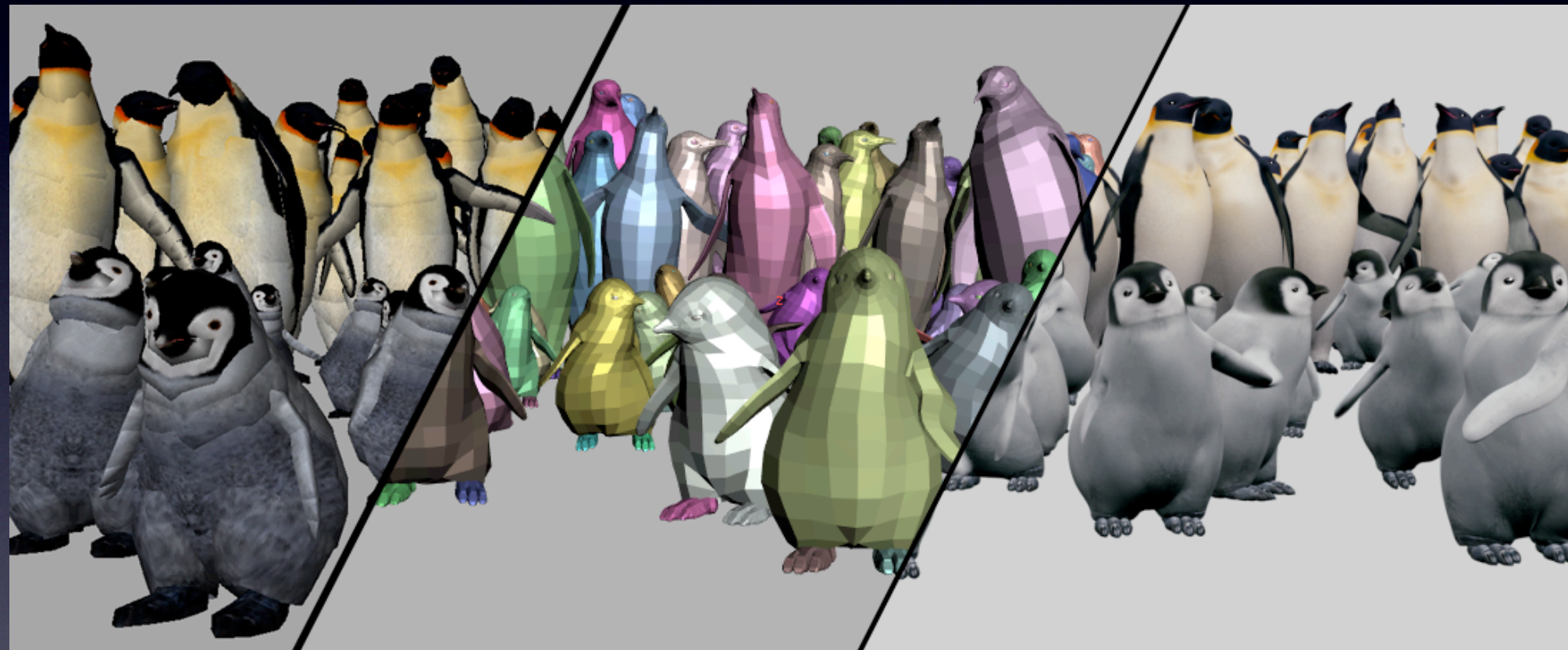


Cortex and the *Happy Feet 2* Crowd Pipeline



Carsten Kolve - Crowd Supervisor - Dr. D Studios
carsten.kolve@drdstudios.com

Overview

- Dr. D Studios & Crowds in *Happy Feet 2*
- Choosing Cortex
- Prototyping
- Crowd System Design
- Integration in 3rd Party Applications
 - Crowd Layout Workflow - Maya
 - FX and Lighting - Houdini
 - Rendering - 3delight
- Conclusions

Dr. D Studios

- Sydney, Australia
- KMM, George Miller / Doug Mitchell
- *Happy Feet 2*



Happy Feet 2 - Crowds

- 600+ crowd shots, 15 unique character types, FG/MG/BG
- 8 months dev time, 3 developers
- 6 months shot production time, 18 artists



