unity

Unite Shanghai 2019

GameDev Partner Practices

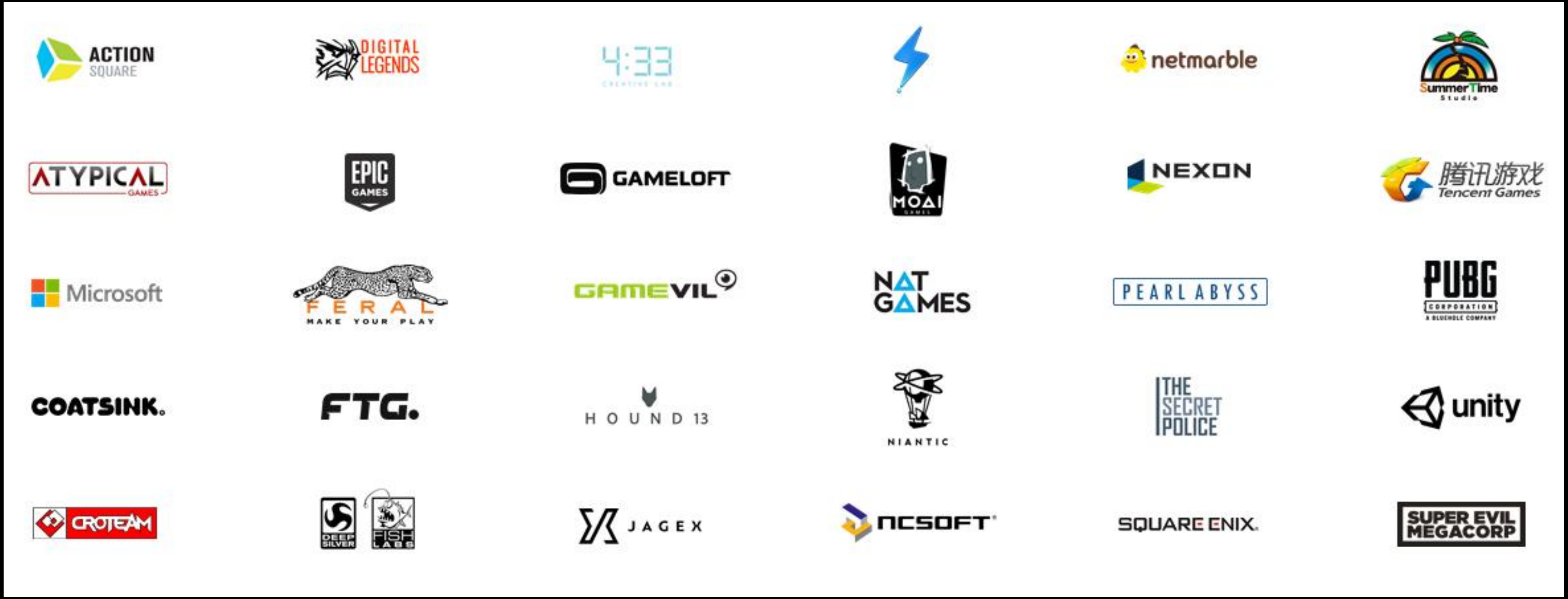
Galaxy GAMEDEV

三星电子（中国）研发中心
姚巍

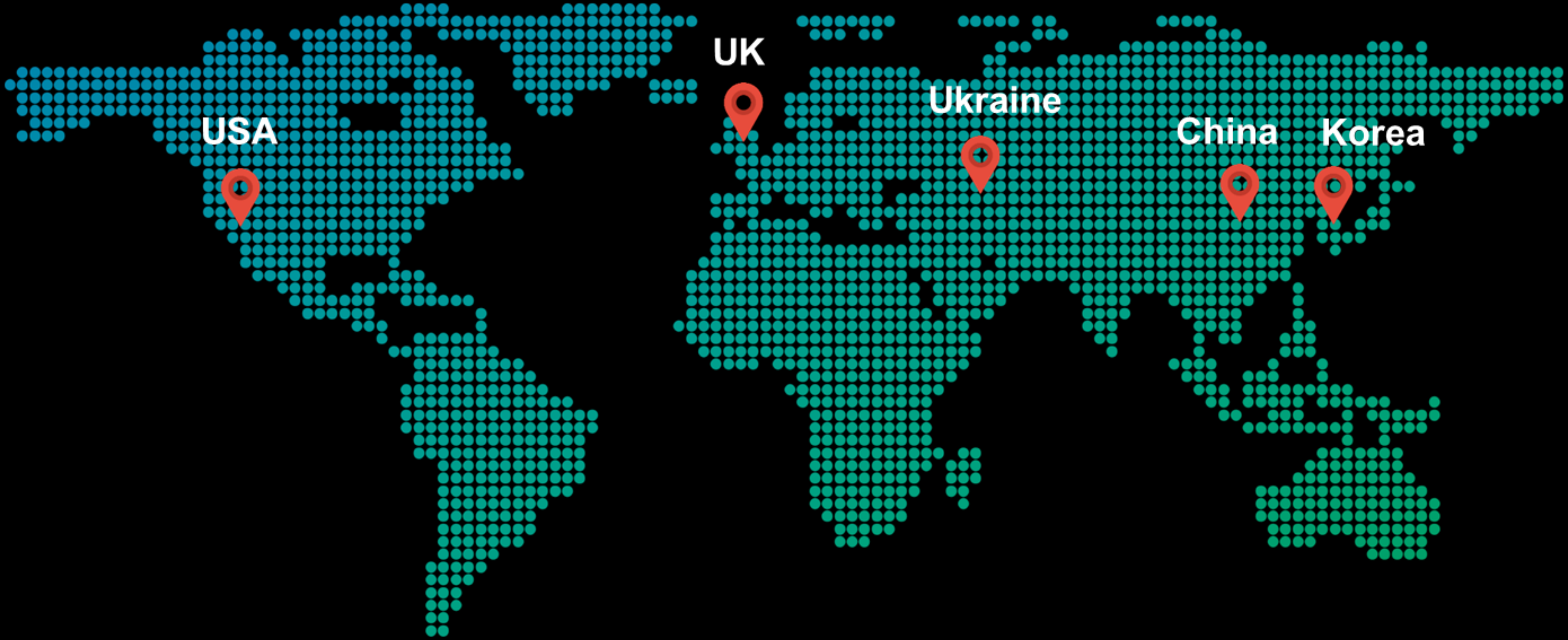
unity

Unite
Shanghai
2019

Partners


ACTION SQUARE
DIGITAL LEGENDS
4:33 CREATIVE LAB
netmarble
SummerTime Studio
ATYPICAL GAMES
EPIC GAMES
GAMELOFT
MOAI GAMES
NEXON
腾讯游戏 Tencent Games
Microsoft
FERAL MAKE YOUR PLAY
GAMEVIL
NAT GAMES
PEARL ABYSS
PUBG CORPORATION A BLUEHOLE COMPANY
COATSINK.
FTG.
HOUND 13
NIANTIC
THE SECRET POLICE
unity
CROTEAM
DEEP SILVER
FISH LABS
JAGEX
NCSOFT
SQUARE ENIX.
SUPER EVIL MEGACORP

Locations & Resources



Team GameDev

GameDev Programs

- Game Engine Optimization
  - Deliver custom game engine
- Device Loaner Program
  - For supporting game dev and QA
- Workload Profiling Support
  - Balance between perf. and quality
- Developer Education / Better Tooling
  - Improve overall game dev environment
- Remote / On-site Tech Support
  - Real & tight co-work with game developer

Recap : 2016 ~ 2018

Recap : 2016 ~ 2018

- Way of Developing High Fidelity Android Games

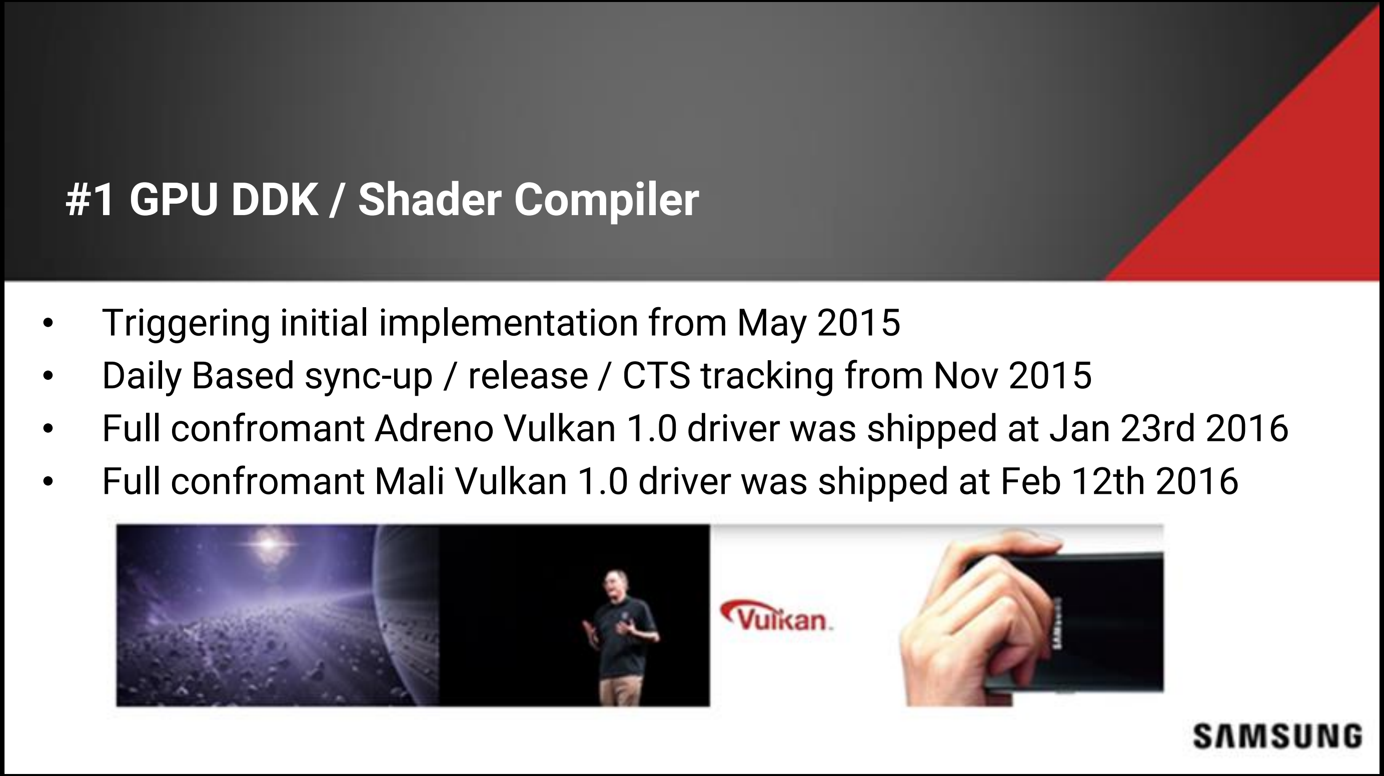
Recap : 2016 ~ 2018

#0 Standardization

The Vulkan and SPIR-V working groups, Seattle, January 2016

* Image from Khronos Group Flickr

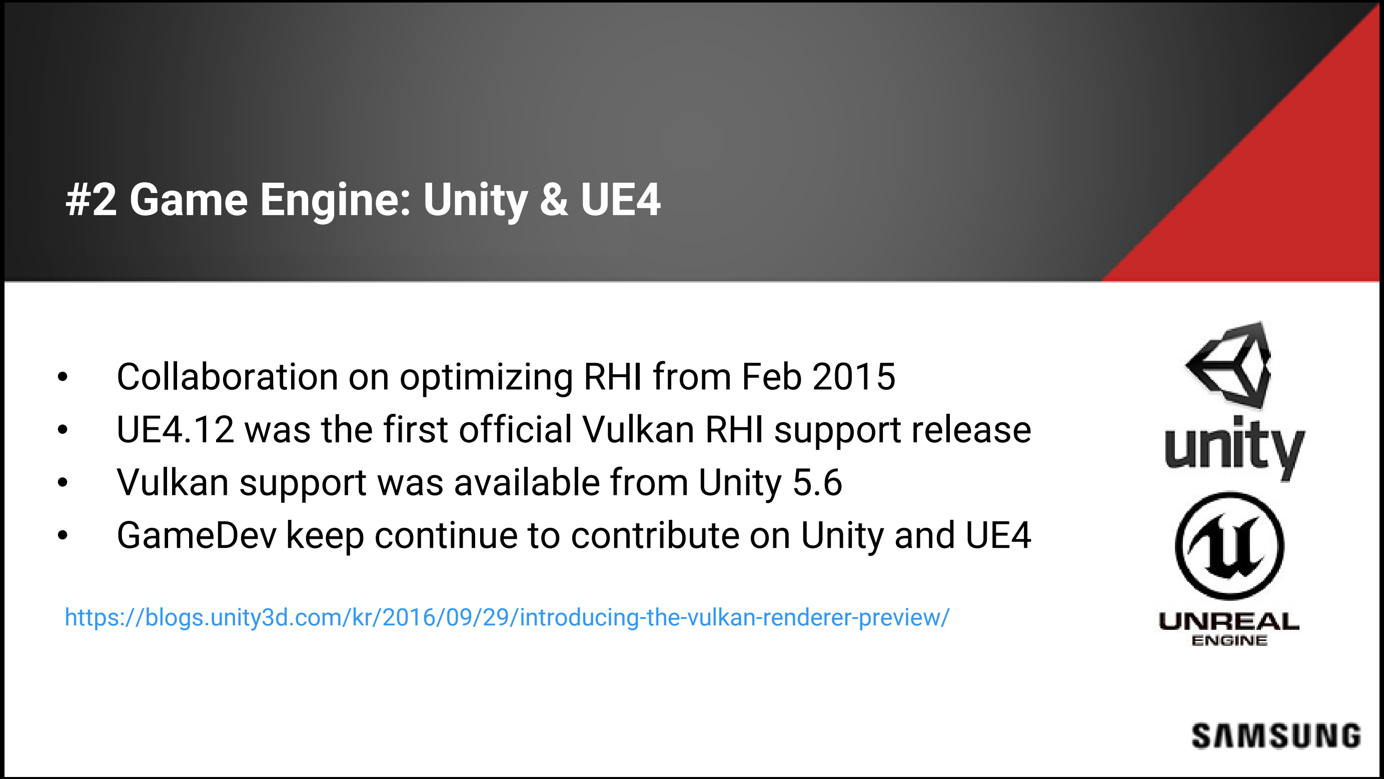
Recap : 2016 ~ 2018



#1 GPU DDK / Shader Compiler

- Triggering initial implementation from May 2015
- Daily Based sync-up / release / CTS tracking from Nov 2015
- Full confromant Adreno Vulkan 1.0 driver was shipped at Jan 23rd 2016
- Full confromant Mali Vulkan 1.0 driver was shipped at Feb 12th 2016

