Open Source Physically Based Rendering with

appleseed

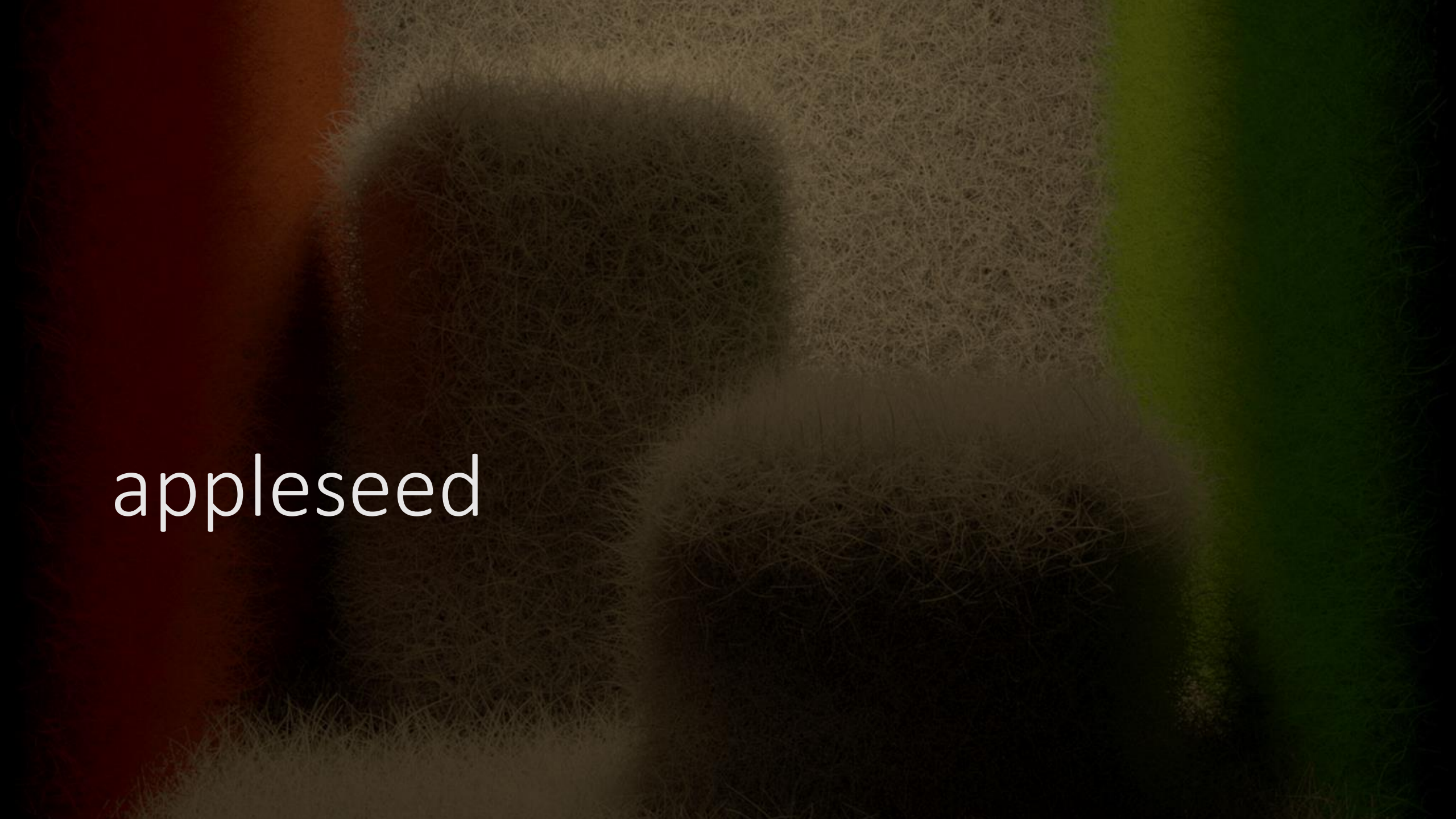
François Beaune

Project Founder

FMX2015

May 05-08, Stuttgart


Fetch
appleseed


appleseed

appleseed

- Open source rendering engine
- Designed for **VFX** and **animation**
- Targeted at individuals and small studios

appleseed

- Started in June 2009
- Small, professional team
- Not our main job

appleseed

- Pure CPU renderer
- Unidirectional path tracing
- Physically-based
- Highly programmable

appleseed

LIGHT TRANSPORT

- Distributed Ray Tracing
- Unidirectional Path Tracing
- Stochastic Progressive Photon Mapping
- Light Tracing

RENDERING MODES

- Multi-pass rendering
- Progressive rendering
- Interactive rendering
- Scene editing during rendering
- **Spectral rendering (31 bands)**
- RGB rendering
- Automatic spectral / RGB switching

CAMERA MODELS

- Pinhole camera
- Spherical camera
- Thin lens camera (depth of field)
- Polygonal diaphragm shapes
- Image-based diaphragm shapes

LIGHT SOURCE MODELS

- Point light
- Spot light
- Gobos
- Directional/parallel light
- Mesh light
- Purely diffuse emission profile
- Cone-shaped emission profile
- Image-based lighting
- Latitude-longitude environment maps
- Mirror-ball environment maps
- Preetham physically-based day sky
- Hosek & Wilkie physically-based day sky
- Physically-based sun