Choosing Cortex

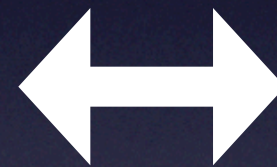
Interface Requirements

Animation / Lensing / Stereo
MAYA

MoCap / MoEdit
NUANCE

Final Layout / FX / Lighting
HOUDINI

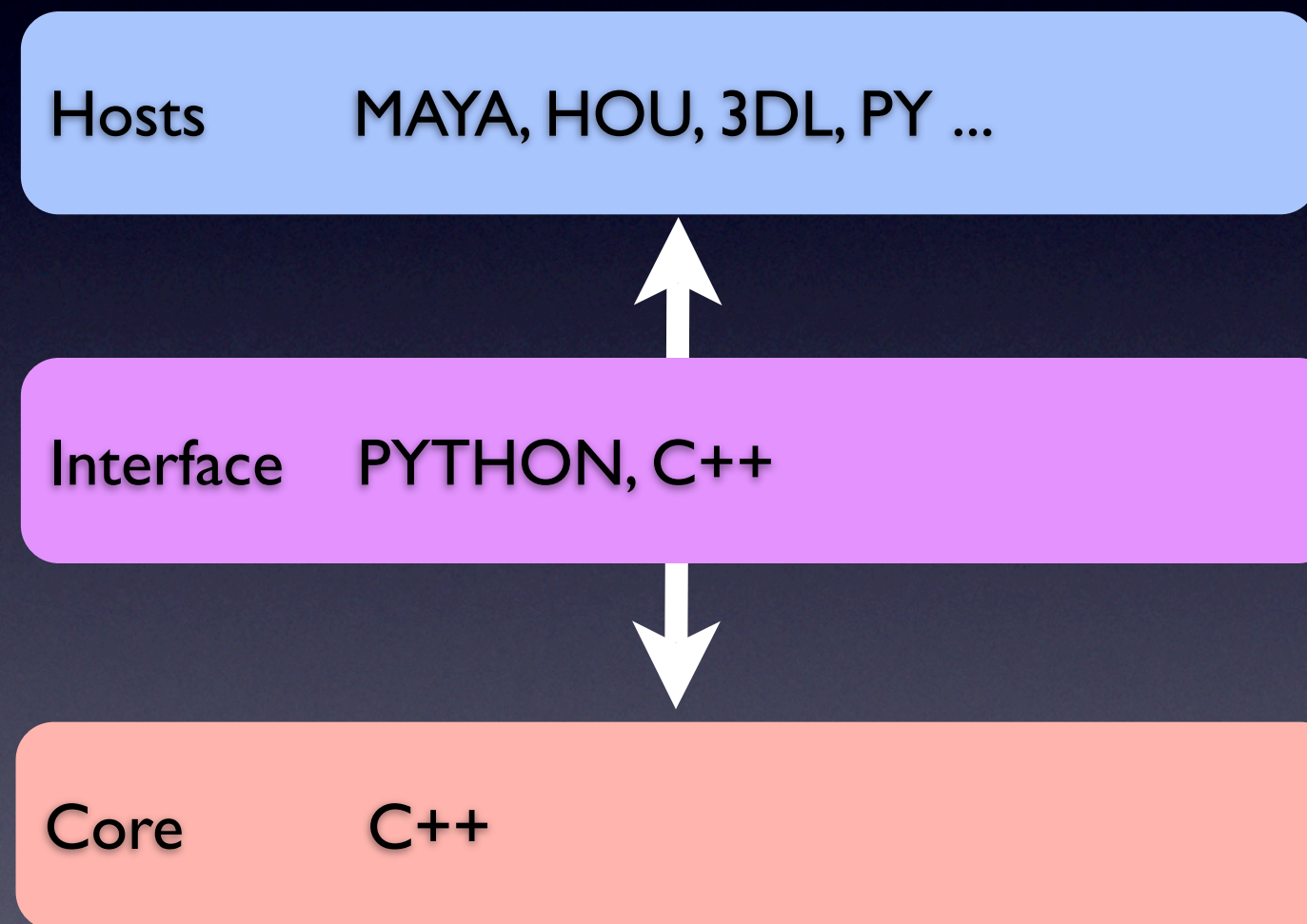
Rendering
3DELIGHT



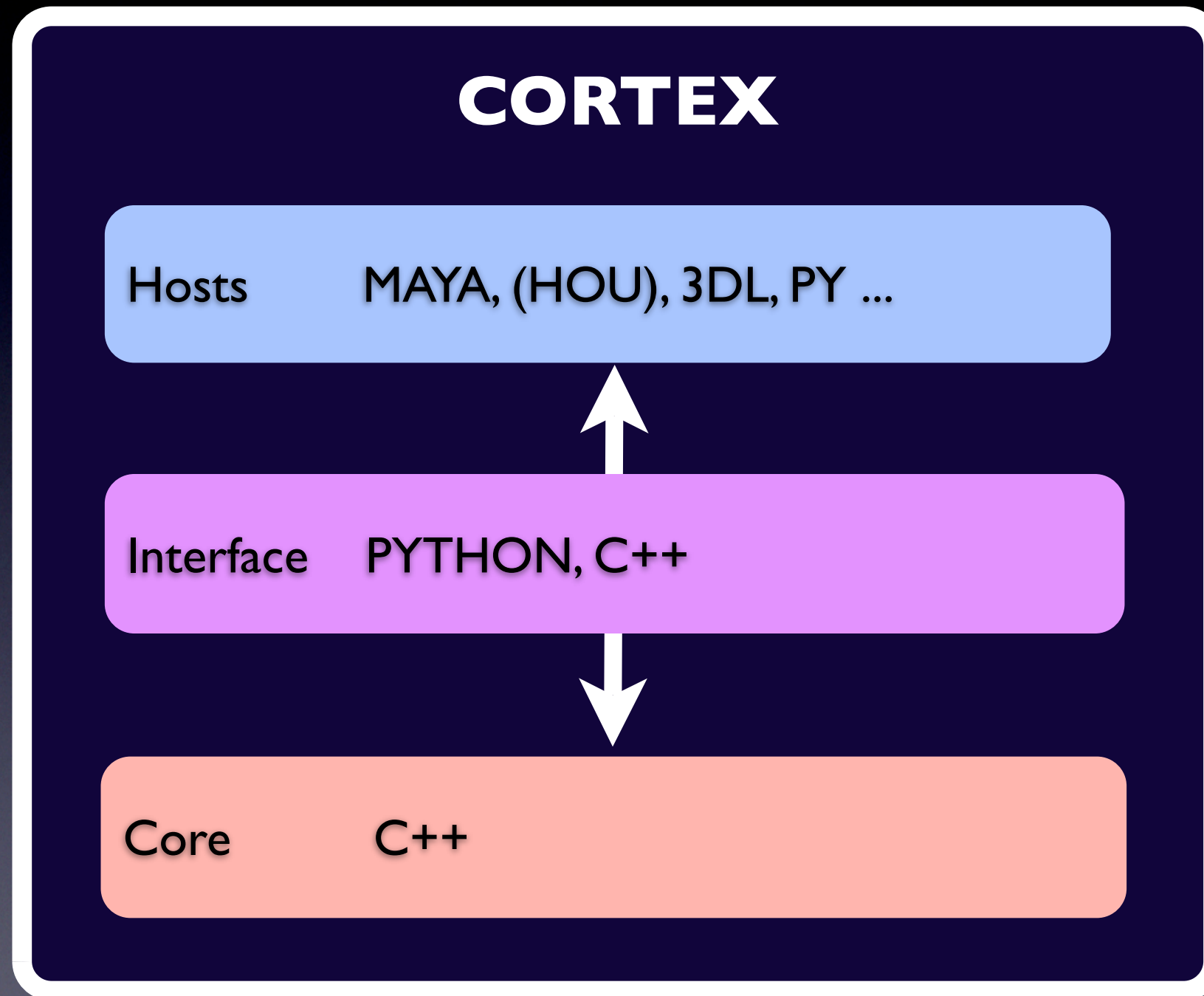
Crowd
MAYA
HOUDINI
PYTHON

?

