Achieving Console-Like Experiences on Mobile with Apex Construct

Daniele Di Donato, Staff Software Engineer, Arm
Kristoffer Benjaminsson, Chief Technology Officer, Fast Travel Games (FTG)

Game Developers Conference
Moscone Center, San Francisco, CA - March 21, 2018
The question

Can we achieve Console-Like Experiences on Mobile?
Agenda

• Apex Construct in a nutshell
• The vision
• What we had
• Moving to mobile
• Chasing performance
  • Profiling & Optimizations
• Gameplay
  • Inside-out tracking using ARCore
Key Takeaways

• Tools and tips from an experienced game developer team
• A workflow process to profile and optimize your game for mobile
• How can you achieve positional tracking on mobile today
Apex Construct in a nutshell

Action Adventure Game built for VR
Built for positional tracking & motion controllers
50/50 Combat vs Exploration and puzzle solving
PS4 + PC
Man’s experiments with alien technology has ripped the world apart and reassembled it as a scrambled mess, empty of organic life.

Two AIs strive to repopulate the planet, one with perfect artificial life, the other with humans, returned from the void.

The player is cast straight into a conflict between the two and wields an upgradable bow in combat and to solve puzzles.
Why Mobile VR?

Part of our technical vision
Play whenever you want, wherever you are
Not bound to a specific location: World Scale tracking

Potential to offer the same immersion as stationary VR!
Many ways of accomplishing the vision but right now low power CPUs/GPUs are the best option

Still waiting for positional tracking and motion controllers
Is Apex Construct on mobile even feasible?

It's a title built for PS4 & PC
Built for motion controllers and positional tracking
But the performance range is already quite broad
And mobile CPUs and GPUs are quite powerful!
Lot of tools and tech already in place to handle scaling
First step towards vision

First reality check

- What is possible today?
- Compare limitations: mobile vs console
- Compare perceived quality
- What has to give way?

“Even though the actual fidelity will be lower, the perceived fidelity will be similar”
Optimizing Apex Construct

Performance needs to be a mindset. Nothing will stick out from a grey mess!

FTG Rule: Our own code should never be in top when profiling

Replaced 3rd party solutions with custom code to gain control and performance

Provide tools to manage scaling across platforms and performance levels

Reduce asset complexity and number of expensive items on lower performance levels

- FX
- Audio
- Ambient animations
- Intractable items

Merged meshes to control batching
Tools to handle scaling between platforms

Sub levels: Controls large amount of objects at **build time**. Think layers in Photoshop. Allows multiple people to work on a single scene at the same time. Sub levels are included in builds based on quality level.

Bucket Streaming: Controls large amount of objects at **runtime**. Group objects belonging together in sets. Define sets that are active together into buckets. Enable/Disable buckets based on player location or events.

Quality Set Selectors: **Runtime** control parts of object hierarchies to be enabled based on quality level.
Sublevels
Bucket Streaming

Example:
4 sets of objects that can be referenced by different buckets

Each bucket defines the sets that are active together

Bucket A: 1 & 2
Bucket B: 2 & 4
Bucket C: 1 & 3
Quality Set Selectors

Operates at runtime by evaluating to low/medium/high setting

What defines the setting is defined in code per project

Enables/disables objects as part of creation using the same mechanism as bucket streaming

Everything referenced by a given set is enabled

Everything in Transforms not referenced is disabled
Mesh Merge

22 STATIC MESHES
184 PROBE LIT MESHES
64 PROBE LIT TRANSPARENT MESHES

22 STATIC MESHES
184 PROBE LIT MESHES
64 PROBE LIT TRANSPARENT MESHES

1 STATIC MESH
6 PROBE LIT MESHES
3 PROBE LIT TRANSPARENT MESHES
Lighting merged meshes

MERGING IS DONE BASED ON LOCATION BY ARTISTS
THIS IS AN EXAMPLE OF A STANDARD MERGE.
Physics

Mesh merge made wonders for rendering but messed with physics performance.

Too big physics meshes will make the broadphase test fail resulting in more expensive midphase tests.

Solution: Extend tool to allow for fine grained physics mesh generation.
Cheaper physics midphase

Apex Construct originally used the visual mesh as mesh collider

Splitting mesh colliders and switching to convex hulls for complex meshes increases chance for broadphase elimination and lowers cost for midphase, narrowphase
What to consider when moving to mobile

Performance

• HW characteristics
• Battery life/Thermals

Gameplay

• Touchscreens, not joypads
• Controllers difference
• Positional-tracking
• Ease of use
Tools: how do we investigate performance on Mali?

