Killzone Shadow Fall

Creating art tools for a new generation of games

Sander van der Steen
Senior Technical Artist
Guerrilla Games

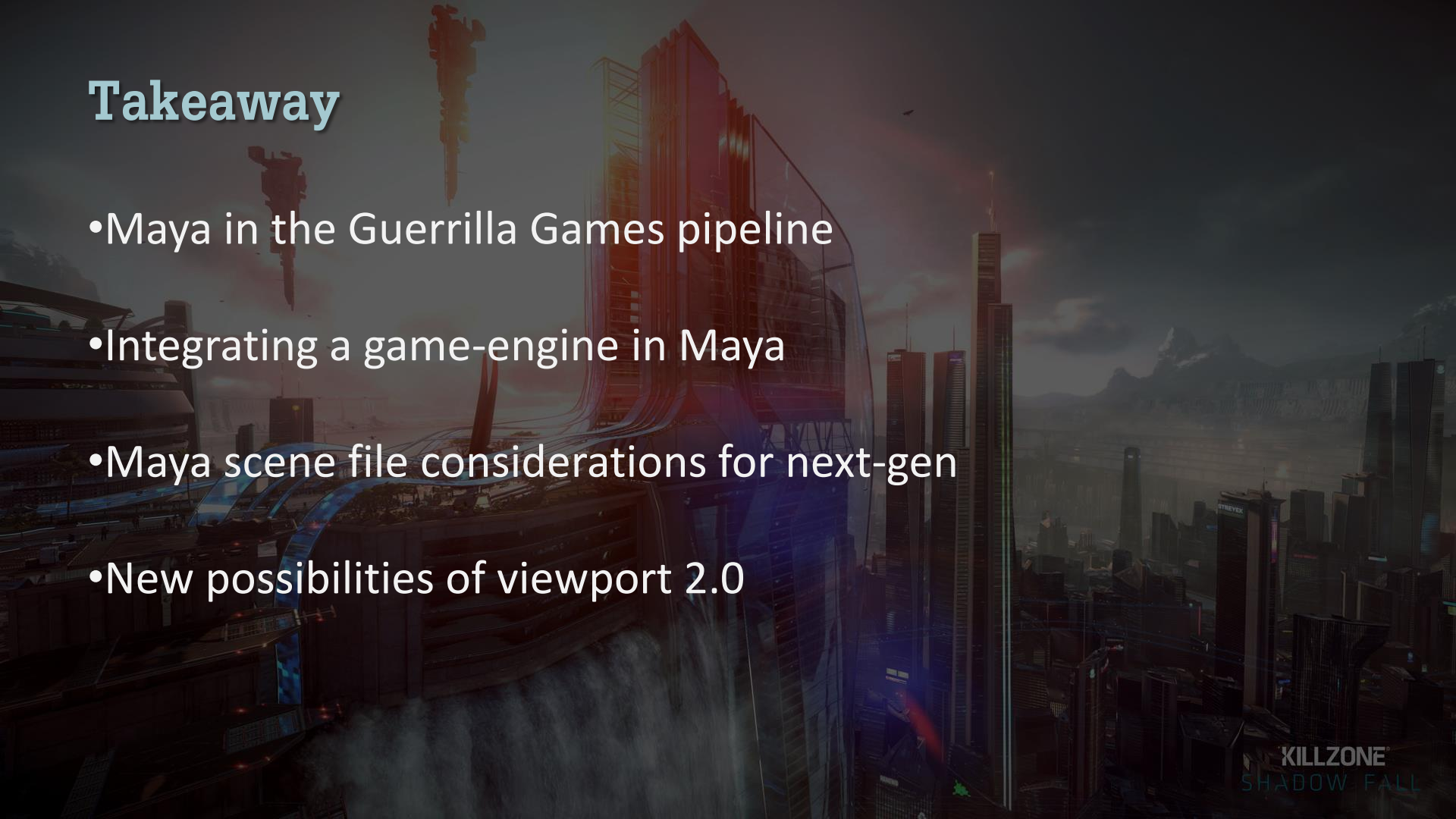
GUERRILLA

Intro

- Guerrilla Games is based in Amsterdam
  - First party Sony studio since 2005
- Killzone Shadow Fall
  - Launch title for the Playstation 4
- Pipeline build around Maya

Takeaway

- Maya in the Guerrilla Games pipeline
- Integrating a game-engine in Maya
- Maya scene file considerations for next-gen
- New possibilities of viewport 2.0

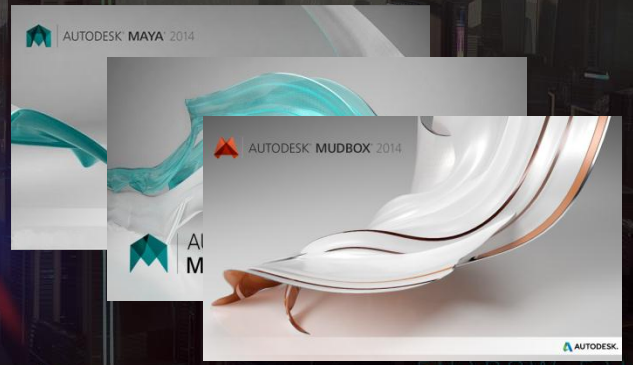


Maya in the Guerrilla Games pipeline

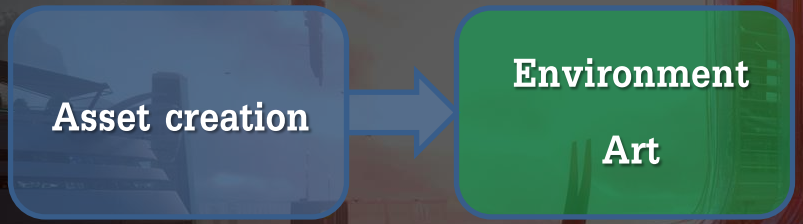
Simple Killzone Shadow Fall pipeline overview

Asset creation

Maya, Mudbox, Motion builder etc.

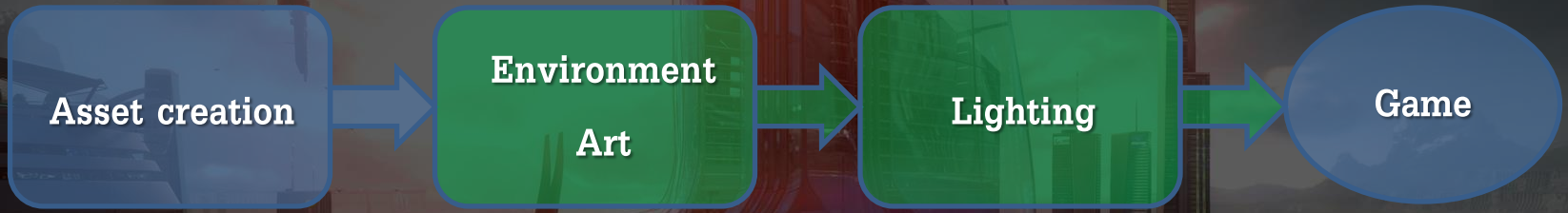


Simple Killzone Shadow Fall pipeline overview



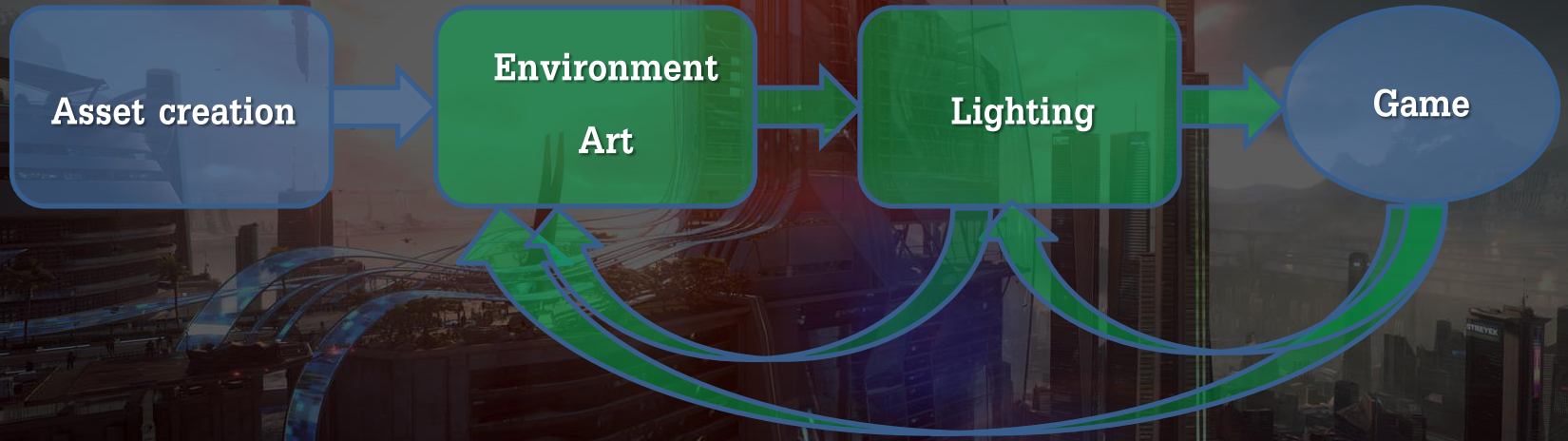
- All assets are imported into Maya and placed in environment
- Individual assets are tweaked

Simple Killzone Shadow fall pipeline overview



- The levels with geometry transfer to the lighting team
  - “Complete” levels will be exported to game

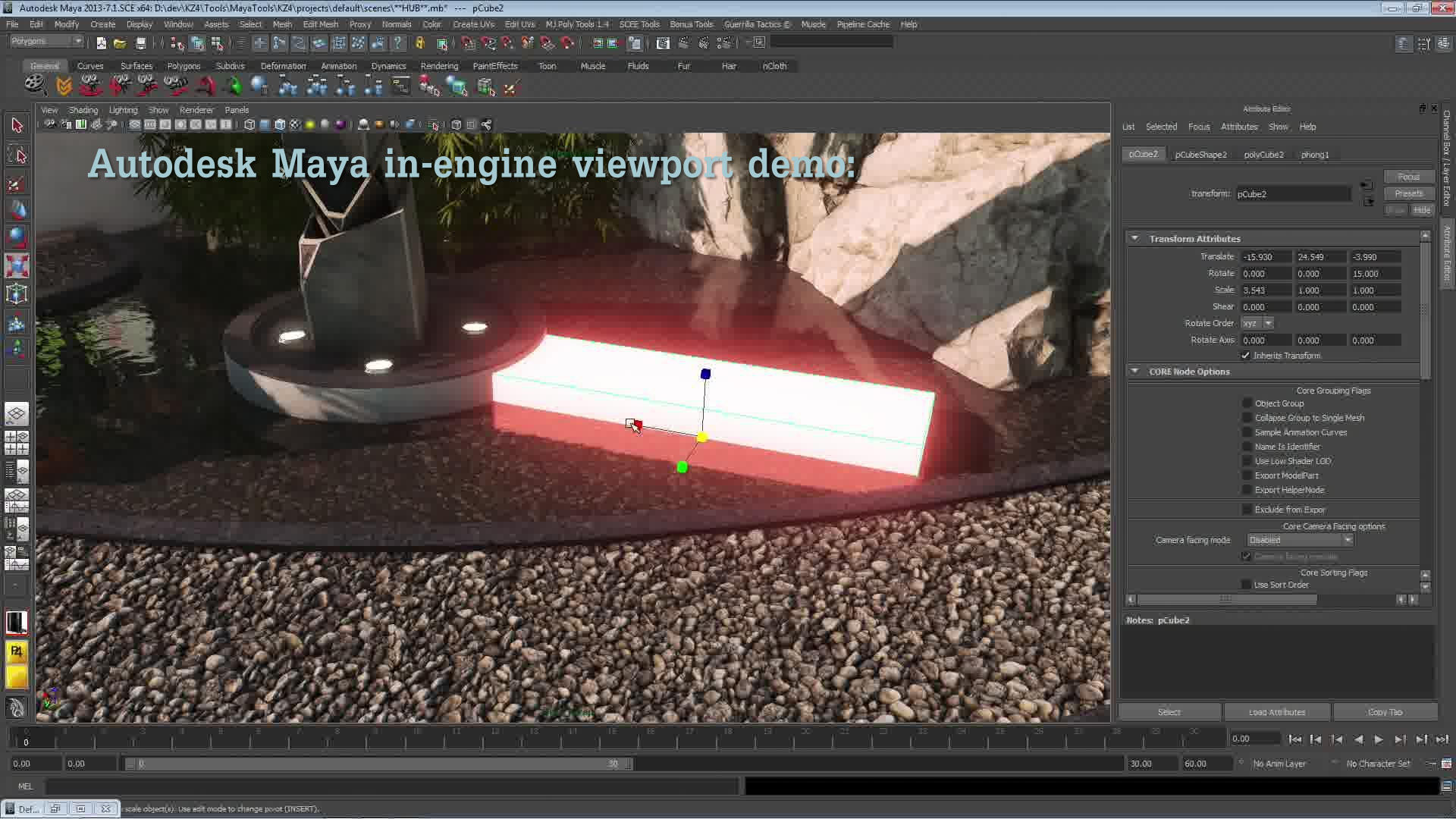
Simple Killzone Shadow Fall pipeline overview