Base Structure

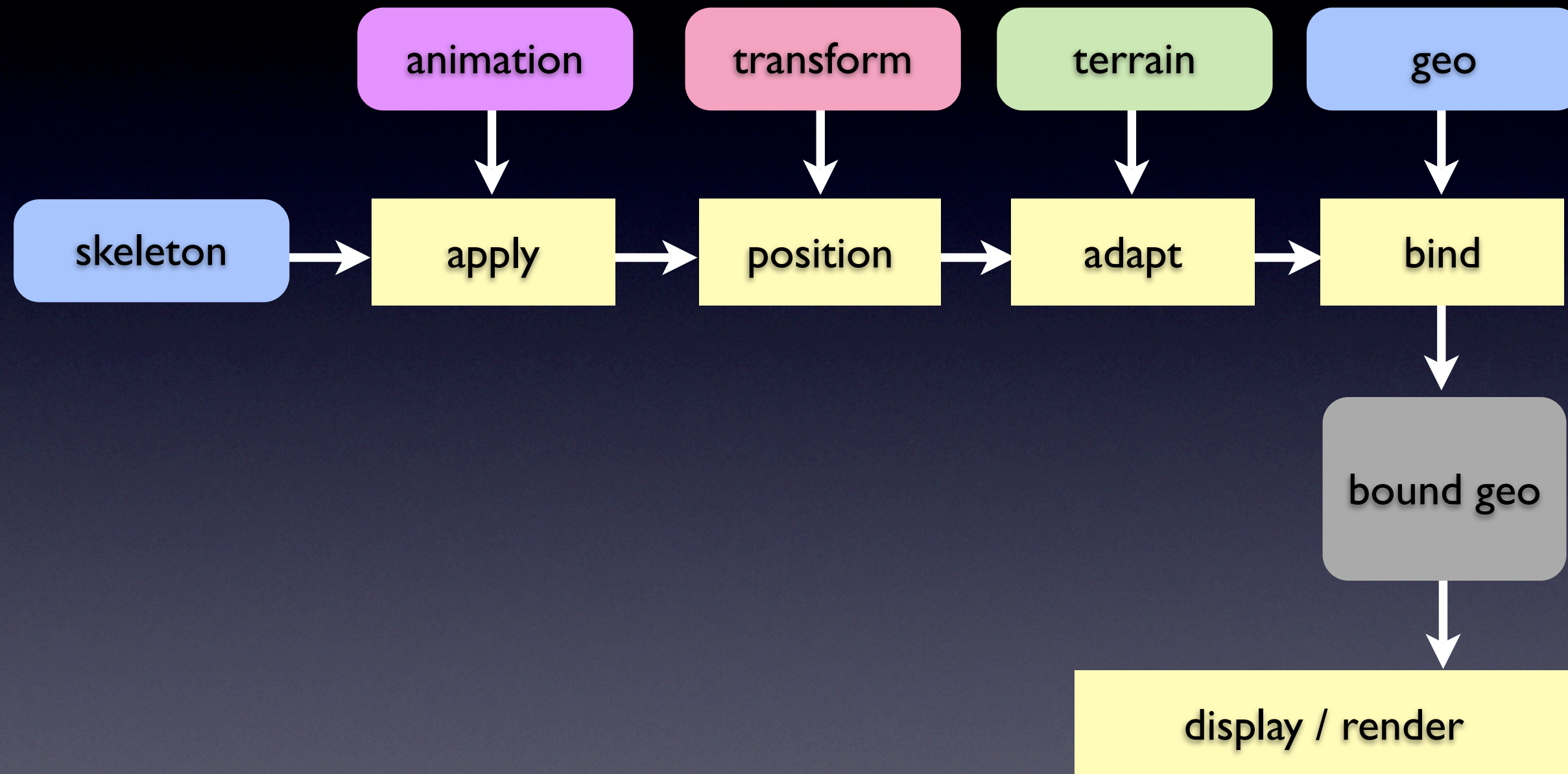


Base Structure



Prototyping

Basic Character Generation



Skeletons & Animation

- Cortex functionality bound in Python to export assets from Maya (FromMayaConverters & CompoundData)
- Python procedural to test and preview character animation process



notes: group of crowd
skeleton in action

870_010_crowd_v013.mov
daniele.niero

#0101
1 260

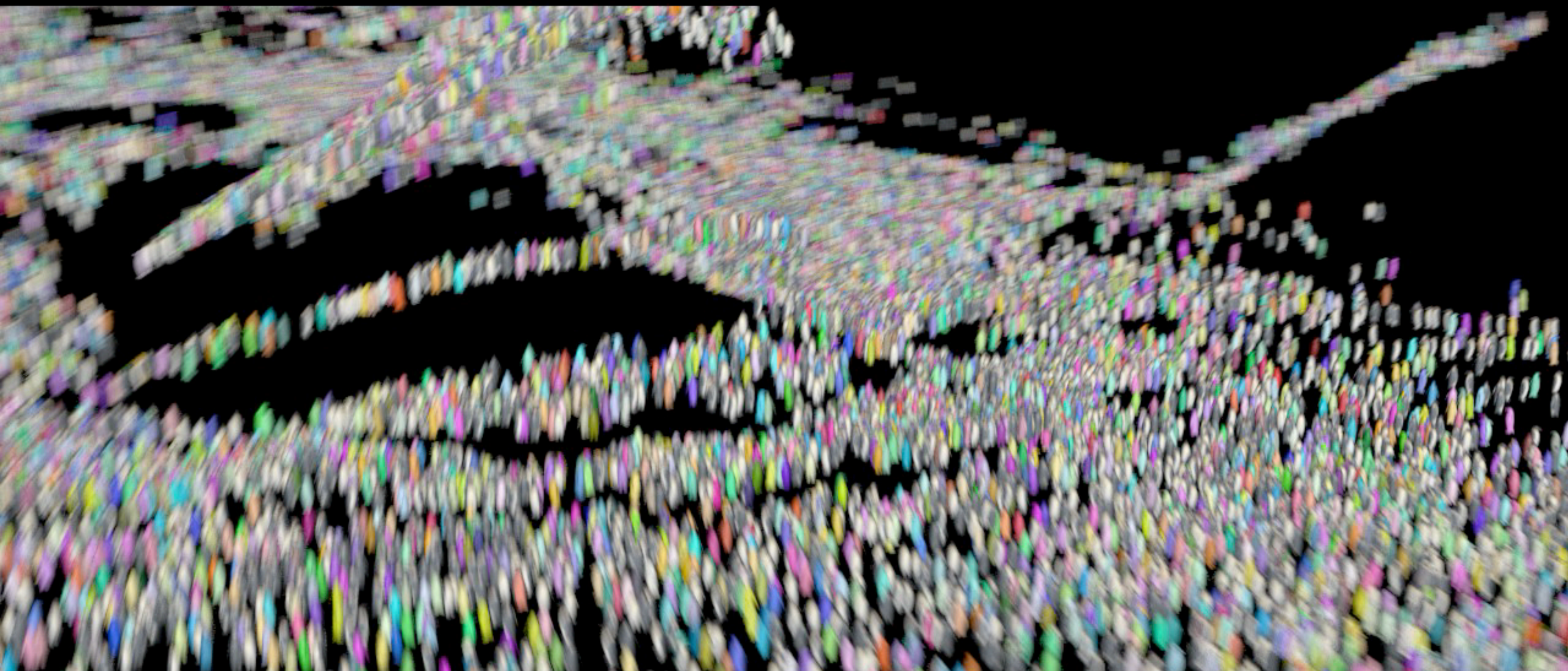
Crowd Layout Rendering

- CompoundData to store layout information
- procedural to render in 3delight

Dr. D Studios
Happy Feet 2

crowd
870_010_crowd_v015

21/05/10
08:36:22 AM



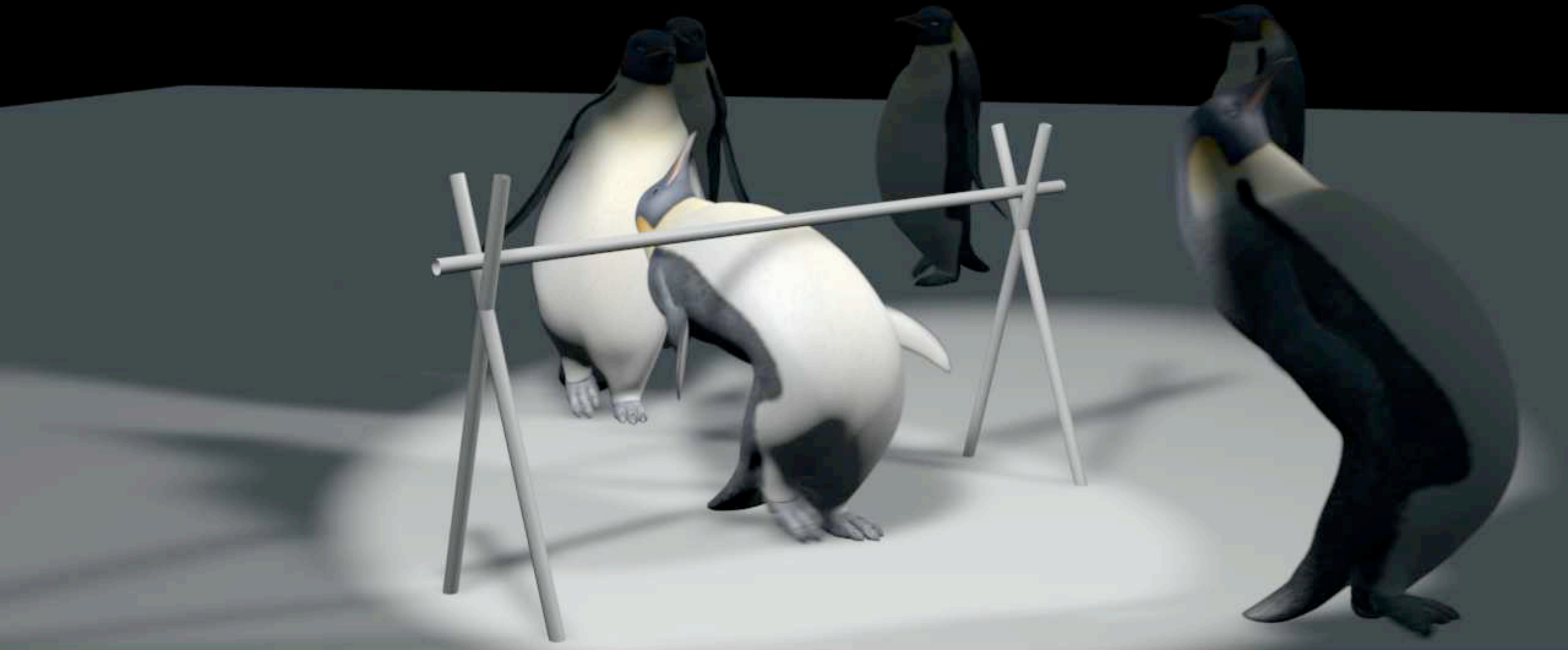
notes: Render test using
callsheet for layout
and render-time
shader assignment

