure, increase occupancy (up to 100%)



## **GLSL** optimisations

PBR shader:

- A-Series: cycle count decreased by 9%
- Rogue (Oppo Reno): cycle count decreased by 12% and utilisation improved by 33%

PBR+IBL shader:

- A-Series: cycle count decreased by 9%
- Rogue (Oppo Reno): cycle count decreased by 20% and utilisation improved by 36%

Use PVRShaderEditor to analyse cycle count and disassembly



#### Analysing Roblox on PowerVR

**Performance improvements** 







# **LIGHTING AND SHADOWS IN ROBLOX**

### Lighting system: phase 2, shadow maps

- Voxel shadows are too coarse
- Shadow maps to the rescue!
- Excellent quality
- Challenge: "soft" shadows
- Challenge: high cost to re-render
- Challenge: many shadow casting lights
- Only released sun shadows to simplify





## Lighting system: rendering shadow maps

- Cascaded tiled shadow maps
  - 1-4 cascades depending on the quality levels
  - Far cascades are split into tiles to be able to throttle shadow updates
- CSM Scrolling (Insomniac @ SIGGRAPH 2012)
  - CSM Scrolling: An acceleration technique for the rendering of cascaded shadow maps
- When we need to update a cascade, it's done in single pass
  - If multiple tiles are marked as dirty, we re-render geometry in these tiles
  - CPU does per-tile frustum culling
- Light direction changes invalidate caching :(
  - Haven't found a good solution to this







## Lighting system: shadow softness

- Want a wide range of shadow map "softness"
  - Wide PCF kernel expensive at high resolutions
  - Screen-space filtering impractical due to transparent geometry and tiler performance
- EVSM to the rescue!
  - Fully decouples shadow rendering from sampling
  - Light leaks...





## Lighting system: making EVSM practical

- Light leaks
  - Reduce Z range!
    - Tightly fit shadow frustum to receiving geometry
    - "Pancaking" when rendering caster geometry
    - VkPipelineRasterizationStateCreateInfo::depthClampEnable
  - Over-darkening when sampling EVSM
- Performance
  - Use two moments (RG16F) on mobile, four (RGBA16F) on desktop
  - Render depth only, reconstruct exponentially warped moments during first blur pass
- Integrating with tiling...
  - Blurring is per tile to make it fully incremental, we cache blurred EVSM contents
  - CSM scrolling might require adjusting the cascade depth
    - exp(z + offset) = exp(z) * exp(offset)



## Lighting system: blurring shadow maps

- Standard two-pass blur (horizontal & vertical)
- First pass reads directly from depth texture (D16/D24)
  - Need to reconstruct EVSM moments from depth
  - Can't use bilinear filter optimizations :(
- Second pass reads from EVSM texture
  - Fewer taps!
  - See "Efficient Gaussian blur with linear sampling"
- Filter width varies from 3x3 to 17x17
  - Depends on developer-configured parameter, "ShadowSoftness"





## Lighting system: shadow MSAA

- Wide blurs are pretty expensive
  - Render depth at full resolution, two pass blur
  - Three full stores of tile contents to memory
- Since shadow moments are filterable, we can use MSAA!
  - Use 4x MSAA to reduce shadow resolution by 2x2
  - Fragment shader directly outputs shadow moments (render pass isn't depth only anymore)
  - Subsequent filters are smaller (2x) and can use bilinear optimizations
- On-chip resolve is crucial
  - *Don't* use vkCmdResolveImage!
  - Instead, use pResolveAttachments in render pass
- MSAA becomes an almost free way to increase shadow quality



## Lighting system: sampling shadow maps

- Find a single cascade per pixel
  - Some ALU to do point-sphere tests per cascade
  - Because of throttling & quality concerns can't use pixel depth
- Sample EVSM texture
  - Requires adjusting UV coordinates
  - All shadow maps are stored in a single shadow atlas
- Compute shadow factor
  - Exponential depth warping
  - 2 or 4 moment VSM
  - Apply over-darkening to reduce leaks







## COMPUTE OPTIMISATIONS FOR POWERYR



#### **Shadow Mapping Performance Analysis**





#### Move gaussian blur to compute?

Compute post-processing allows more scheduling flexibility



What about work in the middle of the frame?



Shadow map cascades 1, 2 and 3



### Separable-kernel gaussian blur

- Perform 2D gaussian blur with two 1D passes
- Mathematically equivalent (rank 1 matrix)
- 2x texture fetches instead of x²





#### **Compute gaussian blur gather algorithm**

Optimal work-group size on PowerVR is 32

8x4 work group size processing an 8x8 area

Experiment with looping shader multiple times (2 texels per thread here)





#### **Compute gaussian blur gather algorithm**

Read depth values including surrounding area into shared memory

Reduces the number of texture fetches



Texture lookups per shadow map texel





#### **Morton ordering**

Images created with VK_IMAGE_TILING_OPTIMAL addressed with Morton ordering Improve cache efficiency by loading an aligned area of texels (512 bit cache line)



(8n, 8n) 32 value fetch aligned to 8x4 coordinate multiple then fetch edge values

Z-order indexing

C Imagination

### Linear indexing

		Baı	1 nk		Bank 2				Bank 3				Bank 4			
dword	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
16																
32																
4080																

Shared memory:

- Read up to 16 adjacent floats 1 cycle
- Read multiple rows in a bank in n cycles
- Read = 1 cycle, write = 4 cycles

```
// our 32 threads write to adjacent memory simultaneously
float warped_depth = warp(texelFetch(depth_buffer,
    gl_LocalInvocationID.xy + offset).x);
shared_storage[gl_LocalInvocationIndex + n*32].x =
    warped_depth;
```



### **Horizontal pass**

Write moments M1 and M2 to shared memory

$$M_1 = E(x) = \int_{-\infty}^{\infty} xp(x)dx$$
$$M_2 = E(x^2) = \int_{-\infty}^{\infty} x^2 p(x)dx$$

VSM algorithm:

http://www.punkuser.net/vsm/vsm_paper.pdf





#### **Vertical pass**

```
Write moments M1 and M2 to the shadow map
```

```
vec2 M1M2 = vec2(0.0);
memoryBarrierShared();
for (uint y = 0u; y < blur_width; y++) {
    uint shared_index = get_shared_index(y,
      gl_LocalInvocationID);
    vec2 M1M2_val = shared_storage[shared_index];
    M1M2 += M1M2_val * get_gaussian_factor(y);
}
```

```
imageStore(shadow_map, global_coord, vec4(M1M2,
 0.0f, 0.0f));
```





#### **Task packing**



Vulkan spec section 6.2:

"The initial order is determined by the order in which vkQueueSubmit commands are executed on the host, for a **single queue**, from first to last."



#### Task packing – alternating queues

Alternate between two queues each frame for submitting commands.

Allows frame independent scheduling and increases task packing!

May require alternating between two sets of resources.

More developer recommendations! <u>https://docs.imgtec.com/</u>



O Imagination

#### **Compute improvements**



#### 40% fps increase for 5x5 blur! **Meizu Pro 7 Plus** - PowerVR 7XTP



### **Compute improvements**

#### Frame times for high-end PowerVR GPUs



Low-end PowerVR GPUs





# THANK YOU!