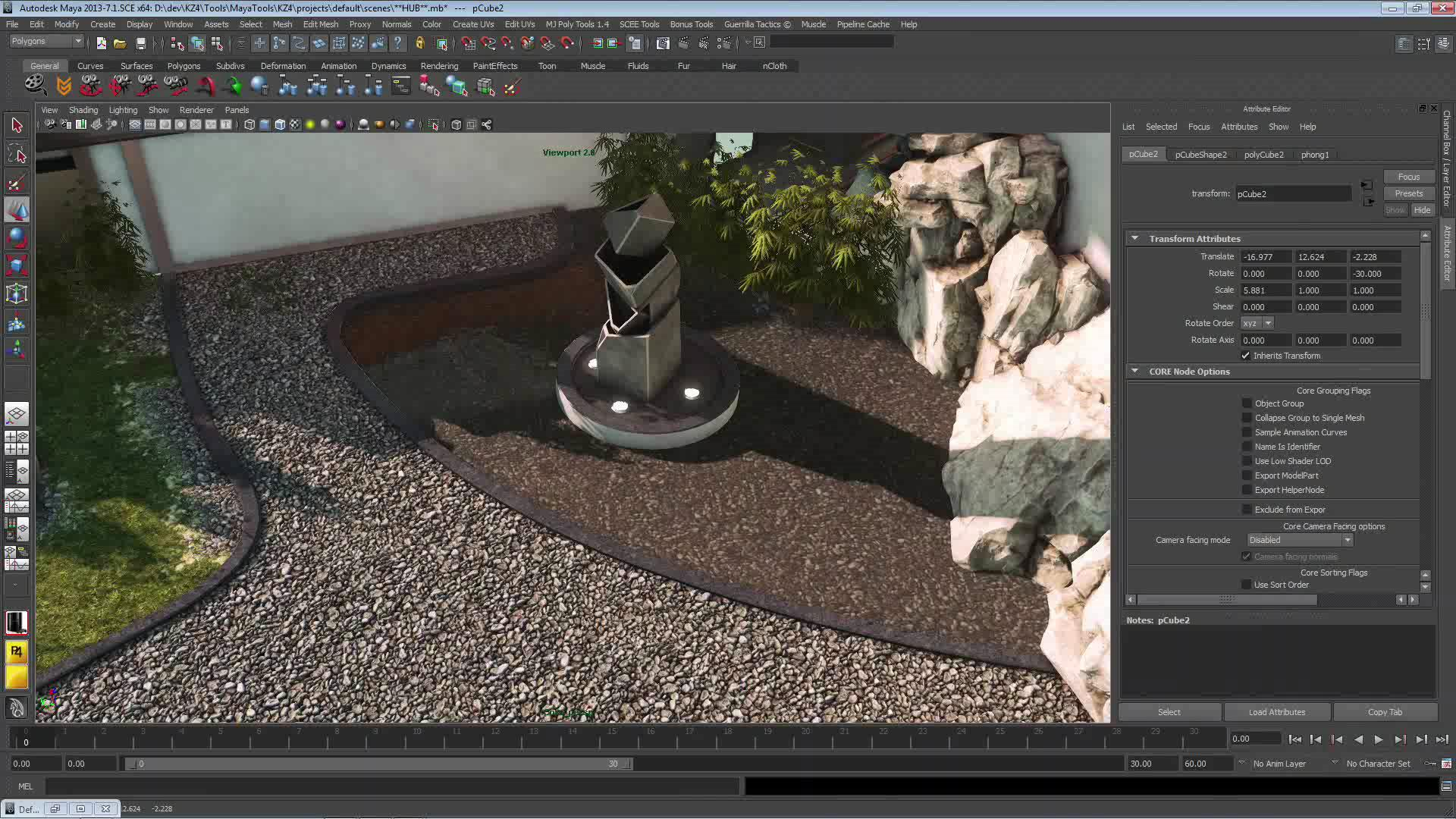


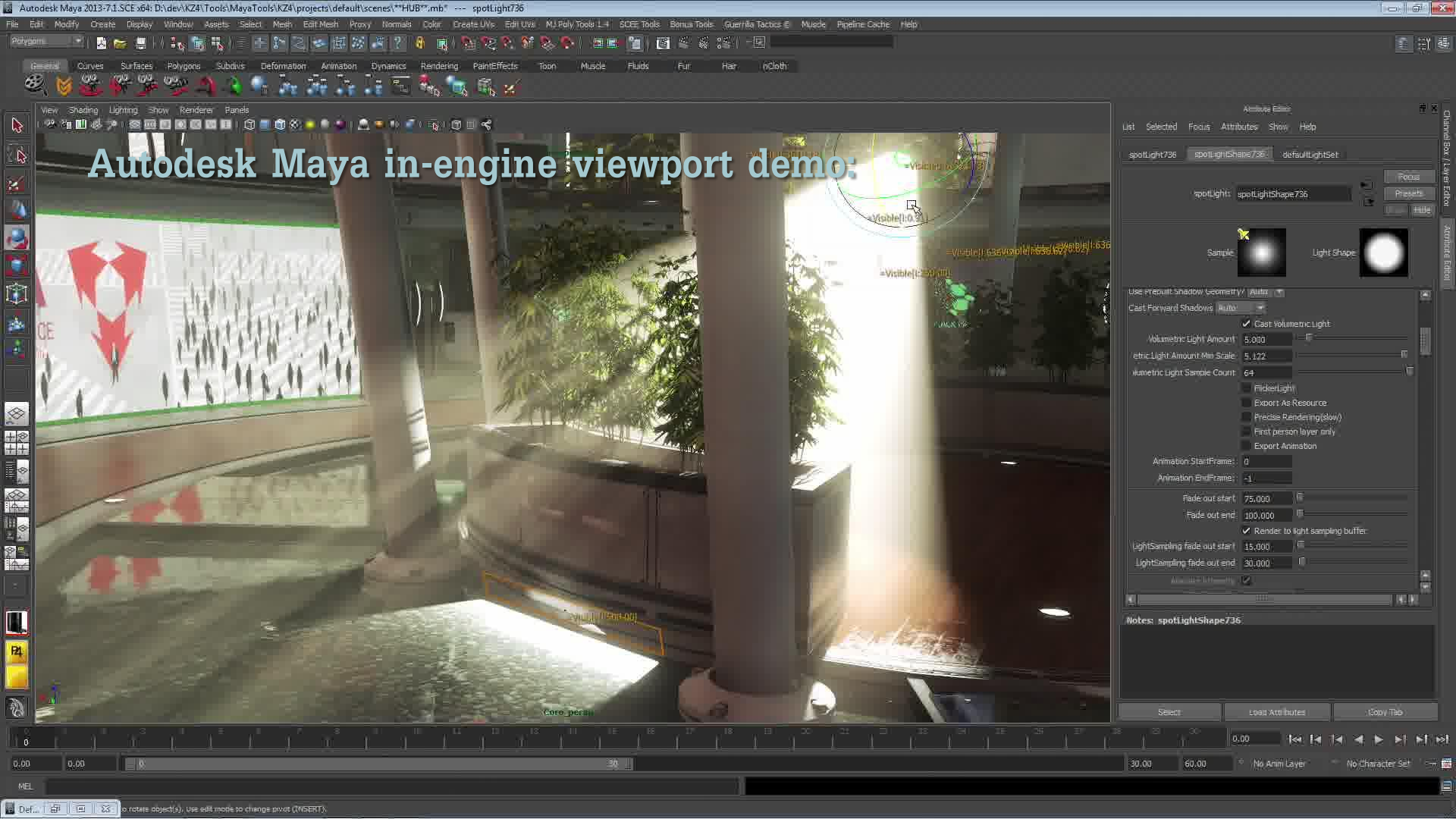
- Many iterations required to reach production quality

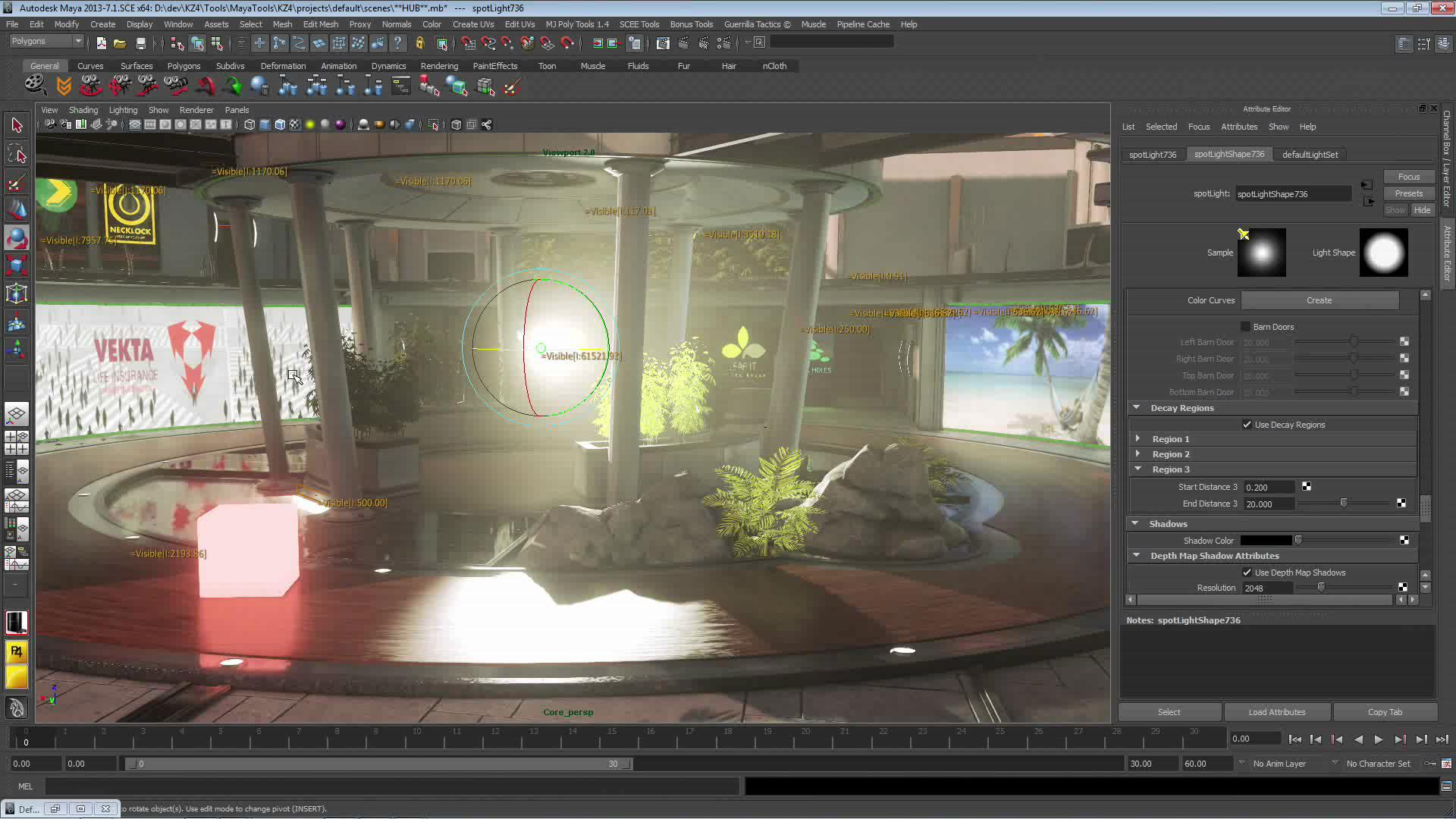
Autodesk Maya advantages

- Artist friendly + familiar
- A lot of *editing* for free
  - Mesh edits *(polygon tools)*
  - Material edits *(hypershade)*
- Less need to *import/export to/from* applications


Autodesk Maya in-engine viewport demo:


Autodesk Maya 2013-7.1.SCE x64: D:\dev\KZ4\Tools\MayaTools\KZ4\projects\default\scenes**HUB**.mb* --- pCube2
File Edit Modify Create Display Window Assets Select Mesh Edit Mesh Proxy Normals Color Create UVs Edit UVs MJ Poly Tools 1.4 SCEE Tools Bonus Tools Guerrilla Tactics © Muscle Pipeline Cache Help
Polygons
General Curves Surfaces Polygons Subdivs Deformation Animation Dynamics Rendering PaintEffects Toon Muscle Fluids Fur Hair nCloth
View Shading Lighting Show Renderer Panels
Viewport 2.0
Attribute Editor
List Selected Focus Attributes Show Help
pCube2 pCubeShape2 polyCube2 phong1
transform: pCube2
Focus Presets Show Hide
Transform Attributes
Translate -16.977 12.624 -2.228
Rotate 0.000 0.000 -30.000
Scale 5.881 1.000 1.000
Shear 0.000 0.000 0.000
Rotate Order xyz
Rotate Axis 0.000 0.000 0.000
Inherits Transform
CORE Node Options
Core Grouping Flags
Object Group
Collapse Group to Single Mesh
Sample Animation Curves
Name Is Identifier
Use Low Shader LOD
Export ModelPart
Export HelperNode
Exclude from Expor
Core Camera Facing options
Camera facing mode Disabled
Camera facing normals
Core Sorting Flags
Use Sort Order
Notes: pCube2
Select Load Attributes Copy Tab
Channel Box / Layer Editor
Attribute Editor
MEL
No Anim Layer
No Character Set


Autodesk Maya in-engine viewport demo:


Autodesk Maya 2013-7.1.SCE x64: D:\dev\KZ4\Tools\MayaTools\KZ4\projects\default\scenes**HUB**.mb* --- spotLight736
File Edit Modify Create Display Window Assets Select Mesh Edit Mesh Proxy Normals Color Create UVs Edit UVs MJ Poly Tools 1.4 SCEE Tools Bonus Tools Guerrilla Tactics © Muscle Pipeline Cache Help
Polygons
General Curves Surfaces Polygons Subdivs Deformation Animation Dynamics Rendering PaintEffects Toon Muscle Fluids Fur Hair nCloth
View Shading Lighting Show Renderer Panels
Viewport 2.0
VEKTA
LIFE INSURANCE
NECKLOCK
Core_persp
Attribute Editor
List Selected Focus Attributes Show Help
spotLight736 spotLightShape736 defaultLightSet
spotLight: spotLightShape736
Focus
Presets
Show Hide
Sample
Light Shape
Color Curves Create
Barn Doors
Left Barn Door 20.000
Right Barn Door 20.000
Top Barn Door 20.000
Bottom Barn Door 20.000
Decay Regions
Use Decay Regions
Region 1
Region 2
Region 3
Start Distance 3 0.200
End Distance 3 20.000
Shadows
Shadow Color
Depth Map Shadow Attributes
Use Depth Map Shadows
Resolution 2048
Notes: spotLightShape736
Select Load Attributes Copy Tab
Channel Box / Layer Editor
Attribute Editor
No Anim Layer
No Character Set
MEL


Integrating a game-engine in Maya
KILLZONE
SHADOW FALL

A lot of tools coding!

Maya needs to know how:

- To render
  - *Game engine specific*
- To communicate attribute changes
  - *Mesh edits, position updates etc.*
- To read game data
  - *Maya scene usage*

A game-engine in Maya: Deferred rendering

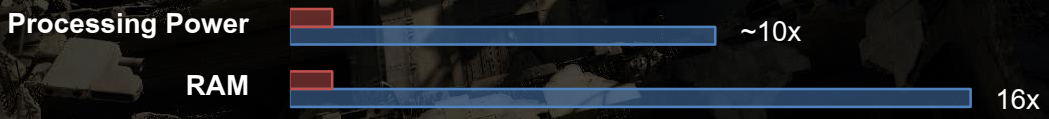
Deferred rendering: Killzone 3

Initial implementation:

- *Brute force* updates each frame *(CPU limited)*
  - Knowing what to update can be difficult
- Limited *integration*
  - *View selected, draw overrides, etc.*

Playstation 3 vs. Playstation 4

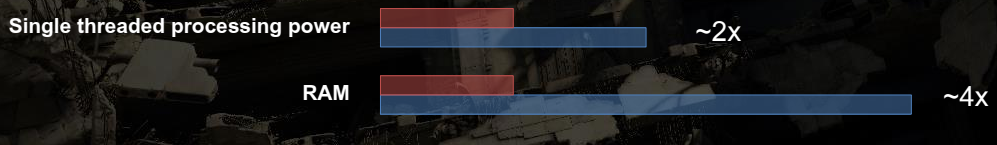
VS



KILLZONE SHADOW FALL

Workstation performance: “Killzone 3” vs “Killzone Shadow Fall”

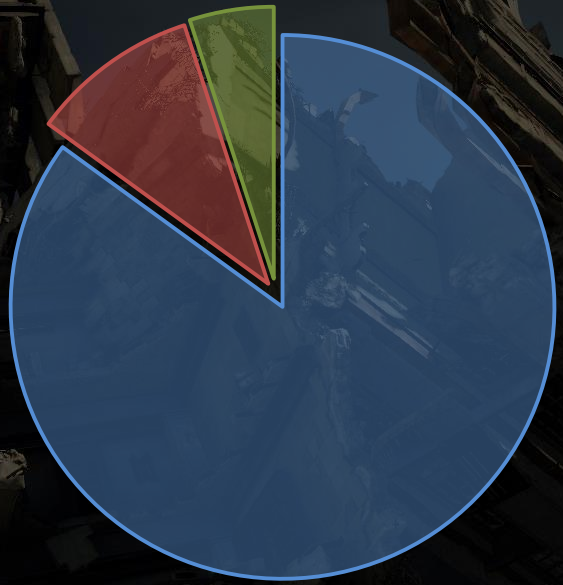
VS



KILLZONE
SHADOW FALL

Deferred rendering: Killzone 3

- Maya Updates
- Maya Render
- Deferred Render



Single frame CPU time distribution

Challenges for pushing updates

- Maya mesh edits?
- Tricky hierarchical updates *(from a parent transform)*
  - Visibility changes
  - Maya LOD groups
- Shader changes?

Challenges for pushing updates 2

Maya's DAG tree might not be represented 1 on 1 in-game.