Recap : 2016 ~ 2018



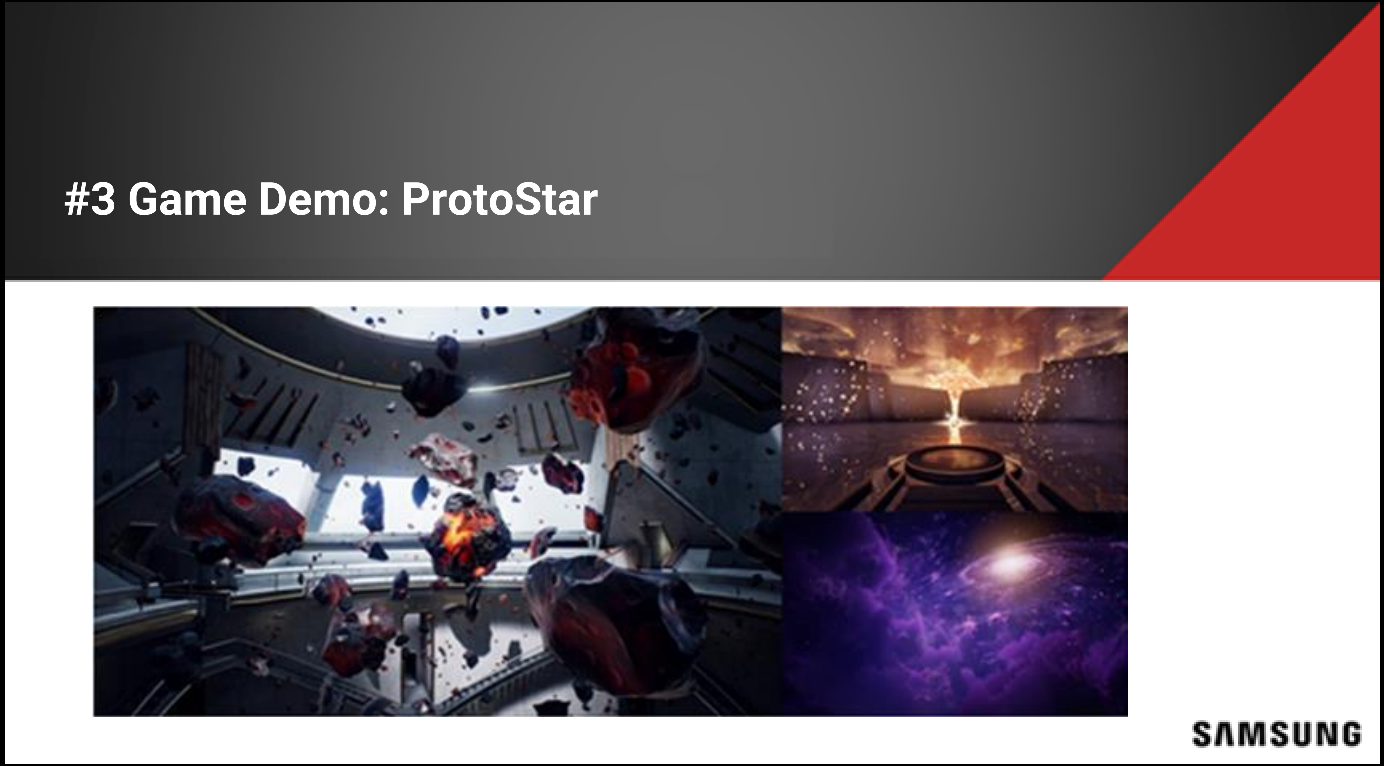
#2 Game Engine: Unity & UE4

- Collaboration on optimizing RHI from Feb 2015
- UE4.12 was the first official Vulkan RHI support release
- Vulkan support was available from Unity 5.6
- GameDev keep continue to contribute on Unity and UE4

https://blogs.unity3d.com/kr/2016/09/29/introducing-the-vulkan-renderer-preview/

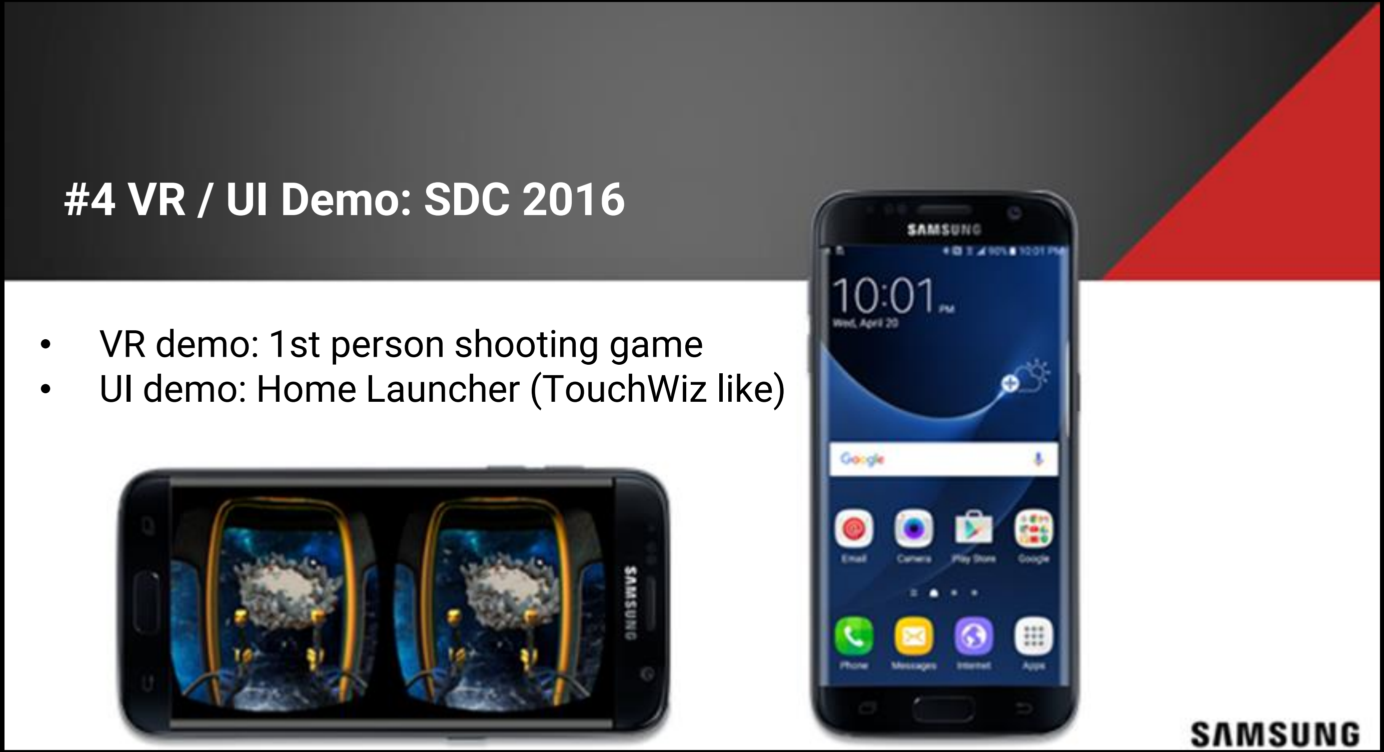
unity
UNREAL
ENGINE
SAMSUNG

Recap : 2016 ~ 2018



#3 Game Demo: ProtoStar

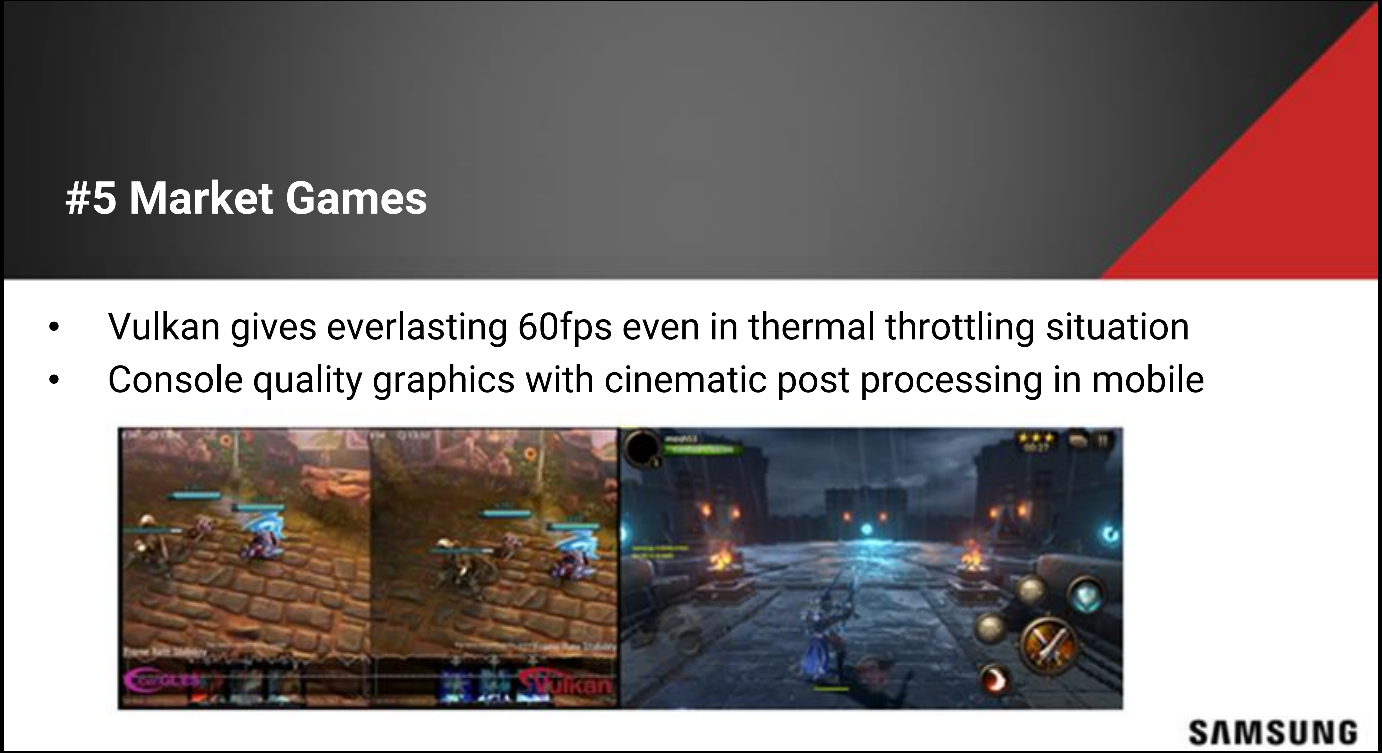
Recap : 2016 ~ 2018



#4 VR / UI Demo: SDC 2016

- VR demo: 1st person shooting game
- UI demo: Home Launcher (TouchWiz like)

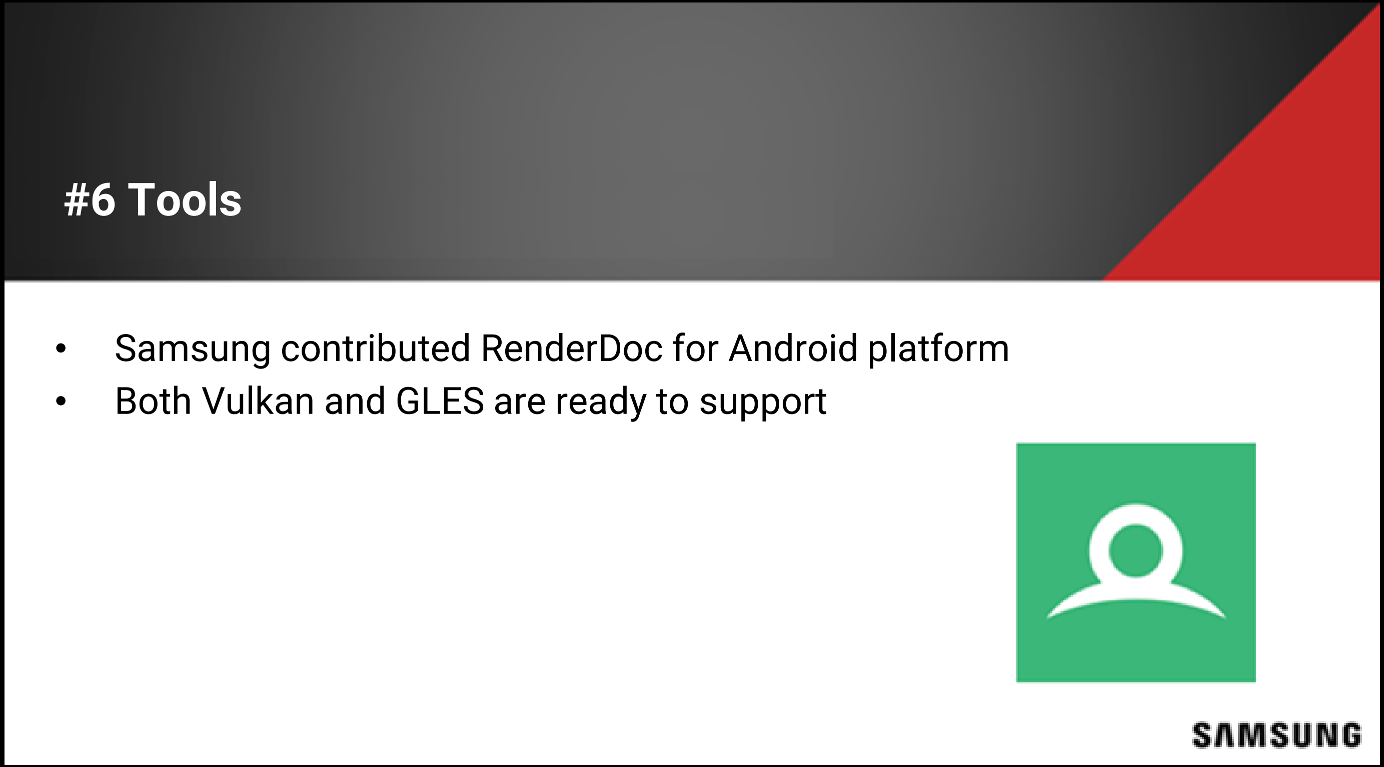
Recap : 2016 ~ 2018



#5 Market Games

- Vulkan gives everlasting 60fps even in thermal throttling situation
- Console quality graphics with cinematic post processing in mobile