REFLECTION MODELS

- Lambertian BRDF (purely diffuse)
- Specular BRDF (perfect mirror)
- Specular BTDF (clear glass)
- Oren-Nayar Microfacet BRDF
- Ward Microfacet BRDF
- Blinn Microfacet BRDF
- GGX Microfacet BRDF
- Microfacet BTDF (rough glass)
- Anisotropic Ashikhmin-Shirley BRDF
- Kelemen BRDF
- **Disney's Layered BRDF**
- Arbitrary mixture of BRDFs

MOTION BLUR

- Camera motion blur
- Transformation motion blur
- Deformation motion blur
- Arbitrarily number of motion steps

PRODUCTION FEATURES

- **Open Shading Language**
- OSL shader library
- **Disney's SeExpr expressions**
- Rule-based render layers
- Hierarchical instancing
- Per-instance visibility flags
- Alpha mapping
- Automatic color space conversions
- Ray bias
- Light Near Start
- Max Ray Intensity
- Dozens of diagnostic modes

INTEROPERABILITY

- Windows, Linux and OS X (64-bit)
- OBJ, Alembic, BinaryMesh (proprietary)
- OpenEXR, PNG
- OSL shaders
- **Gaffer integration**
- Maya integration
- Blender integration

HACKABILITY

- Fully open source, MIT license
- Very clean code
- CMake build system
- Full featured C++ API
- Full featured Python 2.x/3.x API
- More than 1200 built-in unit tests
- Hundreds of built-in performance tests
- Rich, automatic functional test suite

PERFORMANCE

- Multithreaded, scalable
- SSE / SSE2 vectorization
- Memory-bounded texture cache
- Multiple Importance Sampling
- Efficient handling of alpha maps

TOOLS

- Graphical tool for scene edition
- Command line renderer
- Dropbox-based render farm tools
- OSL compiler and tools

71.8 million triangles
2.4 GB of textures
Disney layered BRDFs
SeExpr expressions
Image-based lighting
Depth of field

Average workstation
Intel Core-i7 5820K (6-core)
16 GB of RAM

* full_fmx_screencast_06_turbosmooth.appleseed - appleseed.studio

File View Rendering Diagnostics Debug Tools Help

Project Explorer

Filter: Clear

Name | Render Layer

- Project
  - Scene
    - camera
    - environment
    - Environment EDFs
    - Environment Shaders
    - Colors
    - Textures
    - Texture Instances
    - Assemblies
      - assembly
        - Colors
        - Textures
        - Texture Instances
        - BSDFs
        - EDFs
        - Surface Shaders
        - Materials
          - basic_orange
          - blade_material
          - carbon_material
          - chrome_material
          - eye_material
          - eyebrow_material
          - eyelashes_material
          - face_material
          - finger_armor
          - floor_material
          - hand_armor
          - palm_armor
        - Lights
        - Objects
        - Object Instances
          - full_fmx.arm_00_L_inst
          - full_fmx.arm_00_R_inst
          - full_fmx.arm_01_L_inst
          - full_fmx.arm_01_R_inst
          - full_fmx.arm_02_L_inst
          - full_fmx.arm_02_R_inst
          - full_fmx.arm_03_L_inst
          - full_fmx.arm_03_R_inst
          - full_fmx.arm_04_L_inst
          - full_fmx.arm_04_R_inst

