GDC | March 20-24, 2023
San Francisco, CA

MODERNIZING RENDERING AT SUPERCELL

Timo Heinäpurola

#GDC23

AGENDA

- Short history of rendering at Supercell
- The process of rewriting our rendering core
- Overview of the new renderer
- Learnings

SPEAKER INTRODUCTION

- Timo Heinäpurola
- Supercellian since March 2020
  - Focused on building our rendering capabilities
- Background
  - IT Industry
  - Bugbear Entertainment
  - Next Games
  - Reforged Studios

SUPERCELL

- Founded in 2010
- 420 people globally
- Known for hit games

HAY DAY

CLASH of CLANS

BOOM BEACH

CLASH ROYALE

BRAWL STARS

FROM FLASH TO NATIVE C++

THE AGE OF FLASH

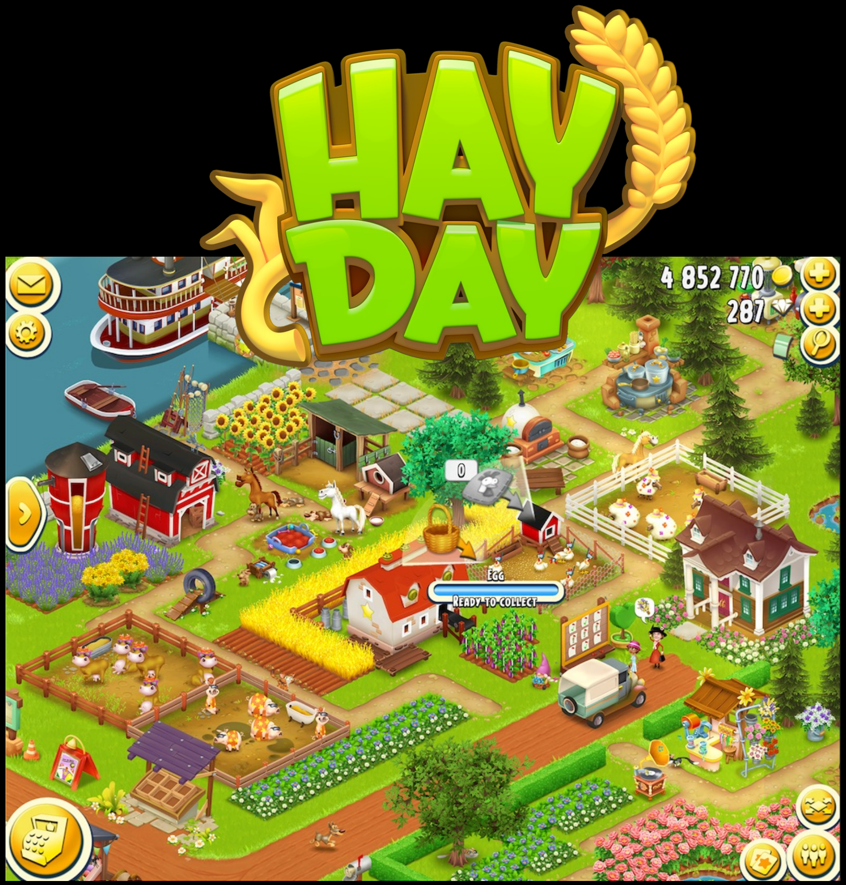
- First game Gunshine.net was made with Flash
- Pivot to tablet oriented mobile
  - Needed something native for iOS and Android
  - Wanted to continue authoring in Flash

THE BIRTH OF TITAN

- Custom Flash to in-house pipeline
- Rendering with in-house system
- Part of Titan
  - Not just rendering
  - Networking, push notifications, compression etc.
  - Maintained by the Tools and Technology team

THE GROWING NEED FOR 3D

- Early games were all isometric with fake pre-rendered 3D



THE GROWING NEED FOR 3D

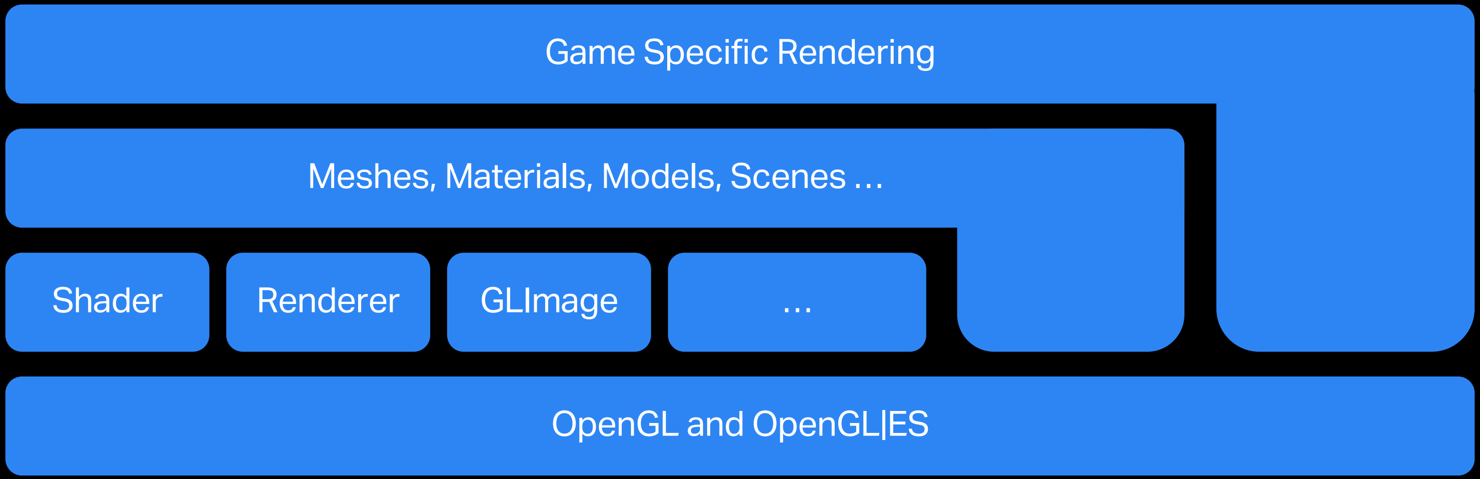
- Early games were all isometric with fake pre-rendered 3D
- Everdale (beta game) and Brawl Stars highlighted the potential of 3D