Recap : 2016 ~ 2018

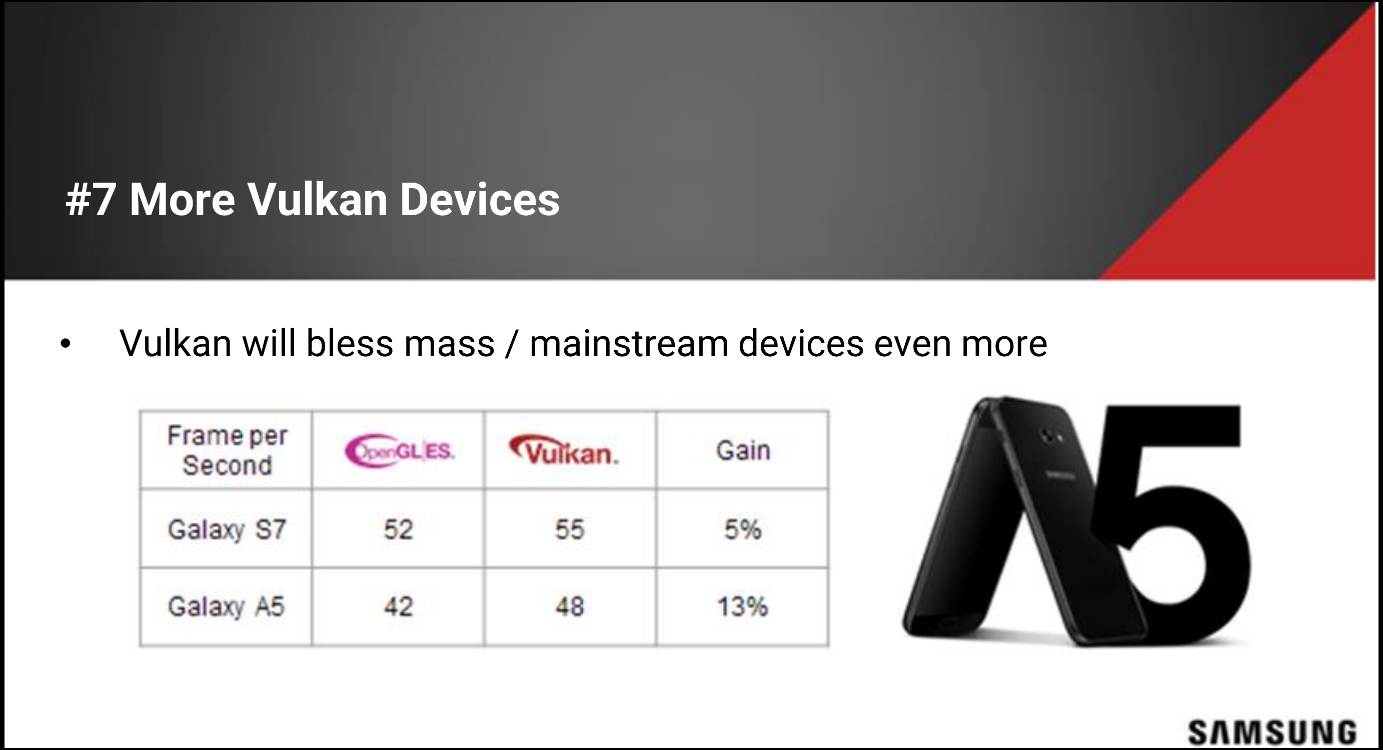


#6 Tools

- Samsung contributed RenderDoc for Android platform
- Both Vulkan and GLES are ready to support

SAMSUNG

Recap : 2016 ~ 2018

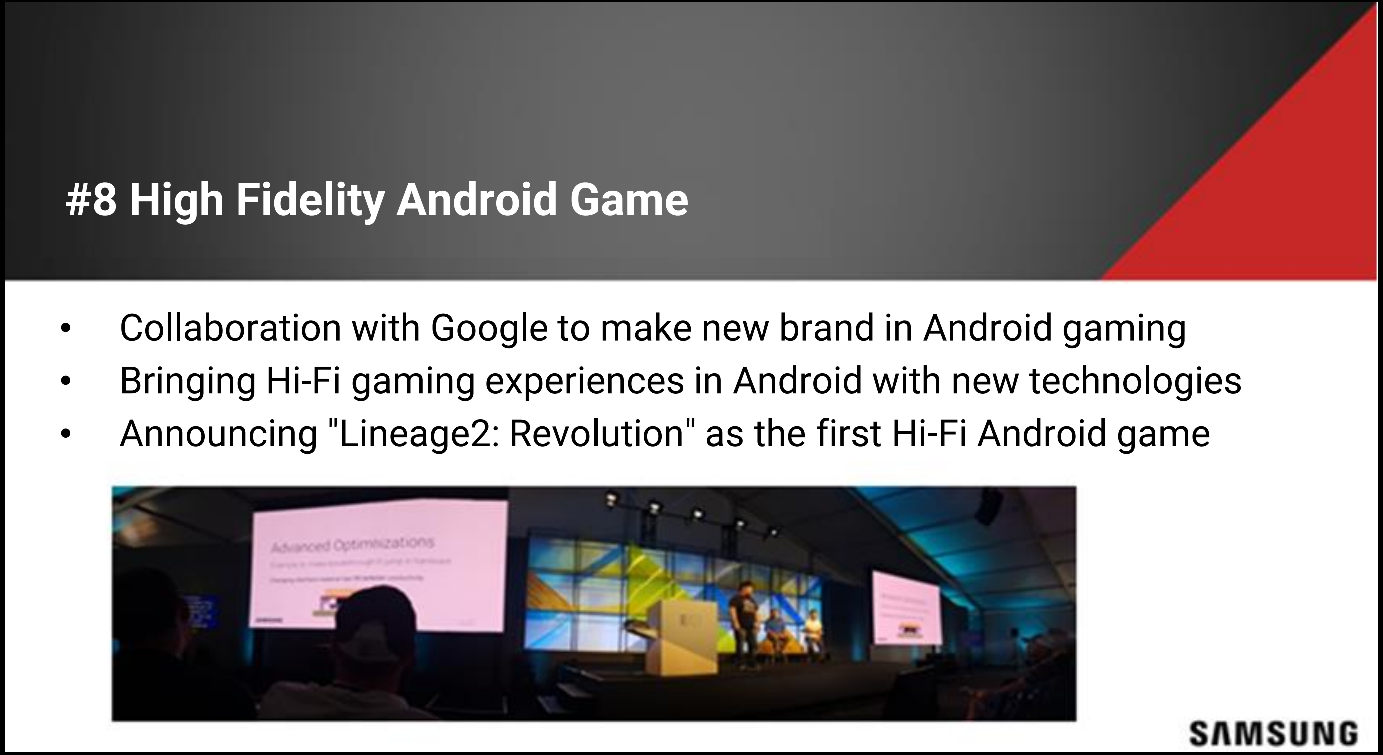


#7 More Vulkan Devices

- Vulkan will bless mass / mainstream devices even more

Frame per Second	OpenGL ES	Vulkan	Gain
Galaxy S7	52	55	5%
Galaxy A5	42	48	13%

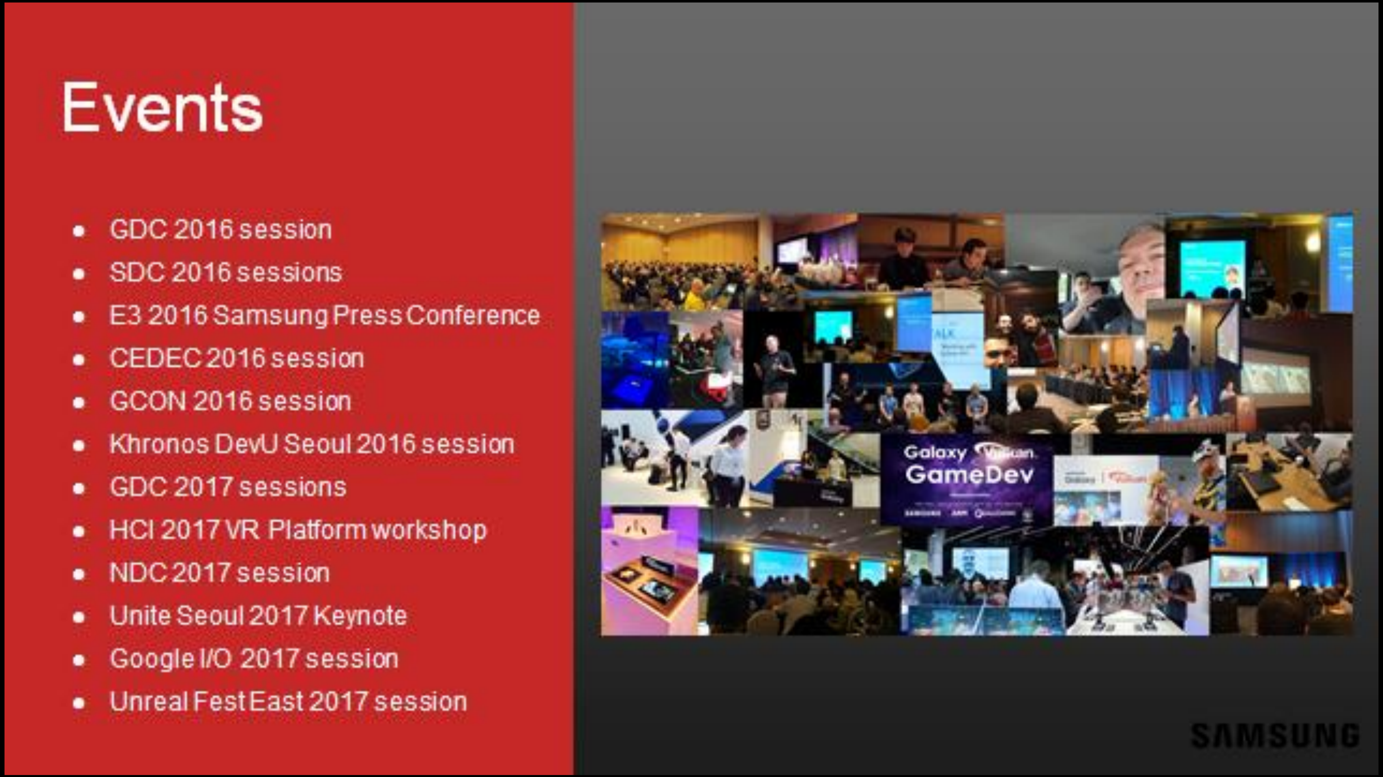
Recap : 2016 ~ 2018



#8 High Fidelity Android Game

- Collaboration with Google to make new brand in Android gaming
- Bringing Hi-Fi gaming experiences in Android with new technologies
- Announcing "Lineage2: Revolution" as the first Hi-Fi Android game

Recap : 2016 ~ 2018



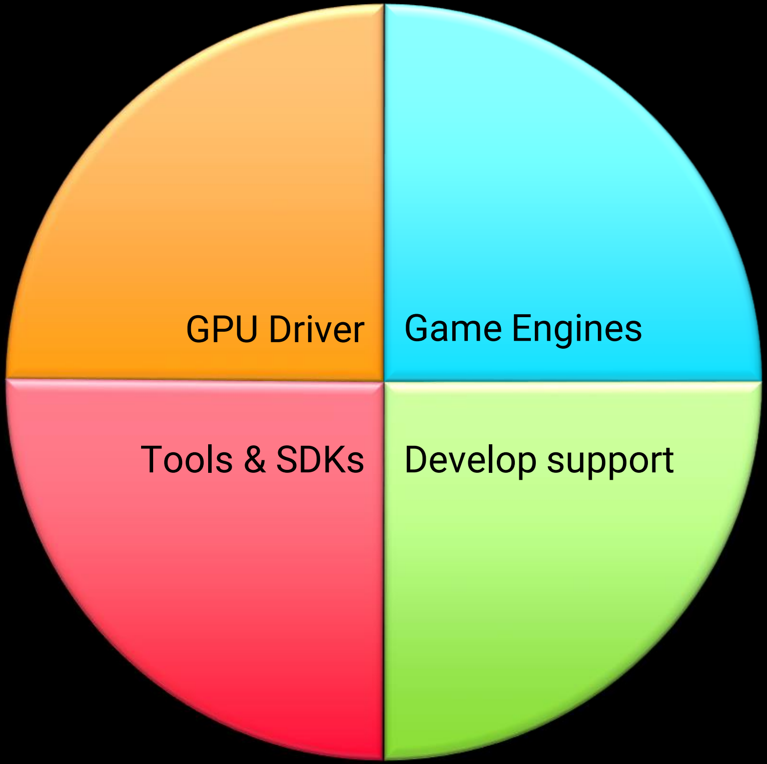
Events

- GDC 2016 session
- SDC 2016 sessions
- E3 2016 Samsung Press Conference
- CEDEC 2016 session
- GCON 2016 session
- Khronos DevU Seoul 2016 session
- GDC 2017 sessions
- HCI 2017 VR Platform workshop
- NDC 2017 session
- Unite Seoul 2017 Keynote
- Google I/O 2017 session
- Unreal Fest East 2017 session

