

# Enhancing Your Unity Mobile VR Experience

Roberto Lopez Mendez – Senior Software Engineer, ARM  
Carl Callewaert – Americas Director & Global Leader of  
Evangelism, Unity Technologies



The Architecture for the Digital World®

# Agenda

- Ice Cave Demo (Roberto)
  - Optimized effects. Notes about the ARM Guide For Unity Developers
- Porting Ice Cave Demo to Samsung Gear VR (Roberto)
  - The changes required for VR
  - Improving VR performance. Dynamic shadows based on local cubemaps
- VR integration into Unity. Best VR practices (Carl)

# Demo Ice Cave



# ARM Guide For Unity Developers: Optimizing Mobile Gaming Graphics

- First released at Unite Seattle 2014
- Initial idea: write about the experience of learning Unity from scratch
- Available for free at [MaliDeveloper.arm.com](http://MaliDeveloper.arm.com)
- V1.0
  - Optimizing Applications. Quality settings in Unity (Chapter 2)
  - Profiling Applications with the Unity Profiler(Chapter 3)
  - Optimizations: Application, GPU and Assets (Chapter 4)
- V2.0
  - Enlighten (Chapter 5) and advanced graphics techniques (Chapter 6)
- V3.0
  - More advanced graphics techniques (Chapter 6)



Download the ARM Guide for Unity Developers at [MaliDeveloper.arm.com](http://MaliDeveloper.arm.com)

# Porting Ice Cave Demo to Samsung Gear VR

# Unity VR support



## VR Plugin support limitations

- Each VR device has a different plugin
- Plugins may conflict with each other
- Each release of newer VR SDKs / Runtimes can break older games
- Lower level engine optimizations are not possible with plugin approach of two separate cameras