NEW WINDS BLOWING FOR 3D

APPLE DROPS THE BOMB

- Apple deprecates OpenGL|ES
- We heavily relied on OpenGL|ES
  - Some lightweight layers on top
  - Many games directly using the underlying API

LEGACY RENDERING ARCHITECTURE



FUTURE PROOFING OUR RENDERING

- We saw this as an opportunity
- Legacy render path was holding us back
- We needed a proper abstraction layer

RISKY MOVE?

- Rewriting the renderer of billion dollar games might be considered risky…
- …we think big changes are often necessary to keep us competitive!
- Moving games to new Titan features helps us …
  - improve our games with every update
  - keep the Tools and Technology team small
  - to share learnings and technology between games

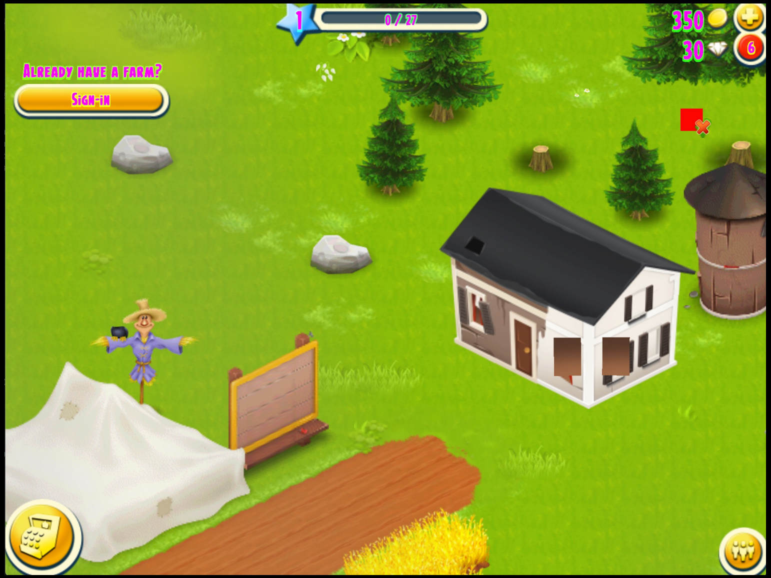
ENTER THOR

THE BIRTH OF THOR

- THOR abstracts Metal, OpenGL, OpenGL|ES and Vulkan
- When I joined there was already a skeleton of what was to become THOR
  - Architecture
  - Basic support for OpenGL, OpenGL|ES and Metal

THE FIRST STEPS OF THE GOD OF THUNDER

- During the spring of 2020 I worked on THOR in isolation
- Hay Day first steps before summer holidays
- Hay Day chosen due to limited rendering feature set



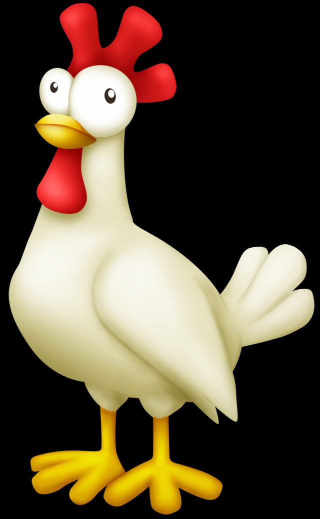
First (buggy) screenshot of Hay Day with THOR

PREPARING FOR LIGHTNING STRIKES

- We set up automated testing for Hay Day
  - Predefined scenarios for smoke testing
  - Video recording
  - Log capturing
  - Executed in Google Firebase
- Helped us find multiple issues in rendering before launch
  - Allowed QA to focus on the more tricky issues

THE HAY DAYS OF THOR

- During fall of 2020 I focused on improving and stabilizing THOR
- Working with the game team was easy
- Hay Day update using THOR was released on the 23rd of November 2020