Galaxy Vulkan
GameDev
SAMSUNG

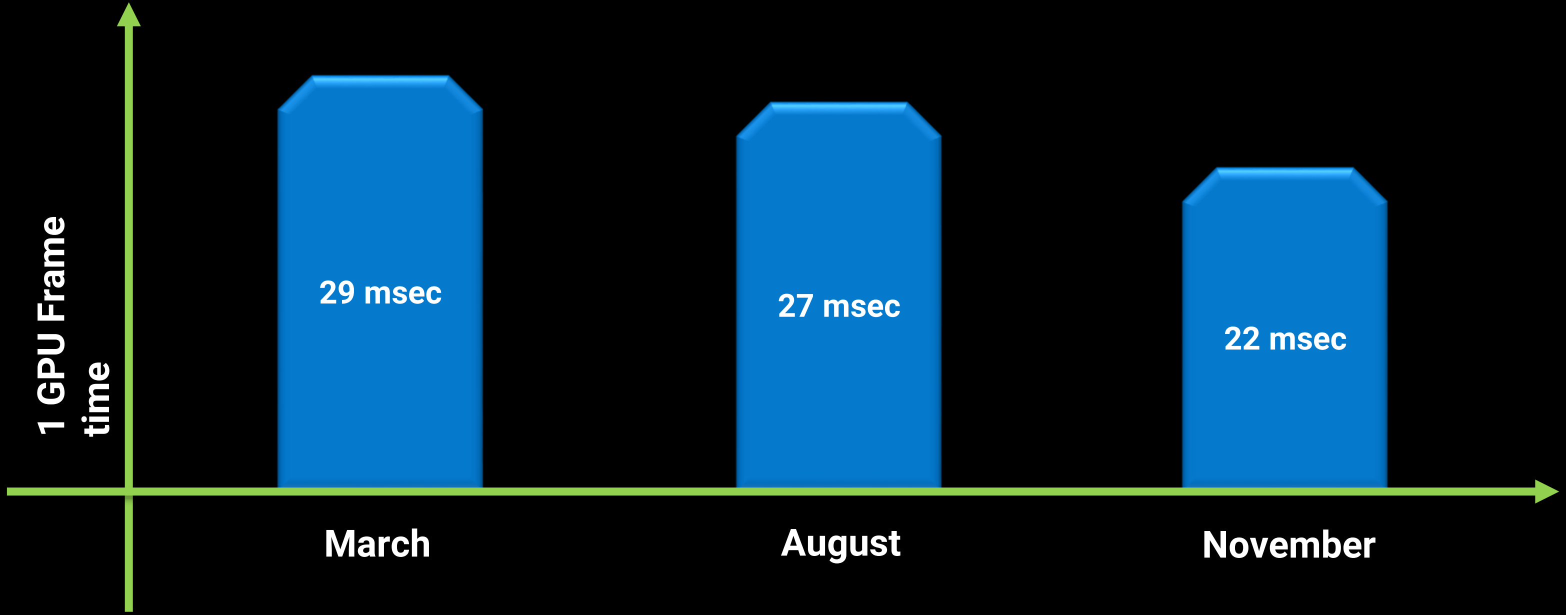
Key Updates

4 key areas about game optimization



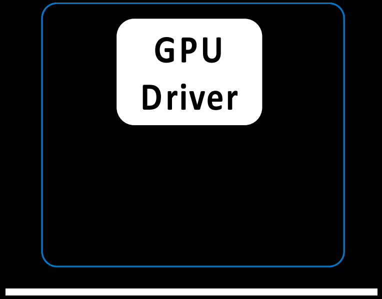
GPU Driver

- Performance : Driver optimization in 2018



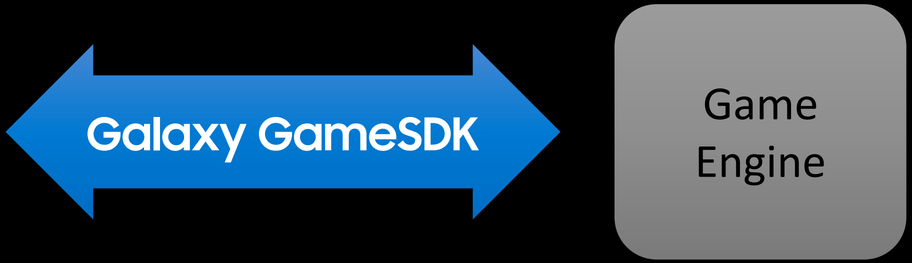
GPU Driver：Updatability

To manage GPU driver quality with more better way Samsung & Google closely have worked together for driver updating with installable packages from Play



We are still working hard to bring this in consumer device target for improving market game support

Galaxy GameSDK



Galaxy GameSDK

Game Engine

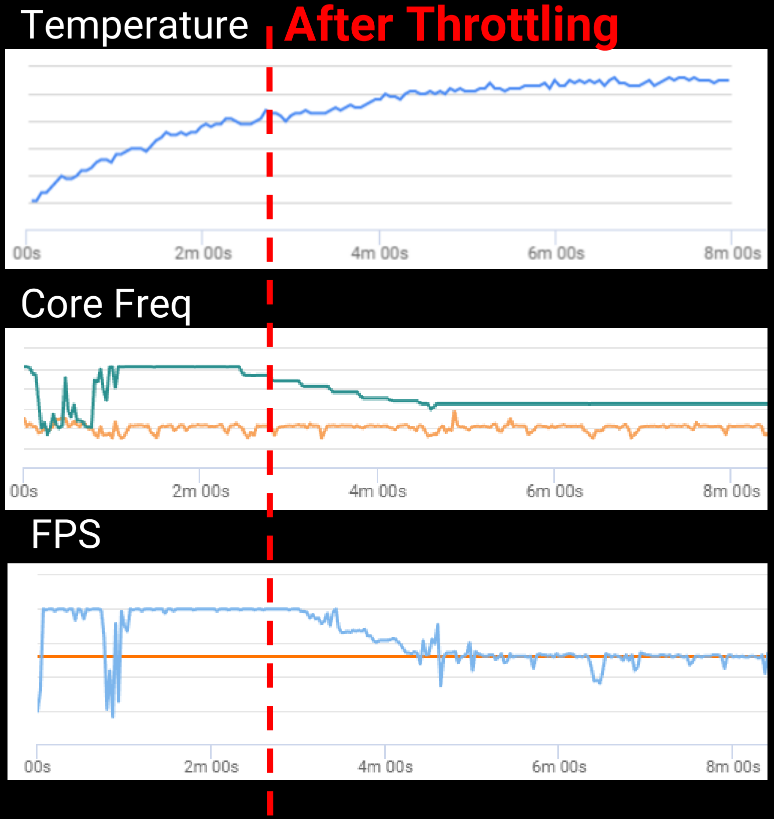
GameSDK

New interface to exchange power and thermal hint between game app and the device to manage gaming performance based on clear context is coming soon

This will allow game developer to tune their games by more predictable and manageable manner

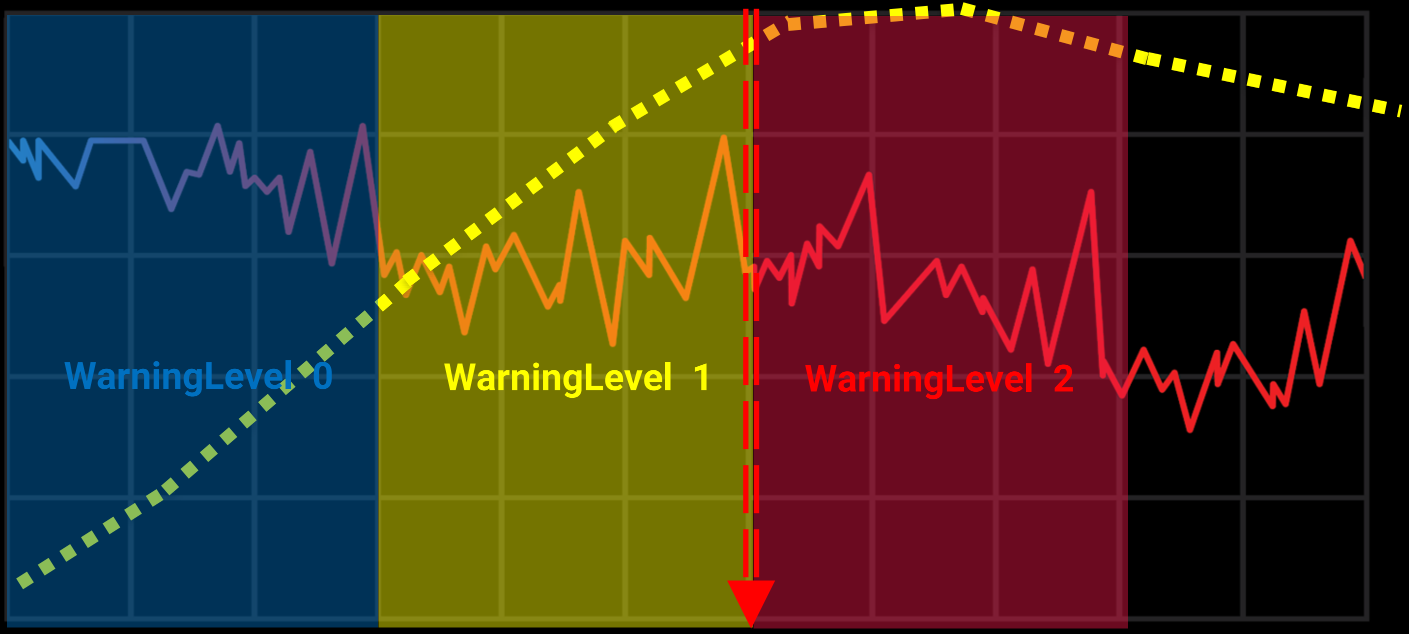
Mobile Games have Problems

- Thermal throttling
  - No active cooling system
  - Performance Throttling
- Power
  - Limitation of battery
  - Use more Power than necessary
- **Developer can’t control**



Thermal throttling

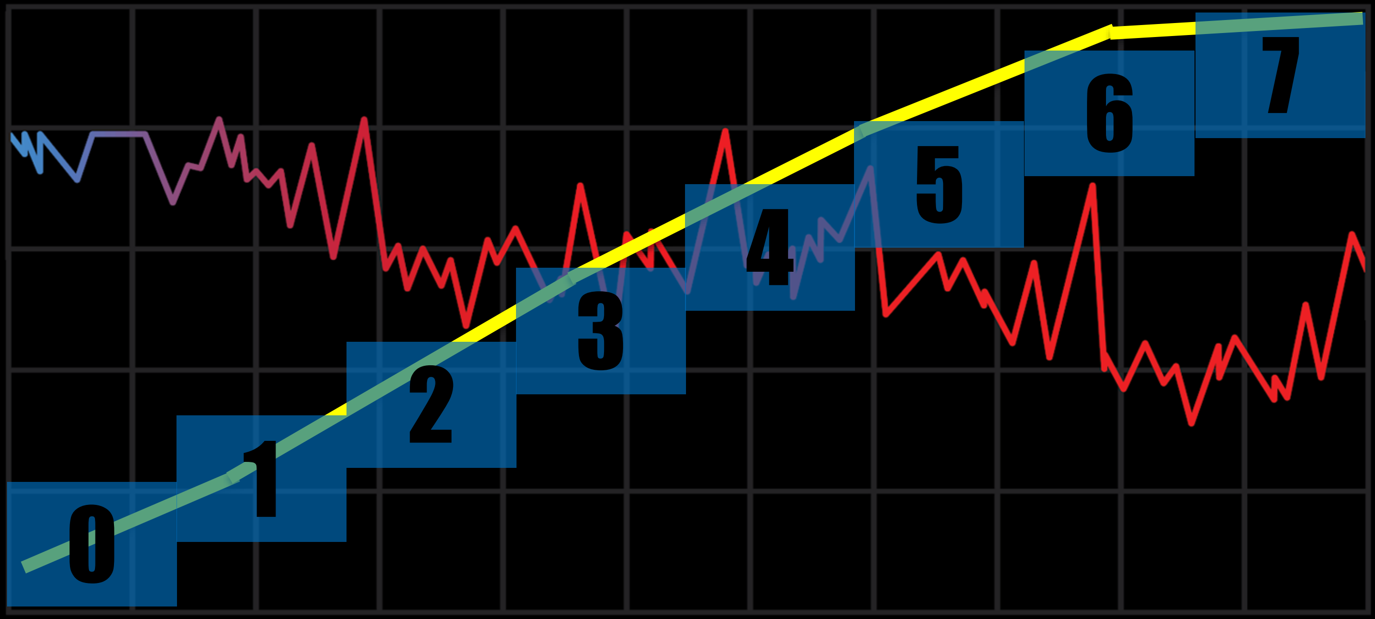
- onHighTempWarning(int warningLevel)



Throttling

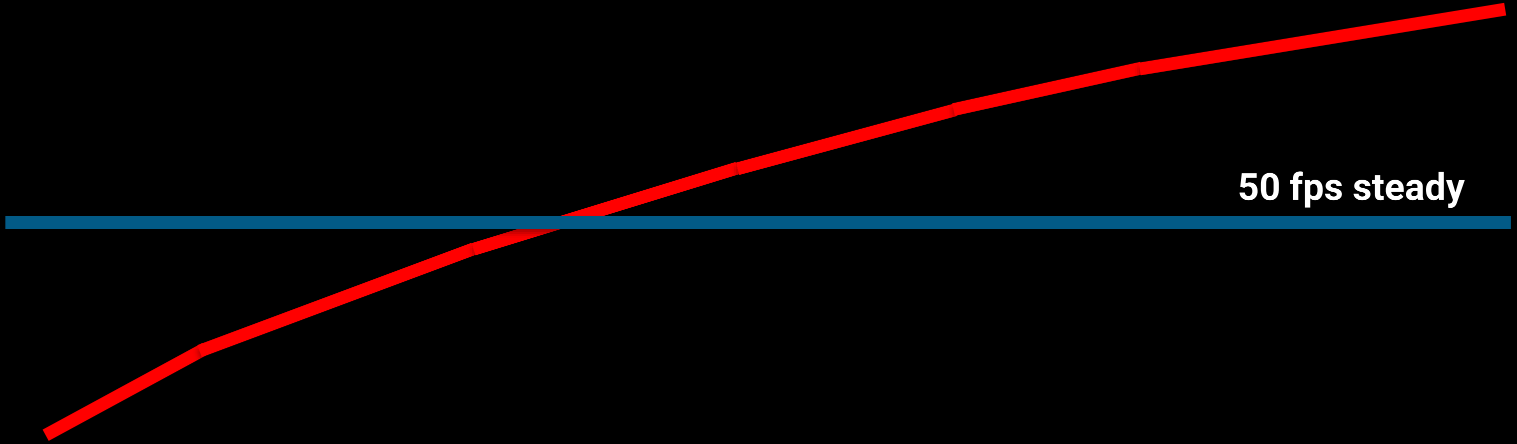
Thermal throttling

- getTempLevel(), getCPU/GPUJTLevel(), getSkinTempLevel()



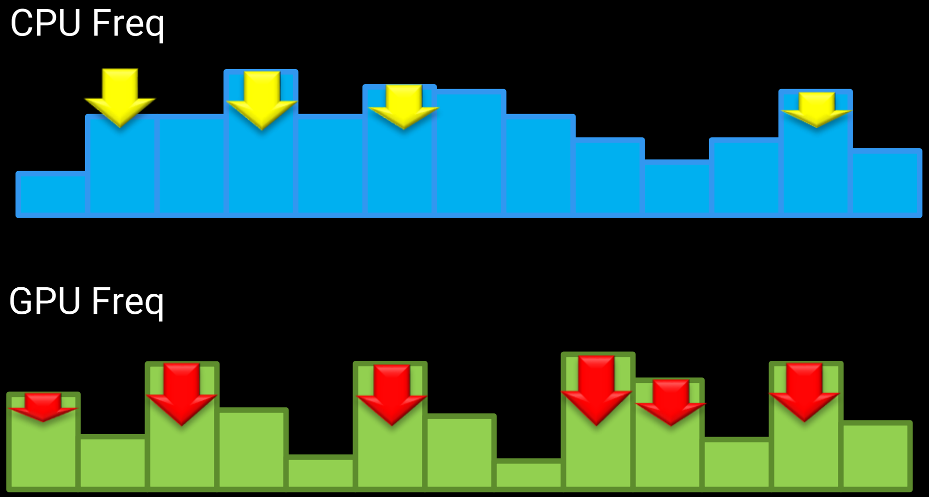
Thermal throttling

- Now you can do !
  - Predict Thermal throttling timing
  - Prevent Thermal throttling
    - Adjust TargeFPS, Effect and can do something