- Callbacks? *(MMessage)*
- Often no notion of *groups* or *hierarchy*
  - Only world-space transforms in-game

Solution: *Guerrilla DAG*

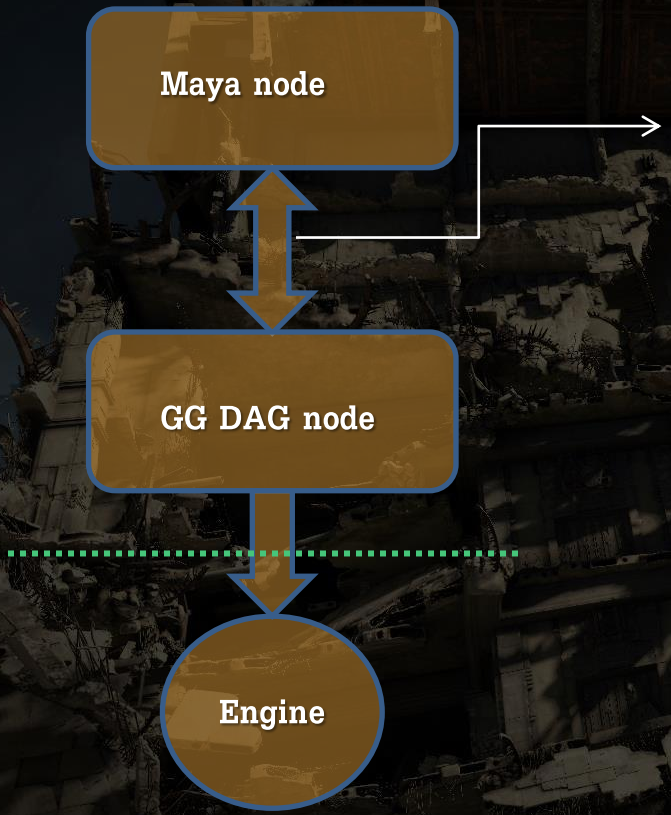
- A lightweight representation of the DAG tree in *Maya*
- It *mirrors* the DAG tree in Maya
- Implemented using C++ callbacks *(API MMessage class)*

Implementing *Guerrilla DAG*

- Node added callback *(MDGMessage)*:
  - Create GG DAGnode
    - *Transforms, lights, maya meshes, building blocks and sets (shaders)*
  - Add GG DAGnode to *database*

- Node removed callback *(MDGMessage):*
  - *Cleanup*

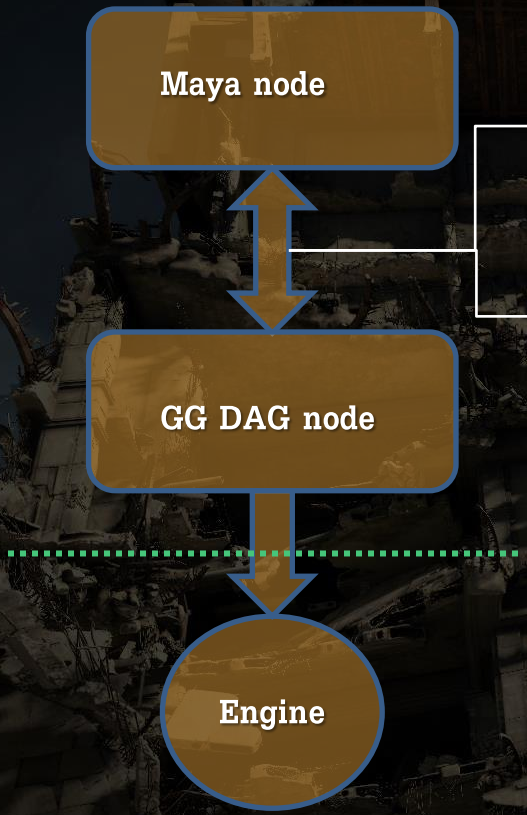
Implementing *Guerrilla DAG*



Each Guerrilla DAG node:

will have *read only* access to it's *sibling* via an *MObjectHandle* to the maya node

Implementing *Guerrilla DAG*



Each Guerrilla DAG node:

will have *read only* access to it's "sibling" via an *MObjectHandle* to the maya node

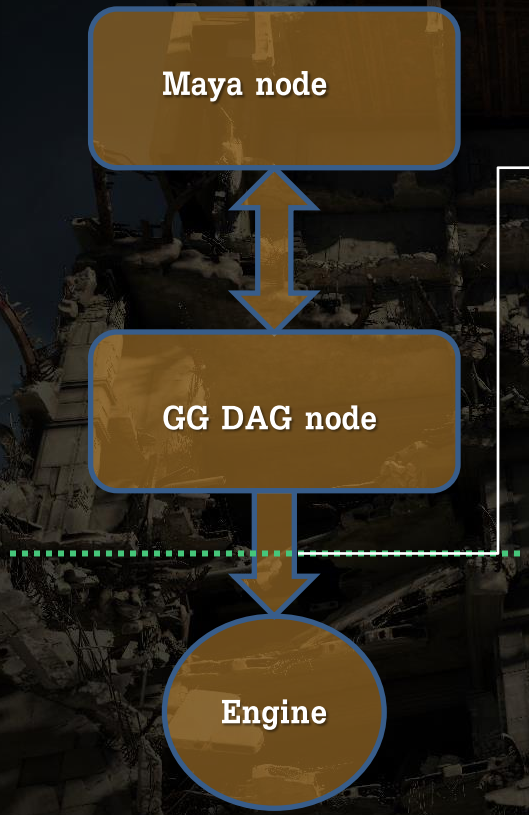
will listen for *AttributeChanged* callback on it's sibling *(MNodeMessage)*

Implementing *Guerrilla DAG*

Each Guerrilla DAG node:

- will have *read only* access to it's "sibling" via an *MObjectHandle* to the maya node
- will listen for *AttributeChanged* callback on it's sibling *(MNodeMessage)*
- deals with *AttributeChanged* as appropriate for the nodetype

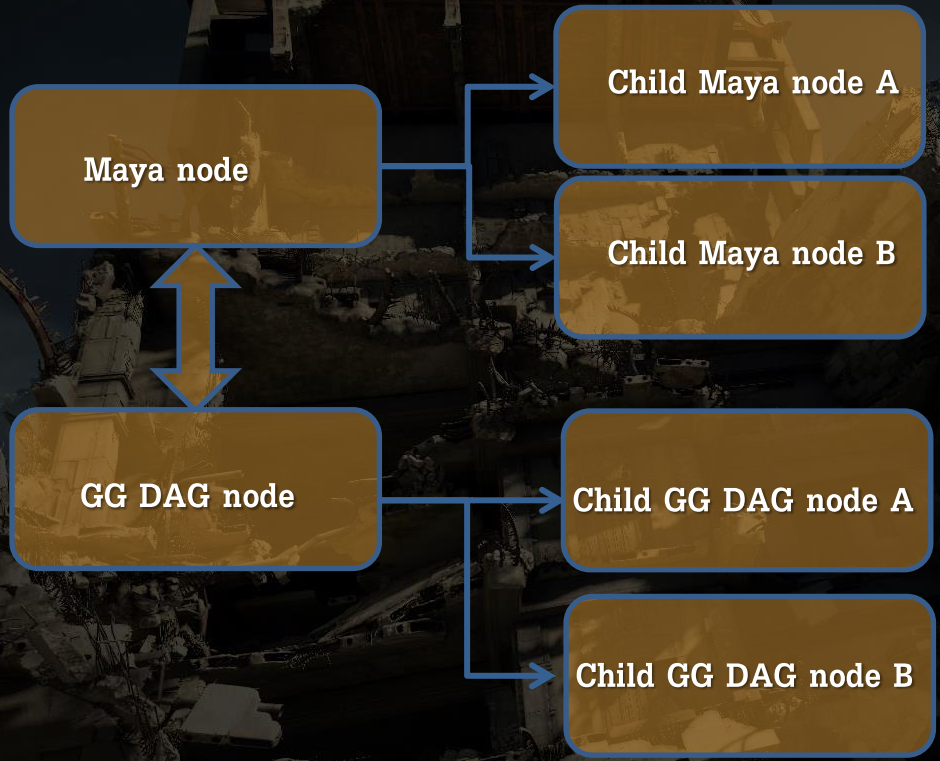
Implementing *Guerrilla DAG*



Each Guerrilla DAG node:

→ *Knows* how to update the game engine

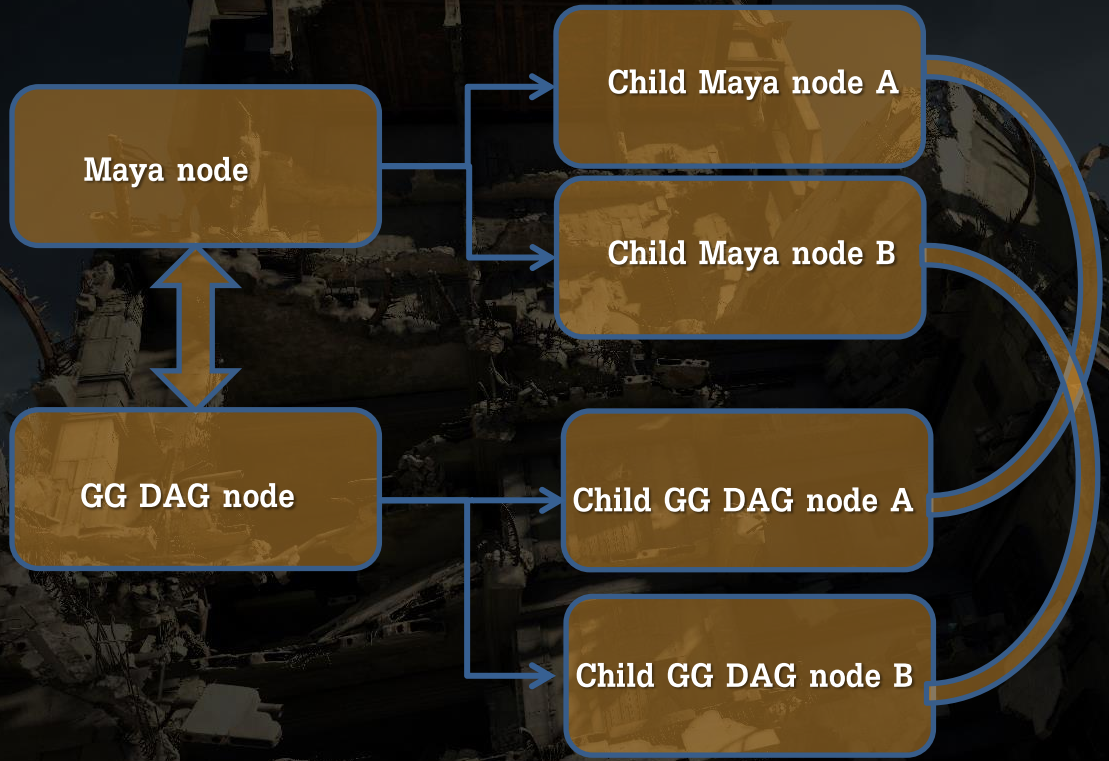
Implementing *Guerrilla DAG*



Transforms are special cases:

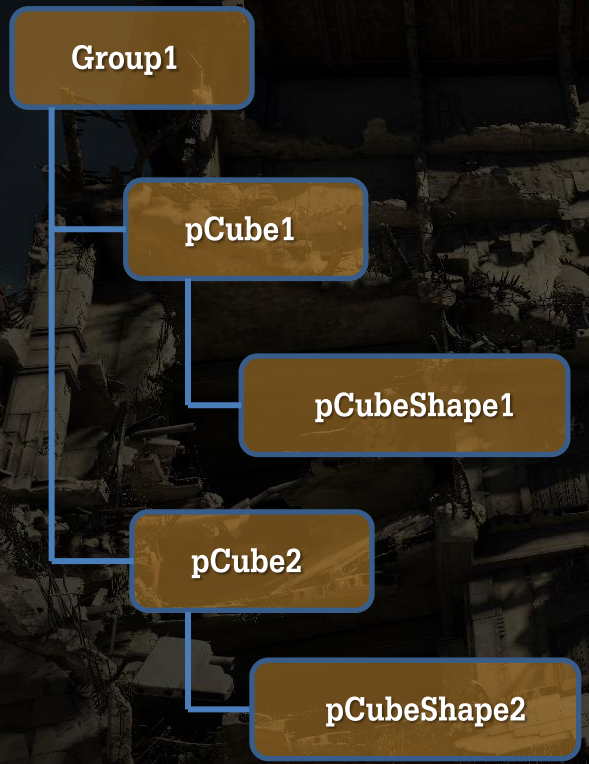
- Monitor child added/removed *(MDagMessage)*
- Keep an up-to-date list of children
- Can push updates to children

Implementing *Guerrilla DAG*



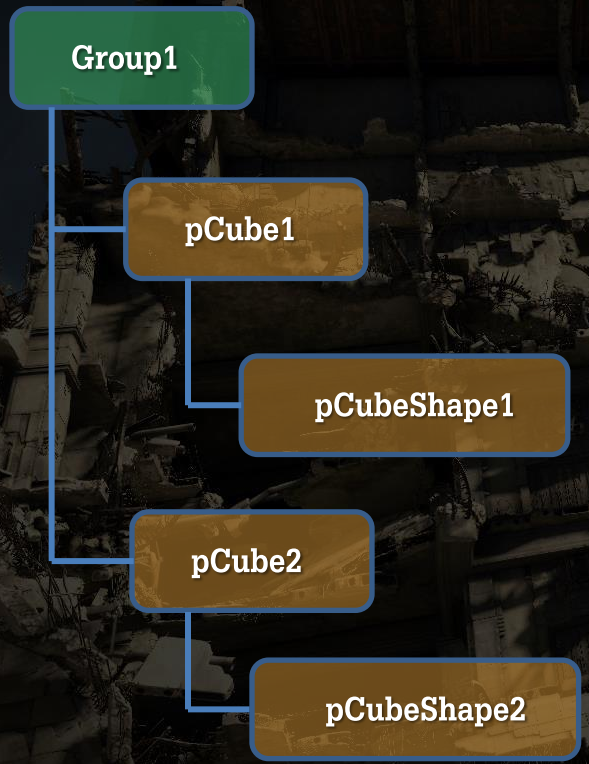
Child nodes can also access siblings!

