

# The Technology of Uncharted: Drake's Fortune

Naughty Dog

Christophe Balestra, Co-President  
Pål-Kristian Engstad, Lead Programmer



UNCHARTED

NAUGHTY DOG



2008 GAME DEVELOPERS CONFERENCE



# Introduction

- Project Code Name: 'BIG'
- 3 year project
- Pre-production: ~ 10 people for 12 months
- ~ 70 people + ~ 5 contractors
- 6 designers, 18 programmers, ~ 50 artists



# Development

- Started the project with ZERO line of code
- Prototyped shaders on PC
- Animation was priority #1
- Over-designed our first set of tools



# Tools

- Keep them SIMPLE!!!
- Few tools rather than a big über tool
- Expand your current set of tools



# Cygwin

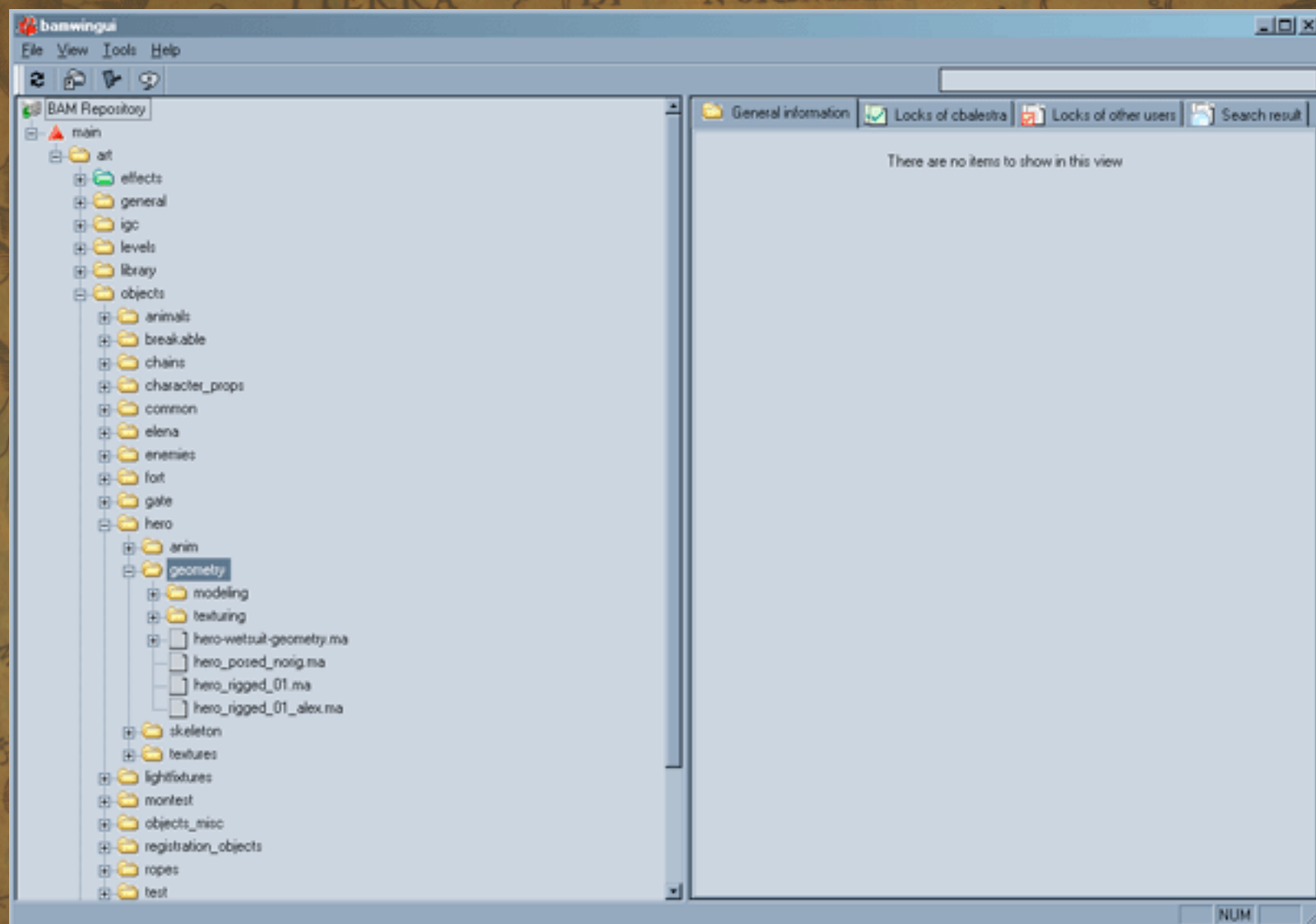
- Like a linux shell
- Helps controlling people's environment
- All tools must be run from Cygwin



# BAM

- Asset Manager
- Everything on the network
- Linux server - symlinks
- Check-out, Check-in but NO SYNC