ARM Streamline

Useful for CPU/GPU profiling

Time in ms = (Work in MCycles / GPU MHz) *1000;
Ex. = (10 MCycles / 338Mhz) * 1000 = 23.6ms
Tools: how do we investigate performance on Mali?

ARM Mali Graphics Debugger (MGD)

Useful for API investigations (drawcalls, triangles, shaders, etc.)
VR Games: Should I profile in VR?

VR poses various constraints to the HW

- Namely: CPU and GPU frequency are lowered to improve performance sustainability
  - In GearVR use "adb logcat -s VrApi" to see the frequencies used.


- Using VR allows better understanding of performance. There is almost none CPU/GPU scaling.

If you profile w/o VR, always assume that running in VR:

- CPU usage will double (unless SinglePass is used)
- Vertex shading time will double (due to twice geometry to be rendered)
- Fragment shading time will increase slightly for a 1080p resolution
  - GearVR uses 2x1Kx1K eye buffer which is ~1920x1080 pixels. Time-warp needs to be considered as well.
- It can sidetrack your investigation
Profiling/Optimization workflow

When profiling, be precise and lean

• It is easy to get lost when you are iterating fast between profiling and optimizing. Keep track of your measurements somewhere.
• You will have self-doubts about older optimizations, especially if they may conflict with new ones.
• Gives answer to: "Where/What/Why/Which?"

When optimizing, be creative

• Create fast experiment to test how the performance will be if you implement that an optimization
• Gives answer to: "How?"
FTG – Iteration 1. No-VR: 43FPS

We started without VR to use boards that don’t support it

Where: Heaviest scene in the mission

What: fragment bound (~21ms, 14cycles per-fragment).

Why: FTG original scene uses standard shaders and custom Surface shaders using standard lighting

Which: Vertex blending material were used for the high-area assets and that access 6 or 9 textures per-fragment

How: We could bake vertex-blended texture offline
  - Prototype: Created a shader that will be used instead and measured perf.

Final fragment time: ~14ms, 9.9cycles per-fragment.
FTG – Iteration 2.a. No-VR: 60FPS
FTG – Iteration 2.b. No-VR: 60FPS

2393 vertices

1056 vertices
FTG – Iteration 3. No-VR: 60FPS

Original shader

Optimized shader
FTG – Iteration 4. VR: <30FPS
FTG – Iteration 5.a. VR: <30FPS
FTG – Iteration 5.b. VR: >30FPS
Gameplay – VR Inside-out tracking using ARCore

- GearVR and Daydream don’t have head positional tracking...
- ...we used Google ARCore 1.0, a SLAM library which uses phone camera
- We sample only the position from ARCore and use the GearVR rotation

```csharp
public void Update ()
{
    ....
    Vector3 currentARPosition = Frame.Pose.position;
    //Remember the previous position so we can apply deltas
    Vector3 deltaPosition = currentARPosition - m_prevARPosePosition;
    m_prevARPosePosition = currentARPosition;
    ....
}
```
Gameplay – VR Head Positional Tracking
Final result

Come to our booth #1523 to try it!

PSVR

GearVR
Conclusions

Achieving console-like experiences on mobile is possible today

- Even easier in the future (Vulkan, new Drivers/GPUs, etc).
- The inverse is not true

If you plan to ship on Mobile, adopt a Mobile mindset:

- Know the rules of the game
- Don’t be sloppy but care about performance from the start
- Performance can affect gameplay
Thank You!
Danke!
Merci!
谢谢!
ありがとう!
¡Gracias!
Kiitos!
감사합니다
धन्यवाद
References

Midgard Perf. Counters
https://community.arm.com/graphics/b/blog/posts/mali-midgard-family-performance-counters

Bifrost Perf. Counters
https://community.arm.com/graphics/b/blog/posts/mali-bifrost-family-performance-counters

GPU Budget
https://community.arm.com/graphics/b/blog/posts/gpu-processing-budget-approach-to-game-development

Inside-out mobile tracking
https://community.arm.com/graphics/b/blog/posts/mobile-inside-out-vr-tracking-now-on-your-phone-with-unity
Want to know more?

ARM Stand
South Hall, Booth #1523

Arm Mali Developer Guides & Tools
https://developer.arm.com/graphics