870_010_crowd_v015.mov
dan.bethell

#0071
1 150

Animation Modification

- procedural modification of animation data
- smooth skinning of skeletons



notes: Test of procedural
skeleton
manipulation based
on external

limboTest_crowd_v002.mov
dan.bethell

#0142
1 350

Attaching Geometry

- attaching geometry to animated skeletons



notes: !!DEBUG MODE!!

closeupTest_DEBUG_crowd_v001.mov
dan.bethell

#0001
1 150

Crowd System Design

Crowd System

- layout system for pre-created skeletal animation data that is skinned on demand
- mass animation instancing system, limited motion blending / no motion synthesis
- focus on fast manual control rather than simulation

Crowd “Callsheet”

- Crowd represented by a *Callsheet* that associates *Characters* with *Animation* and *Terrains*

Crowd “Callsheet”

- **Characters:** skeleton, geometry, bind, materials, grooms etc.
- **Animation:** skeletal poses over time (lots)
- **Terrain:** heightfield information
- **Callsheet:** static lookup table

uid	char	anim	transform	terrain
0	emperor	danceA	M44f	mountB
1	adelie	danceC	M44f	mountB

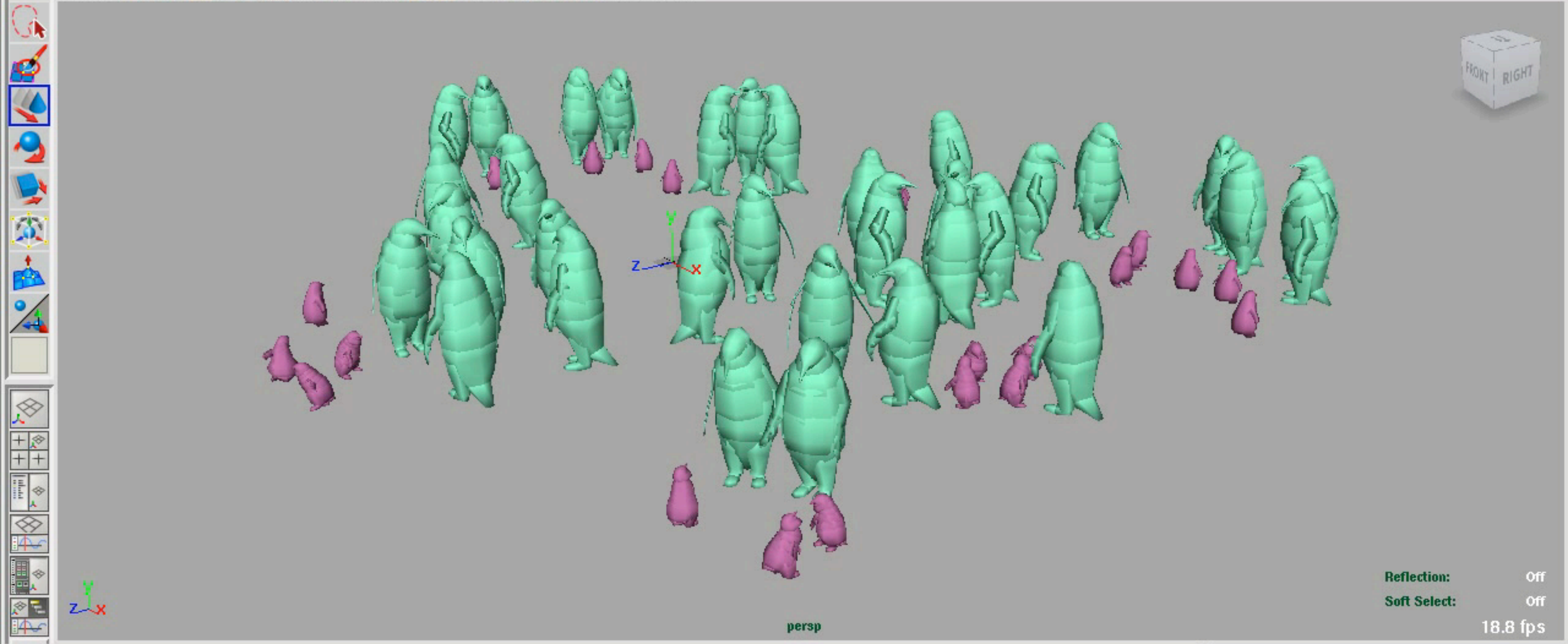
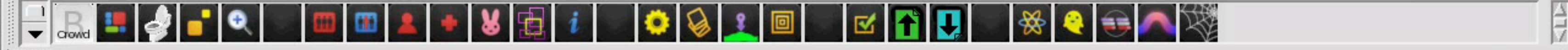
Emerging Structures

- Maya-based interface for most layout work
- Houdini for procedural crowd effects (swarms / flocks)
- Python prototypes (where slow) moved to Cortex dependent C++ libraries or submitted directly to Cortex
- modularisation of functionality into Python function sets & Ops

Crowd Layout Generation (Maya)

Maya Layout Workflow

- Cortex Op networks for procedural layout modification
- custom Maya MPxSurfaceShape for manual callsheet editing and display, crowd characters match to components

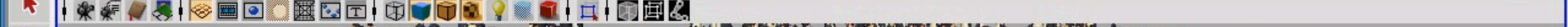
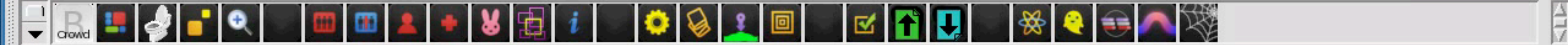