Practical example: Moving a group



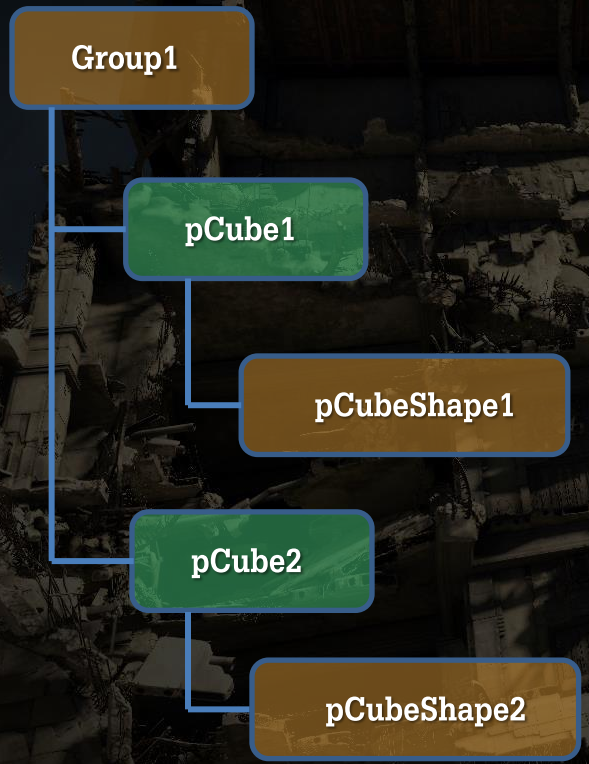
Produces an identical Guerrilla DAG hierarchy

Practical example: Moving a group



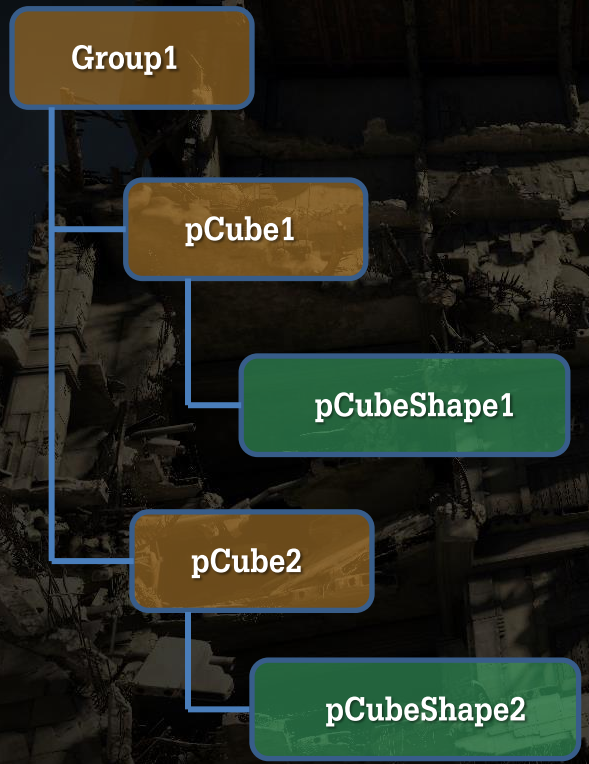
Attribute changed on group1 triggered

Practical example: Moving a group



Update cached matrix and propagate

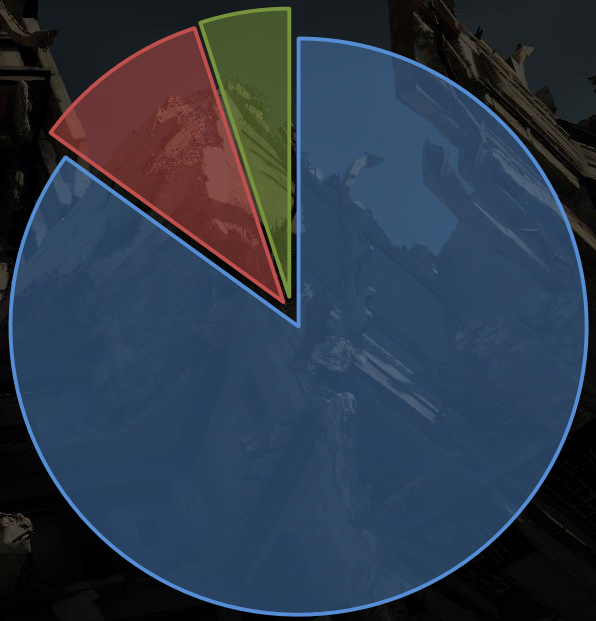
Practical example: Moving a group



Finally update the game

Results, single frame

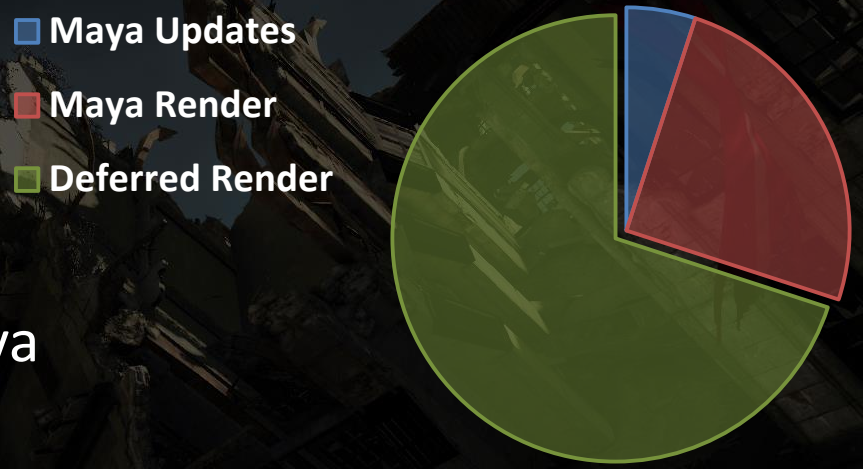
- Maya Updates
- Maya Render
- Deferred Render



Single frame CPU time distribution

Results, single frame

- Performance increase 40x
- Most CPU time spend rendering
- Much *better integration* with Maya
  - Hidden object groups
  - view selected
  - Maya LOD groups



Single frame CPU time distribution


Autodesk Maya 2013-7.1.SCEx64: D:\dev\KZ4\Tools\MayaTools\KZ4\projects\default\scenes**HUB**.mb* --- spotLight715
File Edit Modify Create Display Window Assets Select Mesh Edit Mesh Proxy Normals Color Create UVs Edit UVs MJ Poly Tools 1.4 SCEE Tools Bonus Tools Guerrilla Tactics © Muscle Pipeline Cache Help
General Curves Surfaces Polygons Subdivs Deformation Animation Dynamics Rendering PaintEffects Toon Muscle Fluids Fur Hair nCloth
View Shading Lighting Show Renderer Panels
Attribute Editor
List Selected Focus Attributes Show Help
spotLight715 spotLightShape715 defaultLightSet
spotLight: spotLightShape715
Focus Presets Show Hide
Sample
Light Shape
Spot Light Attributes
Type Spot Light
Color
Intensity 1.345
Illuminates by Default
Emit Diffuse
Emit Specular
Decay Rate No Decay
Cone Angle 67.671
Penumbra Angle -10.000
Dropoff 0.000
CORE Lighting Options
Shader Input
Cast Prelit Shadows
Influence Accuracy: Auto
Layer operation: ADD
Cast Dynamic Shadows
Use Prebuilt Shadow Geometry? Auto
Cast Forward Shadows Auto
Cast Volumetric Light
Notes: spotLightShape715
Select Load Attributes Copy Tab
MEL
No Anim Layer
No Character Set

Scene file considerations

Killzone 3 (Playstation 3)

KILLZONE SHADOW FALL

- Released in February 2011
- Average team size ~120
- 5TB of working data

Killzone Shadow Fall (Playstation 4)

KILLZONE
SHADOW FALL

- Released in November 2013
- Average team size ~160
- 12TB of working data
  - *Larger Maya files*
  - *Longer load/export time*

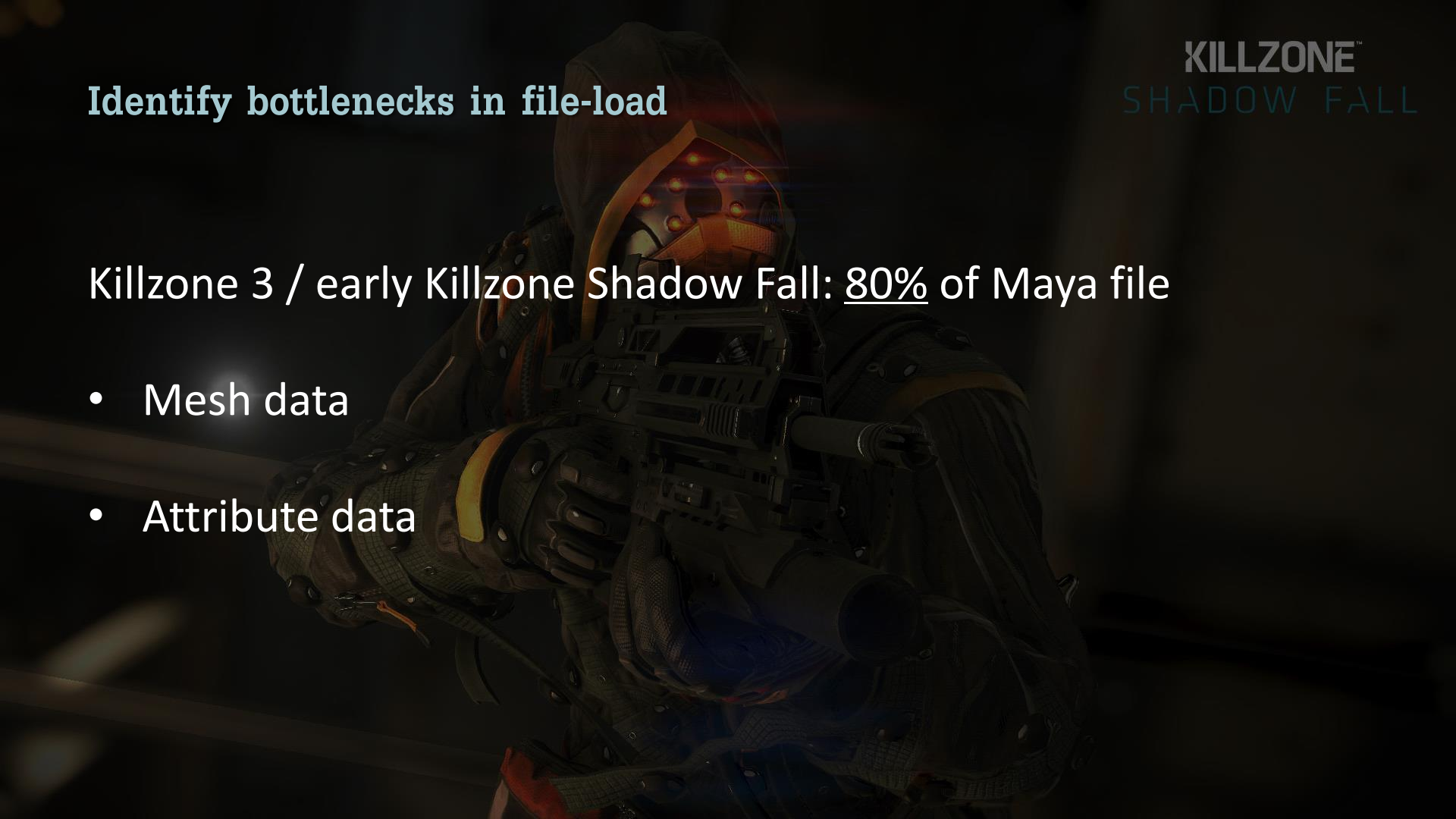
Reducing file-load times

Identify bottlenecks in file-load

KILLZONE SHADOW FALL

In general: Small files load faster

```
4   //Codeset: 1252
5   requires maya "2014";
6   currentUnit -l centimeter -a degree -t film;
7   fileInfo "application" "maya";
8   fileInfo "product" "Maya 2014";
9   fileInfo "version" "2014";
10  createNode transform -s -n "persp";
11      setAttr ".v" no;
12      setAttr ".t" -type "double3" 19.261586240612957 14.51297538424279 18.471197103062892 ;
13      setAttr ".r" -type "double3" -28.538352729602373 46.19999999999996 -2.2976132695769557e-015 ;
14  createNode camera -s -n "perspShape" -p "persp";
15      setAttr -k off ".v" no;
16      setAttr ".fl" 34.999999999999993;
17      setAttr ".coi" 30.37795716351587;
18      setAttr ".imn" -type "string" "persp";
19      setAttr ".den" -type "string" "persp_depth";
```



Identify bottlenecks in file-load

Killzone 3 / early Killzone Shadow Fall: 80% of Maya file

- Mesh data
- Attribute data

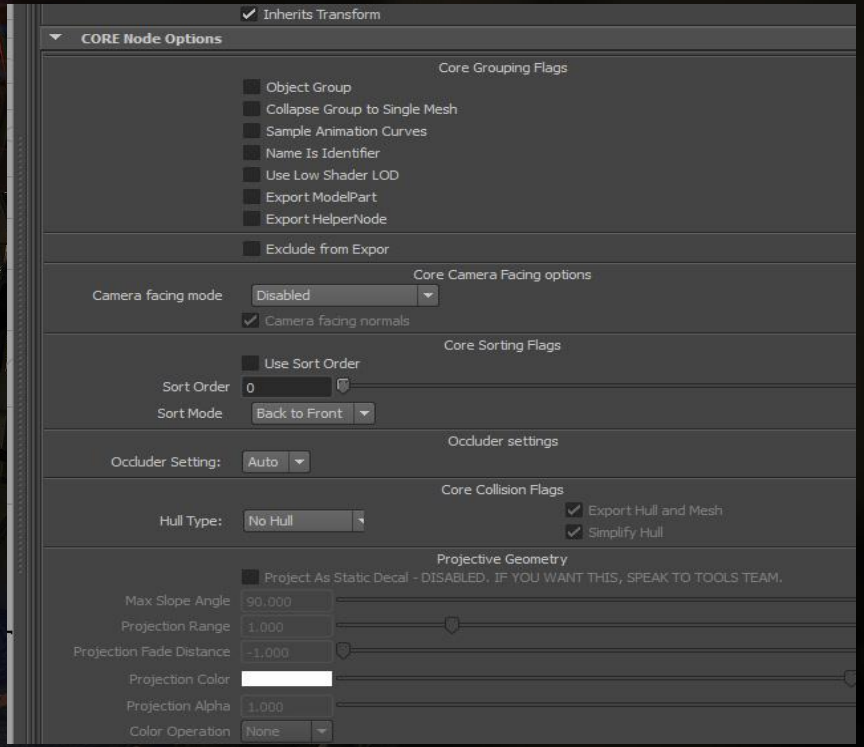
Attribute data

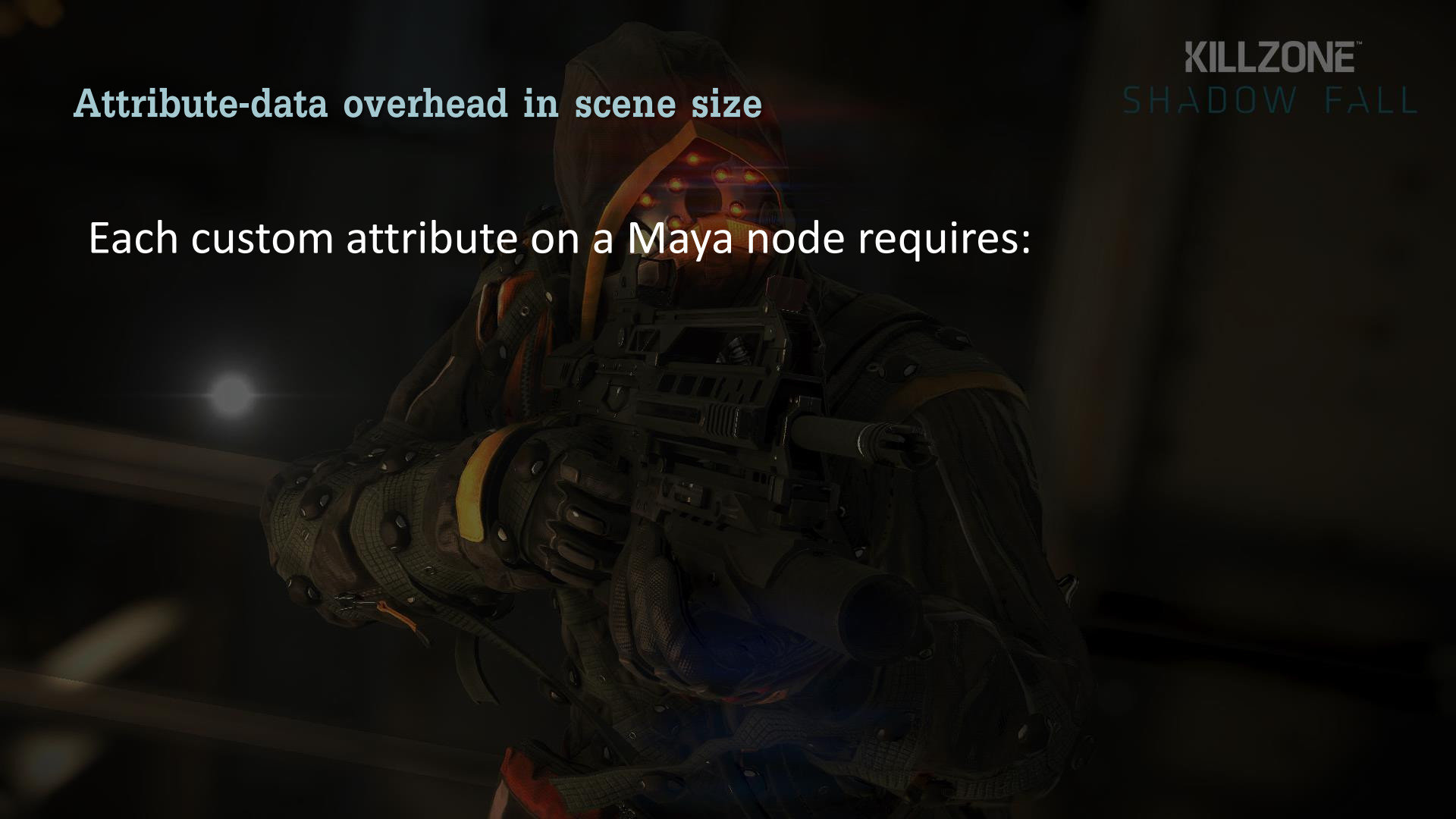
KILLZONE SHADOW FALL

Custom attributes on native Maya nodes

Export-time state:

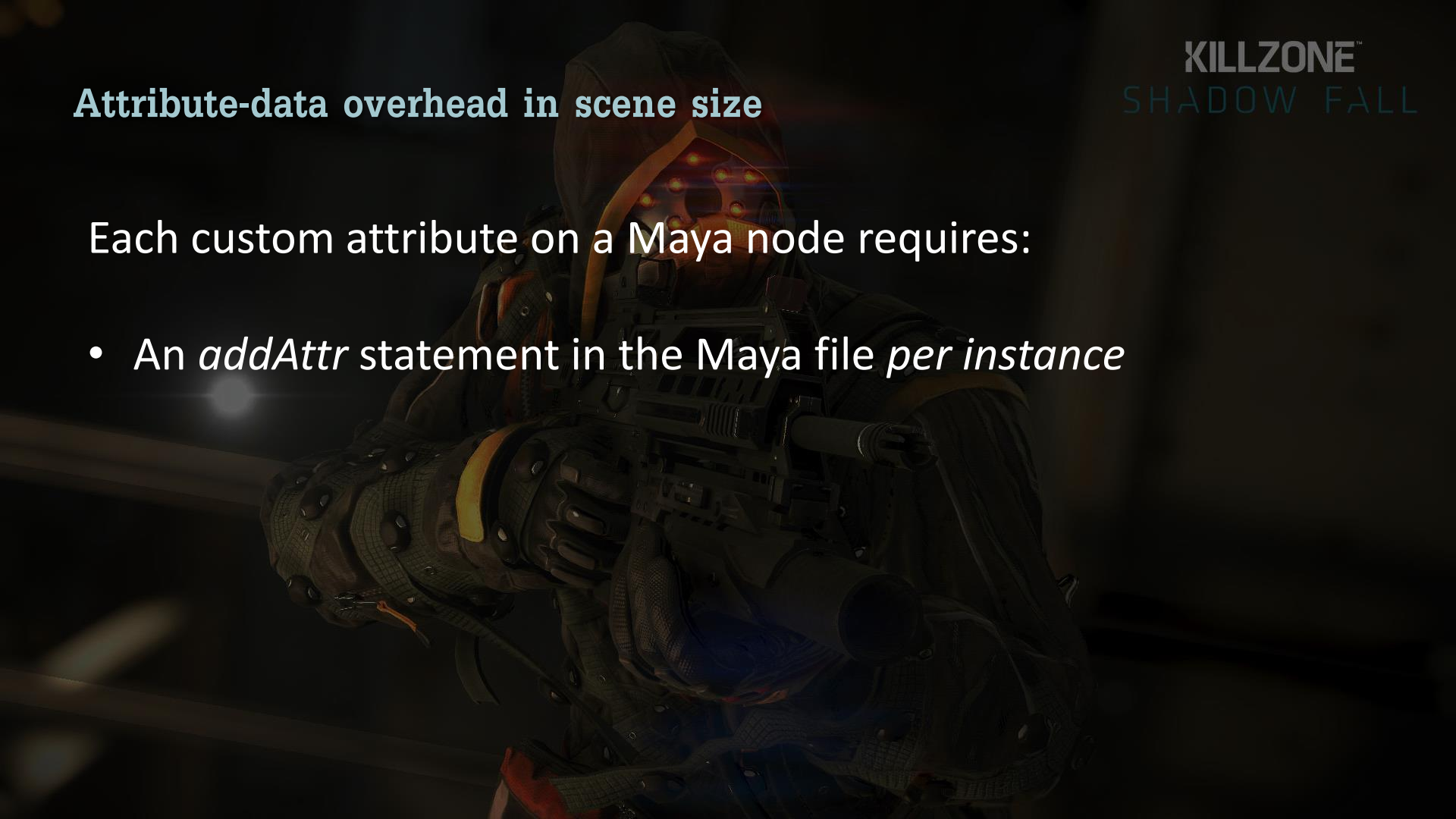
- Collapsing
- Physics
- Destructability
- Etc.





Attribute-data overhead in scene size

Each custom attribute on a Maya node requires:



Attribute-data overhead in scene size

Each custom attribute on a Maya node requires:

- An *addAttr* statement in the Maya file *per instance*

Attribute-data overhead in scene size

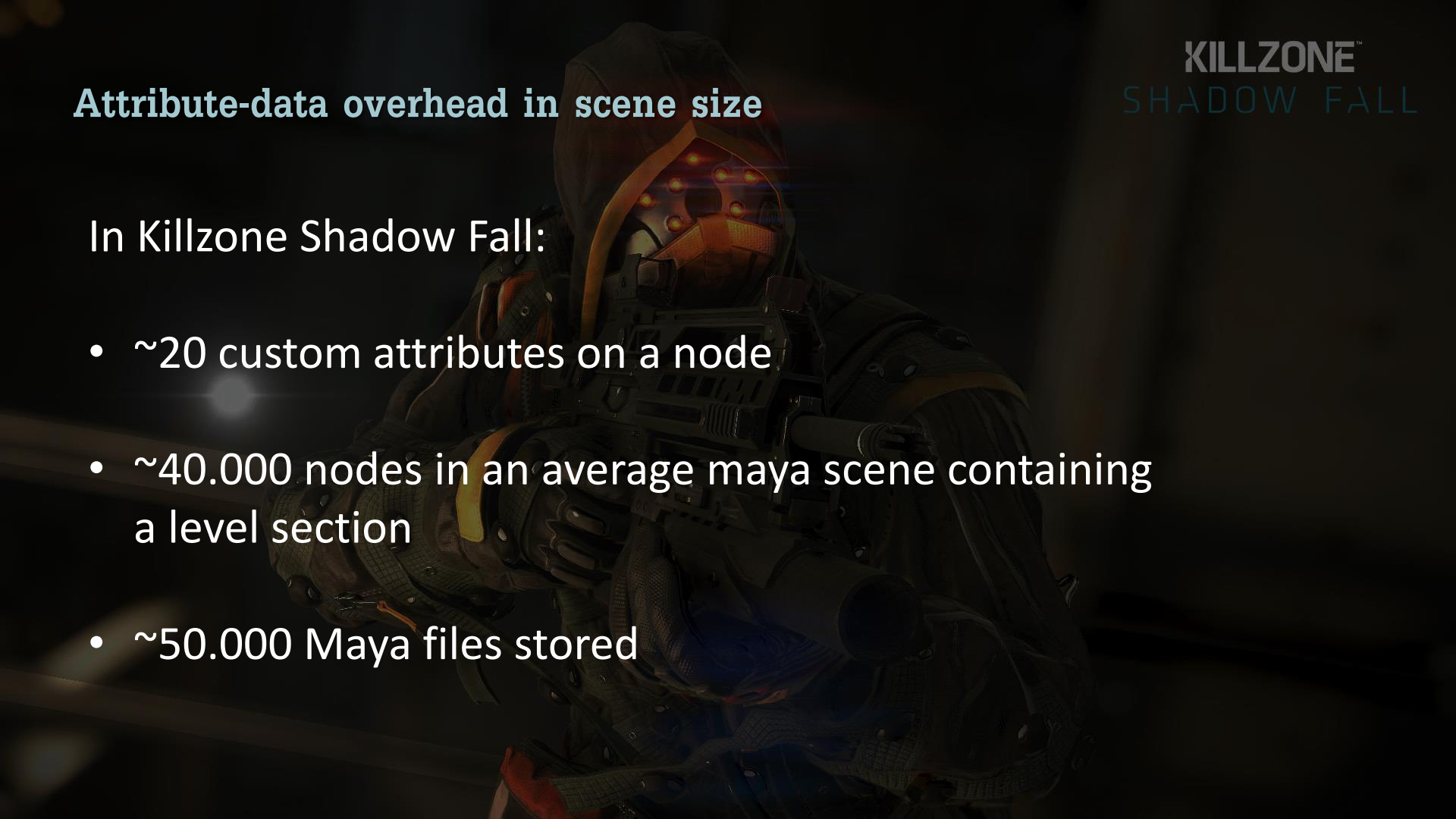
Each custom attribute on a Maya node requires:

- An *addAttr* statement in the Maya file *per instance*
- A *setAttr* statement in the Maya file *per instance*(optional)

Attribute-data overhead in scene size

Each custom attribute on a Maya node requires:

- An *addAttr* statement in the Maya file *per instance*
- A *setAttr* statement in the Maya file *per instance*(optional)
- A *nodeAdded* callback to install the attributes on a newly instanced maya node

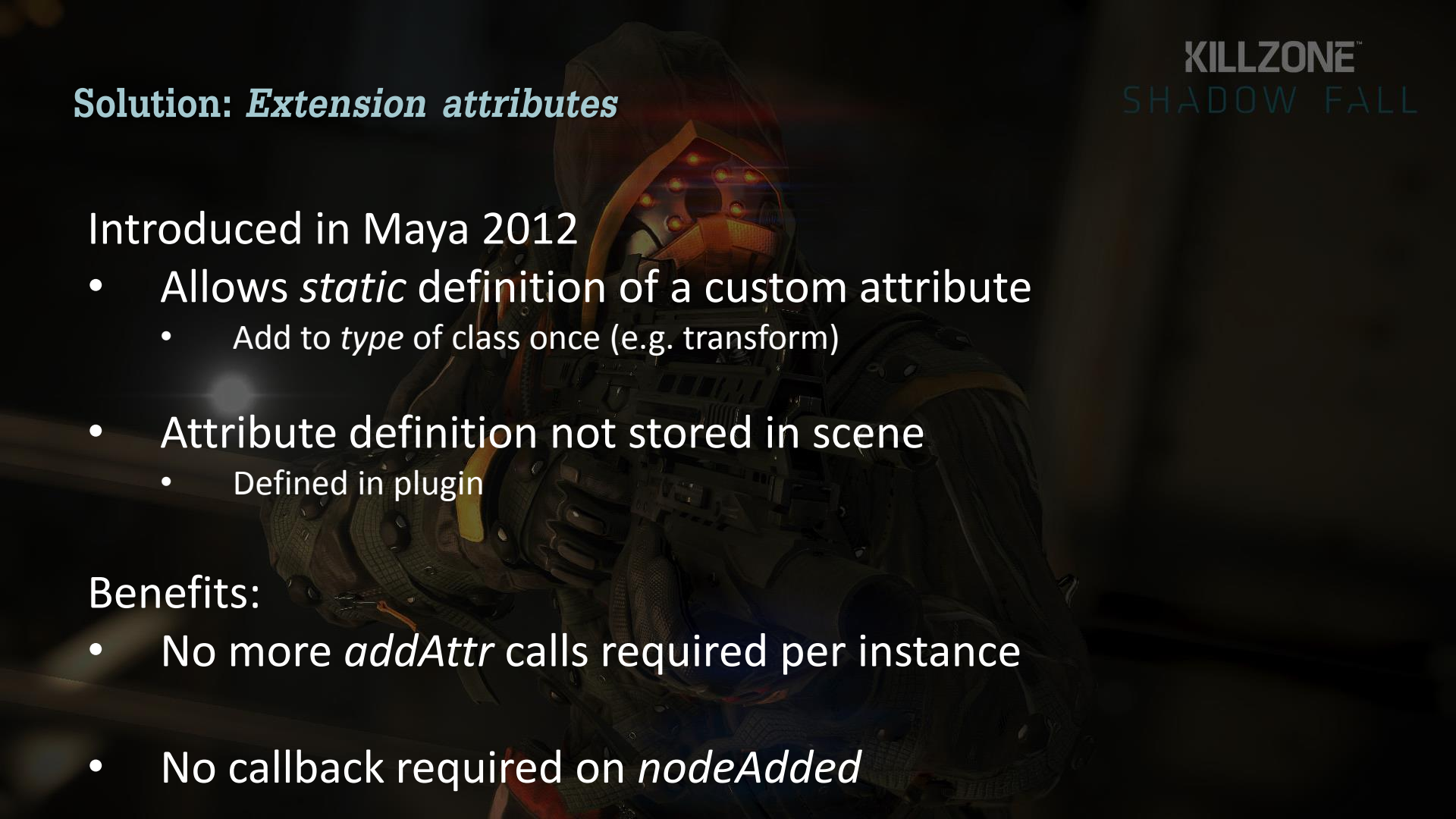


Attribute-data overhead in scene size

KILLZONE SHADOW FALL

In Killzone Shadow Fall:

- ~20 custom attributes on a node
- ~40.000 nodes in an average maya scene containing a level section
- ~50.000 Maya files stored



Solution: *Extension attributes*

KILLZONE SHADOW FALL

Introduced in Maya 2012

- Allows *static* definition of a custom attribute
  - Add to *type* of class once (e.g. transform)
- Attribute definition not stored in scene
  - Defined in plugin

Benefits:

- No more *addAttr* calls required per instance
- No callback required on *nodeAdded*

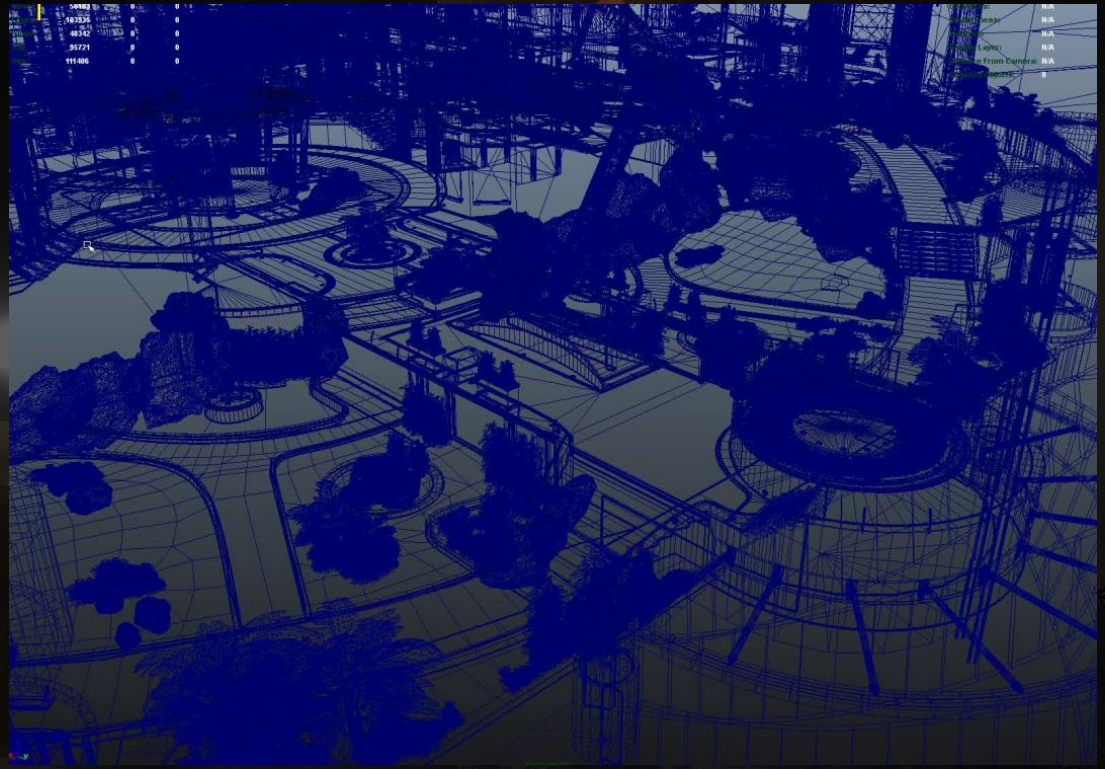
Result: Switching to *Extension attributes*