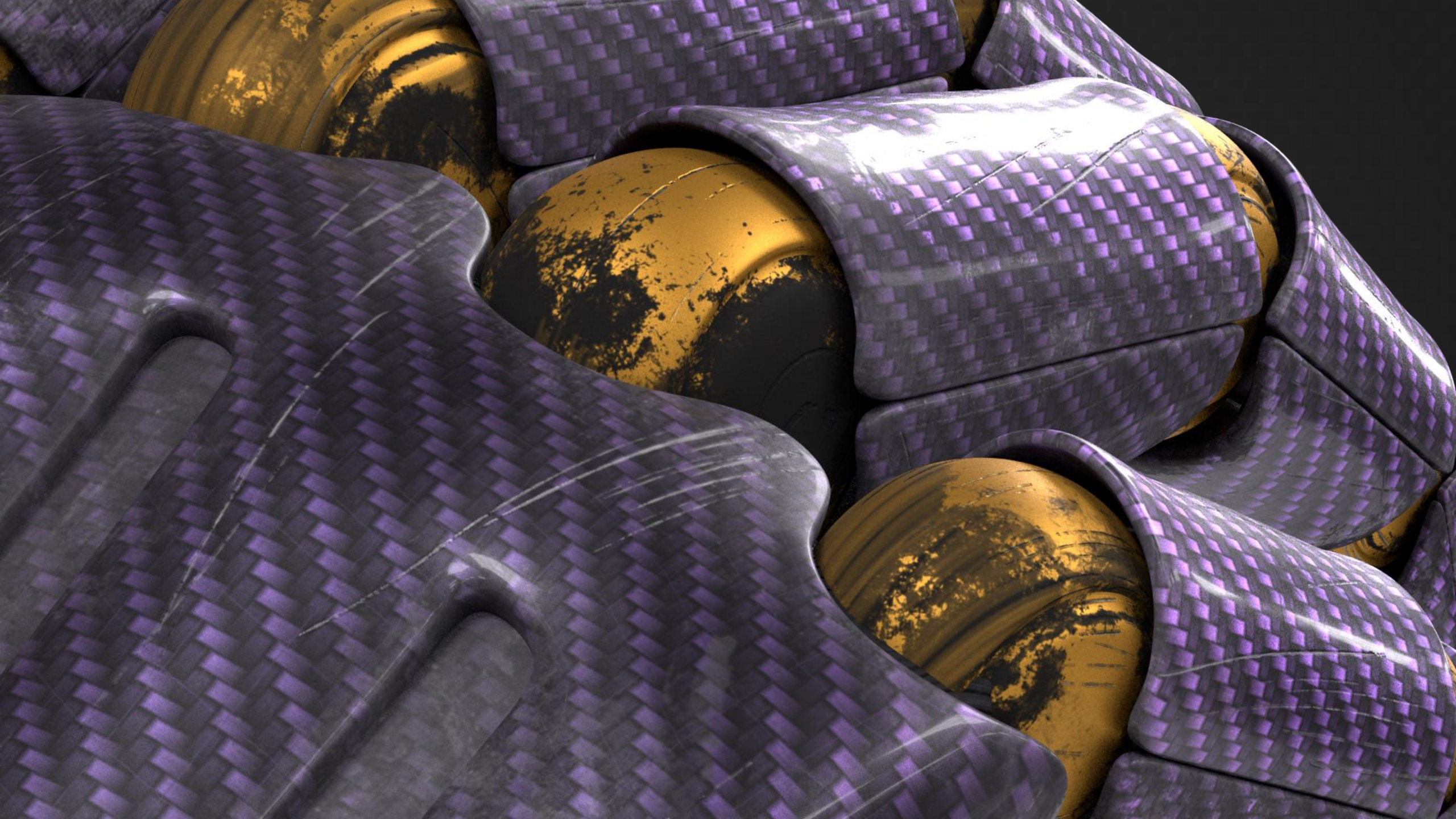
RGB

Picking Mode: Material

Attribute Editor

Log

```
2015-05-01T12:31:02.183267Z <174> 17866 MB info    | 2,007,040 samples, 4.1 samples/pixel, 661,298 samples/second
2015-05-01T12:31:03.183226Z <174> 17866 MB info    | 2,695,168 samples, 5.5 samples/pixel, 667,947 samples/second
2015-05-01T12:31:04.183186Z <174> 17866 MB info    | 3,481,600 samples, 7.1 samples/pixel, 691,479 samples/second
2015-05-01T12:31:05.188151Z <174> 17866 MB info    | 4,366,336 samples, 8.9 samples/pixel, 722,903 samples/second
2015-05-01T12:31:06.224145Z <174> 17866 MB info    | 5,103,616 samples, 10.4 samples/pixel, 724,945 samples/second
2015-05-01T12:31:07.229109Z <174> 17866 MB info    | 5,709,824 samples, 11.7 samples/pixel, 707,011 samples/second
2015-05-01T12:31:08.231071Z <174> 17866 MB info    | 6,430,720 samples, 13.1 samples/pixel, 708,151 samples/second
2015-05-01T12:31:09.286084Z <174> 17866 MB info    | 7,610,368 samples, 15.5 samples/pixel, 754,772 samples/second
2015-05-01T12:31:10.380134Z <174> 17867 MB info    | 8,142,848 samples, 16.6 samples/pixel, 731,087 samples/second
2015-05-01T12:31:11.380094Z <174> 17867 MB info    | 8,888,320 samples, 18.1 samples/pixel, 726,644 samples/second
```



appleseed

- Modern
  - Interactive
  - Single pass
  - Tessellation-free
  - Flicker-free

appleseed

- Reliable
  - Avoid (bad) surprises
  - Avoid crashes
  - Avoid regressions
  - Value correctness
  - Incremental change = incremental effect

appleseed

- Flexible
  - Avoid arbitrary limitations
  - Provide tons of public extension points
  - Maximize programmability
    - OpenShadingLanguage
    - Disney's SeExpr
    - Full C++ API
    - Full Python 2.x / 3.x API

appleseed

- Hackable
  - Fully open source
  - Liberal license (MIT) from the start
  - Everything hosted on GitHub
  - Development fully in the open
  - Using only open source or free tools
  - Welcoming, helpful, mature community

appleseed

Team & Process

François Beaune

Esteban Tovagliari

François Gilliot

Jonathan Topf

Hans Hoogenboom

Joel Daniels

Dorian Fevrier

Haggi Krey

Srinath Ravichandran

Marius Avram

R&D

François Beaune

Esteban Tovagliari

GSoC ‘14 Students

Srinath Ravichandran

Marius Avram

Exporters & Integrations

Esteban Tovagliari

Jonathan Topf

Hans Hoogenboom

Joel Daniels

Haggi Krey

Fetch

François Beaune

François Gilliot

Jonathan Topf

appleseed

- Core practices and values
  - Collective code ownership
  - Continuous refactoring
  - Pull requests reviews
  - Unit tests
  - End-to-end tests
  - Performance regression tests

Selected Works

Light & Dark (BBC Four Documentary)

Light & Dark (BBC Four Documentary)

Light & Dark (BBC Four Documentary)