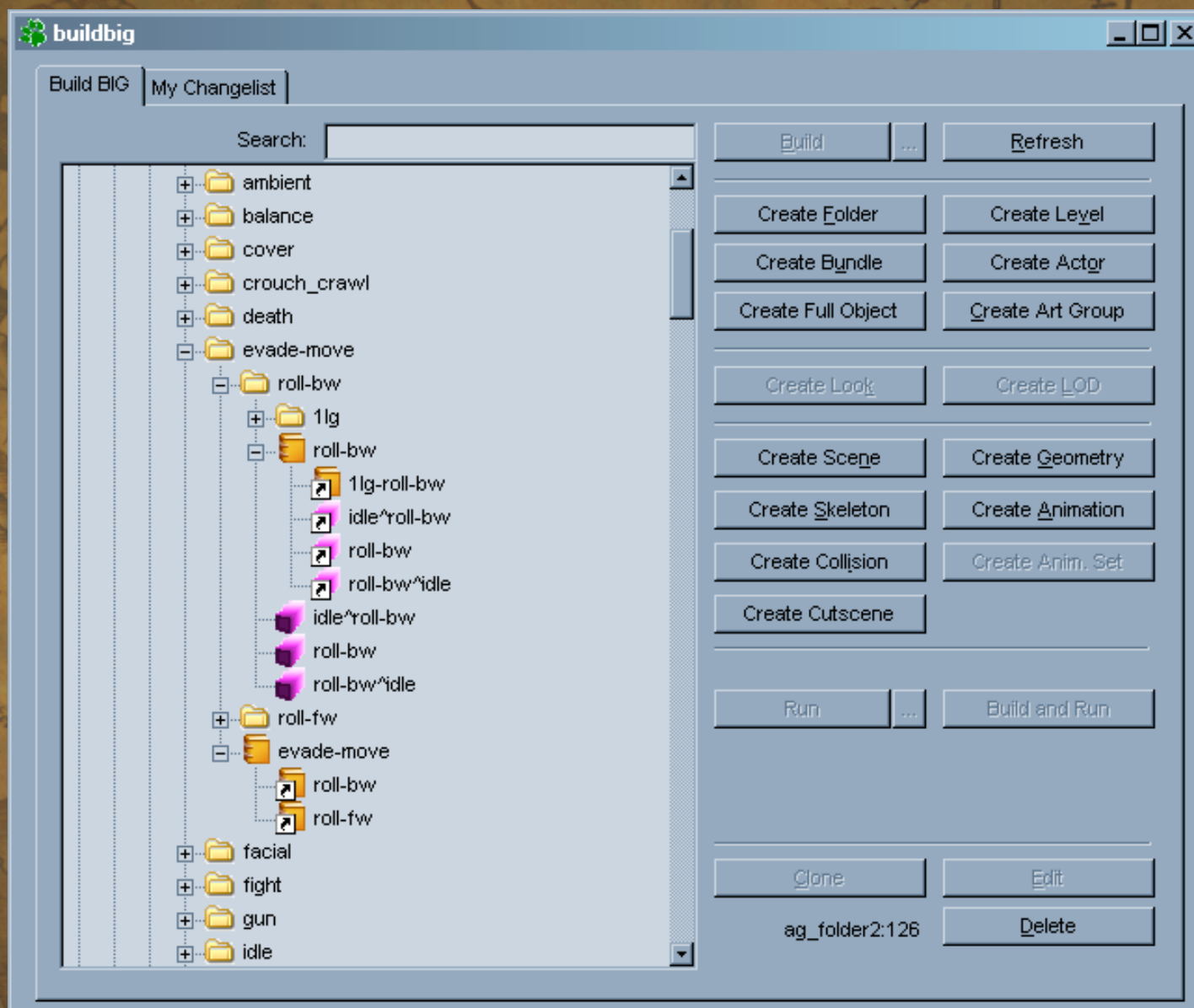




# BuildBig

- GUI tool to describe what actors and levels are made of
- All the data and change lists stored in MySQL:  
Bad idea!!
- Switching to text files and Perforce







# BA / BL

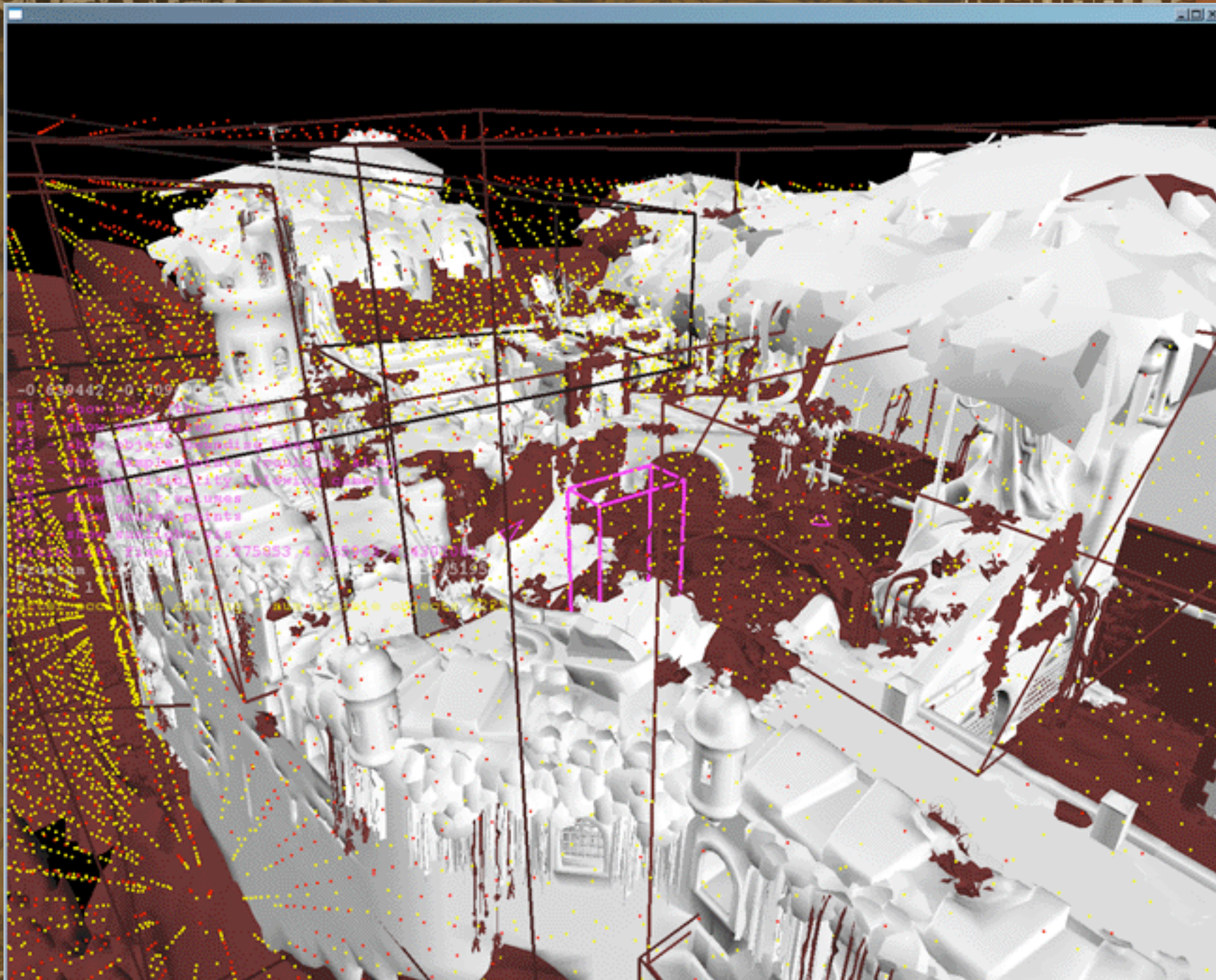
- Very simple command line tools
- BA builds actors
- BL builds levels
- Stats