Reflection: Off
Soft Select: Off

18.8 fps

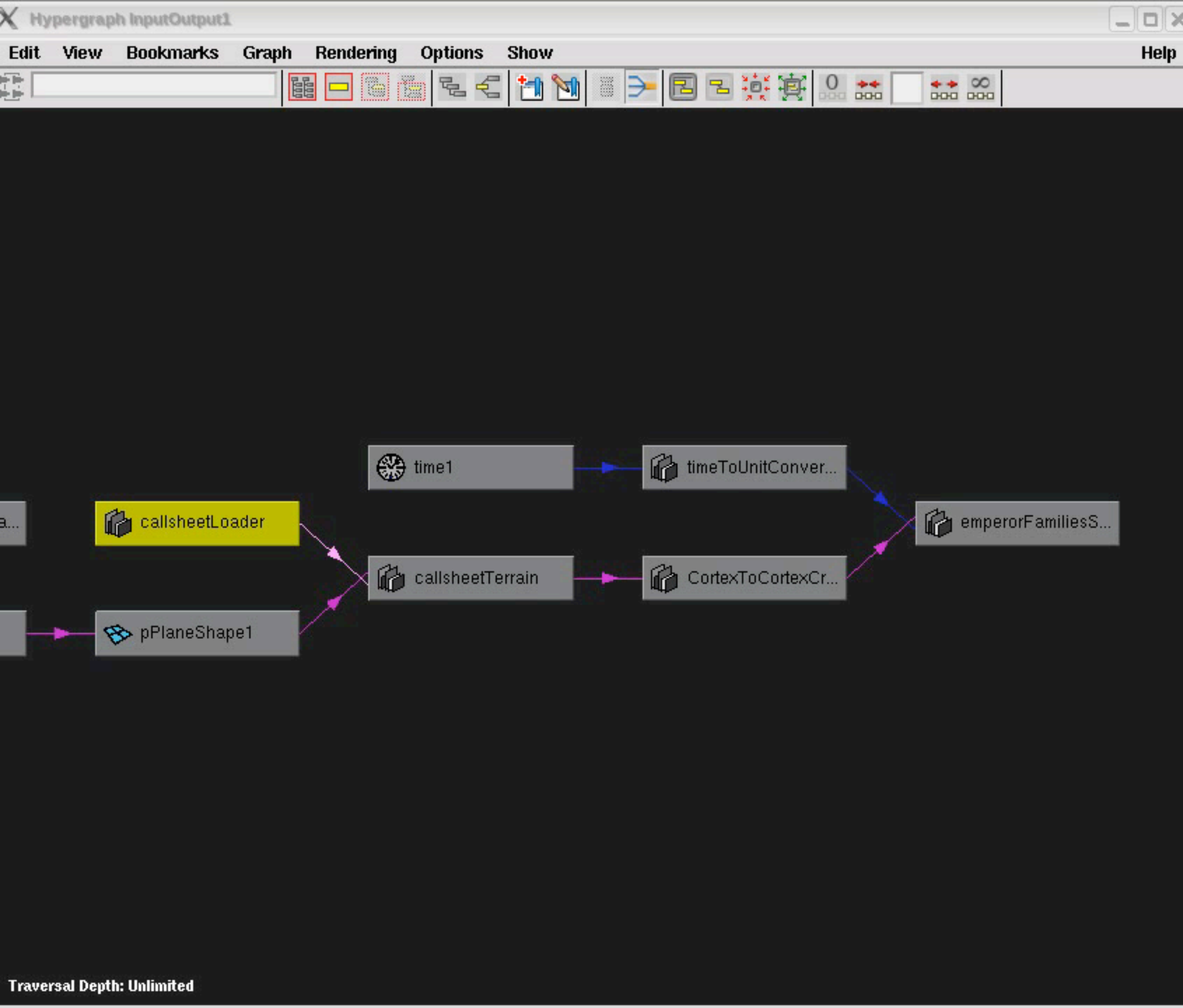
Maya Layout Workflow

- working with terrains



Maya Layout Workflow

- Maya dependency graph networks of cortex ops



Loader

Drd Help

drd_CrowdCore drd_util drd_crowd_pipeline

callsheetLoader callsheetTerrain

ieOpHolderNode: callsheetLoader

Focus Presets Show Hide

Class

Parameters

File Path: wd/testScenes/emperorFamilies.cob

Extra Attributes

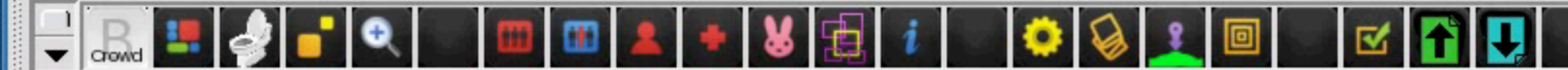
Notes: callsheetLoader

Select Load Attributes Copy Tab

3450 3 3347.00

Maya Layout Workflow

- procedural layout examples



label: creation

Callsheet Terrain

Input Mesh: pPlaneShape1.worldMesh

File Path: []

Init Z: 1000.0000

Resolution: 2000

Disable

Replace Existing

Replace Slot: 1

Callsheet Loader

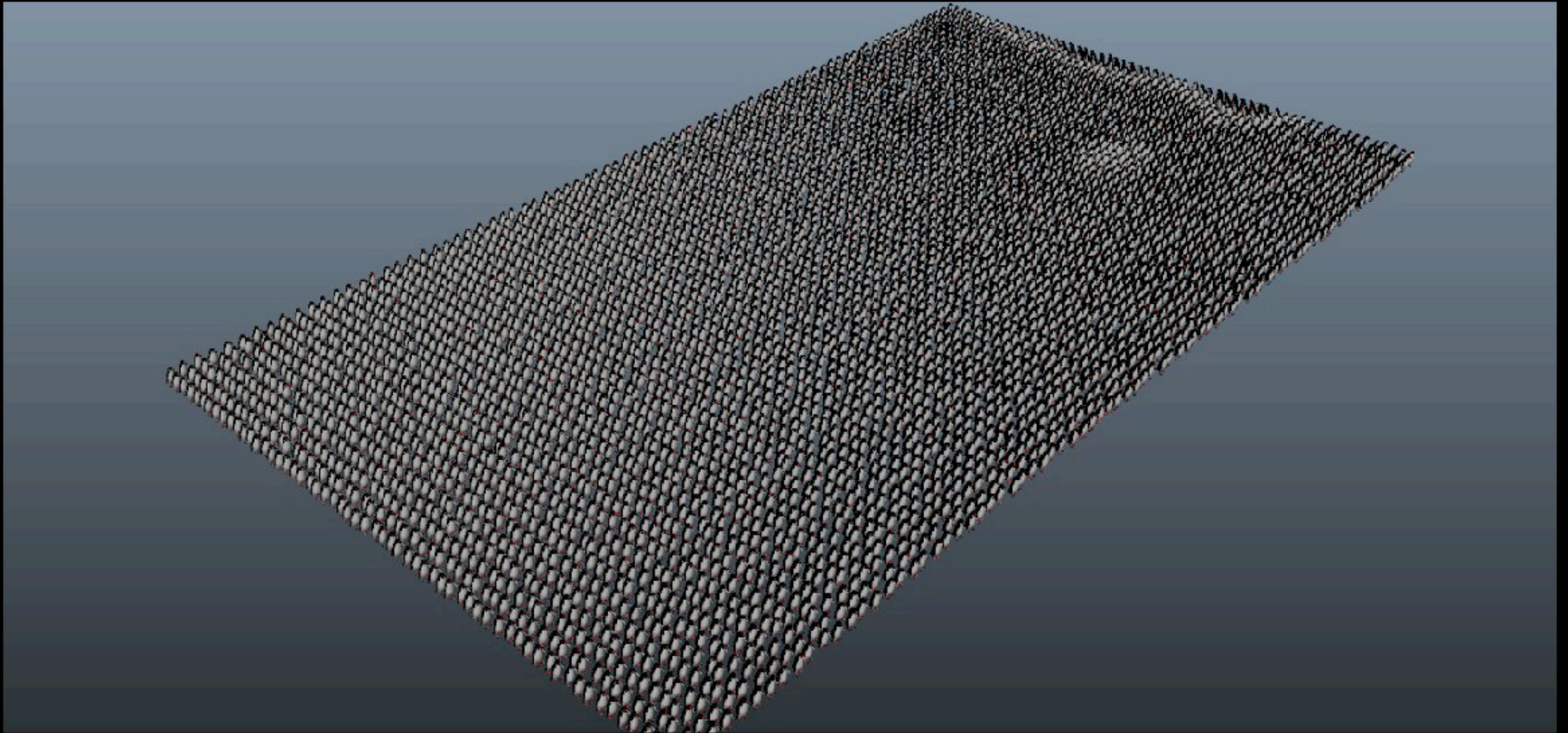
File Path: /d/testScenes/emperorFamilies.cob