Character designs by appleseed users

Fetch, a very short film

Fetch, a very short film

appleseed

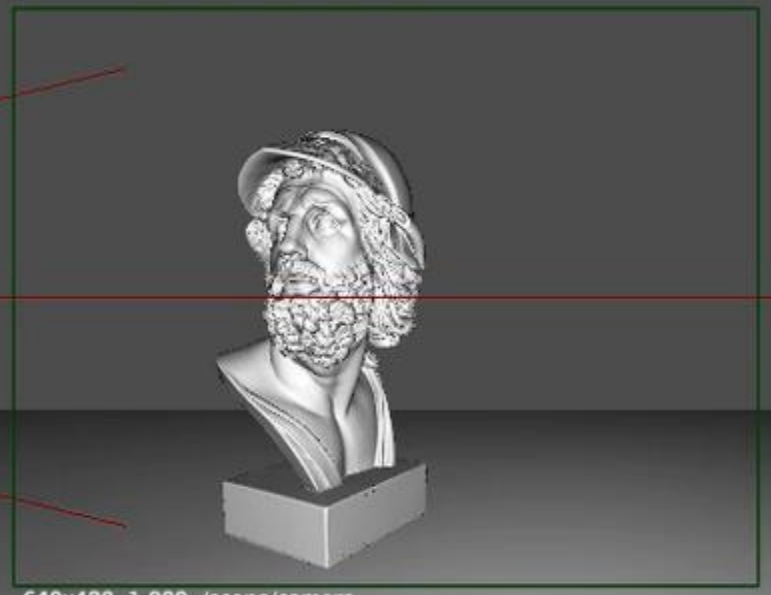
appleseed now fully integrated into Image Engine’s Gaffer

Gaffer : scene_template.gfr * - /home/est/gaffer/projects/default/scripts

Gaffer File Edit Layout Execute Help

Viewer : Outputs

Render Camera



640x480, 1.000, /scene/camera

Node Editor : Outputs | Scene Inspector : Outputs

Node Name Outputs Outputs

Settings Node User

Interactive/Beauty

Name	Interactive/Beauty
File Name	beauty
Type	ieDisplay
Data	rgba
displayHost	localhost
remoteDisplayType	GafferImage::GafferDisplayDriver
quantize	Float
displayPort	1559
driverType	ClientDisplayDriver

Scene Hierarchy : Outputs | Script Editor | Node Editor : Inte

Node Name eractiveAppleseedRender InteractiveAppleseedRender

Settings User

State Stopped

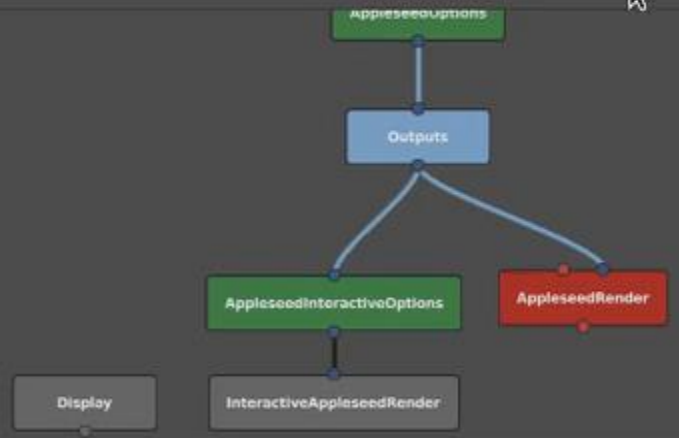
Update Lights

Update Attributes

Update Cameras

idate Coordinate Systems

Node Graph



1 1 1 100 100

appleseed

Welcoming contributions!

appleseed

Home
http://appleseedhq.net/

GitHub
https://github.com/appleseedhq/appleseed

Development Mailing List
https://groups.google.com/forum/#!forum/appleseed-dev

Twitter
https://twitter.com/appleseedhq

Making Fetch

Making Fetch

- Initiated “Project Mescaline” in June 2012
- Goals:
  - Test & validate appleseed on a small production
  - Showcase & promote appleseed
  - Sharpen our skills
  - Have fun with friends
- Constraints:
  - Final render 100% appleseed
  - Tiny budget

Making Fetch

- Small team:
  - 1 for direction & art
  - 1 for pipeline & render
  - 1 for sound effects & soundtrack (late in project)
  - Help from friends
- Strictly free-time / rainy days project
- Effort:
  - Planned: 8 months
  - Actual: 19 months ☺

Making Fetch

- “Fetch, a very short film”
- 2 minutes hand-animated short
- Targeted at kids
- Miniature look
- Fully rendered with appleseed