Extension attributes results:

- Static scene (20.000 cubes)

	Dynamic attributes	Extension attributes	Difference
Scene size	85MB	15MB	- 80%
File load time	45 sec.	14 sec.	- 66%

Mesh-data overhead in scene size



Mesh usage in Killzone 3

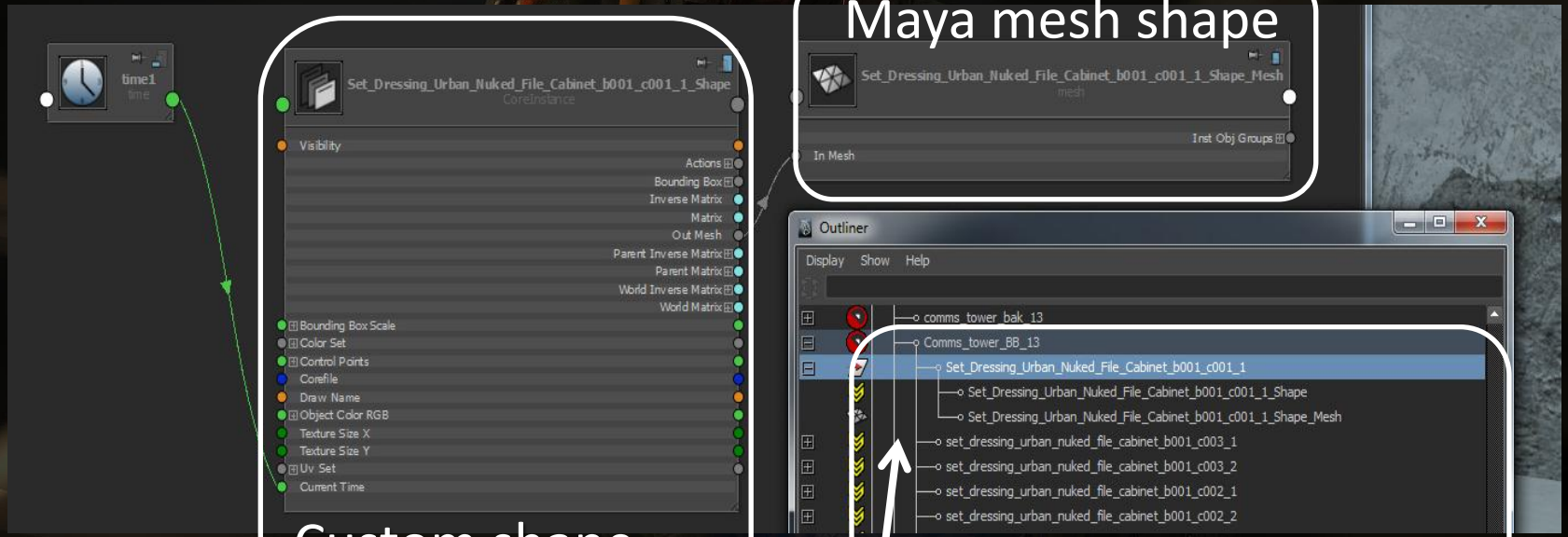
KILLZONE SHADOW FALL

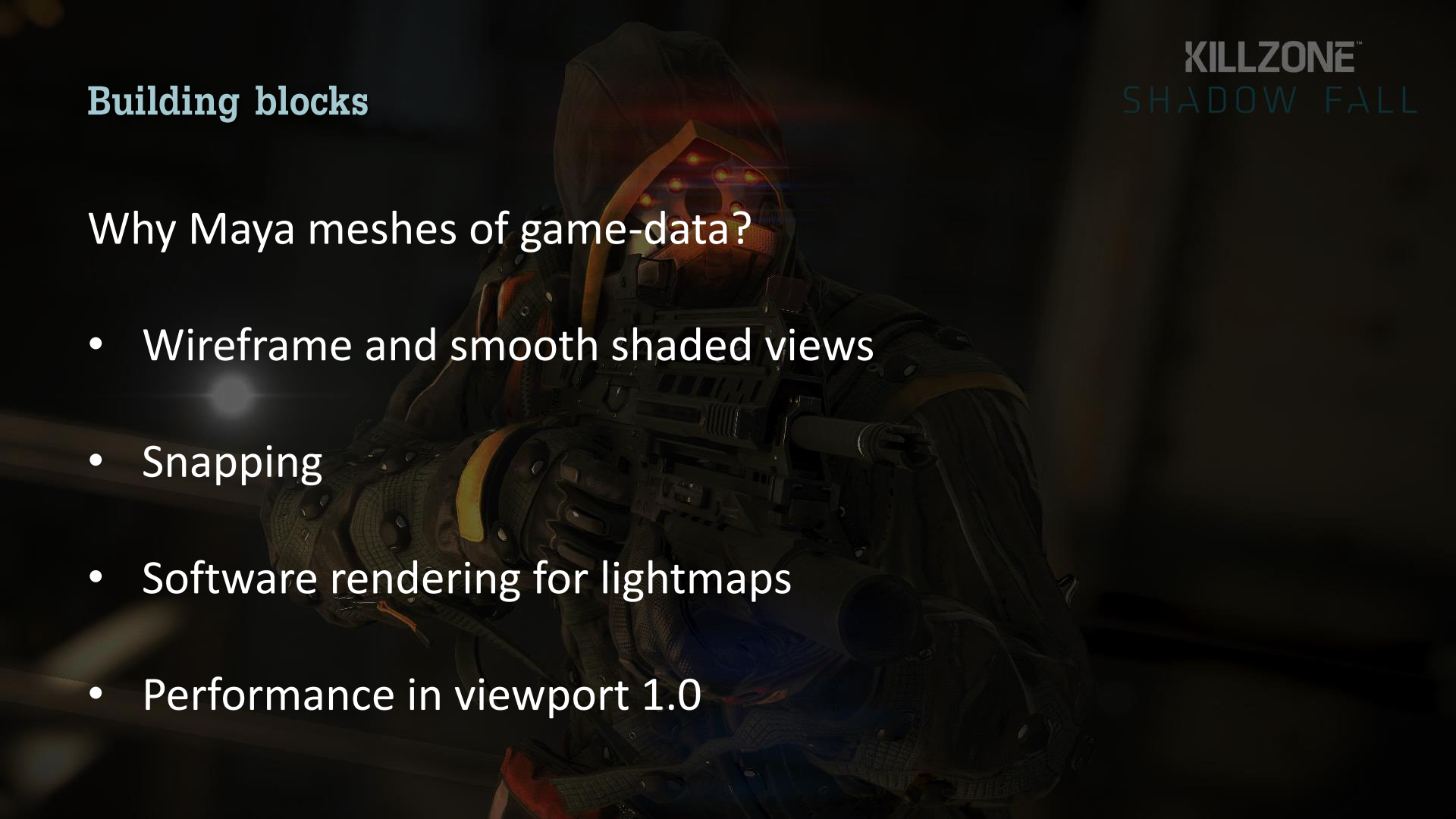
Building blocks

- Developed during Killzone 2
- Referencing of game-data
  - Small section of re-usable static geometry
- Custom shape *(MpxSurfaceShape)*

Building blocks

Custom shape generates a Maya mesh





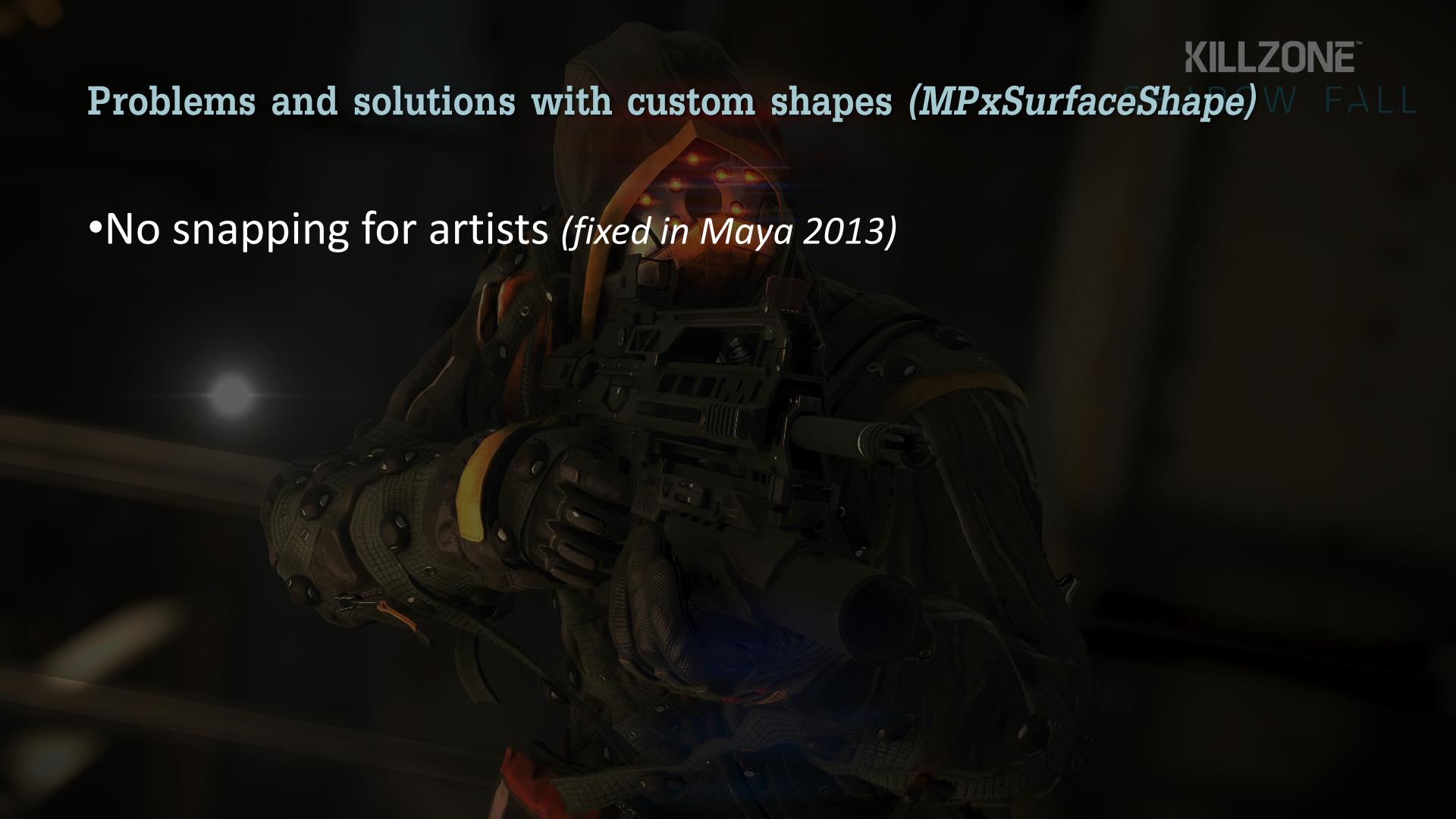
Building blocks

Why Maya meshes of game-data?

- Wireframe and smooth shaded views
- Snapping
- Software rendering for lightmaps
- Performance in viewport 1.0

Problems and solutions with custom shapes *(MPxSurfaceShape)*

- No snapping for artists *(fixed in Maya 2013)*



Problems and solutions with custom shapes *(MPxSurfaceShape)*

- No snapping for artists *(fixed in Maya 2013)*
- Custom shapes need to *feel* as Maya meshes for artists.
  - Draw overrides, display layers, colors

Problems and solutions with custom shapes *(MPxSurfaceShape)*

- No snapping for artists *(fixed in Maya 2013)*
- Custom shapes need to *feel* as Maya meshes for artists.
  - Draw overrides, display layers, colors
- Create maya meshes on demand for lightmaps
  - Do not save these to file *(delete after use)*

Problems and solutions with custom shapes *(MPxSurfaceShape)*

- No snapping for artists *(fixed in Maya 2013)*
- Custom shapes need to *feel* as Maya meshes for artists.
  - Draw overrides, display layers, colors
- Create maya meshes on demand for lightmaps
  - Do not save these to file *(delete after use)*
- Drawing slow due to OpenGL state switching
  - Use *MpxModelView*, or **viewport 2.0**


Viewport 2.0
KILLZONE
SHADOW FALL

What is viewport 2.0?