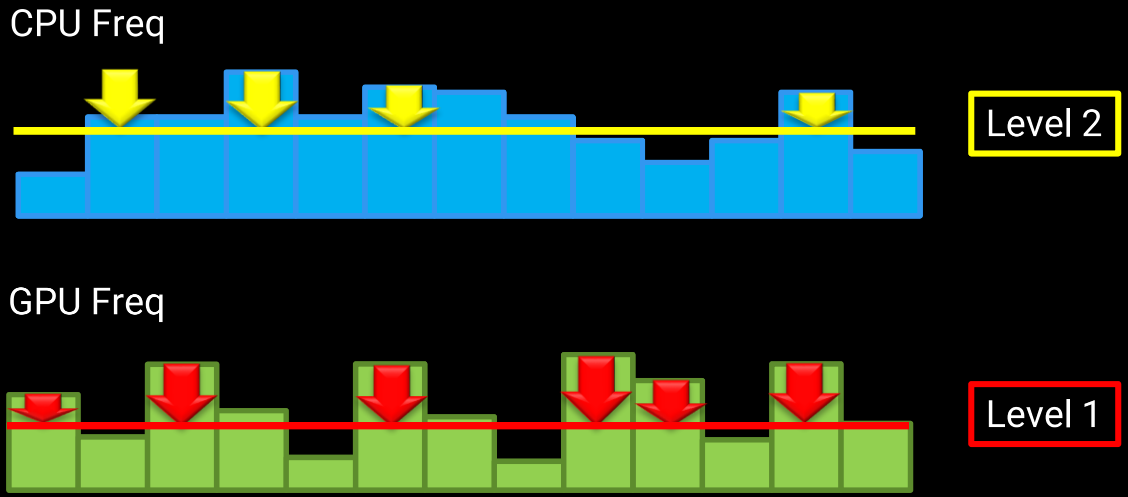
Power Management

- Frequency over react



Power Management

- setLevelWithScene(String scene, int cpuLevel, int gpuLevel)



CPU Freq

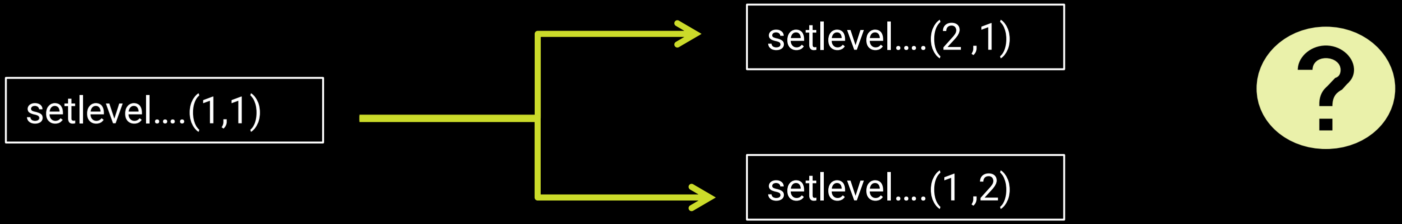
Level 2

GPU Freq

Level 1

Bottleneck Identify

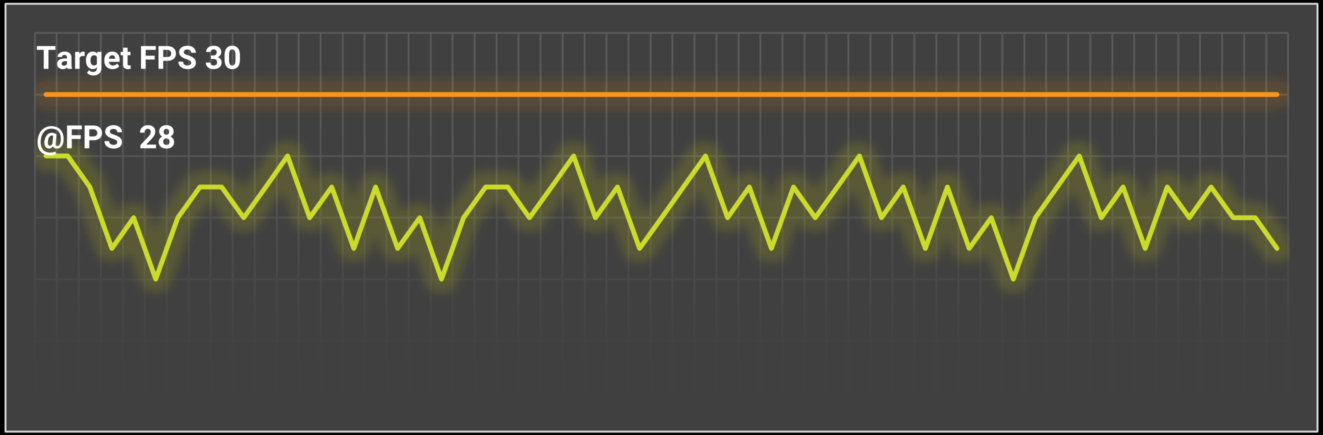
- Precise decision making to control frequency



setlevel….(1,1)

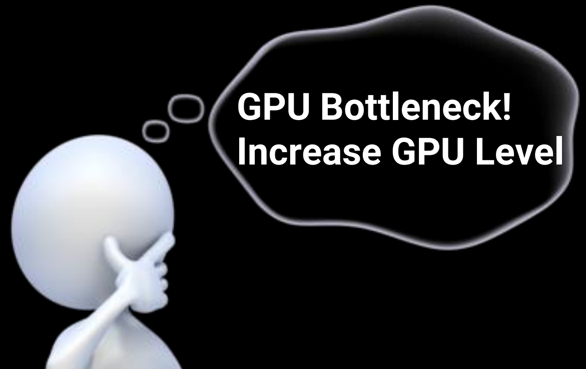
setlevel….(2 ,1)

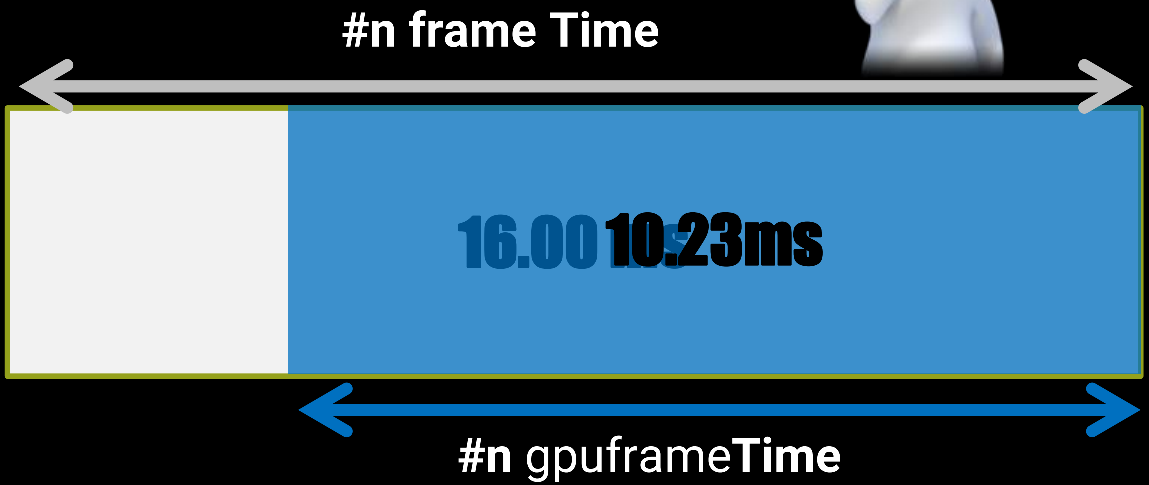
setlevel….(1 ,2)

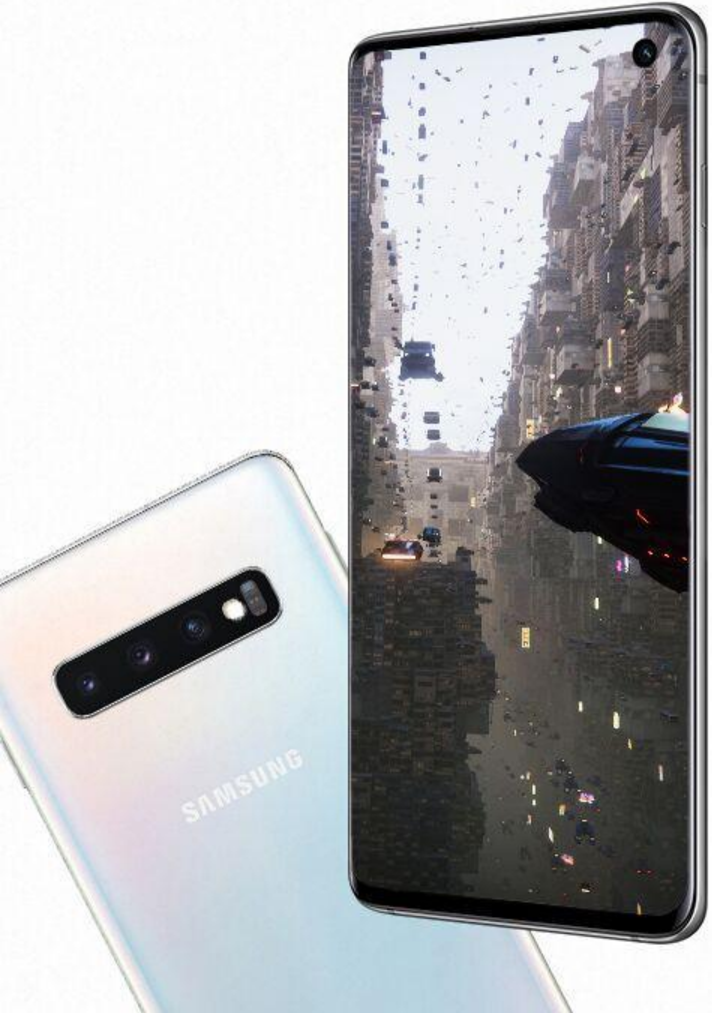


Bottleneck Identify

- getGpuFrameTime()
  - #n frame’s GpuFrameTime







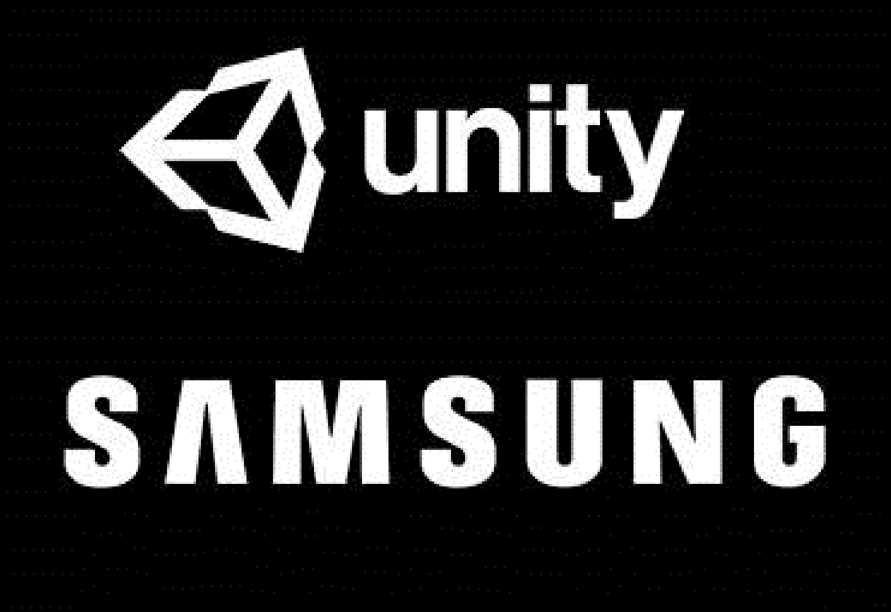
Adaptive Performance

Adaptive Performance

Get device performance status and thermal trends

Proactively adjust performance and quality settings on the fly

First Integration into popular Game Engine --- Unity



Maximum Quality + Maximum Performance

=

Smoother Gameplay

MEGACITY

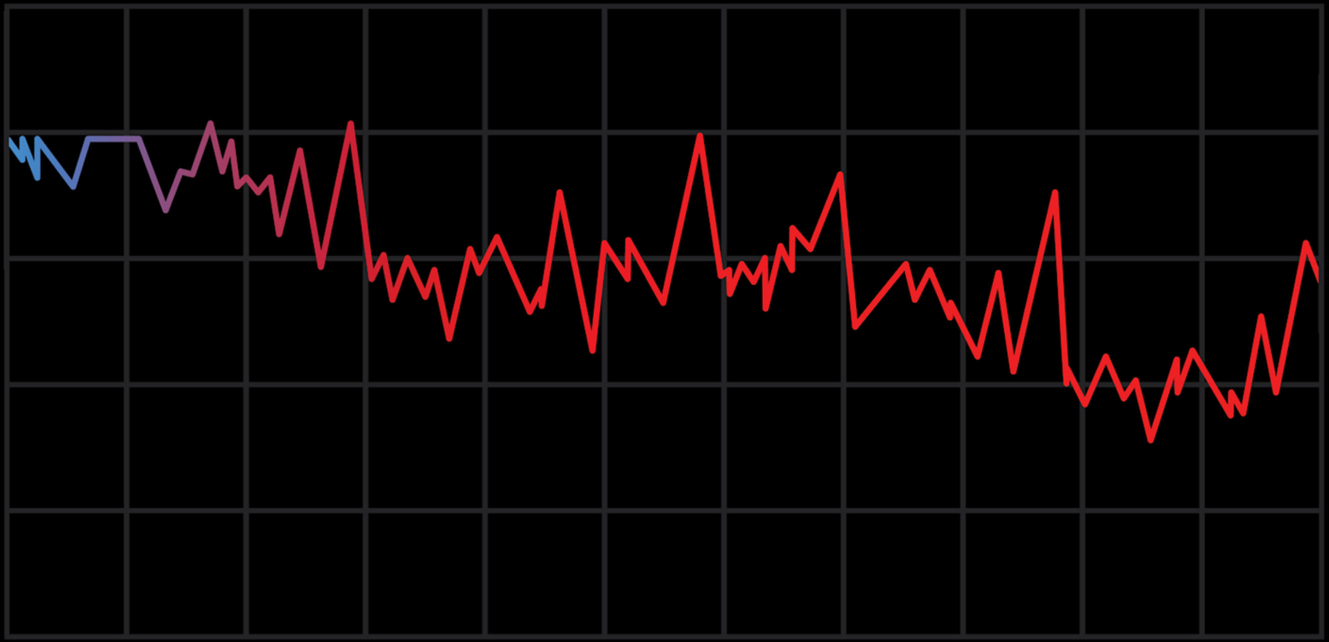
Unruly Heroes by Magic Design Studios – Made with Unity