Making Fetch

- Pipeline
- Render Setup
- Render Farm
- Conclusion

Making Fetch

Pipeline

Making Fetch – Pipeline

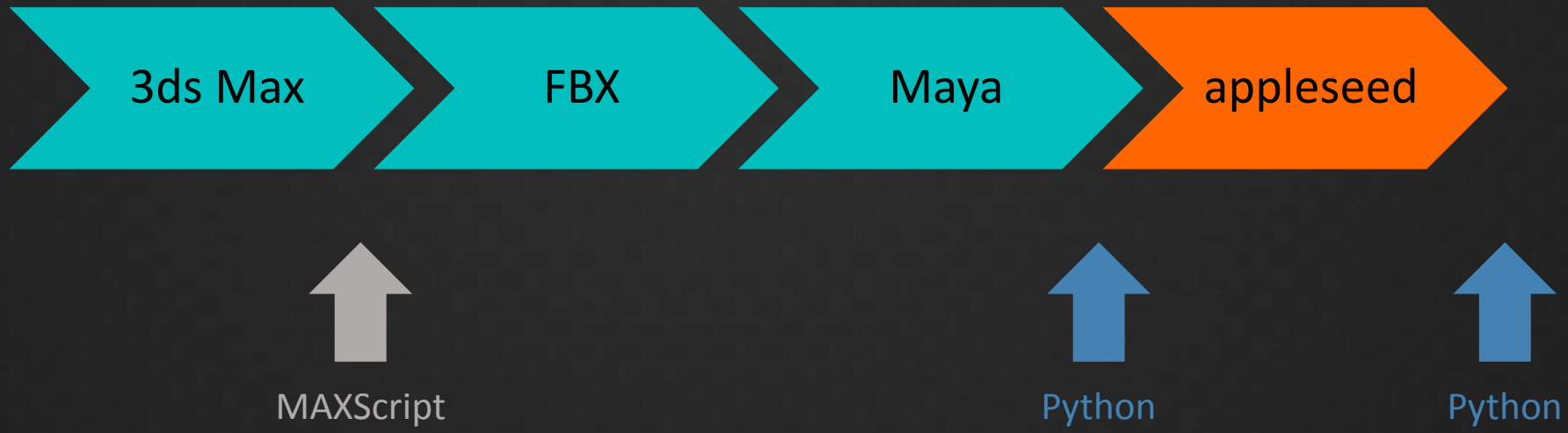
- Modeling, animation, lookdev in 3ds Max
  - Tool of choice for the artist
- Lookdev mostly with V-Ray
  - Integrated in 3ds Max

Making Fetch – Pipeline

- Problem: no 3ds Max-to-appleseed exporter
- Writing a full-featured exporter for 3ds Max too big of a project
- Solution:

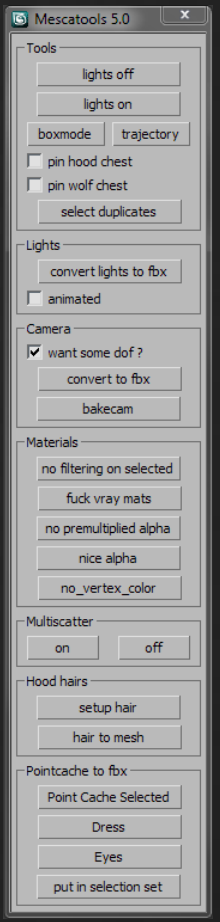
Making Fetch – Pipeline

- Problem: no 3ds Max-to-appleseed exporter
- Writing a full-featured exporter for 3ds Max too big of a project
- Solution:



Making Fetch – Pipeline

- FBX format would lose lots of information
  - Area lights
  - Gobos
  - DOF parameters…
- Several custom scripts to remedy this
  - 3ds Max side (MAXScript)
    - Store various info into custom attributes
    - Prepare the scene before FBX export
  - Maya side (Python)
    - Retrieve info from custom attributes
    - Adjust materials



Making Fetch – Pipeline

- Initial lookdev mostly with V-Ray 3
- Materials translated to appleseed
  - Automatic translation during export
  - Lots of post-export tweaks
    - Automatic tweaks via Python scripts


TITLE
END

Making Fetch

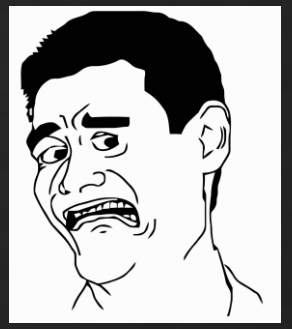
Render Setup

Making Fetch – Render Setup

- Art direction called for:
  - Miniature look = realistic lighting + shallow DOF
  - Mostly forest shots with almost no direct illumination
  - Millions of grass blades and tree leaves in nearly every shot
    - All translucent (thin translucency)
    - All using alpha cutouts
  - Image-based lighting in 25% of the shots
  - Many scenes with really strong motion
    - Transformation and deformation

Making Fetch – Render Setup

- Art direction called for:
  - Miniature look = realistic lighting + shallow DOF
  - Mostly forest shots with almost no direct illumination
  - Millions of grass blades and tree leaves in nearly every shot
    - All translucent (thin translucency)
    - All using alpha cutouts
  - Image-based lighting in 25% of the shots
  - Many scenes with really strong motion
    - Transformation and deformation



Making Fetch – Render Setup

- Physically-based materials & lighting
- Unidirectional path tracing, 2 bounces
- 64-400 samples/pixel depending on DOF and MB
- Single pass, no baking whatsoever
- One AOV per light (4-6 lights per shot)
- Plus a few special AOVs
  - Girl’s hair
  - Wolf’s eyes…

Making Fetch – Render Setup

- Full HD resolution (1920x1080)
- 24 frames/second
- 2767 frames (~ 115 seconds)

Making Fetch – Render Setup

- 3120 individual scenes to render
  - 2767 frames + a couple backgrounds rendered separately
- 32 GB of final render data
  - OpenEXR textures (RLE-compressed)
  - Proprietary geometry format (LZ4-compressed)
- Tens of thousands of files

Making Fetch

Render Farm

Making Fetch – Render Farm

- Obviously too much work for one or even a couple machines
- No money meant:
  - Not buying additional machines
  - Not renting a render farm
  - Not paying for Amazon Web Services
- So?