THE REST OF THE LIVE GAMES

CLASH OF CLANS	was the second game and also the first game to use any 3D	12th of April 2021
CLASH ROYALE	was started in parallel with CoC, but took longer to ship due to the game's release schedule	25th of Oct 2021
BRAWL STARS	was perhaps the most complicated one due to its heavy use of 3D	16th of Dec 2021
BOOM BEACH	took the longest to go live due to their release schedule, but didn't provide any major surprises	2nd of Nov 2022

...THAT WAS JUST THE BEGINNING

- The development of THOR has continued non-stop
- Now building on top of THOR
  - New rendering techniques
  - Easier sharing
  - Better tooling

THOR

THOR DESIGN GOALS

- Support multiple graphics APIs
- Powerful and efficient, yet easy to use
- Futureproof

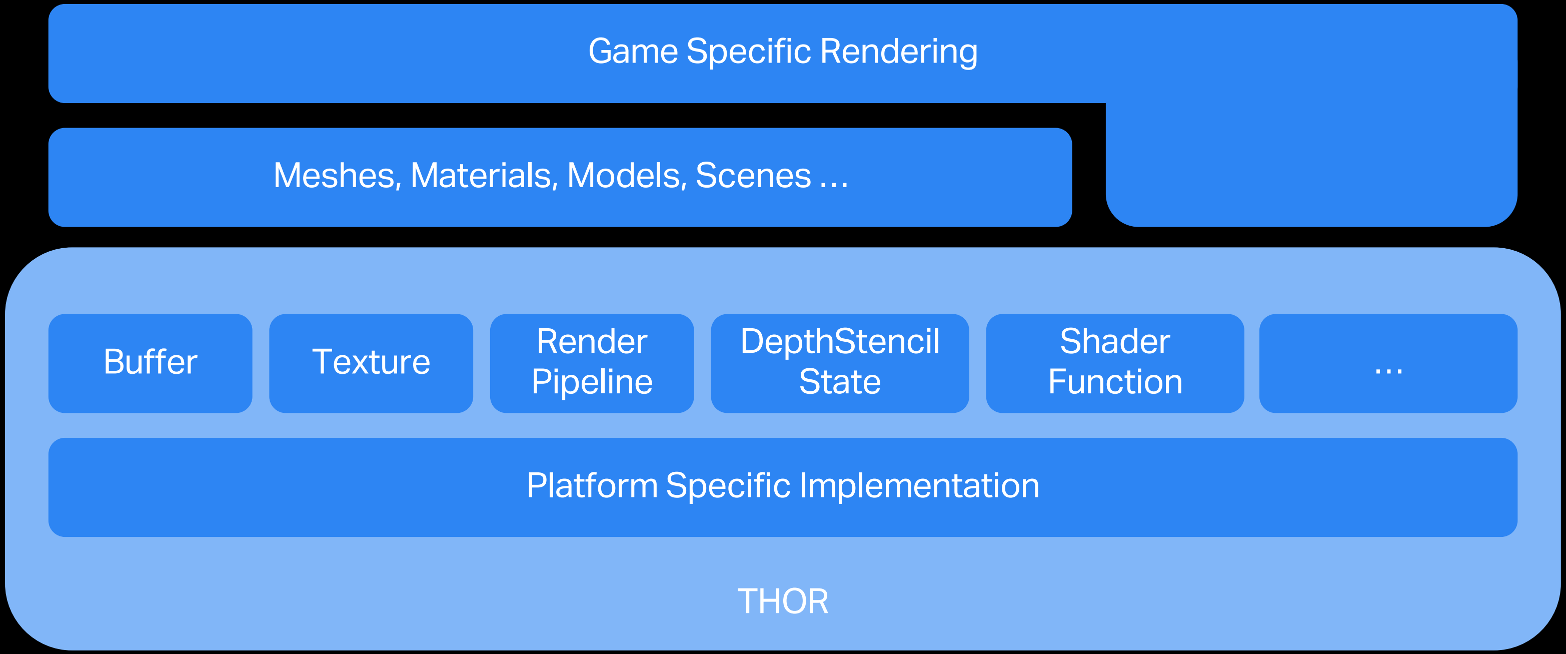
THOR IN A NUTSHELL

- Currently supports Metal, OpenGL, OpenGL|ES and Vulkan
- Architectured around the core concepts of Metal
- THOR is built to be multithreaded