# Visibility

- During BL a PVS is pre-computed per level
- Rendering the level with OpenGL from sample points
- Computed once a day and re-used cache
- Split boxes
- Can easily add sample points







# Distributed build

- Using Naughty Dog Distributed System
- Very simple
- Runs command line tools
- Everything stored in MySQL



# Shaders

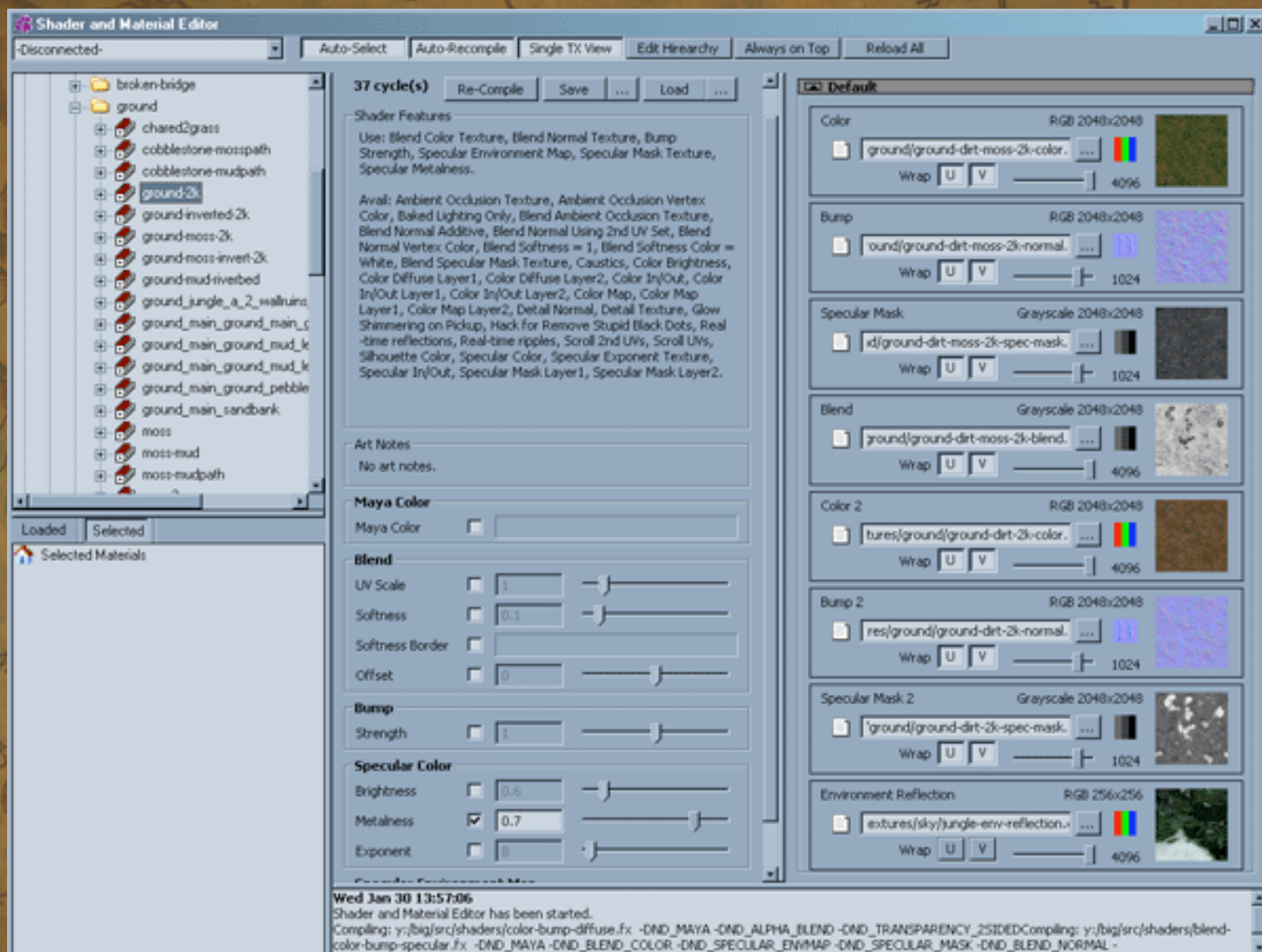
- Über Shader file
- #define = feature
- Not great for dependencies
- Compiled during BA/BL
- Each actor or level contains its shader code



# Material Editor

- GUI tool connected to Maya to create shaders
- Artists select an .fx file and choose features
- Artists loved it but ...
- Not very stable, data & change list stored in MySQL
- Working on connecting it with the game