## Unity 5.1 VR Native Support

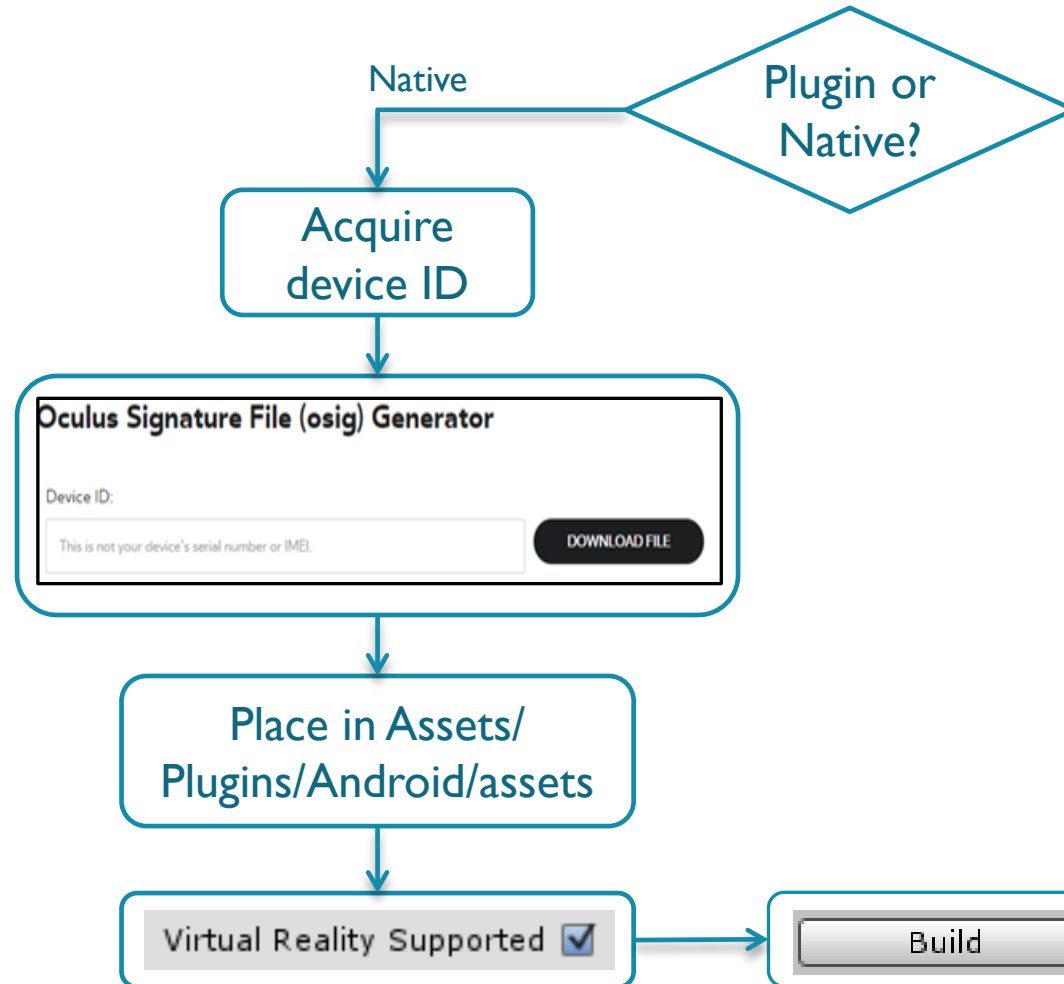


# Samsung Gear VR

- Sensors: Accelerator, Gyrometer, Geomagnetic, Proximity
- Motion to Photon Latency < 20ms
- Manual Focal Adjustment
- Main Physical UI: TouchPad
- Available for Samsung Galaxy S6 and Samsung Note4
- Oculus's Asynchronous TimeWarp technology



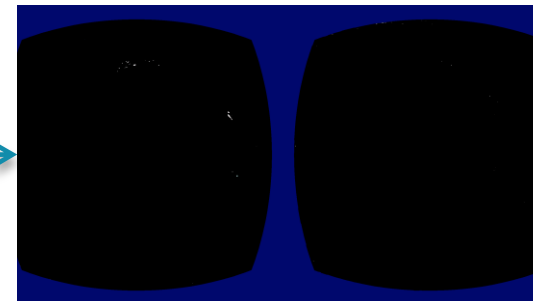
# Porting Ice Cave for Samsung Gear VR



## Enabling Gear VR developer mode

- Go to your device Settings → Application Manager → Gear VR Service
- Tap on "Manage storage"
- Tap on the "VR Service Version" six times
- Wait for scan process to complete and you should now see the Developer Mode toggle

Developer mode allows launching the application without the headset and also docking the headset at any time without having Home launch.



# Considering VR specifics. UI.

- Removed existing UI. Not suitable for VR
- Added very simple UI through Gear VR touchpad:
  - Pressed: camera moves in the direction the user is looking
  - Not pressed: camera stops
  - Double tap: camera resets to initial position



# Considering VR specifics. Motion sickness.

- Removed camera animation
  - Users experienced motion sickness particularly when going backwards
- Carefully set the camera speed after many test with different users
  - Users experience motion sickness when the camera moved just a little too fast
- Lucky that Ice Cave is big
  - Users don't feel claustrophobic and can explore different areas of the cave

# Considering VR specifics. Application features.

- Removed dirty lens effect
  - Looks bad in VR
- Added camera collision and sliding
  - Bad VR experience when going through geometry



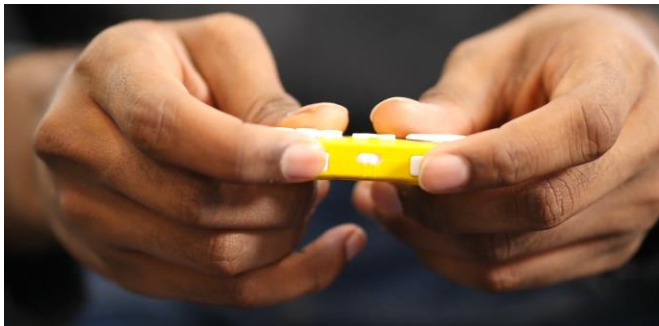
## Ice Cave VR extras

- Added streaming to another device to show what the user is actually experiencing (camera position and orientation)