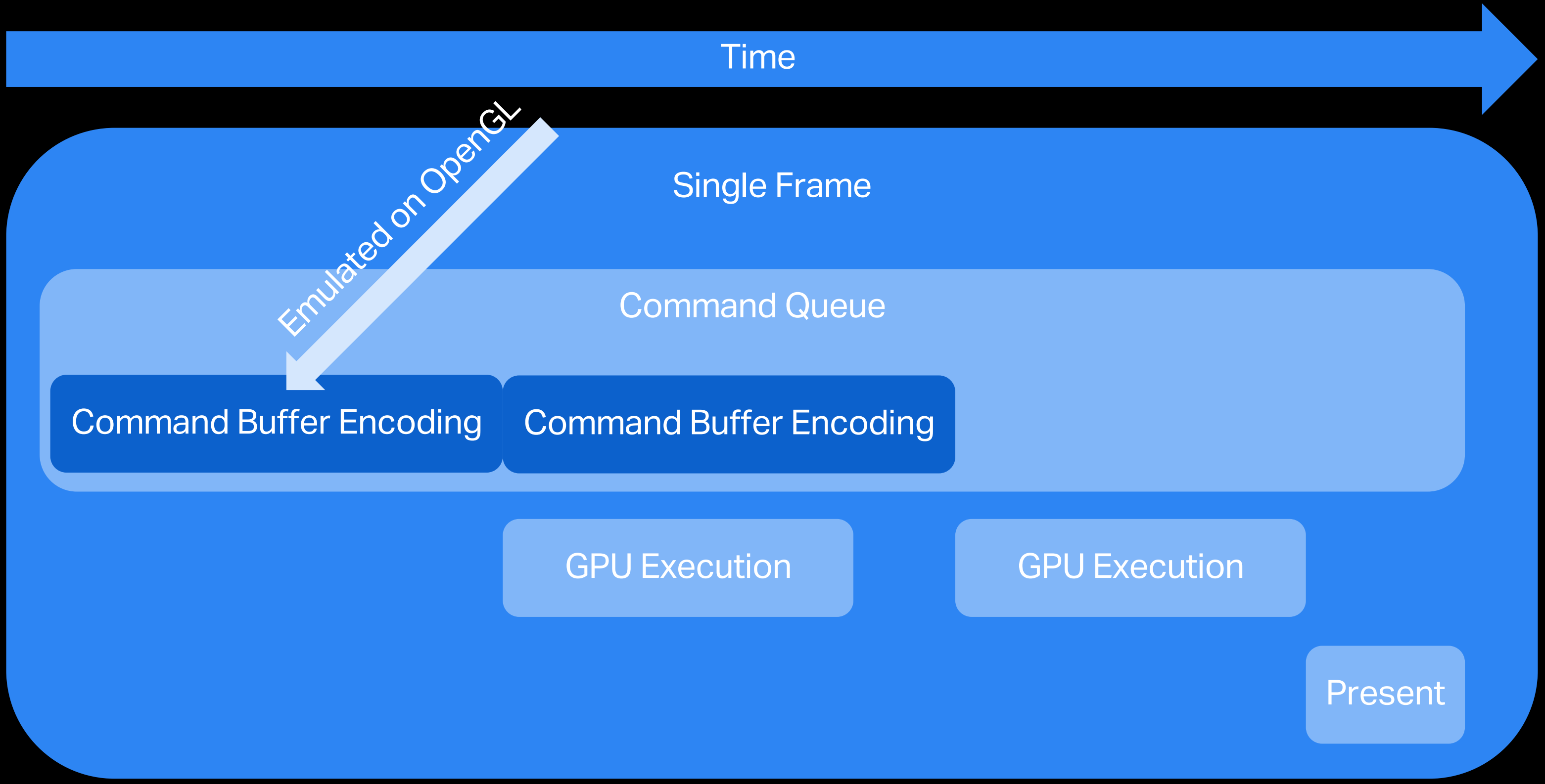
DESIGN CHALLENGES

- Must support all devices that the legacy path supported
  - *Supercell's mission is to create great games that as many people as possible play for years and that are remembered forever.*
- Added abstractions should be efficient
- Must coexist with the legacy path