A new viewport for Autodesk Maya

- Designed for current generation hardware
  - *Less state switching, much less CPU overhead*
- A new API for plugin drawing
  - *No more MPxLocatorNode::draw, MPxSurfaceShapeUI, etc*

New possibilities of viewport 2.0

The new viewport 2.0 API offers:

- More performance
  - *Maya meshes at least x10*
- More controlled integration with Maya
  - *Depth buffer sharing*
- New features
  - *Mix and match MRenderOperations*

Viewport 2.0: Integration

Deferred renderer based on “Viewport 1”

- (*MViewportRenderer* class)
- Little control/integration with Maya

Artists lacked:

- Selection highlighting
- Depth between Maya and game geometry incorrect (locators/lights)
- No display of Maya meshes

Comparing *deferred* and *deferred_vp2*


Vertices: N/A
Smoothness: N/A
Instance: N/A
Display Layer: N/A
Distance From Camera: N/A
Selected Objects: 0
BACK


Viewport 2.0
Instance:
Display Layer:
Distance From Camera:
Selected Objects:
N/A
N/A
N/A
N/A
0
=Visible[I:8795.40]
=Visible[I:4397.70]
=Visible[I:5968.97]
=Visible[I:990.75]
=Visible[I:4069.03]
=Visible[I:4953.74]
=Visible[I:263.43]
=Visible[I:2490.45]

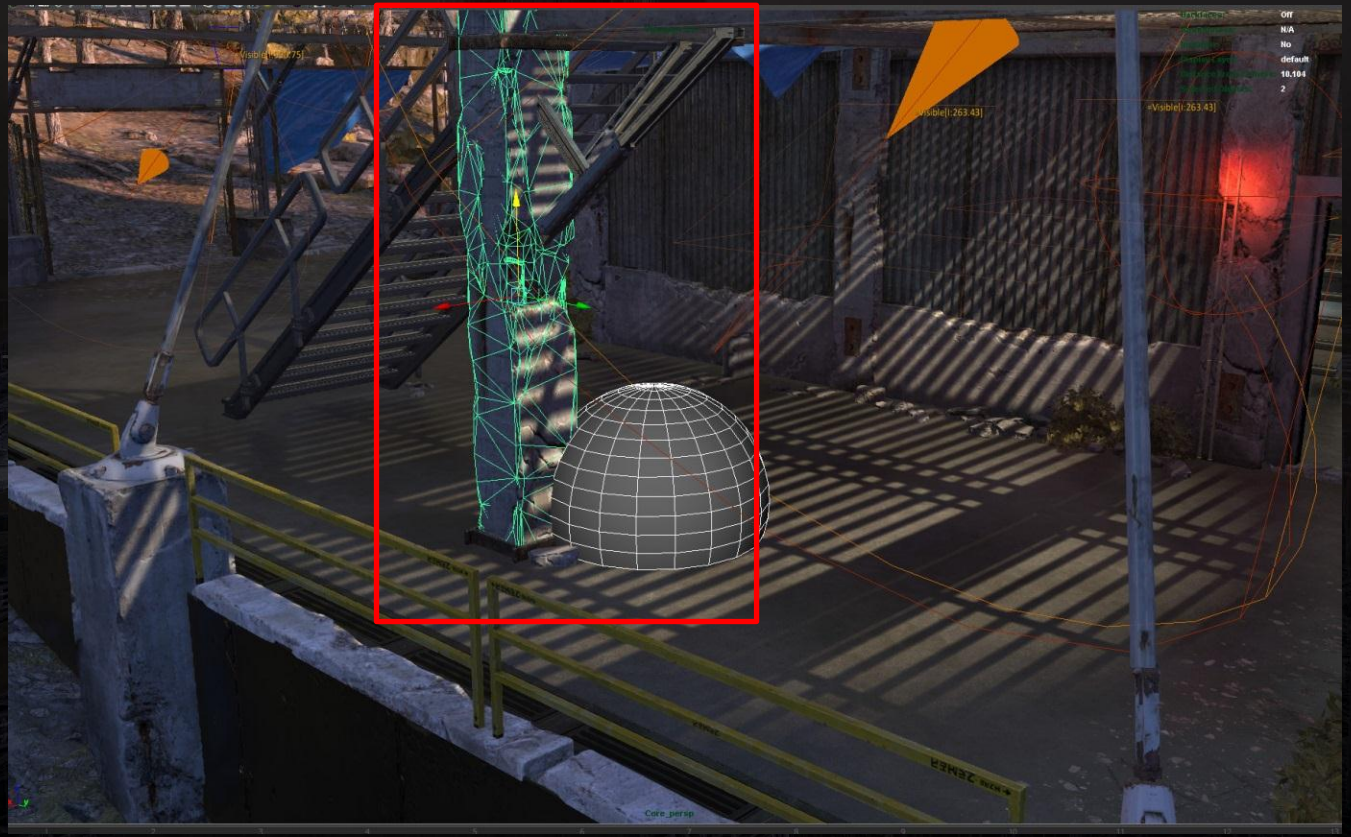
Comparing *deferred* and *deferred_vp2*

Proper depth support, no more clutter!

Comparing *deferred* and *deferred_vp2*

Proper depth support, no more clutter!

Comparing *deferred* and *deferred_vp2*



Viewport 2.0: High level API overview

How does it work?

MRenderOverride

Custom Viewport 2.0 renderer

- **MRenderOperation**

  A pass in MRenderOverride

  - **MSceneRender**

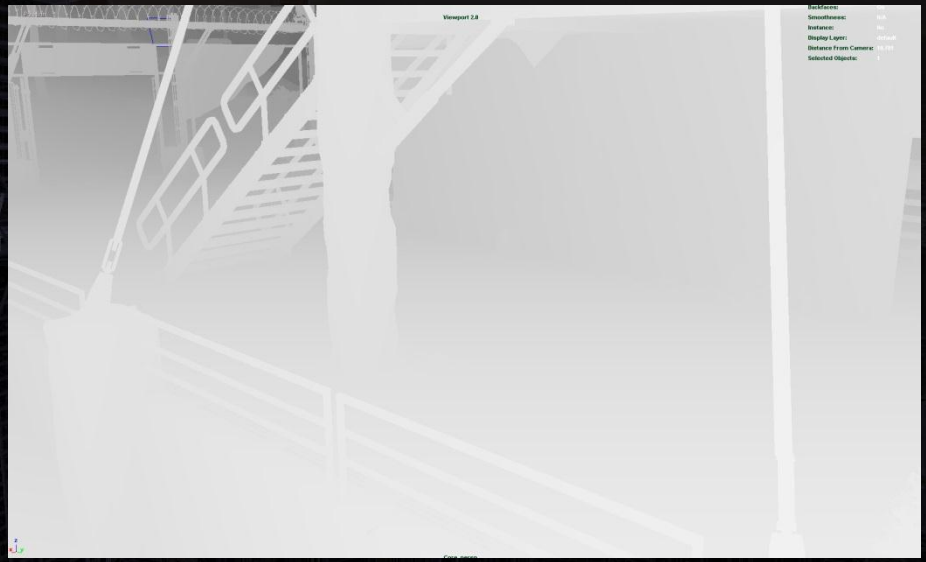
    Maya's viewport 2 rendering pass

  - **MUserRenderOperation**

    Do your own!

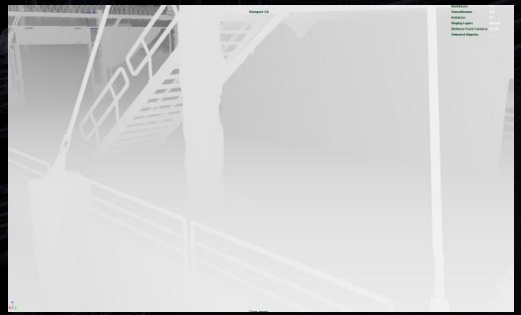
Deferred Viewport 2.0: Rendering operations

Render deferred beauty and depth *(MUserRenderOperation)*



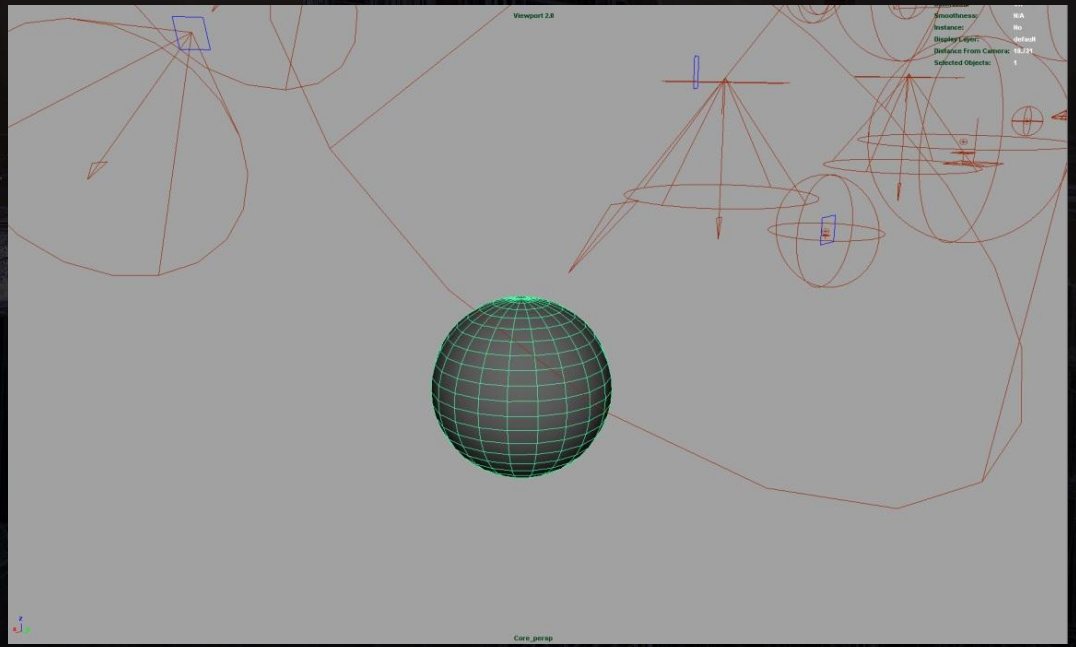
Deferred Viewport 2.0: Rendering operations