# Ice Cave VR extras

- Added an alternative UI by means of a mini Bluetooth controller

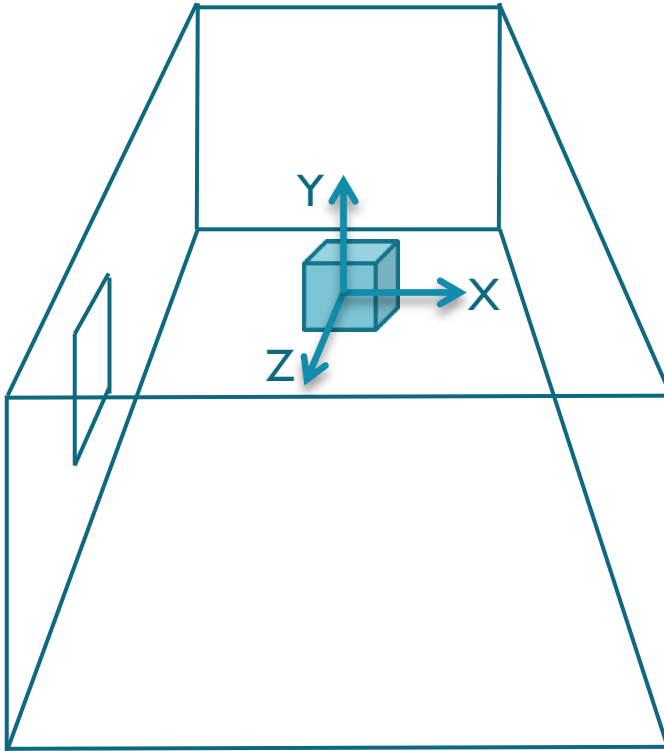


# Dynamic shadows based on local cubemaps

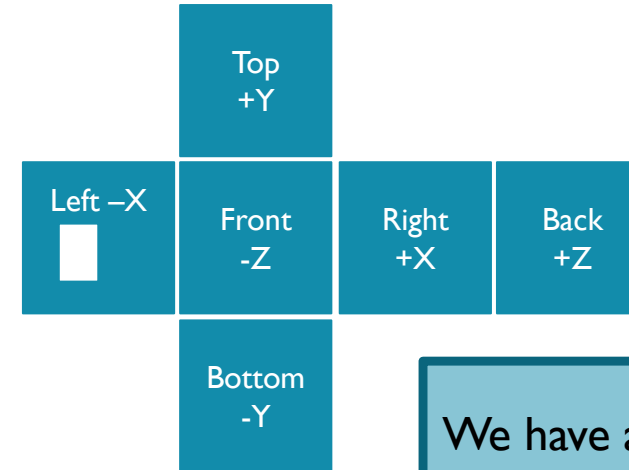
## Performance for VR

# Dynamic Soft Shadows Based on Local Cubemaps

## Generation stage



Render the transparency of the scene in the alpha channel



Camera background alpha color = 0.

Opaque geometry is rendered with alpha = 1.

Semi-transparent geometry is rendered with alpha < 1.

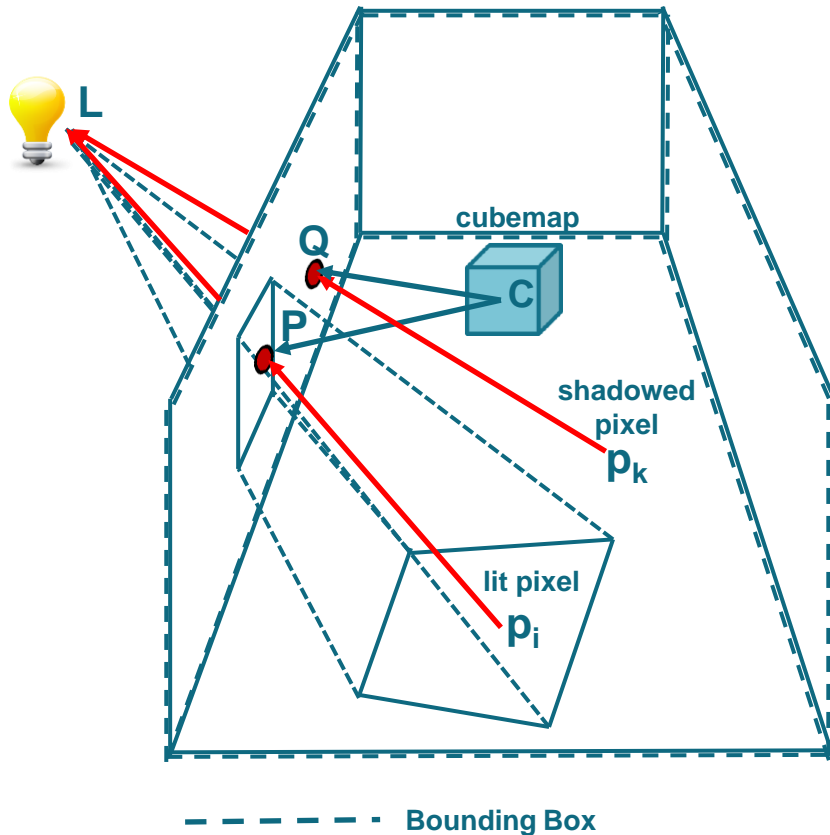
Fully transparent geometry is rendered with alpha 0.

We have a map of the zones where light rays can potentially come from and reach the geometry.

No light information is processed at this stage.

# Dynamic Soft Shadows Based on Local Cubemaps

## Runtime stage



- Create a vertex to light source  $L$  vector in the vertex shader.
- Pass this vector to the fragment shader to obtain the vector from the pixel to the light position  $\mathbf{p}_i L$ .
- Find the intersection of the vector  $\mathbf{p}_i L$  with the bounding box.
- Build the vector  $\mathbf{CP}$  from the cubemap position  $C$  to the intersection point  $P$ .
- Use the new vector  $\mathbf{CP}$  to fetch the texture from the cubemap.