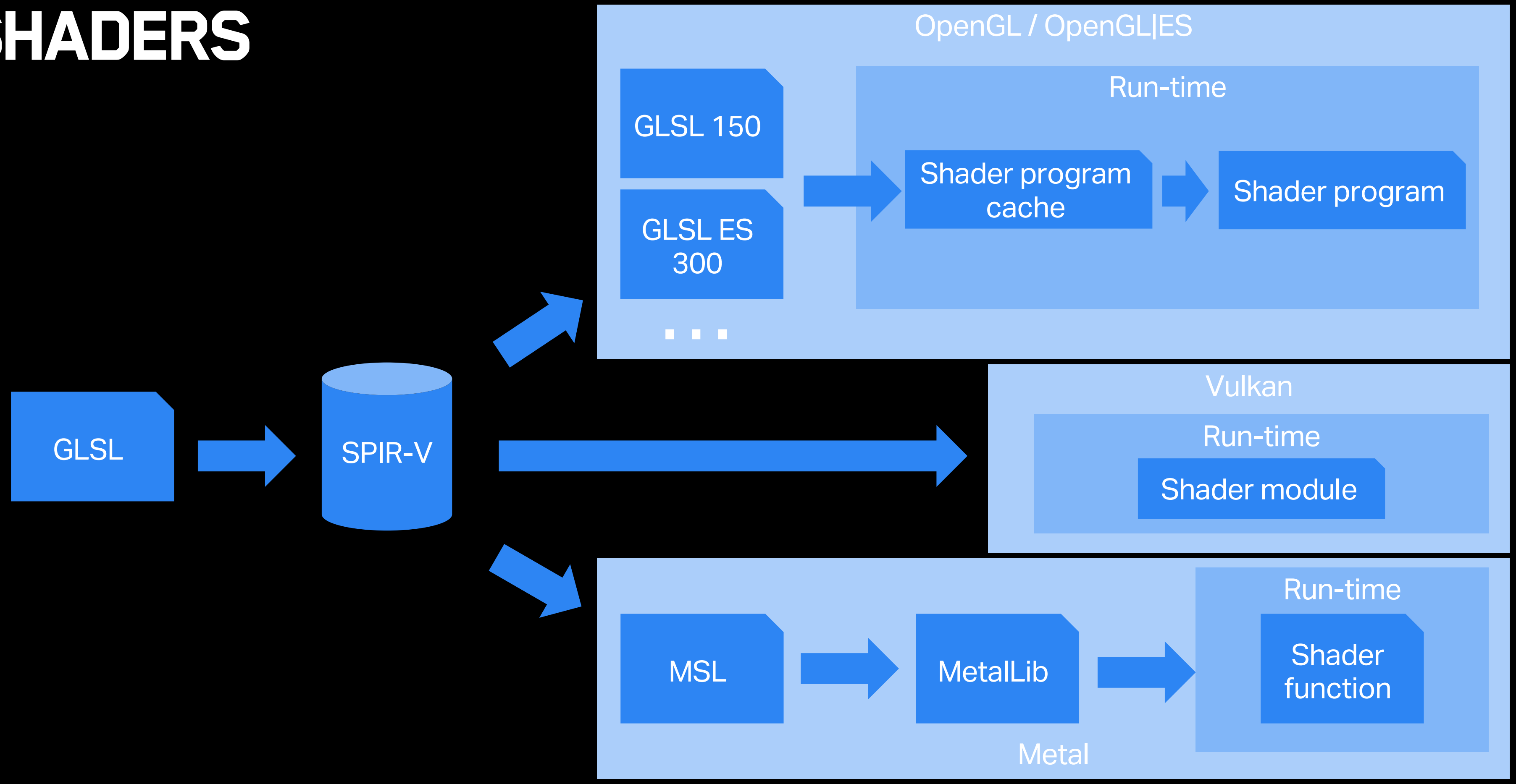
CORE ARCHITECTURE



CORE ARCHITECTURE



SHADERS



OpenGL / OpenGL|ES
GLSL 150
GLSL ES 300
. . .
Run-time
Shader program cache
Shader program
GLSL
SPIR-V
Vulkan
Run-time
Shader module
MSL
MetalLib
Run-time
Shader function
Metal

EXTENDING THOR

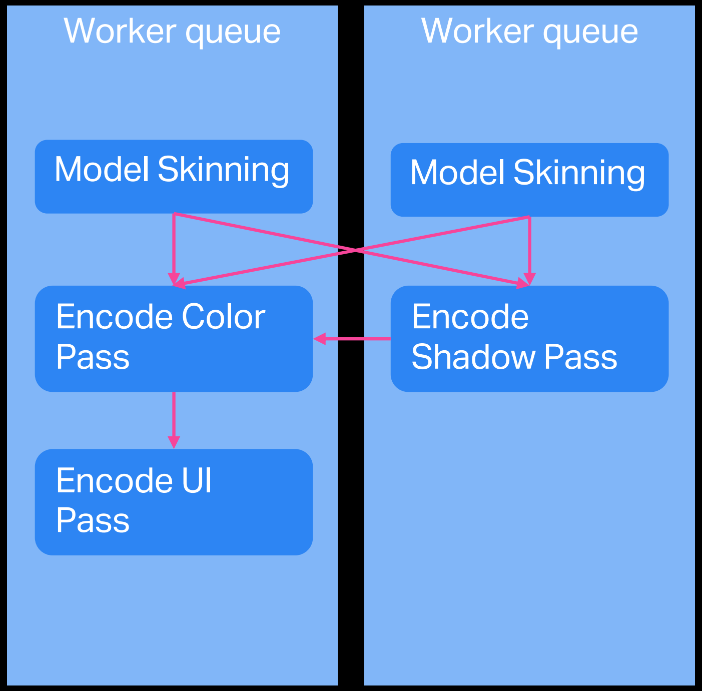
- Some features of THOR not available on all platforms
- Games can check feature availability
- Some functionality is emulated
  - Allows for uniform base implementation
  - Example: uniform buffers