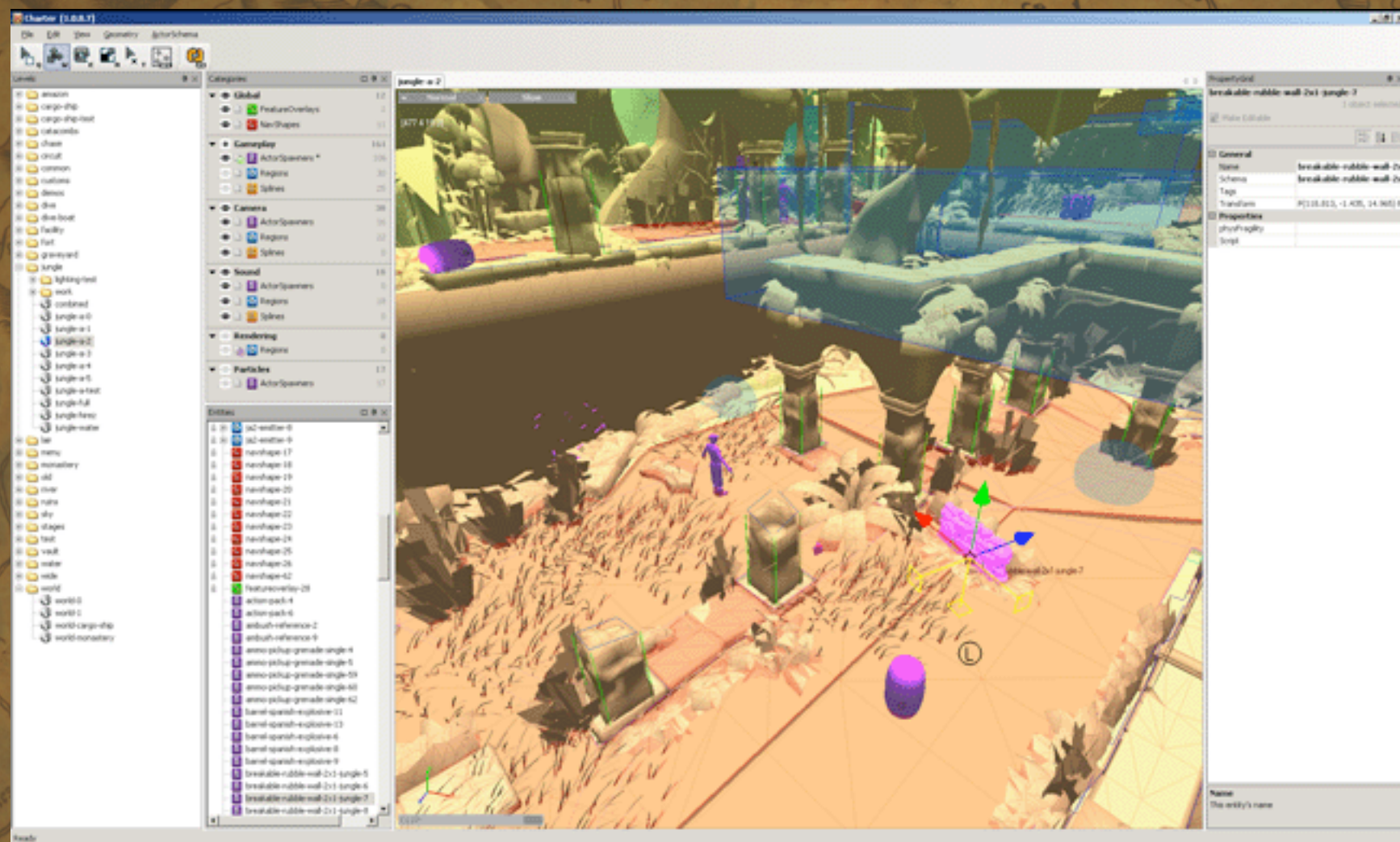
# Charter

- GUI tool to create game-play contents
- Maya too slow
- Regions, Nav-Meshes, Spawners, Cover points...
- Quick iteration time



# UNCHARTED

# NAUGHTY DOG



## 2008 GAME DEVELOPERS CONFERENCE



# DC

- Data Compiler
- LISP based
- Original design was to create data structures
- Added scripting
- Render settings, AI, animation, region scripts
- Realtime linking



# Static Lighting

- Global illumination
- Direct color, indirect color and direction per vertex
- Light probes for objects
- Too slow, couldn't distribute



# TAME

- GUI tool to manage text localization
- HELPED US A LOT!!!
- Producers in Europe and Japan can manage everything without us
- MySQL



Add String

Refresh

Export to Game

Export

Filter

Import

Name Id	Text	Last Update
00A_01A_SEQUENCE_01	I'm here off the coast of Panama,	2007-06-05 11:47:29
00A_01A_SEQUENCE_02	where we've just recovered what we believe	2007-08-06 17:09:08
00A_01A_SEQUENCE_03	to be the coffin of legendary explorer Sir Francis Drake,	2007-08-06 17:09:31
00A_01A_SEQUENCE_04	who was buried at sea over 400 years ago.	2007-08-06 17:09:58
00A_01A_SEQUENCE_05	Are you sure you wanna be defiling	2007-08-06 17:10:13
00A_01A_SEQUENCE_06	your ancestor's remains like that?	2007-08-06 17:10:36
00A_01A_SEQUENCE_07	You make it sound so dirty. (laughs)	2007-08-06 17:10:50
00A_01A_SEQUENCE_08	Besides, I thought you didn't believe me.	2007-08-06 17:11:10
00A_01A_SEQUENCE_09	Well, I did do my research.	2007-08-06 17:11:25
00A_01A_SEQUENCE_10	And apparently, Francis Drake didn't have any children.	2007-08-06 17:11:41
00A_01A_SEQUENCE_11	Well, history can be wrong, you know.	2007-08-06 17:11:57
00A_01A_SEQUENCE_12	For example - you can't defile an empty coffin.	2007-08-06 17:12:10
00A_01A_SEQUENCE_13	What the hell?	2007-08-06 17:12:25
00A_01A_SEQUENCE_14	(laughs) You devil.	2007-08-06 17:12:37
00A_01A_SEQUENCE_15	What is it? C'mon, hold it up -	2007-08-06 17:12:57
00A_01A_SEQUENCE_16	No, no, no - no way.	2007-08-06 17:13:12
00A_01A_SEQUENCE_17	The deal was for a coffin, that's it.	2007-08-06 17:13:26
00A_01A_SEQUENCE_18	Wait a minute, if my show	2007-08-06 17:13:44
00A_01A_SEQUENCE_19	-hadn't've funded this expedition, you wouldn't've- \ -Hey, hey...	2007-10-08 00:38:11
00A_01A_SEQUENCE_20	You got your story, lady.	2007-08-06 17:14:21
00A_01A_SEQUENCE_21	-Look, Mr. Drake, you signed a contract. \ -(laughs)	2007-08-06 17:14:40
00A_01A_SEQUENCE_22	-I have a right to see every single thing that- \ -Whoa whoa...	2007-08-06 17:15:25
00A_01A_SEQUENCE_23	Could you hold that thought?	2007-08-06 17:16:35
00A_01A_SEQUENCE_24	Sully? Uh, we got some trouble. Hurry it up.	2007-08-06 17:18:12
00A_01A_SEQUENCE_25	Okay, okay - what's going on?	2007-08-06 17:18:31
00A_01A_SEQUENCE_26	Uh... Pirates.	2007-08-06 17:18:51
00A_01A_SEQUENCE_27	Pirates?!	2007-08-06 17:19:08
00A_01A_SEQUENCE_28	Yeah, the modern kind. They don't take prisoners.	2007-08-06 17:20:04
00A_01A_SEQUENCE_29	...least not male prisoners.	2007-08-06 17:20:33
00A_01A_SEQUENCE_30	Wait, what are you talking about?	2007-08-06 17:20:50
00A_01A_SEQUENCE_31	Uh, sh- shouldn't we call the authorities or something?	2007-08-06 17:21:04

Welcome to TAME



# Fileserver

- We love linux
- Linux is fast
- The game connects to our own file system running under linux using the debug port
- 80 MB/sec