Outgoing Crowd

Hidden

Notes: []

Select Crowd Shape



notes :

funky noise wave

888_crowd_v005.mov

moe.elali

#3957

3957 4020

Crowd FX (Houdini)

Houdini FX Workflow

- adding Cortex support to Houdini, Ops & Procedurals + Converter
- Cortex Op networks - reusing the same ops in Houdini written for Maya
- procedural geometry conversion for fx integration

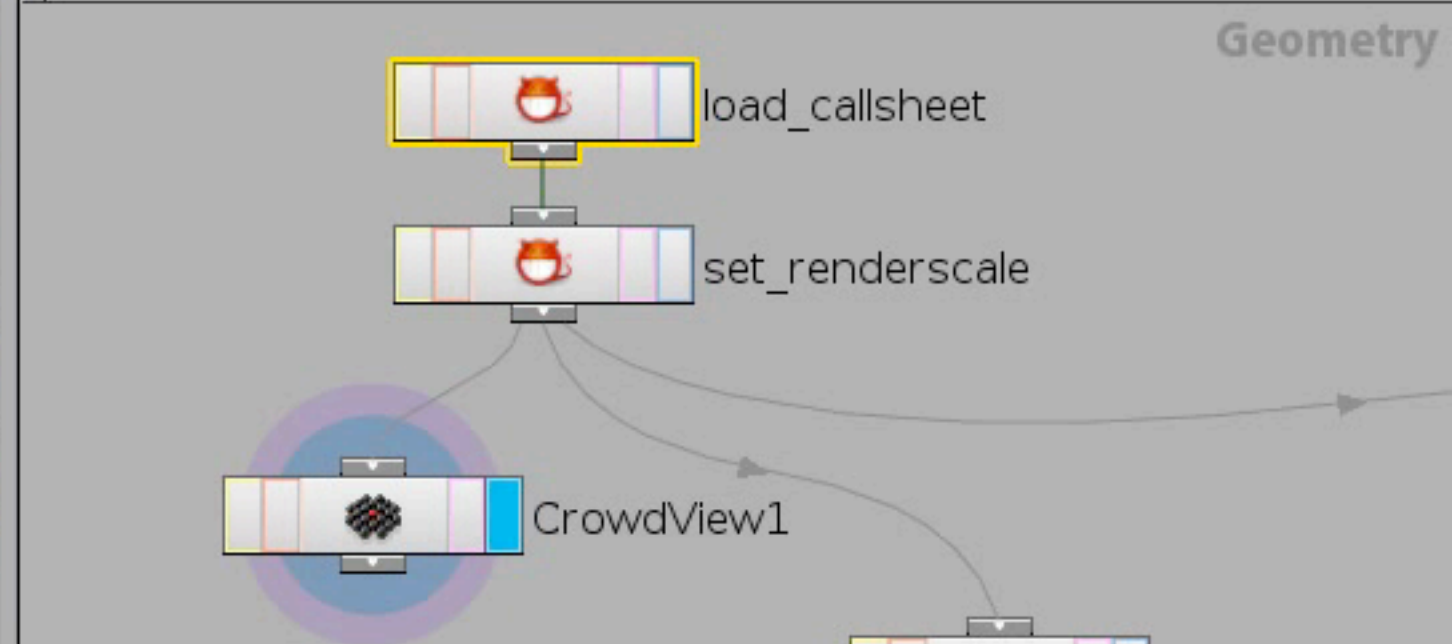


Category: < No C... Class: callshe... Version: 1

Reload

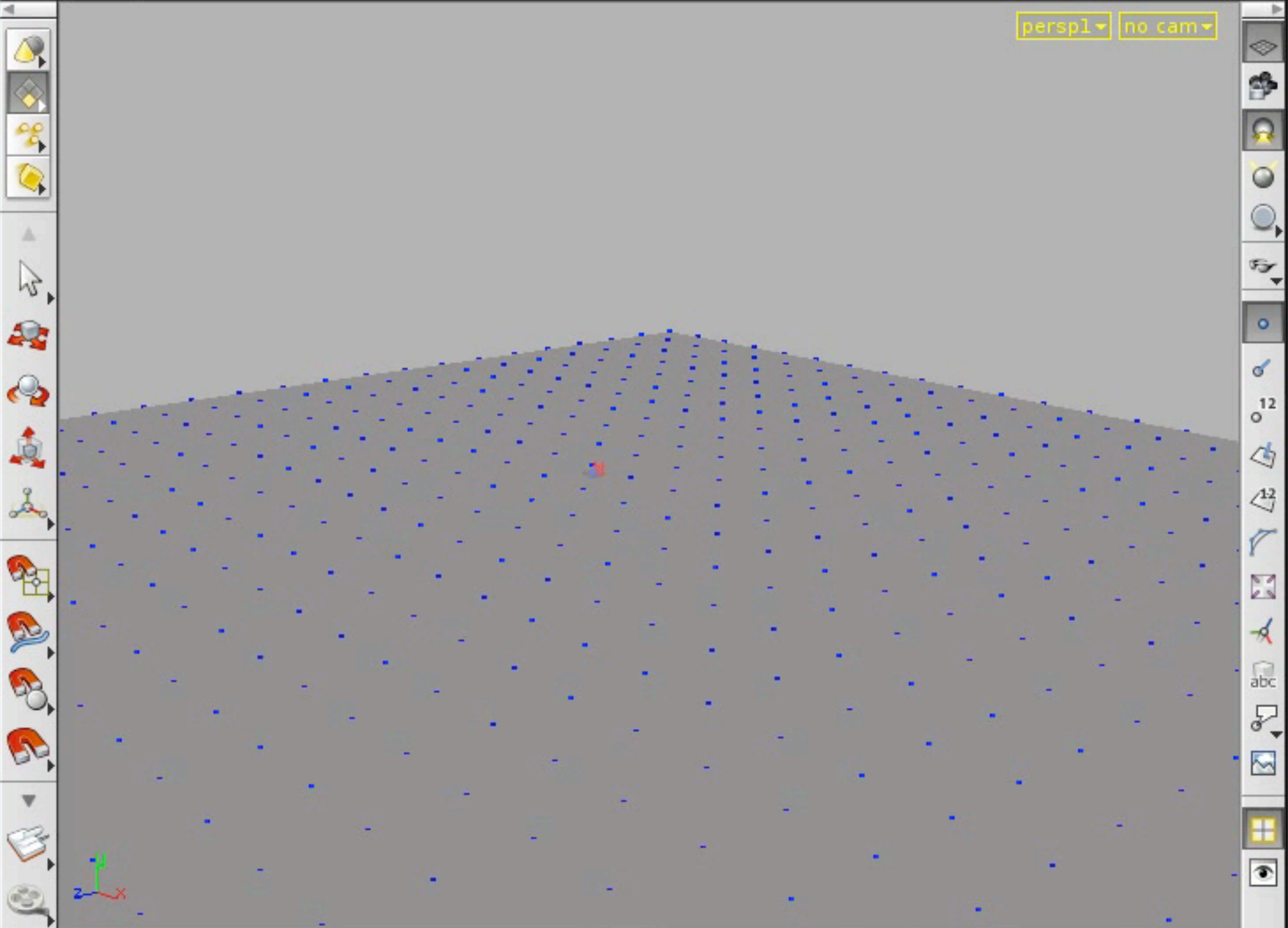
Parameters

File Path /drd/jobs/hf2/wip/scenes/870/crowd/testScenes/e



