PLAYCANVAS

January 2022 Update
PLAYCANVAS

Browser-based 3D content creation platform
PLAYCANVAS

Browser-based content creation platform
PLAYCANVAS

Browser-based content creation platform
PLAYCANVAS

Browser-based content creation platform

Viewer
Editor
Shader Editor
Runtime
- Games
- VR
- Visualizers
- Playable ads
- e-commerce
PLAYCANVAS

- Games
- VR
- Visualizers
- Playable ads
- e-commerce

- Desktop & mobile
- WebGL 1.0 & 2.0
1. Clustered Lighting

What is it?

- Solution for rendering lots of lights
- Traditionally combined with deferred shading
- Used to great effect in games like Doom 2016 and many others
1. Clustered Lighting

PlayCanvas Engine

- Simple forward renderer
- Shaders generated at runtime
1. Clustered Lighting

Old Method

- Lights baked into shader code
- Limitations:
  - Slow
  - Shader recompilation
  - Unpredictable resource usage
  - :(  

1. Clustered Lighting

New method
1. Clustered Lighting

New method

- Clustered lighting
1. Clustered Lighting

New method

- Clustered lighting
- 3 Steps
1. Clustered Lighting

Step 1 - generate a 3d cluster grid
1. Clustered Lighting

Step 1 - generate a 3d cluster grid
1. Clustered Lighting

Step 1 - generate a 3d cluster grid
1. Clustered Lighting

Step 1 - generate a 3d cluster grid
1. Clustered Lighting

Step 2 - store light parameters
1. Clustered Lighting

Step 2 - store light parameters in light buffer
1. Clustered Lighting

Step 3 - Render the scene

- Per pixel:
  - Given world-space fragment location
  - Calculate grid cell
  - In a loop:
    - Read light index from cluster cell
    - If light index == 0 return;
    - Read light parameters from light buffer texture
    - Calculate and sum lighting contribution
1. Clustered Lighting

Texture Atlas

World

Shadow Atlas
1. Clustered Lighting

Wins

- Fast!
- Fewer shader recompiles
- Predictable resource usage
1. Clustered Lighting

Future plans

- Migrate completely to clustered solution
- Decals
- Volumetric lighting
2. Image Based Lighting

- Multiple rendering paths
- Based on runtime hardware
2. Image Based Lighting

sh + dp
2. Image Based Lighting
2. Image Based Lighting
2. Image Based Lighting

Similar limitations to old lighting

- Varying resource usage
- Looks different depending on hardware support
- Complicated code to debug and maintain
- :\(\)
2. Image Based Lighting

Atlas!
2. Image Based Lighting

Roughness
2. Image Based Lighting

Sharp Reflection
2. Image Based Lighting

Prefiltered Ambient
2. Image Based Lighting

\texttt{lerp(a, b, roughness);}
2. Image Based Lighting

\text{lerp}(a, b, dF);
2. Image Based Lighting

1 tap

2 tap
2. Image Based Lighting

Updated prefiltering

- GGX
- Importance sampling
- Shout out to the Khronos glTF sample viewer
- https://github.com/KhronosGroup/glTF-Sample-Viewer
2. Image Based Lighting

Summary

- Consistent resource usage
- Looks the same
Online examples

https://playcanvas.github.io/
Online demos

https://github.com/playcanvas/engine
Thanks!

PlayCanvas Team <3

Martin  Elliott  Iakov  Joao  Steven  Zach  Will
Thanks!

PlayCanvas Team <3