Copy to Maya's buffers *(MQuadRender)*

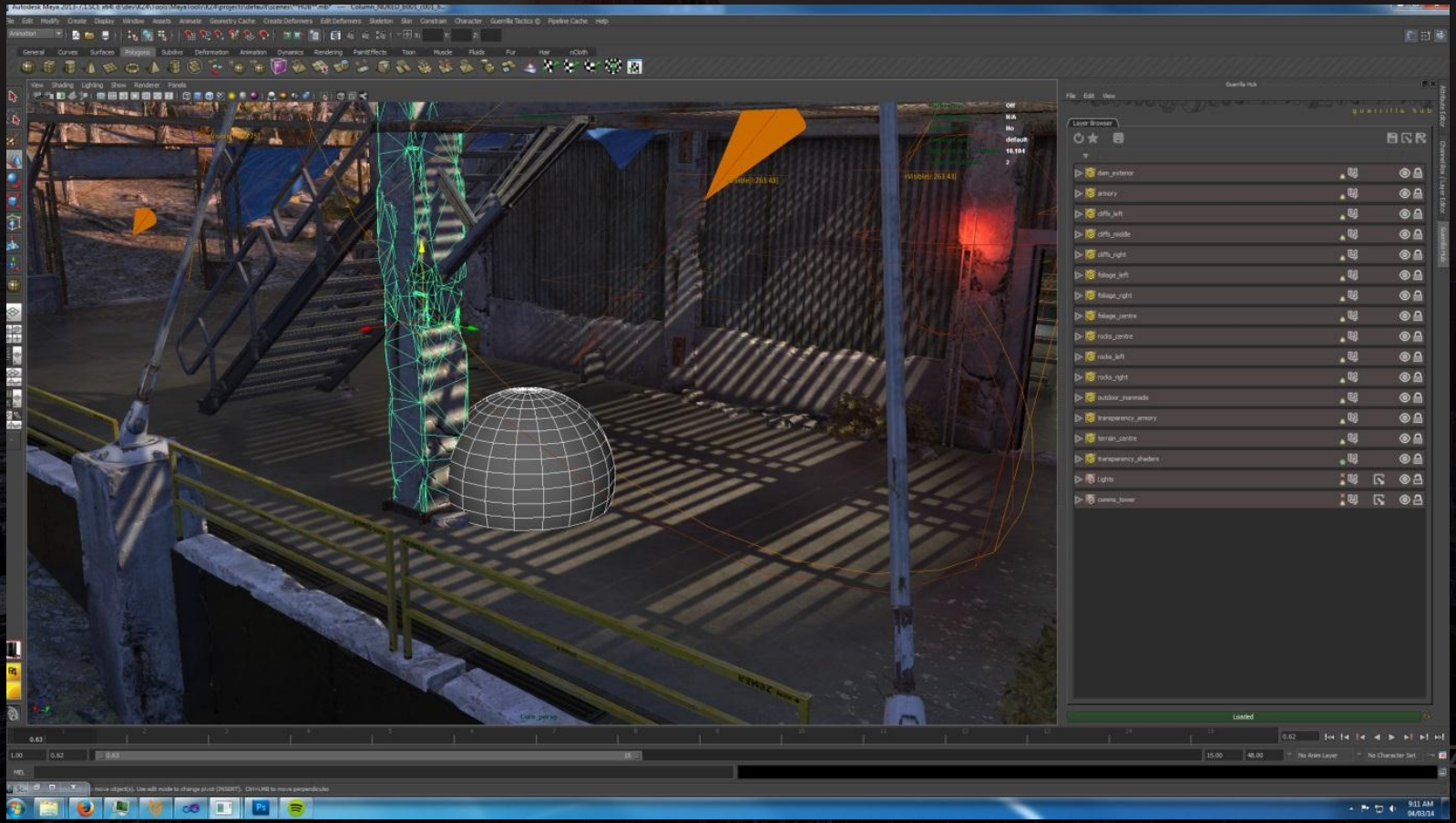


Deferred Viewport 2.0: Rendering operations

Render Maya's view on top, preserving depth *(MSceneRender)*

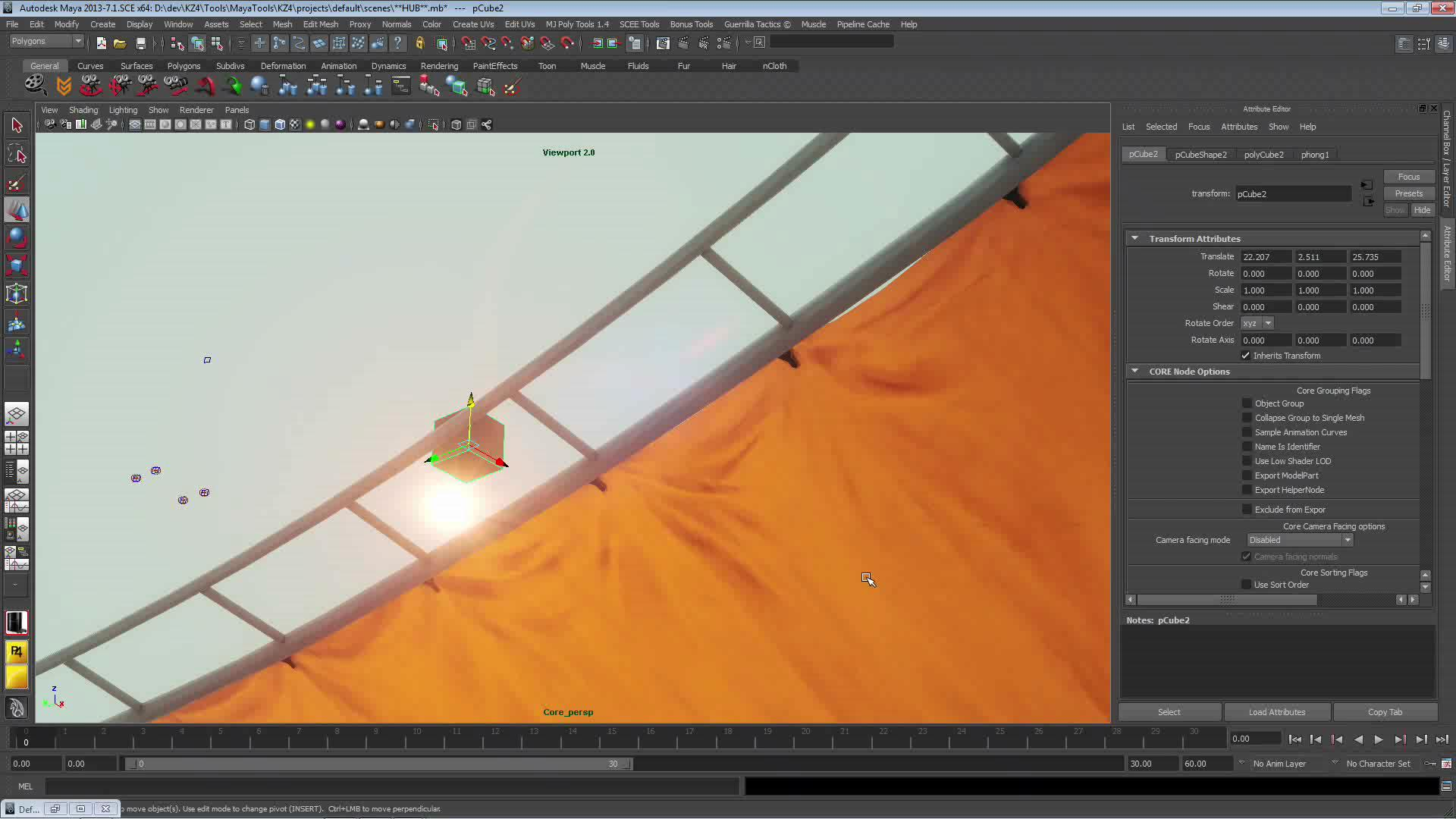


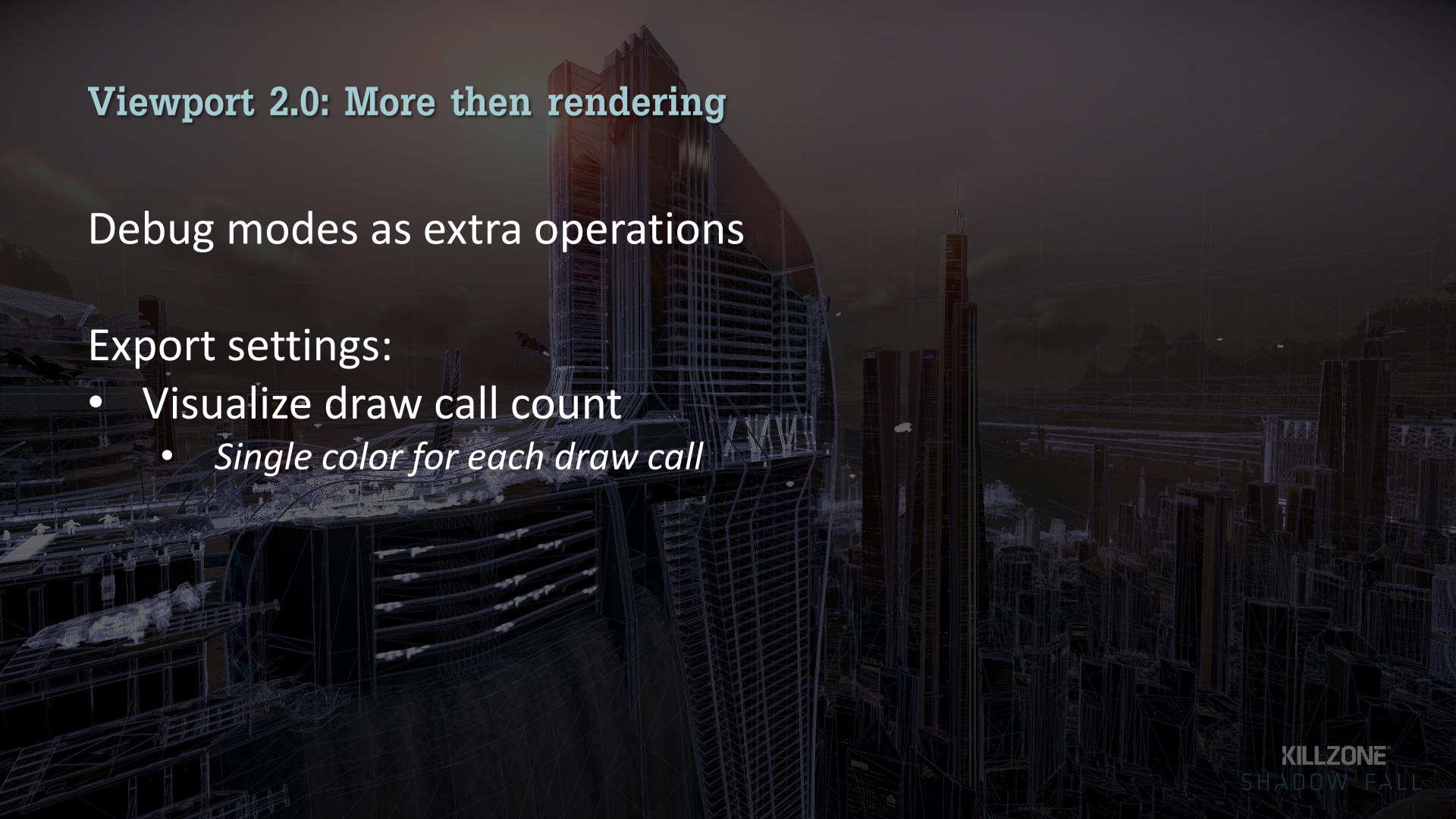
Deferred Viewport 2.0: Rendering stages



Viewport 2.0: Integration advantages

- Allows artists to work directly with *in-game* rendering
  - *Selection highlighting, depth*
- Full support for Maya native geometry/manipulators etc.
  - *Full control over what maya draws*
- Rendering API Agnostic (DirectX or OpenGL)


Autodesk Maya 2013-7.1.SCE x64: D:\dev\KZ4\Tools\MayaTools\KZ4\projects\default\scenes**HUB**.mb* --- pCube2
File Edit Modify Create Display Window Assets Select Mesh Edit Mesh Proxy Normals Color Create UVs Edit UVs MJ Poly Tools 1.4 SCEE Tools Bonus Tools Guerrilla Tactics © Muscle Pipeline Cache Help
Polygons
General Curves Surfaces Polygons Subdivs Deformation Animation Dynamics Rendering PaintEffects Toon Muscle Fluids Fur Hair nCloth
View Shading Lighting Show Renderer Panels
Viewport 2.0
Core_persp
Attribute Editor
List Selected Focus Attributes Show Help
pCube2 pCubeShape2 polyCube2 phong1
transform: pCube2
Focus
Presets
Show Hide
Transform Attributes
Translate 22.207 2.511 25.735
Rotate 0.000 0.000 0.000
Scale 1.000 1.000 1.000
Shear 0.000 0.000 0.000
Rotate Order xyz
Rotate Axis 0.000 0.000 0.000
Inherits Transform
CORE Node Options
Core Grouping Flags
Object Group
Collapse Group to Single Mesh
Sample Animation Curves
Name Is Identifier
Use Low Shader LOD
Export ModelPart
Export HelperNode
Exclude from Expor
Core Camera Facing options
Camera facing mode Disabled
Camera facing normals
Core Sorting Flags
Use Sort Order
Notes: pCube2
Select Load Attributes Copy Tab
Channel Box / Layer Editor
Attribute Editor
No Anim Layer
No Character Set
MEL
move object(s). Use edit mode to change pivot (INSERT). Ctrl+LMB to move perpendicular.

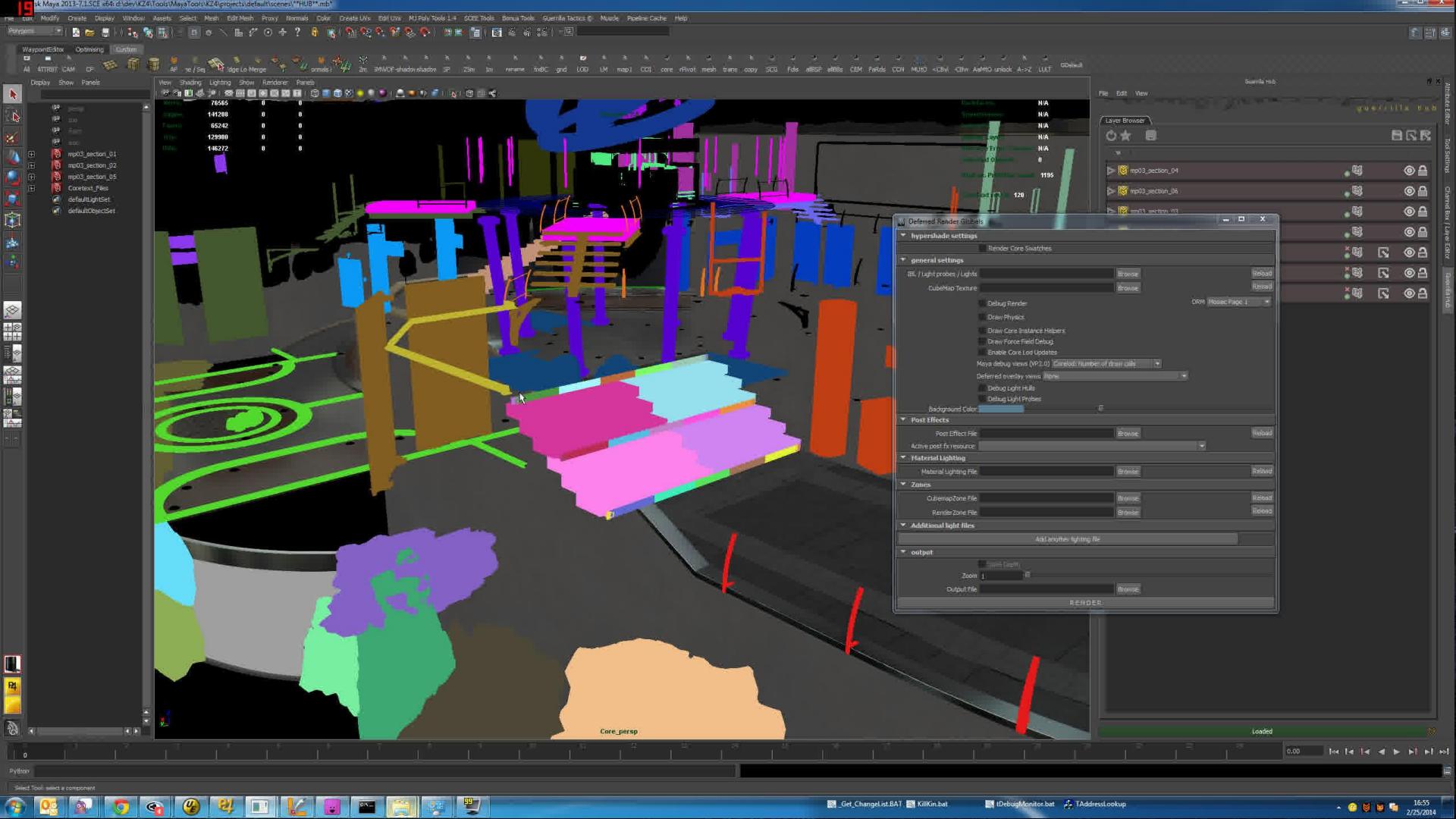


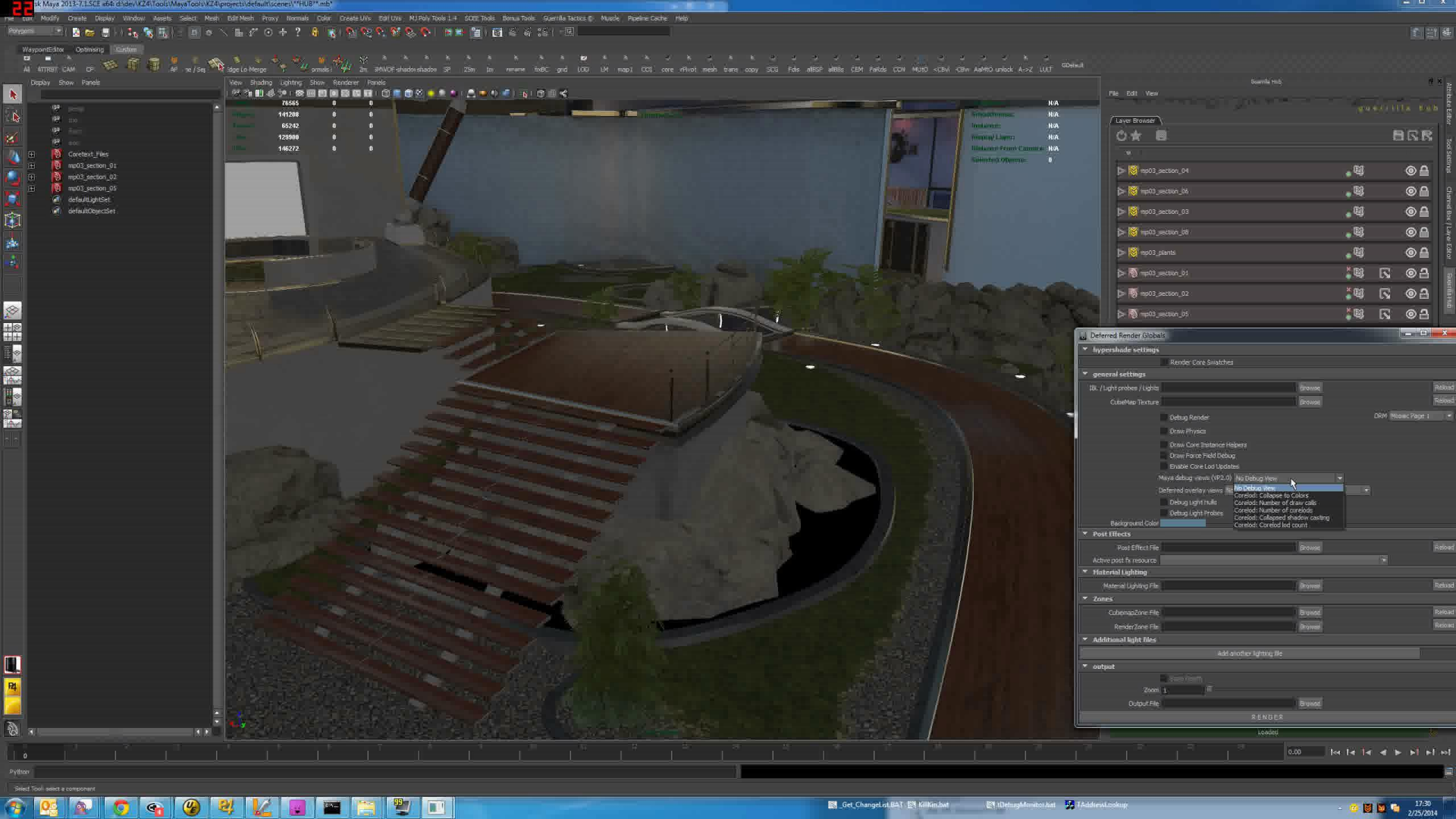
Viewport 2.0: More then rendering

Debug modes as extra operations

Export settings:

- Visualize draw call count
  - *Single color for each draw call*


Deferred Render Globals
hypershade settings
Render Core Swatches
general settings
IBL / Light probes / Lights
CubeMap Texture
Debug Render
Draw Physics
Draw Core Instance Helpers
Draw Force Field Debug
Enable Core Lod Updates
Maya debug views (VP1.0)
Deferred overlay views
Debug Light Hulls
Debug Light Probes
Background Color
Post Effects
Post Effect File
Active post fx resource
Material Lighting
Material Lighting File
Zones
CubemapZone File
RenderZone File
Additional light files
Add another lighting file
output
Zoom
Output File
RENDER
Browse
Reload
Layer Browser
Loaded
Core_persp


Deferred Render Globals
hypershade settings
Render Core Switches
general settings
IBL / Light probes / Lights
CubeMap Texture
Debug Render
Draw Physics
Draw Core Instance Helpers
Draw Force Field Debug
Enable Core Lod Updates
Maya debug views (VP2.0)
No Debug View
Coreload: Collapse to Colors
Coreload: Number of draw calls
Coreload: Number of coreloids
Coreload: Collapsed shadow casting
Coreload: Coreload lod count
Debug Light Hulls
Debug Light Probes
Background Color
Post Effects
Post Effect File
Active post fx resource
Material Lighting
Material Lighting File
Zones
CubemapZone File
RenderZone File
Additional light files
Add another lighting file
output
Zoom
Output File
RENDER
Layer Browser
mp03_section_04
mp03_section_06
mp03_section_03
mp03_section_00
mp03_plants
mp03_section_01
mp03_section_02
mp03_section_05
Coretext_Files
defaultLightSet
defaultObjectSet

The Future

- Bringing Maya and game closer together
  - Background scene loading (streaming)
  - Storing scene elements in game-format only
  - Multithreading
- Direct X rendering

Conclusions

- Integrating game-engine in Maya improves productivity
- Carefully thinking on scene configuration saves loads!
- Viewport 2.0 provides flexibility and new possibilities


Questions?
KILLZONE
SHADOW FALL


Thank you
Visit us at guerrilla-games.com
KILLZONE
SHADOW FALL