Houdini FX Workflow

- feeding Houdini data into the crowd system via cortex



Primitive Type: Mesh

Connectivity: Quadrilaterals

Orientation: ZX plane

Size: 1626 1626

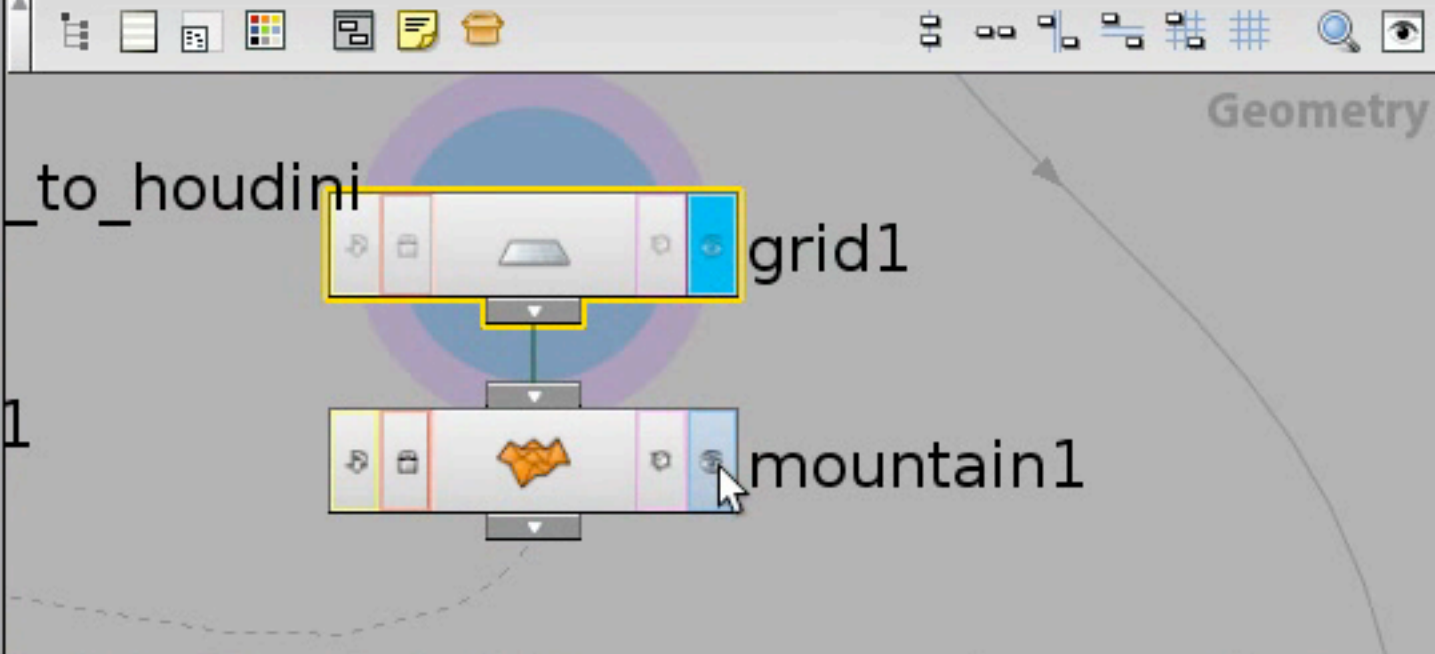
Center: 0 -52 0

Rows: 25

Columns: 24

U Order: 4

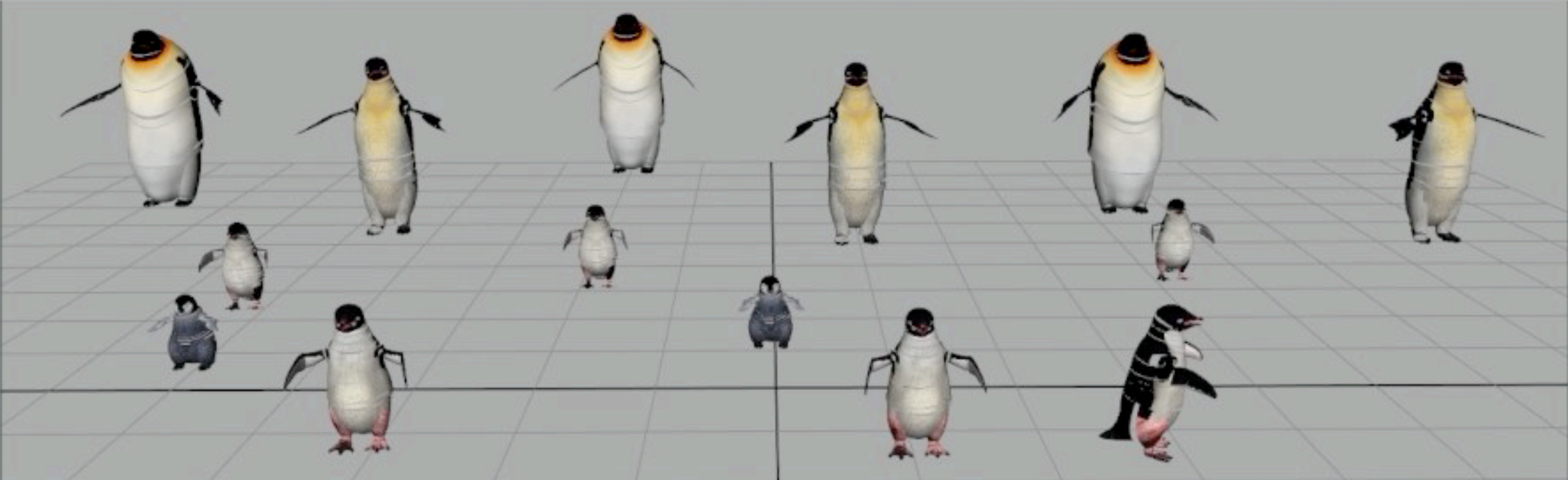
V Order: 4



Crowd Lighting / Rendering (Houdini / 3delight)

Rendering Workflow

- Cortex to bridge gap between Maya, Houdini, 3delight
- nested Python Cortex procedurals distribute the crowd rendering load
- emulation of hero-rendering pipeline: reuse of geometry, shaders, fur-grooms etc.