LOOKING INTO THE FUTURE

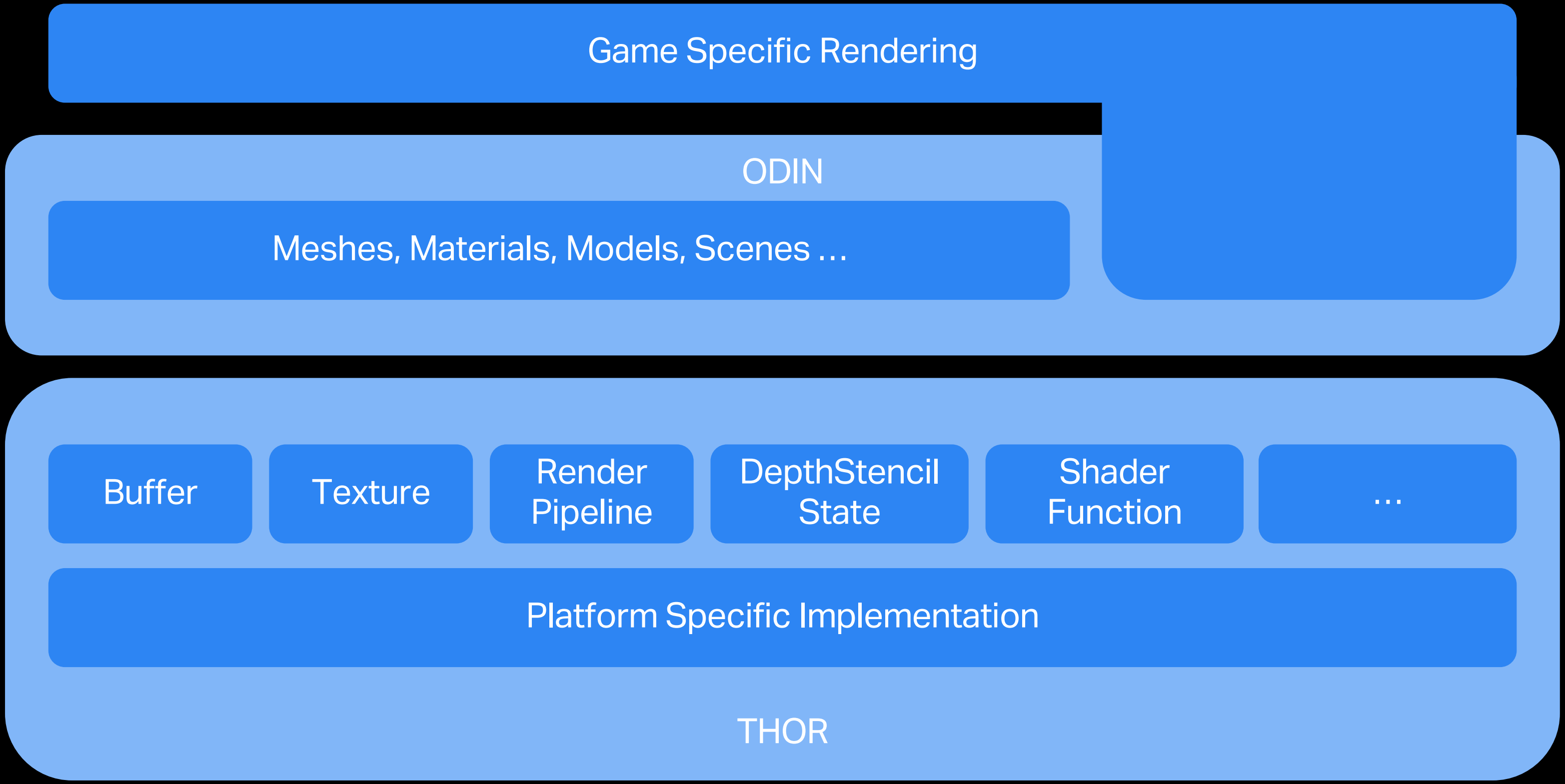
- Focus has shifted to building on top of THOR
- Examples
  - Adding features to THOR (compute shaders, mesh shaders, etc.)
  - Render graph
  - ODIN

RENDER GRAPH

- Render task management system
- Each task implements a piece of the frame rendering
  - Generic tasks: Like skinning
  - Render passes
  - Executed on worker queues
- Synchronization based on inputs and outputs



ODIN


Game Specific Rendering
ODIN
Meshes, Materials, Models, Scenes …
Buffer
Texture
Render Pipeline
DepthStencil State
Shader Function
…
Platform Specific Implementation
THOR

LEARNINGS FROM ALONG THE WAY

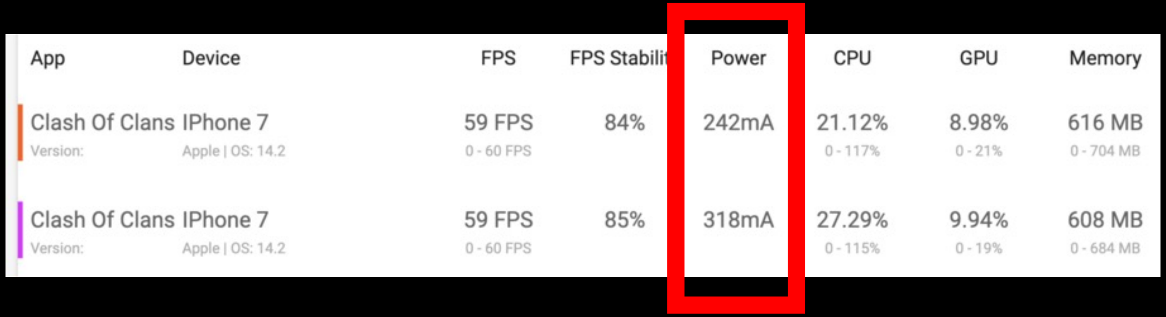
QUALITY!