# Programmer Pipeline

- Linux again!!
- 1 linux box for every 2 programmers
- Putty
- It's FAST!!!
- Multithreaded build
- SN Debugger & Gcm Replay



# Artist Pipeline

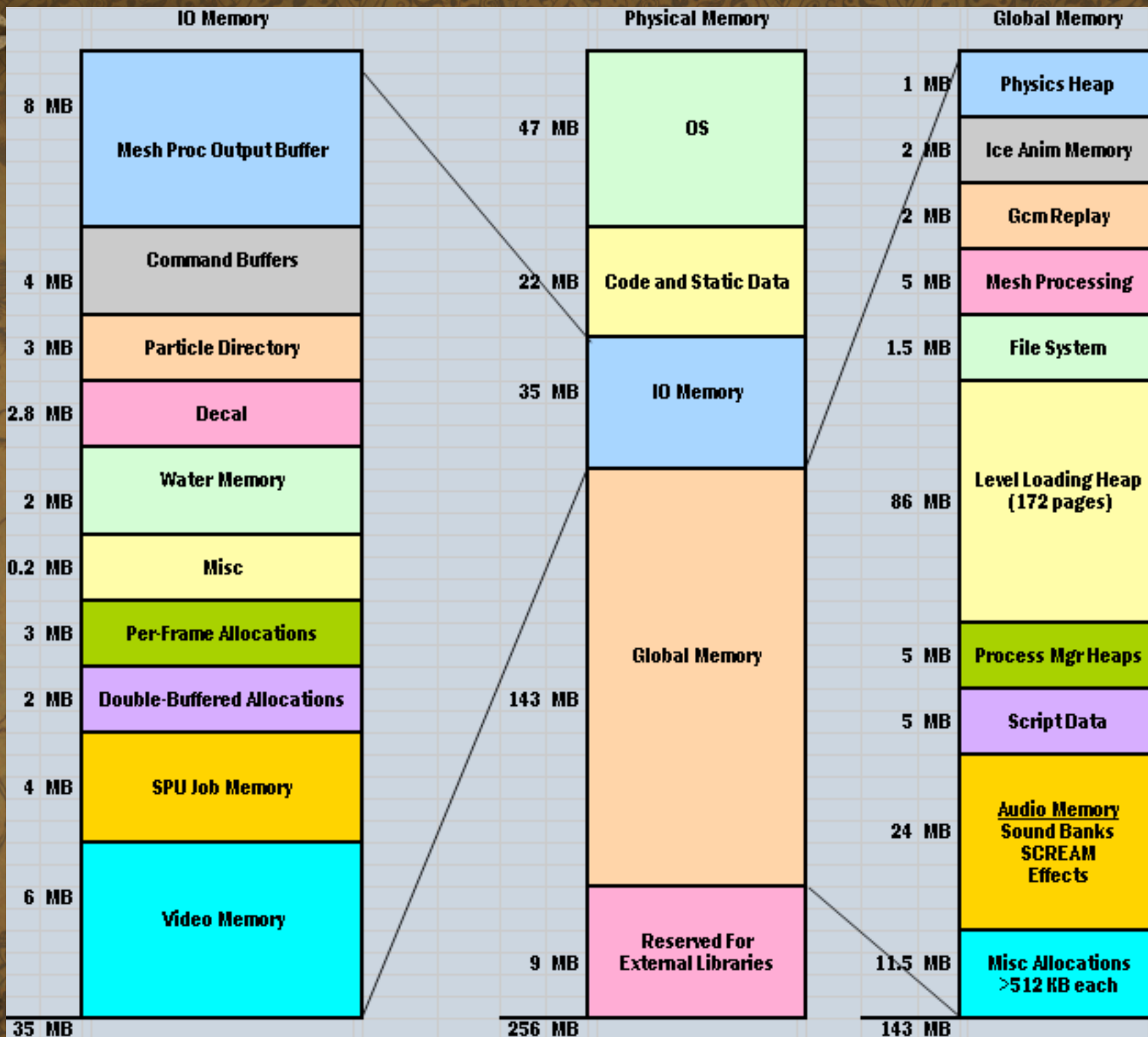
- Maya & ZBrush
- Material Editor
- BuildBig
- BA/BL



# Designer Pipeline

- Charter
- BuildBig
- BA/BL
- DC







# FIOS

- All IO done with FIOS
- Everything is compressed using Edge
- We stream a lot and all the time: levels, actors, sounds, music, textures
- Pre-cache



# Loading Scheme

- Levels and actors are .pak files
- .pak file is made of pages
- Heap of 172 pages
- No fragmentation



# Texture Streaming

- Having a hard-drive on every PS3 is HUGE
- Very easy to do ... 3 days of work
- Adds a lot to the quality of your game
- Everyone must stream textures!
- Defragment memory every frame



# Texture Defragmentation

Step 1



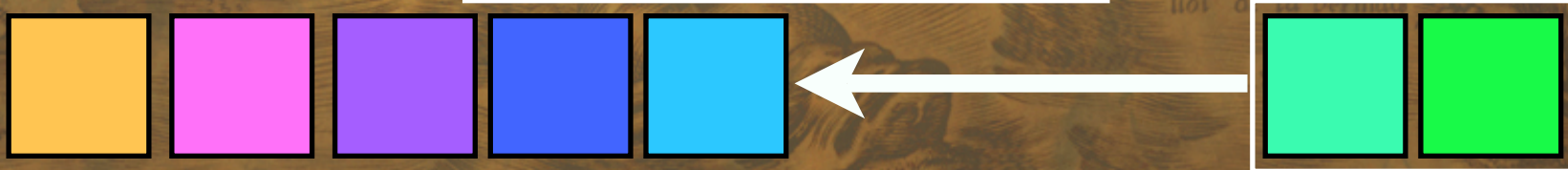
Step 2



Step 3



Step 4



Step 5





# Collisions

- Sphere, Capsule, Convex Polytope
- Concave Geometry, kd-tree Polygon soup
- SPU elf per Shape
- 32 Objects per Batch
- ~ 500 asynchronous ray-cast every frame



# Physics

- World split into Islands
- One SPU job (solver) per island
- Successive Relaxation LCP solver
- General constraint system: rag-doll
- Constraints & limits derived from “range-of-motion” animation



# SPUs

- Scene traversal
- Geometry processing
- Spherical Harmonics to cube maps
- Particle simulation
- Water
- Animations
- Decompression
- Collisions
- Physics
- Path Finding