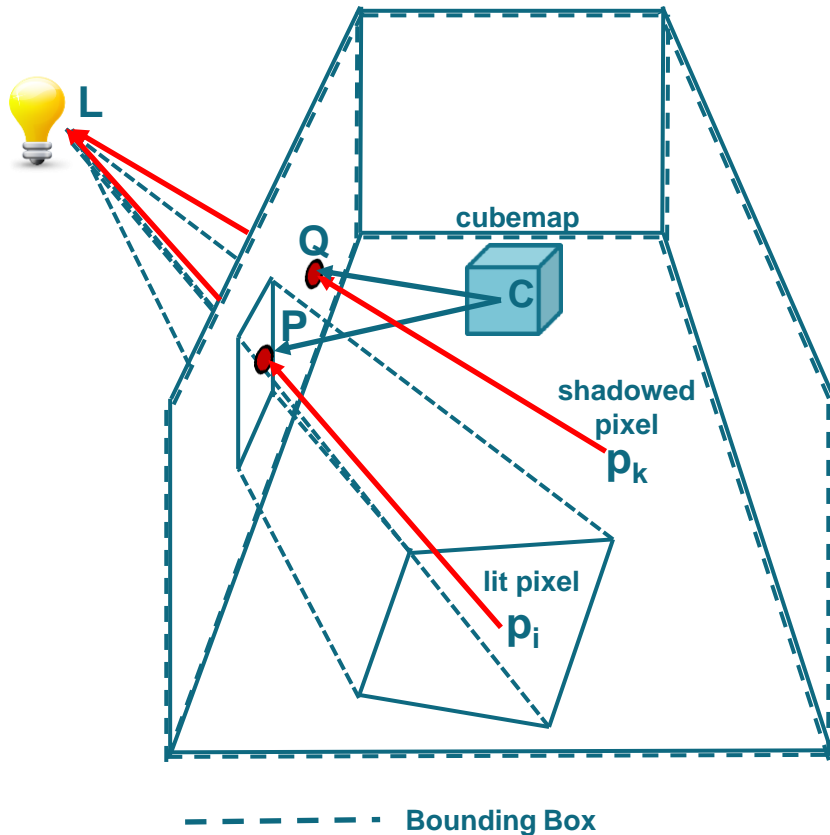
```
float texShadow = texCUBE(_CubeShadows, CP).a;
```



Source code is in the ARM Guide for Unity Developers at [MaliDeveloper.arm.com](http://MaliDeveloper.arm.com)

# Dynamic Soft Shadows Based on Local Cubemaps

Why soft?



~~float texShadow = texCUBE( \_CubeShadows, CP).a;~~

float4 newVec = float4(CP, factor \* length( $p_iP$ ))

float texShadow = texCUBElod(\_CubeShadows, newVec ).a;



Source code is in the ARM Guide for Unity Developers at [MaliDeveloper.arm.com](http://MaliDeveloper.arm.com)

# Dynamic soft shadows based on local cubemaps



## Wrap Up

- Unity has made a great contribution to VR democratization.
- VR is a new boost to mobile games. The user experience is no longer limited to the mobile screen. The user is now embedded in a virtual world.
- It is possible to run high quality VR and non-VR content in mobile devices using optimized rendering techniques.
- Check out The ARM Guide for Unity Developers for optimizations tips, recommendations and very efficient rendering techniques to make the most out of Unity when developing mobile games.



More source code in the ARM Guide for Unity Developers V3.0 at [MaliDeveloper.arm.com](http://MaliDeveloper.arm.com)

# VR integration into Unity

## Best VR practices

Carl Callewaert  
Americas Director & Global Leader of Evangelism  
Unity Technologies



# Thank You



*The trademarks featured in this presentation are registered and/or unregistered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Any other marks featured may be trademarks of their respective owners*

# To Find Out More....



- Find out more about shadows based on local cubemaps at:
  - <http://community.arm.com/groups/arm-mali-graphics/blog/2015/04/13/dynamic-soft-shadows-based-on-local-cubemap>
- Find out more about other techniques based on local cubemaps at:
  - <http://community.arm.com/groups/arm-mali-graphics/blog/2014/08/07/reflections-based-on-local-cubemaps>
  - <http://community.arm.com/groups/arm-mali-graphics/blog/2015/04/13/refraction-based-on-local-cubemaps>
  - <http://community.arm.com/groups/arm-mali-graphics/blog/2015/05/21/the-power-of-local-cubemaps-at-unite-apac-and-the-taoyuan-effect>