Making Fetch – Render Farm

- Friends to the rescue!
- Challenges:
  - 32 shots, tens of thousands of files, GB of data
  - Friends all around the place in Europe
  - Random machines
  - Random OS
  - Machines only available occasionally
  - Many machines behind firewall / NAT
  - No technical expertise or rendering experience for most of them

Making Fetch – Render Farm

Solution:

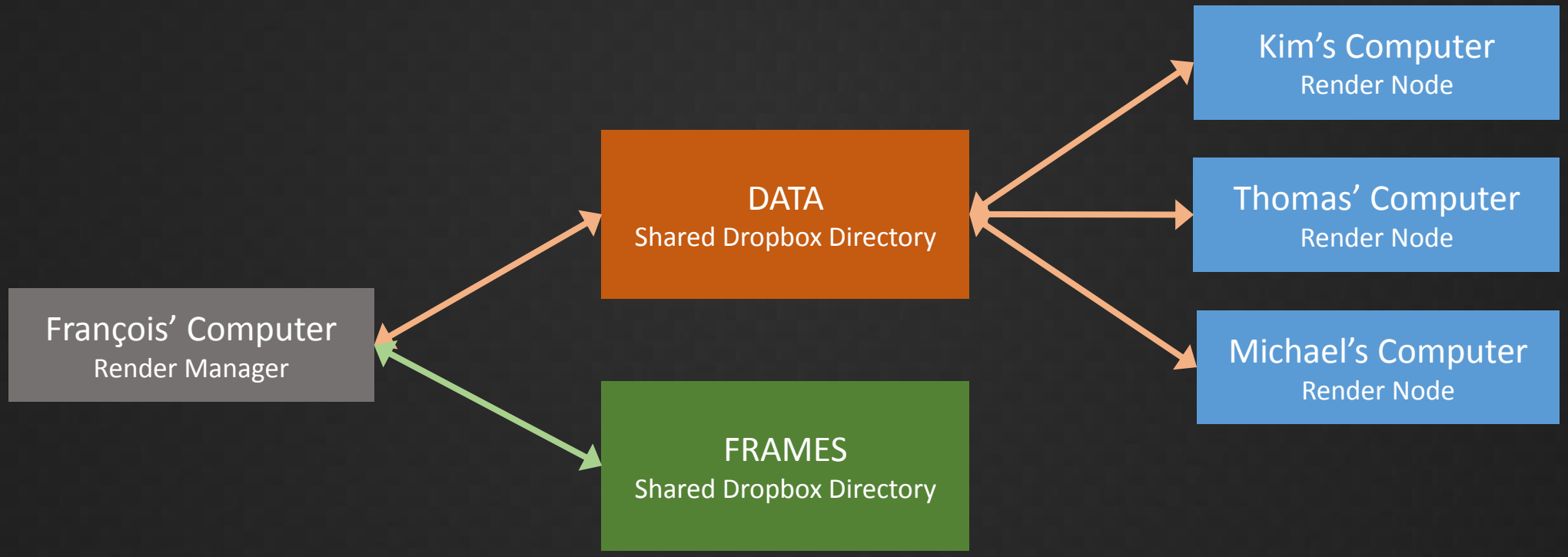
DYI render farm based on Dropbox

Making Fetch – Render Farm

Use Dropbox as **delivery channel**,

and for **command & control**

Making Fetch – Render Farm



Making Fetch – Render Farm

- Shared directory
- Assume Dropbox Basic accounts (free!) = 2 GB
- Hosts:
  - appleseed binaries for Windows, Linux and OS X
  - Data for one or multiple partial shots

Making Fetch – Render Farm

- Shared directory on Dropbox Pro accounts
- Hosts all rendered frames
  - Ended up with 140 GB worth of OpenEXR files
- Only shared between team members

FRAMES
Shared Dropbox Directory

Making Fetch – Render Farm

- A variety of 64-bit machines
  - Windows Vista, 7, 8
  - Linux
  - OS X
- Mostly quad core machines
- Typically available nights and week-ends
- Render nodes run the render node script
- Users free to kill render node script at any time

Kim's Computer
Render Node

Thomas' Computer
Render Node

Michael's Computer
Render Node

Making Fetch – Render Farm

- Render nodes run a Python script:

Loop:

"Acquire" scene by appending a per-machine suffix to scene file

Render scene

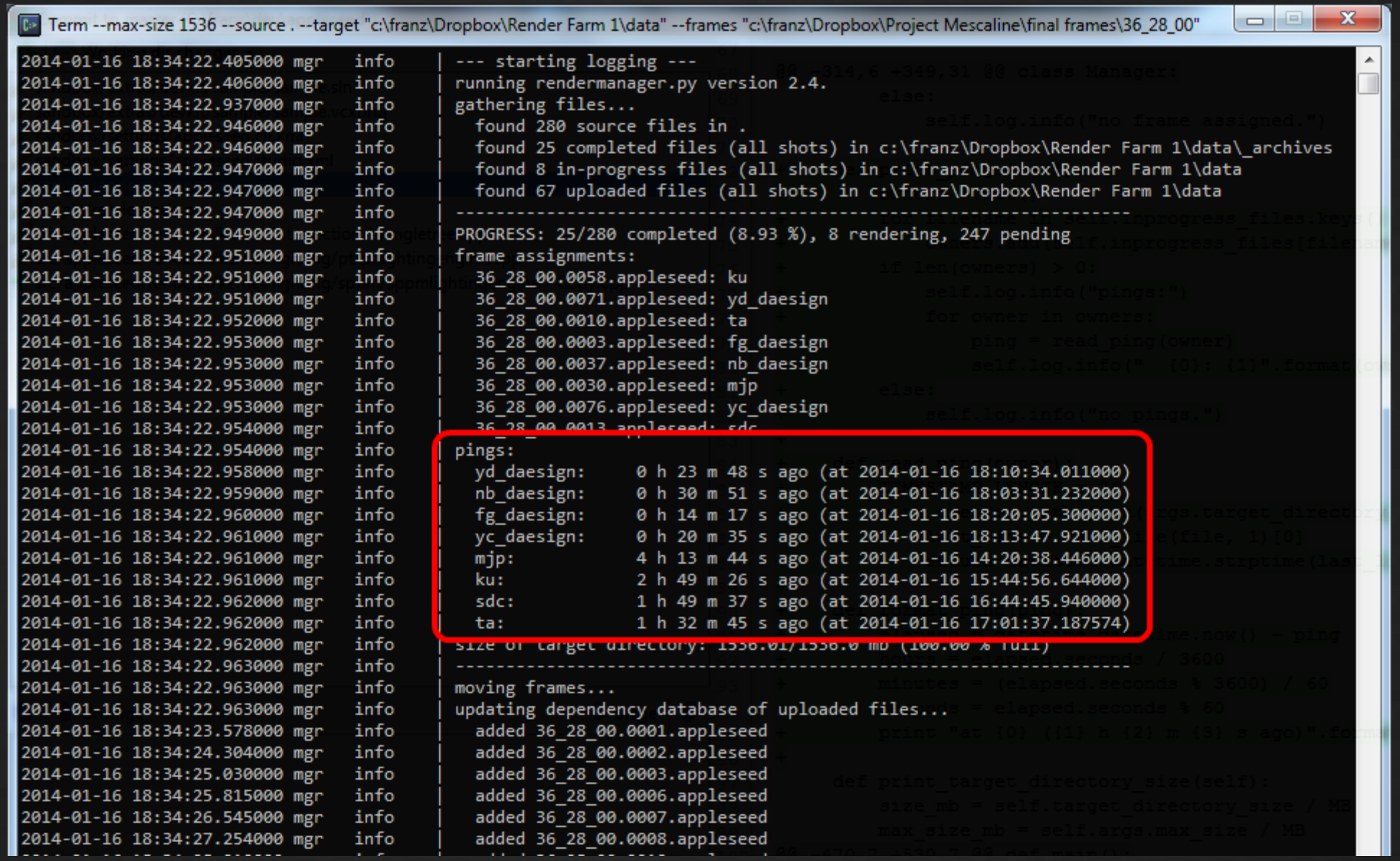
Move rendered frame files to "frames" subdirectory in DATA

Move rendered scene file to "archive" subdirectory in DATA

Making Fetch – Render Farm

François' Computer
Render Manager

- Underpowered Core i5 laptop
- Managing rendering:
  - Upload/remove shot data as required
  - Honor 2 GB size limitation of DATA at all times
  - Move rendered frames from DATA to FRAMES
  - Monitor and print render farm health, activity and progress
- Running 24/7