- Our games are played on a wide range of devices (250 million MAU)
  - Testing everything manually is impossible
  - Automated smoke testing takes pressure off from QA
- Proper self-review of implementation before commit
- Respect your QA! They are superheroes that help you sleep at night ☺

ENERGY EFFICIENCY

- Batteries and heat dissipation limiting factors on mobile
- Rendering needs to pay extra attention to this
- Important for fulling out mission!
- Examples
  - Use cache efficient data layouts on hot memory
  - Favour using multiple cores
  - Aim to use low power cores
  - Write sensible code from the get go

THOR METAL BACKEND - ENERGY SAVINGS

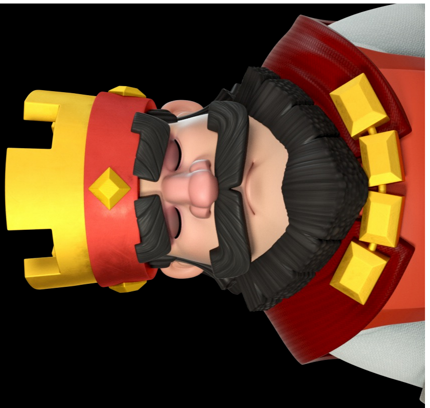


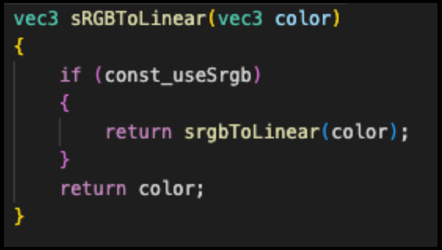
App	Device	FPS	FPS Stabilit	Power	CPU	GPU	Memory
Clash Of Clans Version:	IPhone 7 Apple | OS: 14.2	59 FPS 0 - 60 FPS	84%	242mA	21.12% 0 - 117%	8.98% 0 - 21%	616 MB 0 - 704 MB
Clash Of Clans Version:	IPhone 7 Apple | OS: 14.2	59 FPS 0 - 60 FPS	85%	318mA	27.29% 0 - 115%	9.94% 0 - 19%	608 MB 0 - 684 MB

WHEN DEVICES DON’T WORK AS EXPECTED

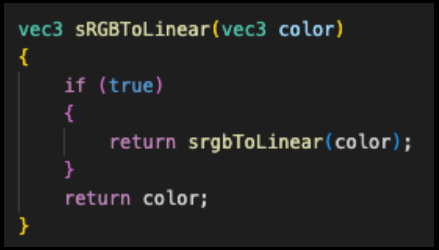
- Many Android devices have buggy drivers or hardware
- Situation not as bad as I had expected, but still significant
- Examples
  - Missing nested code blocks
  - Using outputs as temporaries
  - Uniform structures

MISSING CODE BLOCKS

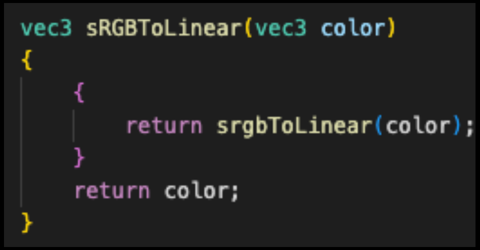




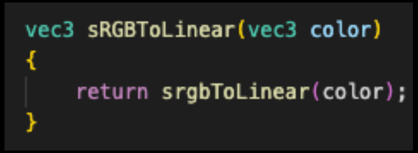
```
vec3 sRGBToLinear(vec3 color)
{
    if (const_useSrgb)
    {
        return srgbToLinear(color);
    }
    return color;
}
```



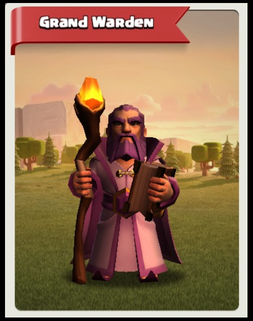
```
vec3 sRGBToLinear(vec3 color)
{
    if (true)
    {
        return srgbToLinear(color);
    }
    return color;
}
```

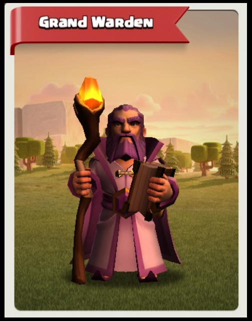


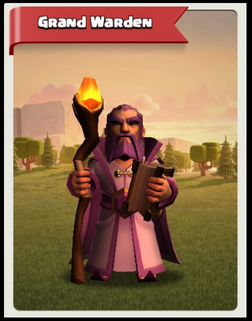
```
vec3 sRGBToLinear(vec3 color)
{
    {
        return srgbToLinear(color);
    }
    return color;
}
```



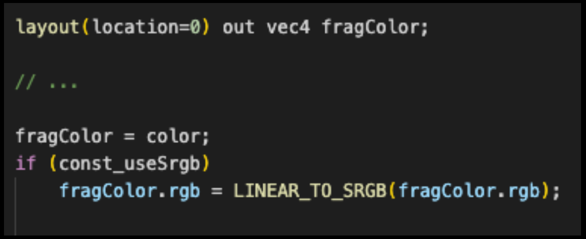
```
vec3 sRGBToLinear(vec3 color)
{
    return srgbToLinear(color);
}
```