# Animation

- Using SPUs to decompress and blend animation tree
- Blend tree up to 25 animations (Drake)
- Use DC to describe animation states
- Layered animation system



# Scene Traversal

- Visibility frustum culling
- PVS lookup
- Sorting
- Render set up
- Mesh processing set up



# Scene Traversal

- Visibility frustum culling - **SPU**
- PVS lookup - **SPU**
- Sorting - **PPU**
- Render set up - **PPU**
- Mesh processing set up - **SPU**



# Mesh Processing

- Use SPU's to offload RSX
- Decompression
- Skinning
- Back-face culling
- Also used to collide with rendered geometry: decals, IK, some gameplay collisions



# Scene Rendering

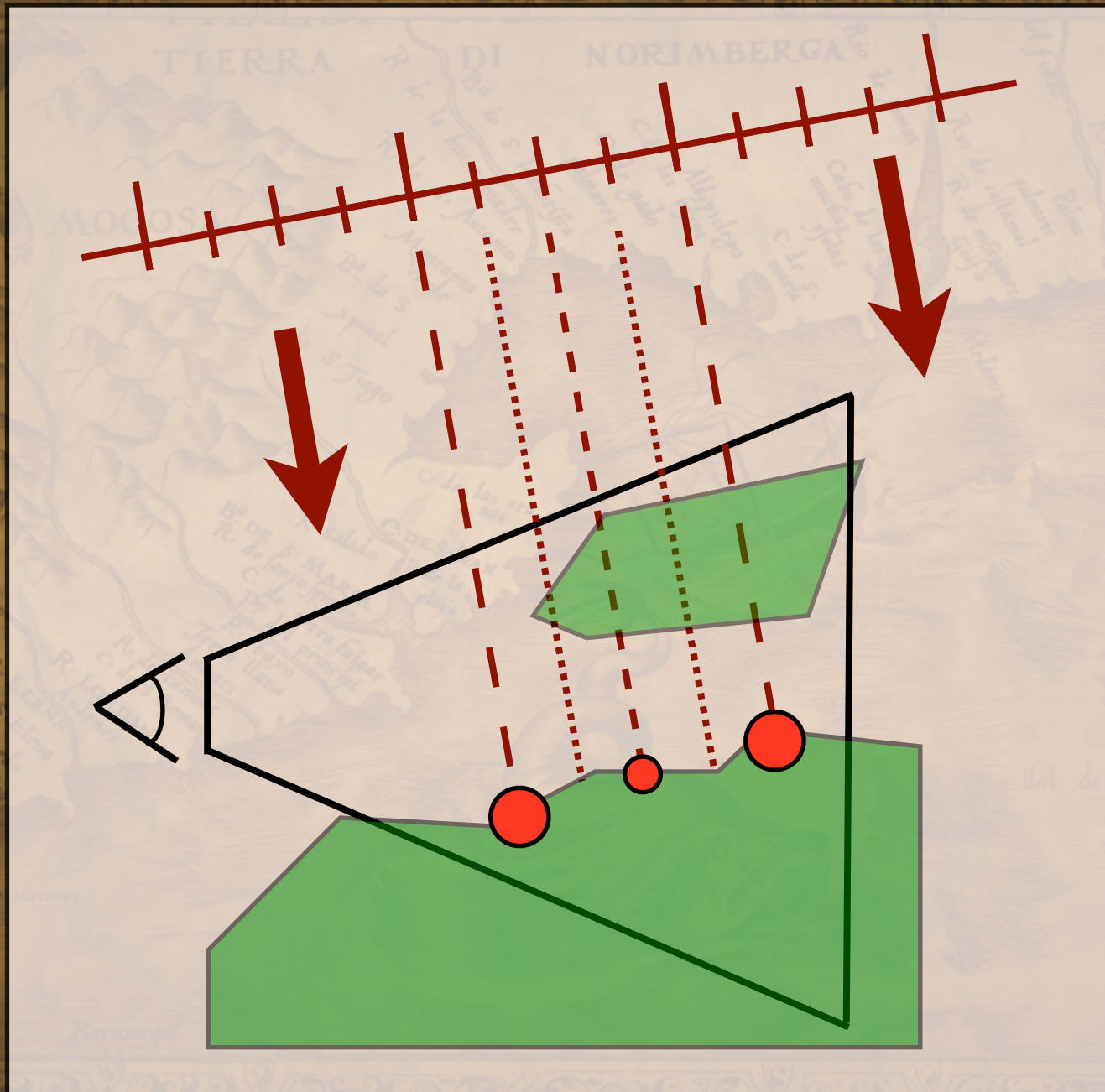
- Shadows
- Dynamic Lighting
- Opaque Geometry
- Alpha Blend Geometry
- Post Processing effects