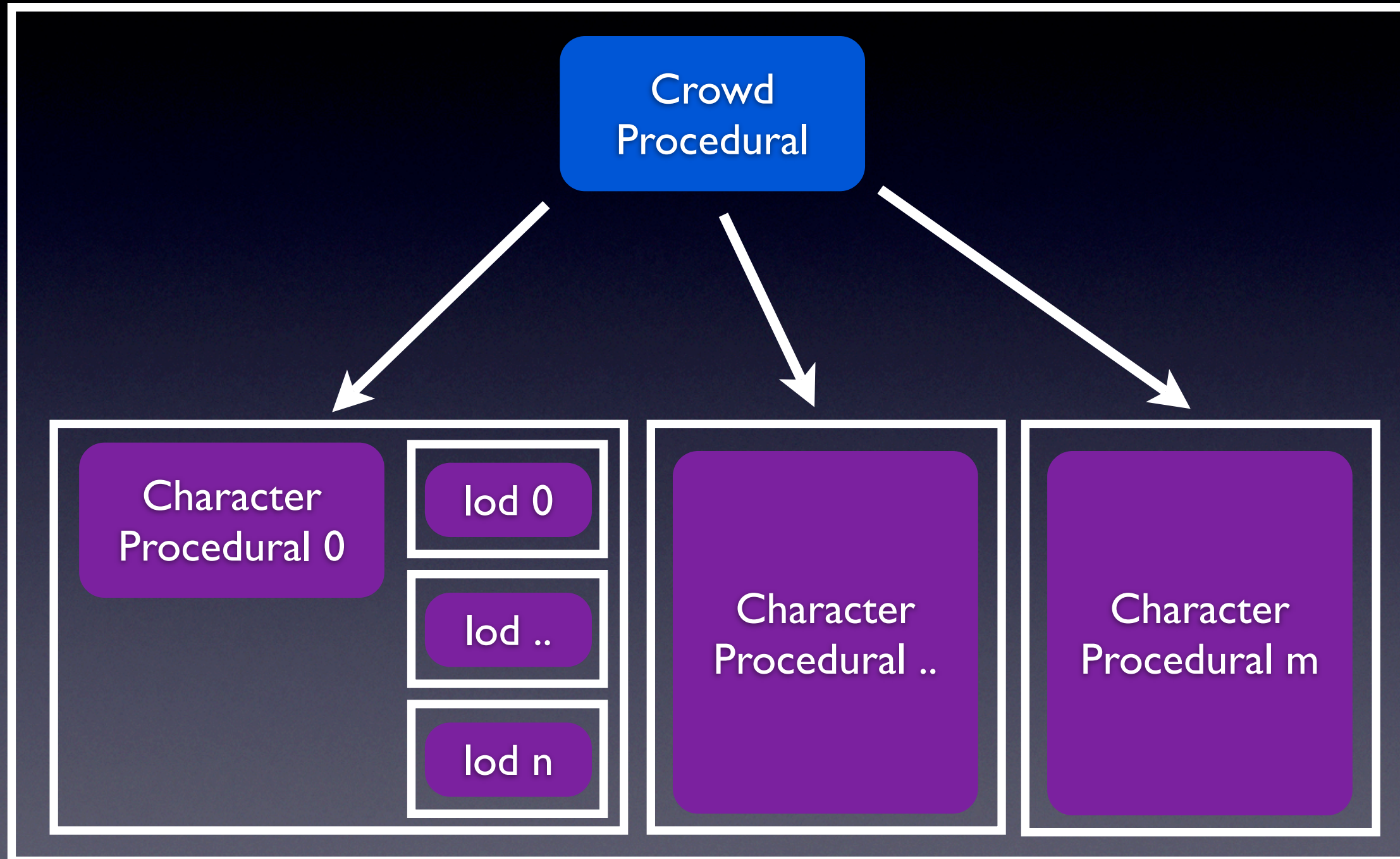


notes: load adb in maya -> duplicate chars ->
change individual attrs -> save as
callsheet ->
load callsheet into houdini -> render

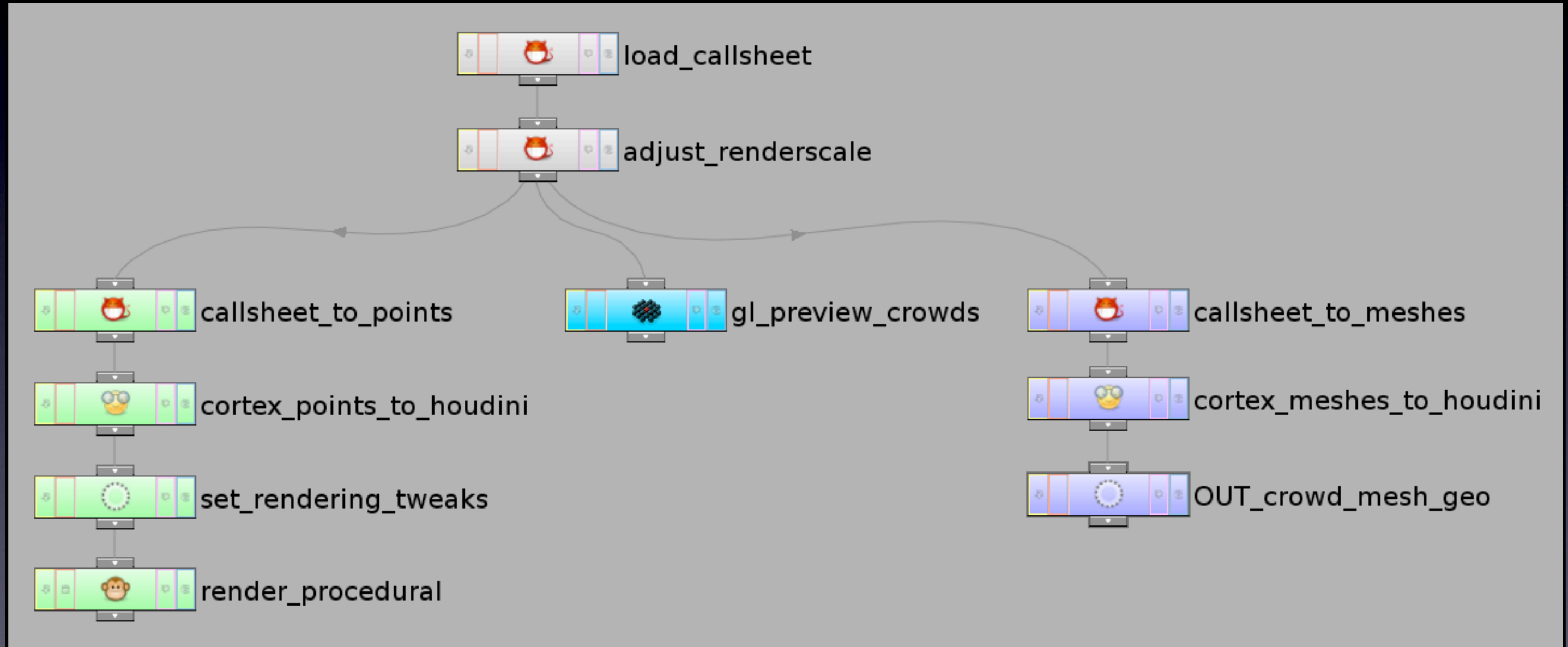
870_020_crowd_v006.mov
mihai.cioroba

#0001
1 1053

Nested Procedurals



Houdini Render Network





Conclusions

- Cortex, flexible and TD-friendly base for vfx / feature animation software development
- well suited to rapid prototyping due to its modular nature and pre-existing functionality
- easy to re-use / exchange data and functionality across multiple hosts
- ability to build workflows that enable artists to benefit from the strength of their preferred host application