8M	6M	2K
Triangles	Entities	Cars

30

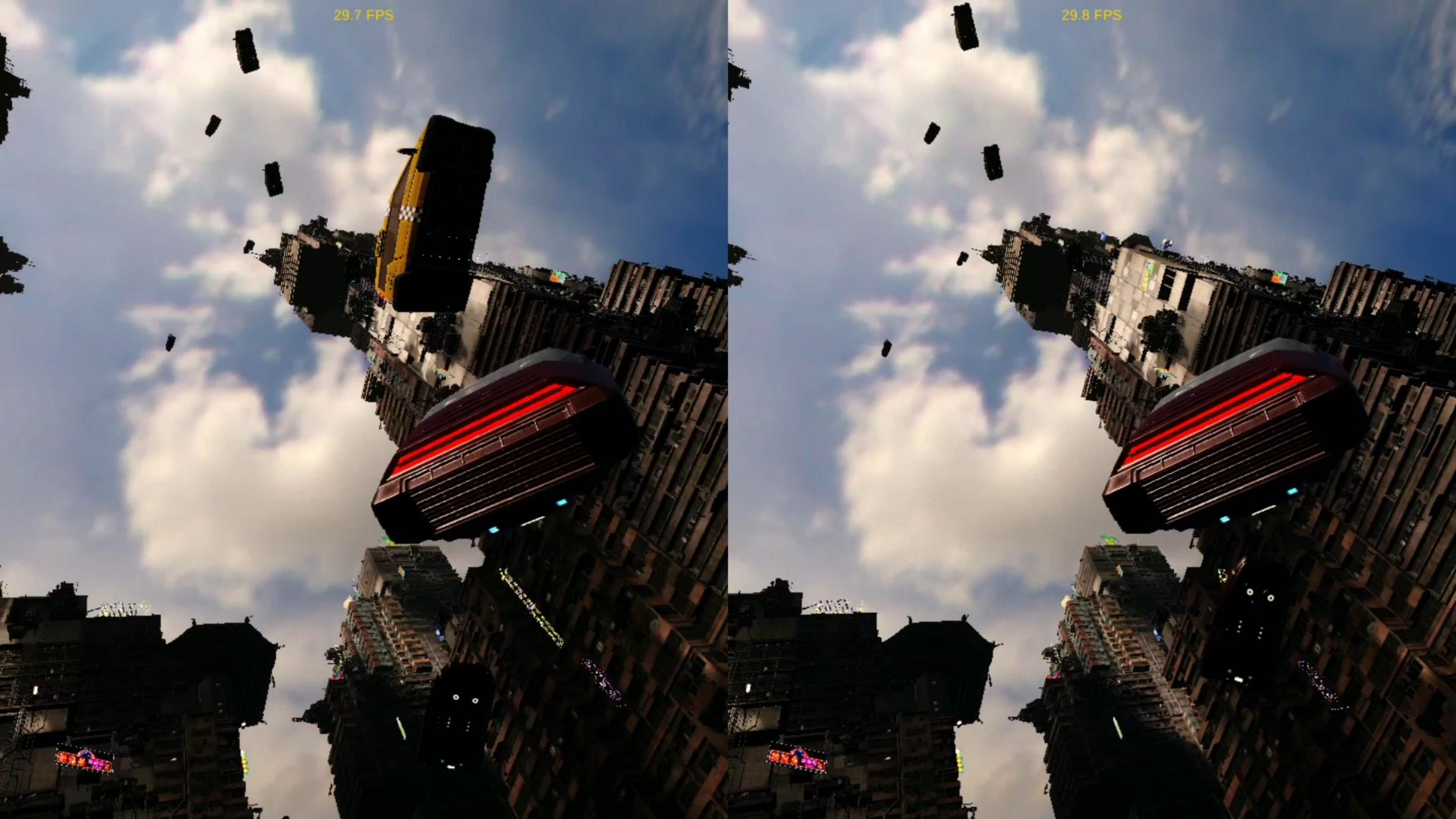
FPS – Stable over time

FrameRate

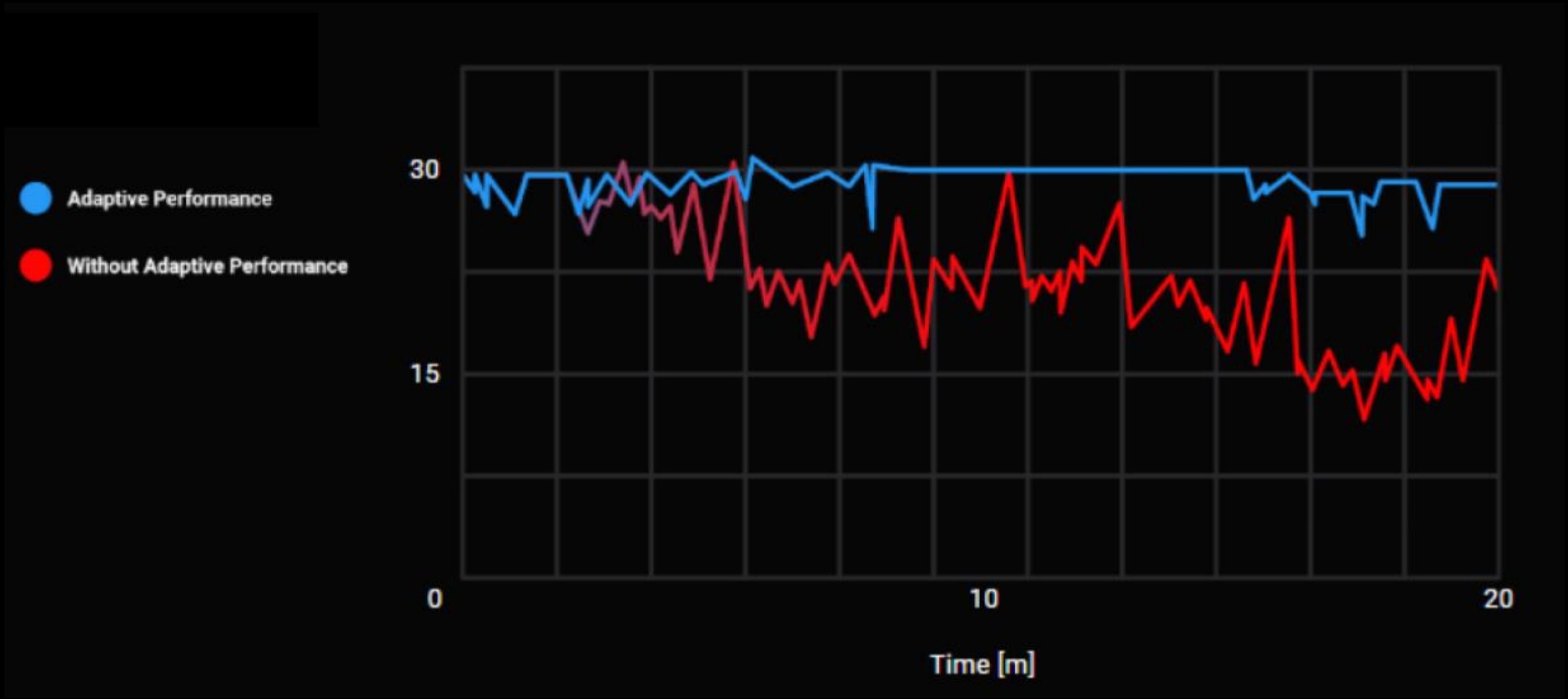


Adaptive Performance

- Start with Low CPU/GPU levels (Menu->Level)
  - Increase levels for CPU or GPU bottleneck respectively
    - Keeps energy consumption low
- Not hitting target frame rate
  - Decrease the LOD bias
    - Decreases GFX load (triangles, memory, draw calls)
- Decrease target framerate when close to throttling

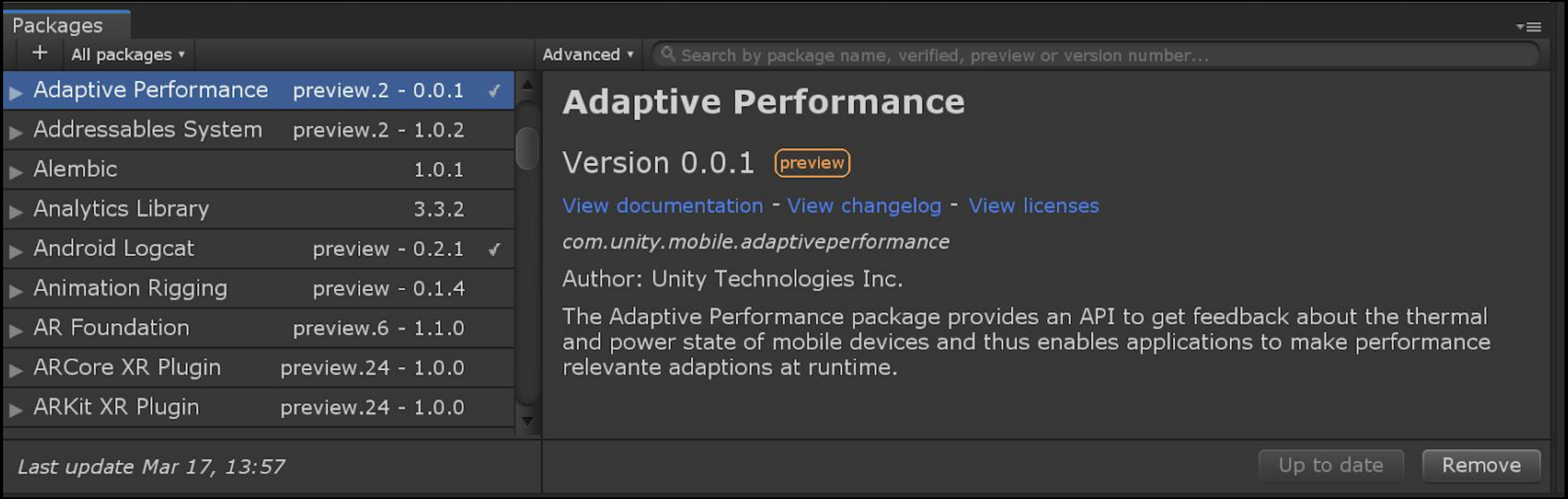

29.7 FPS
29.8 FPS

FrameRate


● Adaptive Performance
● Without Adaptive Performance
30
15
0
10
20
Time [m]

Adaptive Performance: How it works

- Install the package


Packages
+
All packages ▾
Advanced ▾
Search by package name, verified, preview or version number...
Up to date
Remove

▶ Adaptive Performance	preview.2 - 0.0.1	✓
▶ Addressables System	preview.2 - 1.0.2	
▶ Alembic	1.0.1	
▶ Analytics Library	3.3.2	
▶ Android Logcat	preview - 0.2.1	✓
▶ Animation Rigging	preview - 0.1.4	
▶ AR Foundation	preview.6 - 1.1.0	
▶ ARCore XR Plugin	preview.24 - 1.0.0	
▶ ARKit XR Plugin	preview.24 - 1.0.0	

Last update Mar 17, 13:57

Adaptive Performance

Version 0.0.1 preview

View documentation - View changelog - View licenses

com.unity.mobile.adaptiveperformance

Author: Unity Technologies Inc.

The Adaptive Performance package provides an API to get feedback about the thermal and power state of mobile devices and thus enables applications to make performance relevante adaptions at runtime.

Adaptive Performance: How it works

- Register to OnThermalEvent, Set initial target level

```
public class AdaptivePerformanceController : MonoBehaviour
{
    IAdaptivePerformance ap = null;

    ap.ThermalEvent += OnThermalEvent;
    ap.cpuLevel = 0;
    ap.gpuLevel = 0;

    ....
}

```

Adaptive Performance: How it works

- Manage your quality settings when Thermal status changes

```
Int targetFps = 30;
switch (warningLevel)
{
    case PerformanceWarningLevel.NoWarning:
            targetFps = 30;
            break;
    case PerformanceWarningLevel.ThrottlingImminent:
            targetFps = 28;
            break;
    case PerformanceWarningLevel.Throttling:
            targetFps = 15;
            break;
}
```

```
Int targetFps = 30;
if (Application.targetFrameRate != targetFps)
{
    Application.targetFrameRate = targetFps;
    lastChangeTimeStamp = Time.time;
}
```

Adaptive Performance: How it works

- Subscribe to performance bottleneck events

```

switch (ap.performanceBottlenect)
{
   case PerformanceBottlenect.GPU:
          if ((preferRaiseLevels || !CanLowerLOD()) && ap.gpuLevel < ap.maxGpuPerformanceLevel)
              RaiseGpuLevel(timestamp);
          else
              LowerLOD(timestamp);
          break;

   case PerformanceBottlenect.CPU:
          if ((preferRaiseLevels || !CanLowerLOD()) && ap.cpuLevel < ap.maxCpuPerformanceLevel)
              RaiseCpuLevel(timestamp);
          else
              LowerLOD(timestamp);
          break;

    …
```

Adaptive Performance: How it works

- Subscribe to performance bottleneck events

```
case PerformanceBottlenect.TargetFrameRate:
    if (timestamp – targetFrameRateHitTimestamp > 5.0f)
    {
        float bounceAvoidanceThreshold = ap.warningLevel != PerformanceWarningLevel.NoWarning ? 15.0f : 60.0f;
        bool allowRaiseLOD = (ap.warningLevel == PerformanceWarningLevel.NoWarning);
        if (allowRaiseLOD && preferRaiseLOD && CanRaiseLOD())
            RaiseLOD(timestamp);
        else if (ap.cpuLevel > 0 && ap.cpuLevel > ap.gpuLevel && timestamp – lastCpuLevelRaiseTimeStamp > bounceAvoidanceThreshold)
            LowerCpuLevel(timestamp);
        else if (ap.cpuLevel > 0 && timestamp – lastCpuLevelRaiseTimeStamp > bounceAvoidanceThreshold)
            LowerGpuLevel(timestamp);
        else if (allowRaiseLOD)
            RaiseLOD(timestamp);
    }
```

Currently we already release Adaptive Performance with Unity.