# Sun Light Shadow

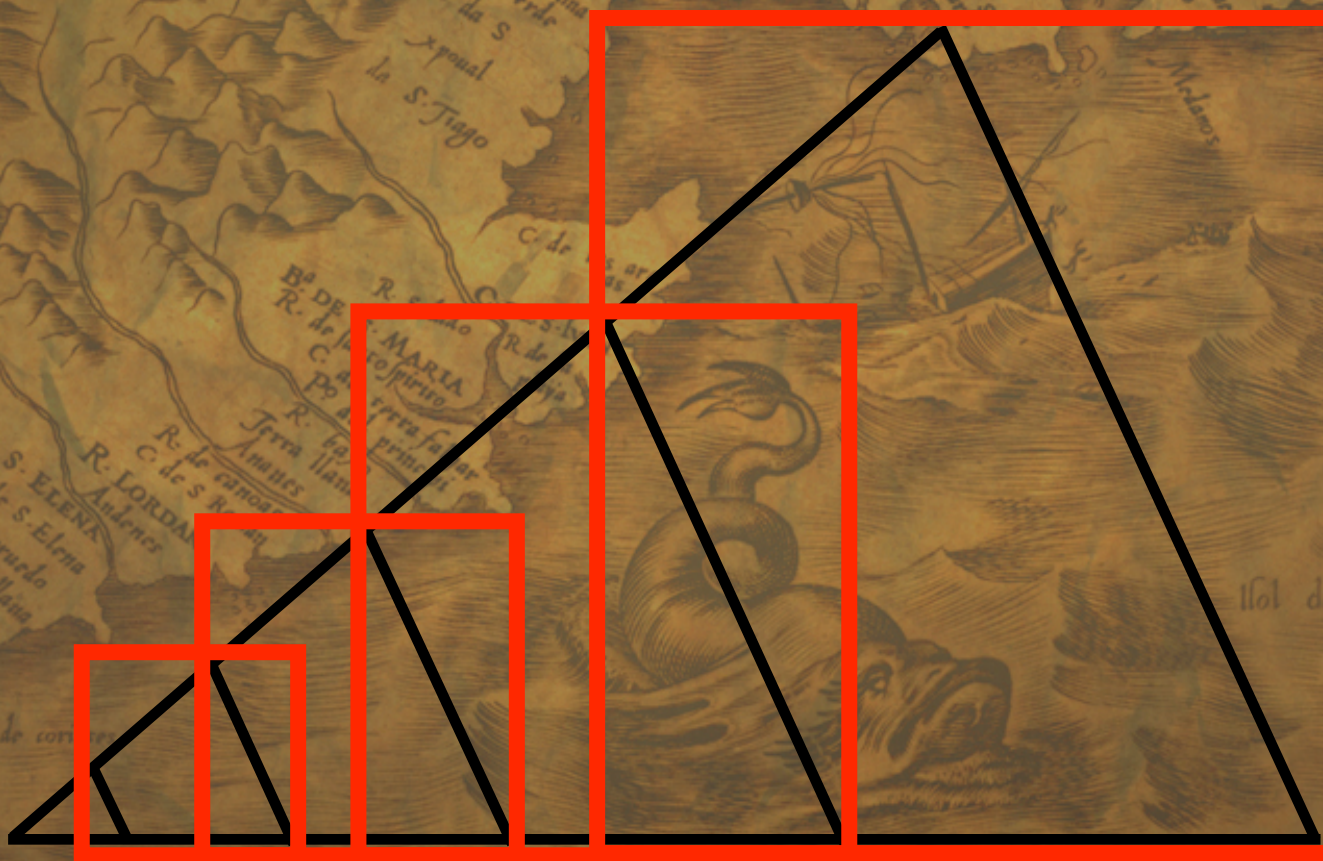
- Tried a lot of solutions: problems!
- Simple idea by Guerilla(Killzone)
- Reduce flickering: fixed world space sample points
- SSM: orthographic shadow map
- Cascaded shadow maps







# Cascaded Shadow Maps





Shadow Blockers

Sunlight Shadow Pass

Shadow Depth Buffer  
1216x1216 : Z16

Opaque Geometry

Depth Only Pass

Depth Buffer  
2x1280x720 : Z24



# Dynamic Lighting

- Render opaque dynamically lit geometry: world normal + specular exponent in screen space
- Divide the screen into a grid
- Find which lights intersect each cell
- Render quads over each cell calculating up to 8 lights per pass: results in a light buffer