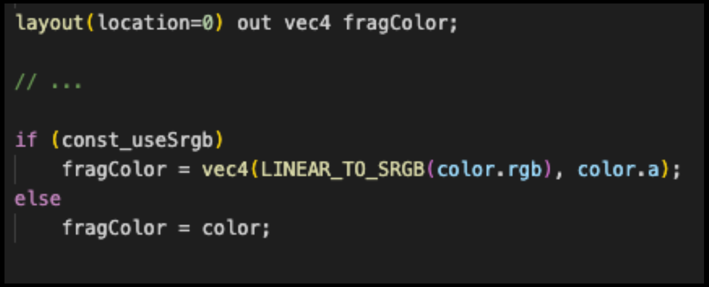




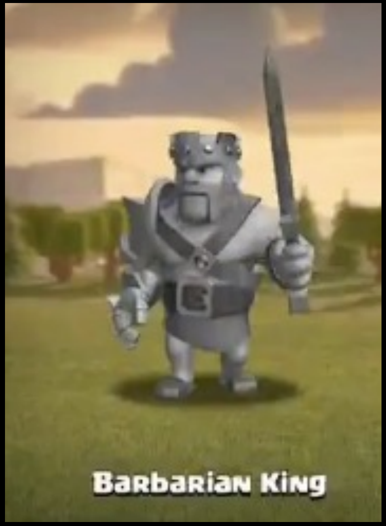
USING FRAGMENT OUTPUT AS TEMPORARY

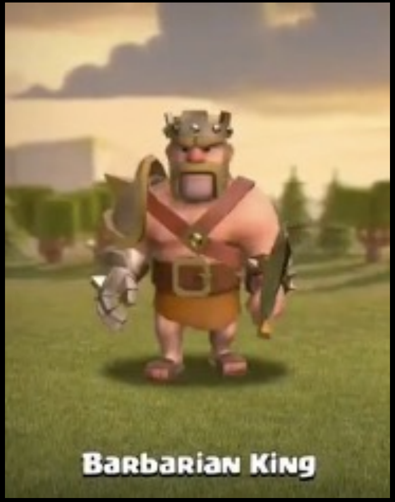


```
layout(location=0) out vec4 fragColor;

// ...

fragColor = color;
if (const_useSrgb)
    fragColor.rgb = LINEAR_TO_SRGB(fragColor.rgb);
```



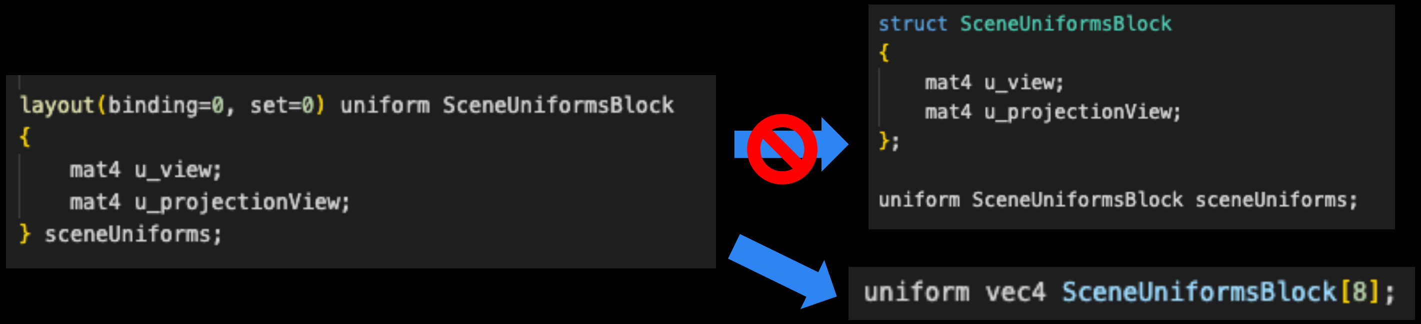
```
layout(location=0) out vec4 fragColor;

// ...

if (const_useSrgb)
    fragColor = vec4(LINEAR_TO_SRGB(color.rgb), color.a);
else
    fragColor = color;
```





UNIFORM STRUCTURES

- Black screen with some flickers in the corner
- GAPID (Android graphics debugging tool) showed inconsistencies in uniform data
- Driver didn't like uniforms defined as structures when updating them with glUniform*



```
layout(binding=0, set=0) uniform SceneUniformsBlock
{
    mat4 u_view;
    mat4 u_projectionView;
} sceneUniforms;
```

```
struct SceneUniformsBlock
{
    mat4 u_view;
    mat4 u_projectionView;
};

uniform SceneUniformsBlock sceneUniforms;
```

```
uniform vec4 SceneUniformsBlock[8];
```

GETTING IT DONE

- Our culture is about empowering people to do their best work
  - No red tape
  - Responsibility and ownership
- Allowed me to work efficiently
- It’s not all roses, but benefits outweigh downsides

CONCLUSIONS

- We build technology for a need
- We had to proceed carefully
- THOR is our platform for the future

WE'RE HIRING!