Vulkan Case Sharing

FINAL FANTASY XV POCKET EDITION

Square Enix / summertime studio

- GameEngine Optimize
- Contents Optimize
- Trouble Support
- Pipeline Barrier Optimize
- Pending RenderPass
- Primitive Buffer

Before	After
42 FPS	60 FPS

S8 Mali – Within Sustainable Power

FINAL FANTASY XV POCKET EDITION

Square Enix / summertime studio

- GameEngine Optimize
- Contents Optimize
- Trouble Support
- Pipeline Barrier Optimize
- Pending RenderPass
- Primitive Buffer

Before	After
42 FPS	60 FPS

S8 Mali – Within Sustainable Power

FINAL FANTASY XV POCKET EDITION

Square Enix / summertime studio

- GameEngine Optimize
- Contents Optimize
- Trouble Support
- Pipeline Barrier Optimize
- Pending RenderPass
- Primitive Buffer

Before	After
42 FPS	60 FPS

S8 Mali — Within Sustainable Power

FINAL FANTASY XV POCKET EDITION

Square Enix / summertime studio

- GameEngine Optimize
- Contents Optimize
- Trouble Support
- Pipeline Barrier Optimize
- Pending RenderPass
- Primitive Buffer

Before	After
42 FPS	60 FPS

S8 Mali — Within Sustainable Power

Honor of Kings

Tencent Timi Studio – L1

- Single Scratch Buffer
- Low Priority Destroy Thread
- Batching UpdateDescriptorSet
- RenderPass Load/Store Optimize
- Shader Module Cache

Before	After
41 FPS	53 FPS

S7 Adreno – Within Sustainable Power

Honor of Kings

Tencent Timi Studio – L1

- Single Scratch Buffer
- Low Priority Destroy Thread
- Batching UpdateDescriptorSet
- RenderPass Load/Store Optimize
- Shader Module Cache

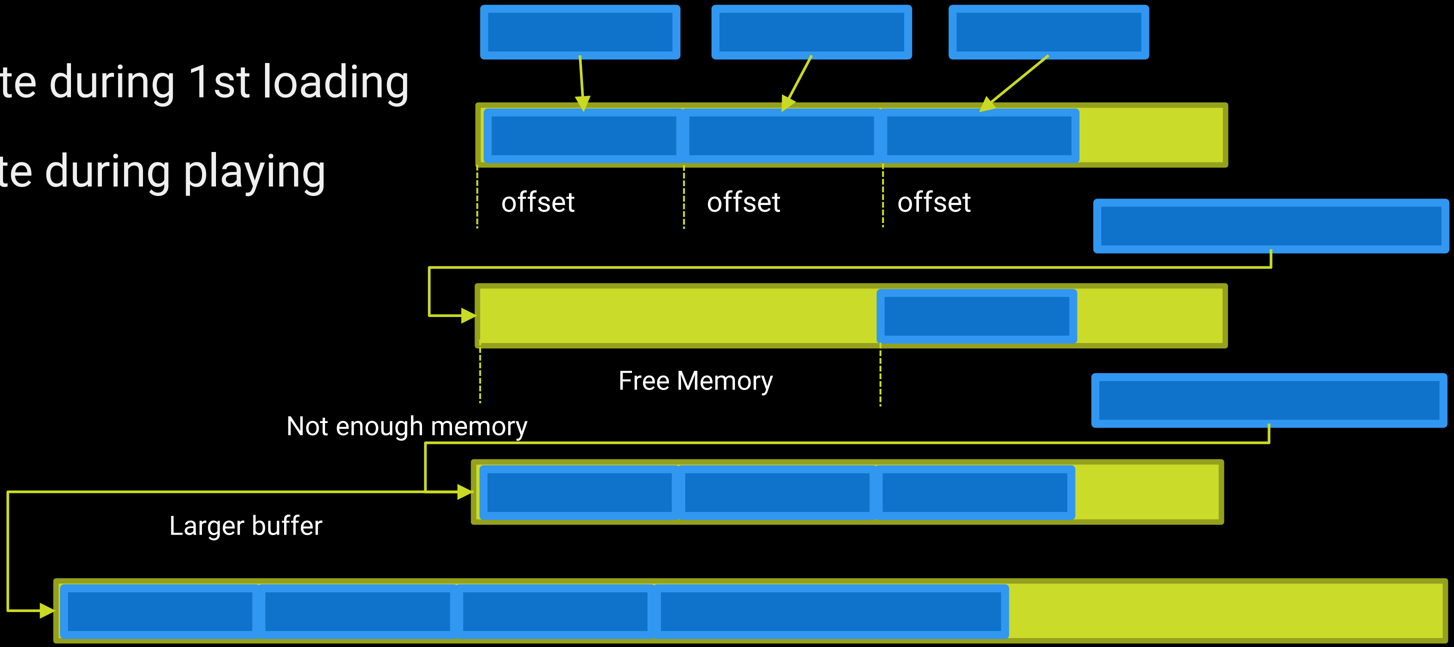
Before	After
41 FPS	53 FPS

S7 Adreno – Within Sustainable Power

Honor of Kings

Tencent Timi Studio – L1

- Only recreate during 1st loading
- 100% hit rate during playing



Honor of Kings

Tencent Timi Studio – L1

- Single Scratch Buffer
- Low Priority Destroy Thread
- Batching UpdateDescriptorSet
- RenderPass Load/Store Optimize
- Shader Module Cache

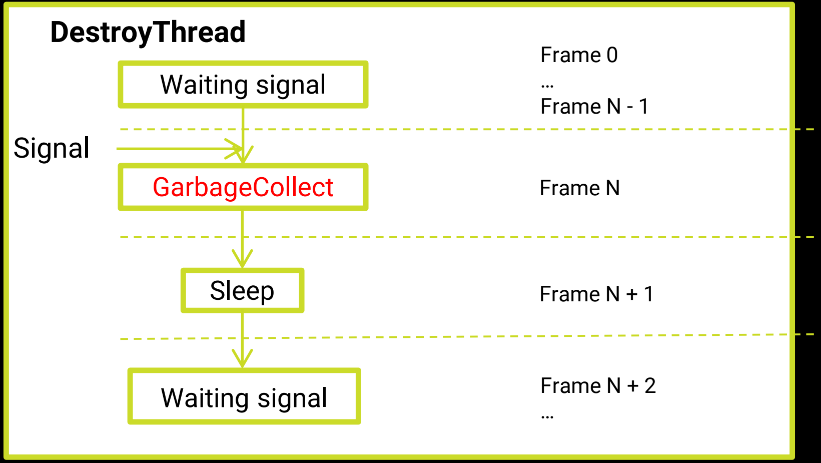
Before	After
41 FPS	53 FPS

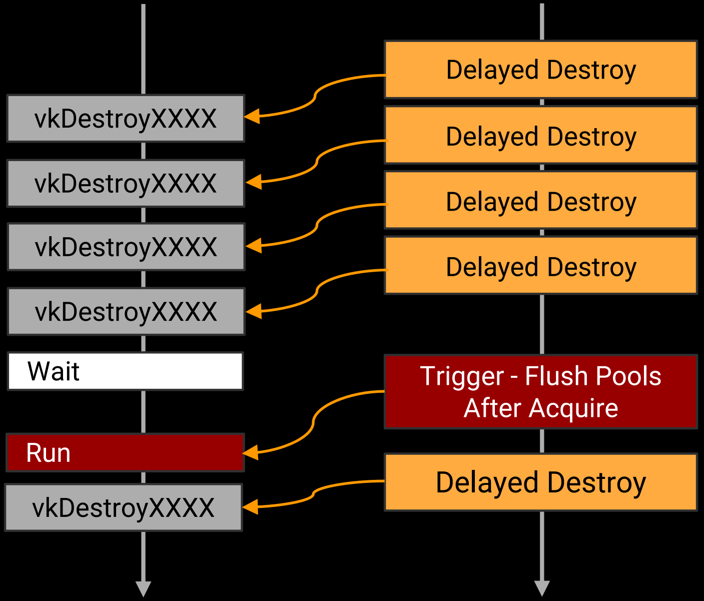
S7 Adreno – Within Sustainable Power

Honor of Kings

Tencent Timi Studio – L1

- Low Priority Destroy Thread





Honor of Kings

Tencent Timi Studio – L1

- Single Scratch Buffer
- Low Priority Destroy Thread
- Batching UpdateDescriptorSet
- RenderPass Load/Store Optimize
- Shader Module Cache

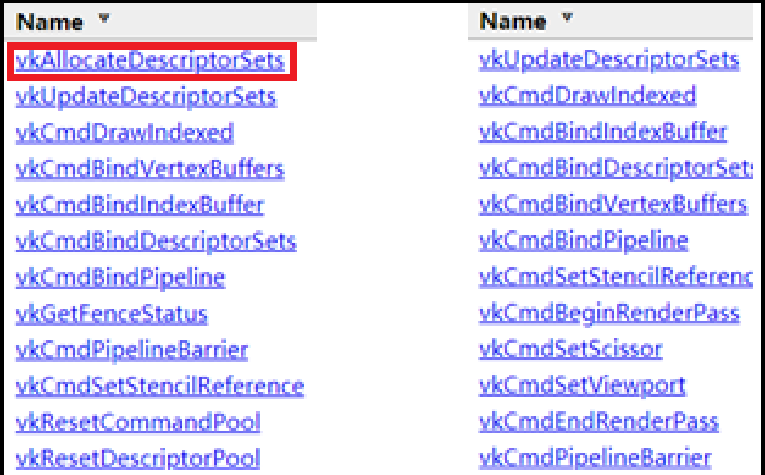
Before	After
41 FPS	53 FPS

S7 Adreno – Within Sustainable Power

Honor of Kings

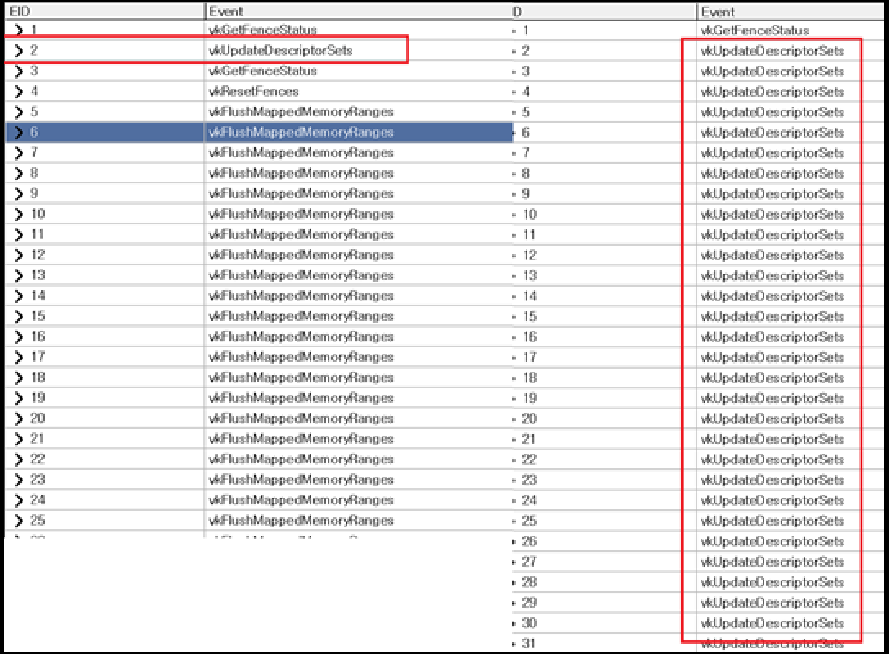
Tencent Timi Studio – L1

– Batching UpdateDescriptorSet



Name
vkAllocateDescriptorSets
vkUpdateDescriptorSets
vkCmdDrawIndexed
vkCmdBindVertexBuffers
vkCmdBindIndexBuffer
vkCmdBindDescriptorSets
vkCmdBindPipeline
vkGetFenceStatus
vkCmdPipelineBarrier
vkCmdSetStencilReference
vkResetCommandPool
vkResetDescriptorPool

Name
vkUpdateDescriptorSets
vkCmdDrawIndexed
vkCmdBindIndexBuffer
vkCmdBindDescriptorSet
vkCmdBindVertexBuffers
vkCmdBindPipeline
vkCmdSetStencilReferenc
vkCmdBeginRenderPass
vkCmdSetScissor
vkCmdSetViewport
vkCmdEndRenderPass
vkCmdPipelineBarrier



EID	Event	D	Event
1	vkGetFenceStatus	1	vkGetFenceStatus
2	vkUpdateDescriptorSets	2	vkUpdateDescriptorSets
3	vkGetFenceStatus	3	vkUpdateDescriptorSets
4	vkResetFences	4	vkUpdateDescriptorSets
5	vkFlushMappedMemoryRanges	5	vkUpdateDescriptorSets
6	vkFlushMappedMemoryRanges	6	vkUpdateDescriptorSets
7	vkFlushMappedMemoryRanges	7	vkUpdateDescriptorSets
8	vkFlushMappedMemoryRanges	8	vkUpdateDescriptorSets
9	vkFlushMappedMemoryRanges	9	vkUpdateDescriptorSets
10	vkFlushMappedMemoryRanges	10	vkUpdateDescriptorSets
11	vkFlushMappedMemoryRanges	11	vkUpdateDescriptorSets
12	vkFlushMappedMemoryRanges	12	vkUpdateDescriptorSets
13	vkFlushMappedMemoryRanges	13	vkUpdateDescriptorSets
14	vkFlushMappedMemoryRanges	14	vkUpdateDescriptorSets
15	vkFlushMappedMemoryRanges	15	vkUpdateDescriptorSets
16	vkFlushMappedMemoryRanges	16	vkUpdateDescriptorSets
17	vkFlushMappedMemoryRanges	17	vkUpdateDescriptorSets
18	vkFlushMappedMemoryRanges	18	vkUpdateDescriptorSets
19	vkFlushMappedMemoryRanges	19	vkUpdateDescriptorSets
20	vkFlushMappedMemoryRanges	20	vkUpdateDescriptorSets
21	vkFlushMappedMemoryRanges	21	vkUpdateDescriptorSets
22	vkFlushMappedMemoryRanges	22	vkUpdateDescriptorSets
23	vkFlushMappedMemoryRanges	23	vkUpdateDescriptorSets
24	vkFlushMappedMemoryRanges	24	vkUpdateDescriptorSets
25	vkFlushMappedMemoryRanges	25	vkUpdateDescriptorSets
		26	vkUpdateDescriptorSets
		27	vkUpdateDescriptorSets
		28	vkUpdateDescriptorSets
		29	vkUpdateDescriptorSets
		30	vkUpdateDescriptorSets
		31	vkUpdateDescriptorSets