Opaque Lit Geometry



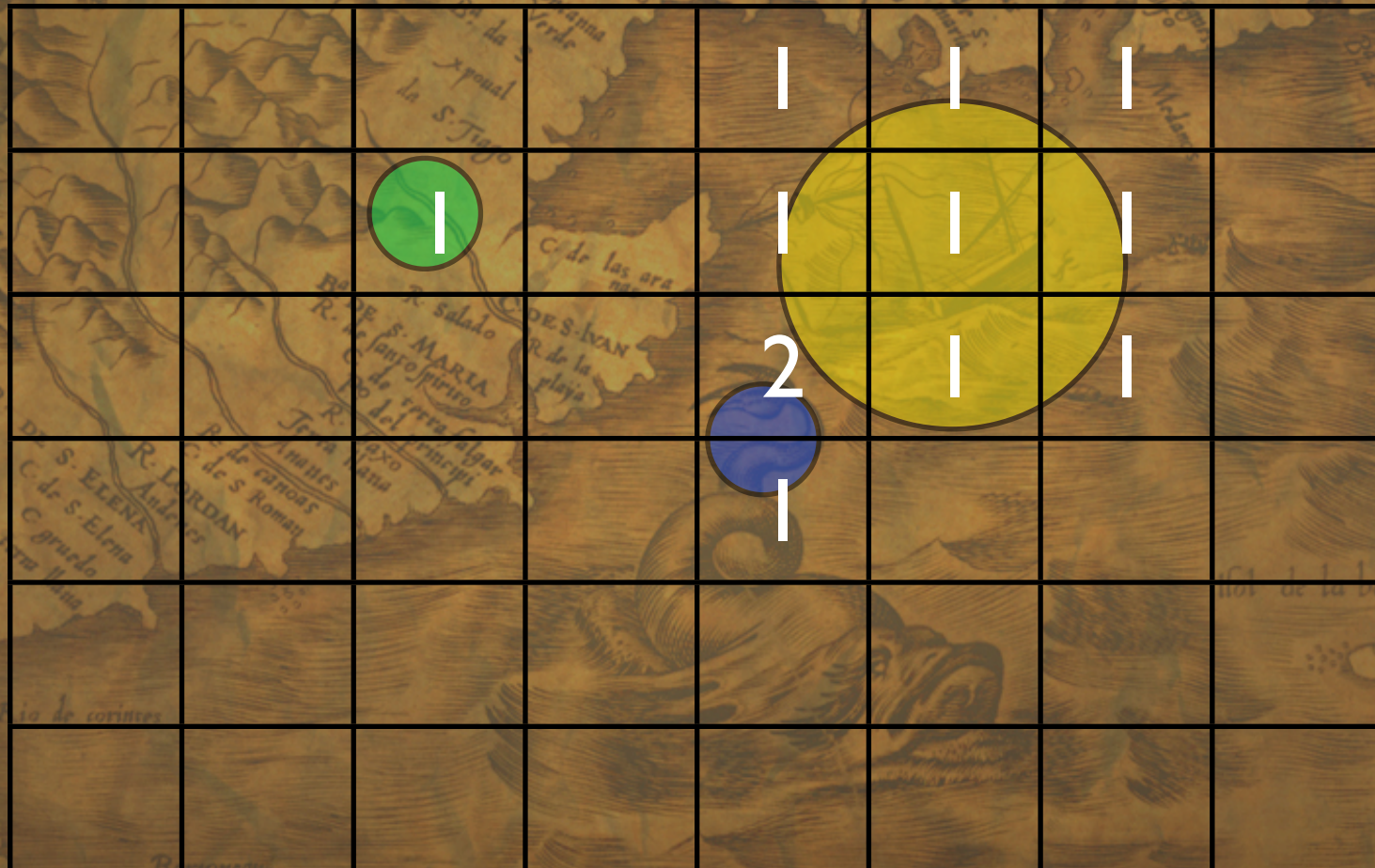
Get World Space  
Normal and Spec Exp



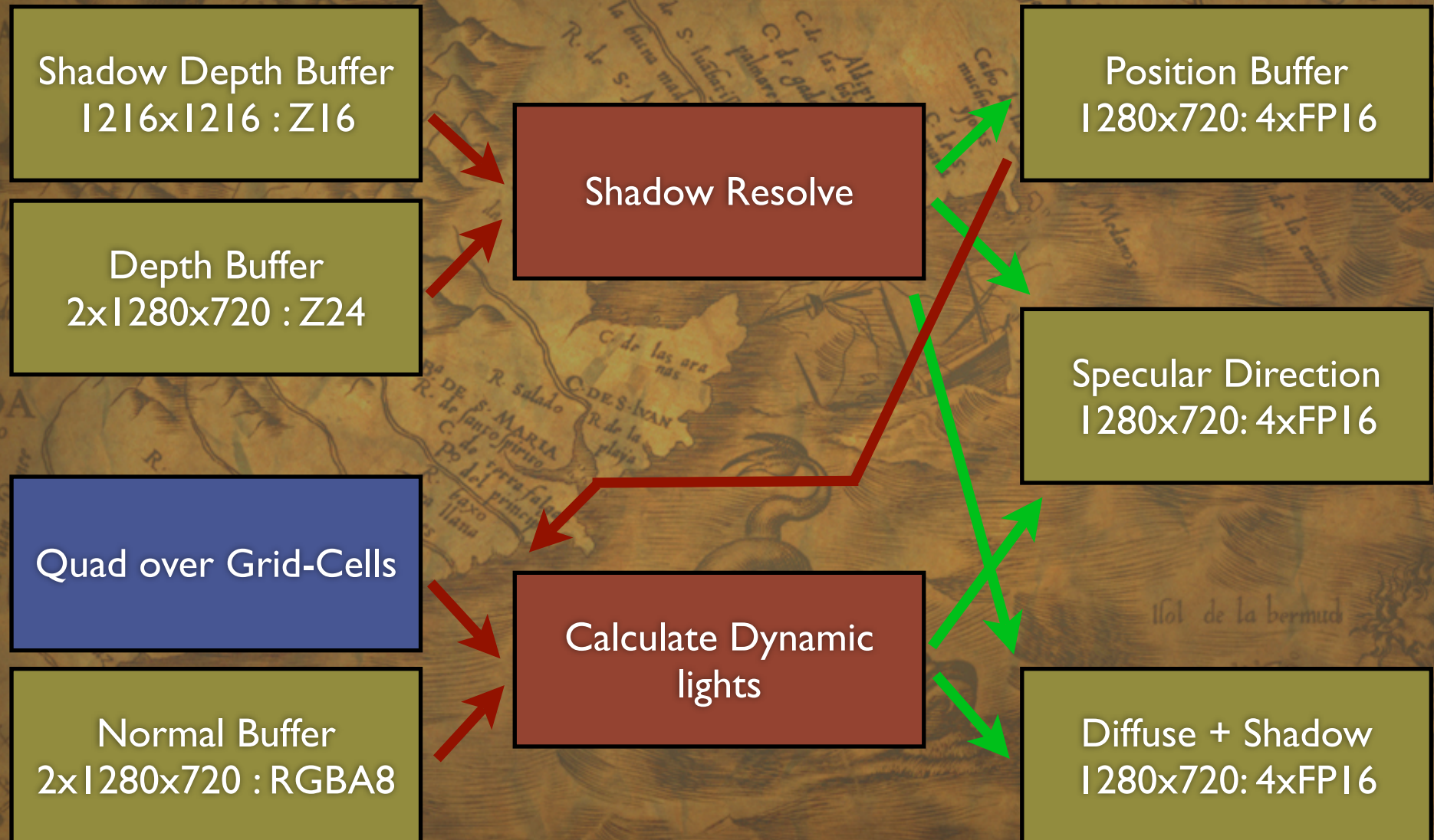
Normal Buffer  
2x1280x720 : RGBA8



# Dynamic Lighting









# Opaque Rendering

- Shadowing and lighting already done
- Material processing + direct lighting
- Output is HDR (logluv)
- Pixel shader bound: vertex processing mostly done on SPUs
- Dithering out some geometry



Opaque Geometry

Depth Buffer  
2x1280x720 : Z24

Specular Direction  
1280x720: 4xFP16

Diffuse + Shadow  
1280x720: 4xFP16

Material  
Direct Lighting

Logluv Buffer  
2x1280x720: RGBA8



Logluv Buffer  
2x1280x720: RGBA8

Depth Buffer  
2x1280x720 : Z24

Resolve color + Fog

Color Buffer  
1280x720: 4xFP16  
w = dist from camera

1x Depth Buffer  
1280x720 : Z24



# Alpha-blended Geometry

- Rendered to FP16 1x buffer
- Water
- Particles
- Glass



Water Meshes

Shadow Depth Buffer  
1216x1216 : Z16

Color Buffer  
1280x720: 4xFP16  
w = dist from camera

Logluv Buffer  
2x1280x720: RGBA8

Water Pass

Color Buffer  
1280x720: 4xFP16  
w = dist from camera

1x Depth Buffer  
1280x720 : Z24



# Particles

- Simulation and rendering set up done on SPU's
- Wrote specific shaders
- Switch to down sample buffer on frame rate spike



# Post Effects

- Depth of field
- Blur
- Tonemap
- Bloom
- Saturate
- Tint
- Distortion
- Motion Blur
- Blend



UNCHARTED

NAUGHTY DOG

KEEP IT SIMPLE!

2008 GAME DEVELOPERS CONFERENCE



# WE'RE HIRING!!!!

- if you're extremely talented ...
- [candace\\_walker@naughtydog.com](mailto:candace_walker@naughtydog.com)



UNCHARTED

NAUGHTY DOG

Thank you

[christophe\\_balestra@naughtydog.com](mailto:christophe_balestra@naughtydog.com)

[pal\\_engstad@naughtydog.com](mailto:pal_engstad@naughtydog.com)

2008 GAME DEVELOPERS CONFERENCE