Honor of Kings – The King's Expedition

Tencent Timi Studio – L1

- Threaded queue present
- VB, IB binding optimization
- Direct buffer access
- Cache optimization
- Async texture load
- Single buffer for image upload

Before	After
49 FPS	59 FPS

Note8 Mali – Within Sustainable Power

Honor of Kings – The King’s Expedition

Tencent Timi Studio – L1

- Threaded queue present
- VB, IB binding optimization
- Direct buffer access
- Cache optimization
- Async texture load
- Single buffer for image upload

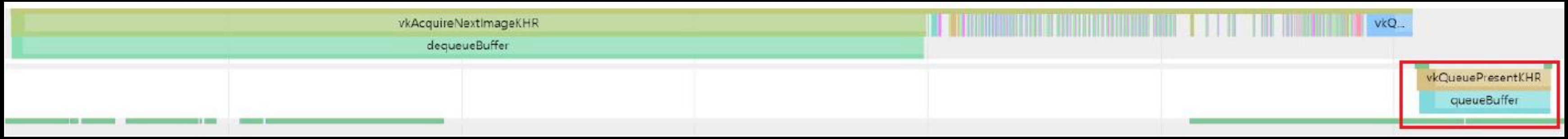
Before	After
49 FPS	59 FPS

Note8 Mali – Within Sustainable Power

Honor of Kings – The King’s Expedition

Tencent Timi Studio – L1

- Threaded queue present



Honor of Kings – The King's Expedition

Tencent Timi Studio – L1

- Threaded queue present
- VB, IB binding optimization
- Direct buffer access
- Cache optimization
- Async texture load
- Single buffer for image upload

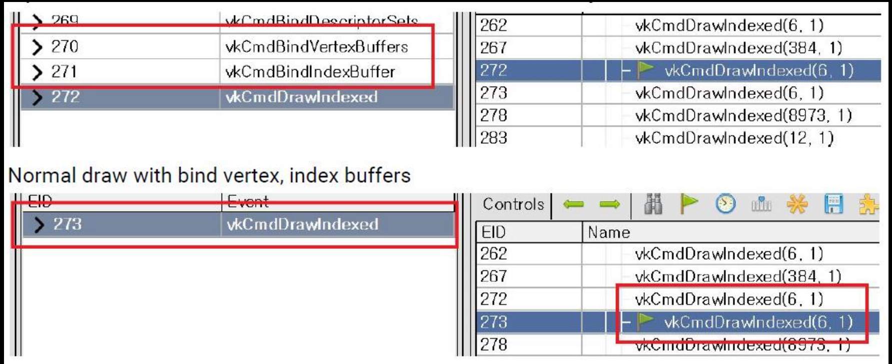
Before	After
49 FPS	59 FPS

Note8 Mali – Within Sustainable Power

Honor of Kings – The King’s Expedition

Tencent Timi Studio – L1

Skip binding buffers, just draw with bound buffers from previous draw call.



269	vkCmdBindDescriptorSets
270	vkCmdBindVertexBuffers
271	vkCmdBindIndexBuffer
272	vkCmdDrawIndexed

262	vkCmdDrawIndexed(6, 1)
267	vkCmdDrawIndexed(384, 1)
272	vkCmdDrawIndexed(6, 1)
273	vkCmdDrawIndexed(6, 1)
278	vkCmdDrawIndexed(8973, 1)
283	vkCmdDrawIndexed(12, 1)

Normal draw with bind vertex, index buffers

EID	Event
273	vkCmdDrawIndexed

EID	Name
262	vkCmdDrawIndexed(6, 1)
267	vkCmdDrawIndexed(384, 1)
272	vkCmdDrawIndexed(6, 1)
273	vkCmdDrawIndexed(6, 1)
278	vkCmdDrawIndexed(8973, 1)

Honor of Kings – The King's Expedition

Tencent Timi Studio – L1

- Threaded queue present
- VB, IB binding optimization
- Direct buffer access
- Cache optimization
- Async texture load
- Single buffer for image upload

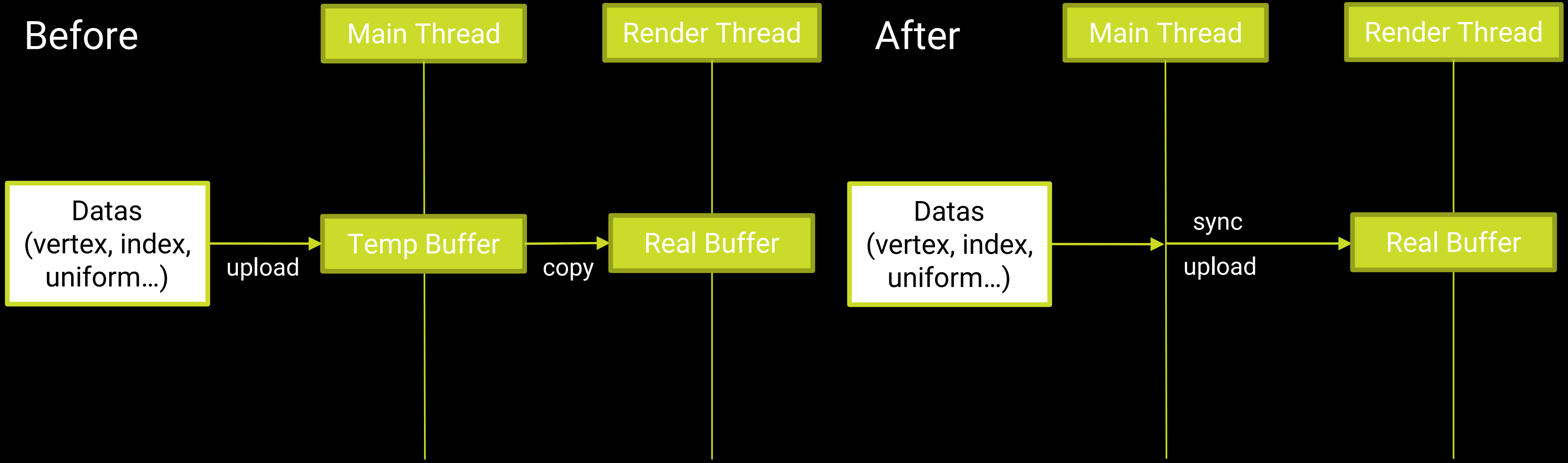
Before	After
49 FPS	59 FPS

Note8 Mali – Within Sustainable Power

Honor of Kings – The King’s Expedition

Tencent Timi Studio – L1

- Direct Buffer Access



Honor of Kings – The King’s Expedition

Tencent Timi Studio – L1

- Threaded queue present
- VB, IB binding optimization
- Direct buffer access
- Cache optimization
- Async texture load
- Single buffer for image upload

Before	After
49 FPS	59 FPS

Note8 Mali – Within Sustainable Power

Hundred Soul

HOUND 13

- Bloom Optimize
- RenderPass Load/Store Optimize
- Paging Allocate Buffer
- Batching UpdateDescriptorSet
- Shader Module Cache

Before	After
41 FPS	52 FPS

Note9 Adreno – Within Sustainable Power

Hundred Soul

HOUND 13

- Bloom Optimize
- RenderPass Load/Store Optimize
- Paging Allocate Buffer
- Batching UpdateDescriptorSet
- Shader Module Cache

Before	After
41 FPS	52 FPS

Note9 Adreno – Within Sustainable Power

Hundred Soul

HOUND 13

- Bloom Optimize
- RenderPass Load/Store Optimize
- Paging Allocate Buffer
- Batching UpdateDescriptorSet
- Shader Module Cache

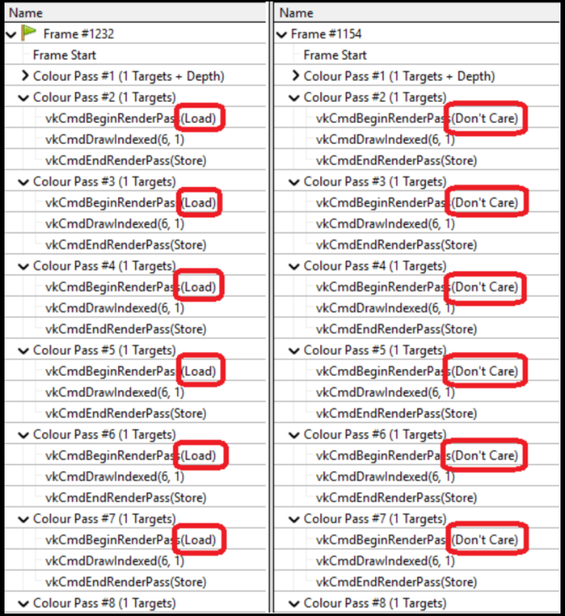
Before	After
41 FPS	52 FPS

Note9 Adreno – Within Sustainable Power

Hundred Soul

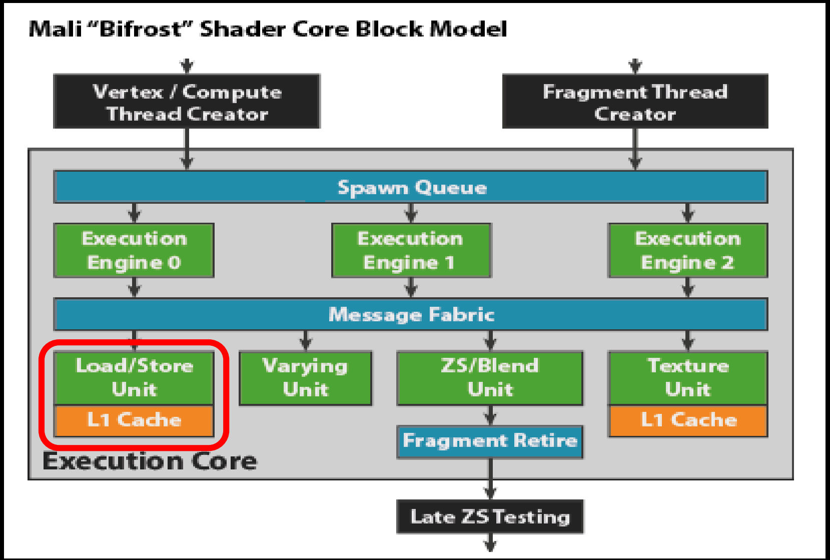
HOUND 13

- RenderPass Load/Store Optimize



```
Name
⌄ Frame #1232
    Frame Start
  › Colour Pass #1 (1 Targets + Depth)
  ⌄ Colour Pass #2 (1 Targets)
      vkCmdBeginRenderPass(Load)
      vkCmdDrawIndexed(6, 1)
      vkCmdEndRenderPass(Store)
  ⌄ Colour Pass #3 (1 Targets)
      vkCmdBeginRenderPass(Load)
      vkCmdDrawIndexed(6, 1)
      vkCmdEndRenderPass(Store)
  ⌄ Colour Pass #4 (1 Targets)
      vkCmdBeginRenderPass(Load)
      vkCmdDrawIndexed(6, 1)
      vkCmdEndRenderPass(Store)
  ⌄ Colour Pass #5 (1 Targets)
      vkCmdBeginRenderPass(Load)
      vkCmdDrawIndexed(6, 1)
      vkCmdEndRenderPass(Store)
  ⌄ Colour Pass #6 (1 Targets)
      vkCmdBeginRenderPass(Load)
      vkCmdDrawIndexed(6, 1)
      vkCmdEndRenderPass(Store)
  ⌄ Colour Pass #7 (1 Targets)
      vkCmdBeginRenderPass(Load)
      vkCmdDrawIndexed(6, 1)
      vkCmdEndRenderPass(Store)
  ⌄ Colour Pass #8 (1 Targets)
```

```
Name
⌄ Frame #1154
    Frame Start
  › Colour Pass #1 (1 Targets + Depth)
  ⌄ Colour Pass #2 (1 Targets)
      vkCmdBeginRenderPass(Don't Care)
      vkCmdDrawIndexed(6, 1)
      vkCmdEndRenderPass(Store)
  ⌄ Colour Pass #3 (1 Targets)
      vkCmdBeginRenderPass(Don't Care)
      vkCmdDrawIndexed(6, 1)
      vkCmdEndRenderPass(Store)
  ⌄ Colour Pass #4 (1 Targets)
      vkCmdBeginRenderPass(Don't Care)
      vkCmdDrawIndexed(6, 1)
      vkCmdEndRenderPass(Store)
  ⌄ Colour Pass #5 (1 Targets)
      vkCmdBeginRenderPass(Don't Care)
      vkCmdDrawIndexed(6, 1)
      vkCmdEndRenderPass(Store)
  ⌄ Colour Pass #6 (1 Targets)
      vkCmdBeginRenderPass(Don't Care)
      vkCmdDrawIndexed(6, 1)
      vkCmdEndRenderPass(Store)
  ⌄ Colour Pass #7 (1 Targets)
      vkCmdBeginRenderPass(Don't Care)
      vkCmdDrawIndexed(6, 1)
      vkCmdEndRenderPass(Store)
  ⌄ Colour Pass #8 (1 Targets)
```



Mali "Bifrost" Shader Core Block Model

https://community.arm.com/developer/tools-software/graphics/b/blog/posts/the-mali-gpu-an-abstract-machine-part-4---the-bifrost-shader-core

Collaboration Titles

- 2016 Epic Games, ProtoStar Galaxy S7 Collaboration
- 2016 NetGames, HIT
- 2016 Super Evil MegaCorp, VainGlory
- 2017 433, HeroDC
- 2017 Netmarble, Lineage 2: Revolution
- 2017 Nexon, AxE
- 2017 XL Games, Archeage: Begins
- 2017 Action Square, Blade II
- 2017 Hound13, HundredSoul
- 2017 Square Enix, FinalFantasy XV Pocket Edition
- 2017 Croteam, Talos Principle
- 2017 Roblox Corporation, Roblox
- 2017 Doragon Entertainment, Danmaku Unlimited 3
- 2017 GameLoft, Asphalt 8
- 2017 Cornfox&Bros, Oceanhorn: Monster of Uncharted Seas
- 2017 Deep Silver, Galaxy On Fire 3: Manticore!
- 2017 Digital Legends Entertainment, Afterpulse
- 2017 First Touch Games, Score! Hero / Dream League Soccer
- 2018 Tencent, Honor of Kings
- 2018 Pearl Abyss, BlackDesert Mobile
- 2018 Epic Games, Fortnite Battle Royale
- 2018 Amazon Lumberyard, Bistro
- 2019 Moai Games, TRAHA
- 2019 PUBG corporation / Tencent, PUBG MOBILE
- 2019 Tencent, QQ Speed

Q&A

Please contact:

✉ gamedev@samsung.com

Galaxy GameDev

unity

Unite Shanghai 2019