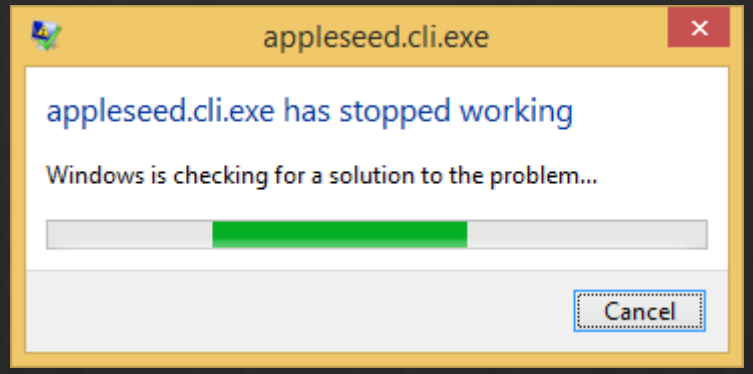

Term --max-size 1536 --source . --target "c:\franz\Dropbox\Render Farm 1\data" --frames "c:\franz\Dropbox\Project Mescaline\final frames\36_28_00"
2014-01-16 18:34:22.405000 mgr info | --- starting logging ---
2014-01-16 18:34:22.406000 mgr info | running rendermanager.py version 2.4.
2014-01-16 18:34:22.937000 mgr info | gathering files...
2014-01-16 18:34:22.946000 mgr info | found 280 source files in .
2014-01-16 18:34:22.946000 mgr info | found 25 completed files (all shots) in c:\franz\Dropbox\Render Farm 1\data_archives
2014-01-16 18:34:22.947000 mgr info | found 8 in-progress files (all shots) in c:\franz\Dropbox\Render Farm 1\data
2014-01-16 18:34:22.947000 mgr info | found 67 uploaded files (all shots) in c:\franz\Dropbox\Render Farm 1\data
2014-01-16 18:34:22.947000 mgr info | ---
2014-01-16 18:34:22.949000 mgr info | PROGRESS: 25/280 completed (8.93 %), 8 rendering, 247 pending
2014-01-16 18:34:22.951000 mgr info | frame assignments:
2014-01-16 18:34:22.951000 mgr info | 36_28_00.0058.appleseed: ku
2014-01-16 18:34:22.951000 mgr info | 36_28_00.0071.appleseed: yd_daesign
2014-01-16 18:34:22.952000 mgr info | 36_28_00.0010.appleseed: ta
2014-01-16 18:34:22.953000 mgr info | 36_28_00.0003.appleseed: fg_daesign
2014-01-16 18:34:22.953000 mgr info | 36_28_00.0037.appleseed: nb_daesign
2014-01-16 18:34:22.953000 mgr info | 36_28_00.0030.appleseed: mjp
2014-01-16 18:34:22.953000 mgr info | 36_28_00.0076.appleseed: yc_daesign
2014-01-16 18:34:22.954000 mgr info | 36_28_00.0013.appleseed: sdc
2014-01-16 18:34:22.954000 mgr info | pings:
2014-01-16 18:34:22.958000 mgr info | yd_daesign: 0 h 23 m 48 s ago (at 2014-01-16 18:10:34.011000)
2014-01-16 18:34:22.959000 mgr info | nb_daesign: 0 h 30 m 51 s ago (at 2014-01-16 18:03:31.232000)
2014-01-16 18:34:22.960000 mgr info | fg_daesign: 0 h 14 m 17 s ago (at 2014-01-16 18:20:05.300000)
2014-01-16 18:34:22.961000 mgr info | yc_daesign: 0 h 20 m 35 s ago (at 2014-01-16 18:13:47.921000)
2014-01-16 18:34:22.961000 mgr info | mjp: 4 h 13 m 44 s ago (at 2014-01-16 14:20:38.446000)
2014-01-16 18:34:22.961000 mgr info | ku: 2 h 49 m 26 s ago (at 2014-01-16 15:44:56.644000)
2014-01-16 18:34:22.962000 mgr info | sdc: 1 h 49 m 37 s ago (at 2014-01-16 16:44:45.940000)
2014-01-16 18:34:22.962000 mgr info | ta: 1 h 32 m 45 s ago (at 2014-01-16 17:01:37.187574)
2014-01-16 18:34:22.962000 mgr info | size of target directory: 1536.01/1536.0 MB (100.00 % full)
2014-01-16 18:34:22.963000 mgr info | ---
2014-01-16 18:34:22.963000 mgr info | moving frames...
2014-01-16 18:34:22.963000 mgr info | updating dependency database of uploaded files...
2014-01-16 18:34:23.578000 mgr info | added 36_28_00.0001.appleseed
2014-01-16 18:34:24.304000 mgr info | added 36_28_00.0002.appleseed
2014-01-16 18:34:25.030000 mgr info | added 36_28_00.0003.appleseed
2014-01-16 18:34:25.815000 mgr info | added 36_28_00.0006.appleseed
2014-01-16 18:34:26.545000 mgr info | added 36_28_00.0007.appleseed
2014-01-16 18:34:27.254000 mgr info | added 36_28_00.0008.appleseed

Making Fetch – Render Farm

- Render Manager Robustness
  - “Rendering state” fully implicit
  - Render manager free to start/stop/crash at any time

Making Fetch – Render Farm

- Render Nodes Robustness
  - Not all geometry files or textures available to render given scene
  - On Windows: appleseed crash = Windows Error Reporting Message Box



Making Fetch – Render Farm

- Advantages
  - Easy for friends to join & participate
  - Reliable transport of scene data and rendered frames
  - Easy to add/remove render nodes
  - Easy to update new appleseed binaries
  - Easy to analyze performance and crashes of render nodes
  - Eventually quite robust

Making Fetch

Conclusion

Mescaline Render Planning

Shot	Version	Description	3ds max start	3ds max end	Needs DOF?	Needs Sky?	Shutter Open Duration	Pixel Samples	Light Samples	Env Samples	Pending Remarks
00	08	opening shot on the valley	0	132	Y	N	0.5	100	1	-	
01	26	she appears on the hill	0	115	Y	YES	0.5	100	8	16	
03	14	she runs down the hill	0	170	Y	YES	0.25	64	16	1	
04	07	she enters the forest	0	105	Y	N	0.5	64	10	-	
04_bg	07	background	0	105	NO	YES	no motion blur	16	1	1	
05	20	she plays in the forest 1	0	135	Y	N	0.5	100	4	-	
06	29	she plays in the forest 2	0	300	Y	N	0.5	80	4	-	
07	12	she sees the wolf	0	90	Y	N	0.5	200	16	-	
09	18	she waits and walks by the wolf	0	168	NO	N	0.5	200	1	-	
10	19	she walks by the wolf	0	118	Y	N	0.5	64	4	-	
11	24	wolf stands up	0	188	Y	N	0.5	100	4	-	
17	28	she jumps over a large root and wolfgang stops	15	130	Y	N	0.5	200	1	-	
26	20	she stops at the edge of the cliff	0	115	Y	YES	0.5	100	1	1	
26_bg	20	background	0	115	NO	N	0.5	4	1	-	
28	12	she looks around to find a way	0	73	Y	YES	0.5	100	1	1	
28_bg	12	background	0	73	NO	N	0.5	4	1	-	
31	15	the girl turns back to the forest	0	50	Y	YES	0.5	100	1	1	
33	07	she puts her hand in the basket	0	61	Y	YES	0.5	100	1	1	
36	28	wolf arrives and wants to play	0	280	Y	YES	0.5	64	1	1	
37	13	she runs towards the exit	0	42	Y	N	0.5	100	1	-	
38	02	closeup hood face	0	45	Y	N	0.5	100	1	-	
39	00	closeup wolf face	0	47	Y	N	0.5	400	1	-	
40	01	closeup hood feet	0	75	Y	N	0.5	200	1	-	
41	01	closeup wolf face	0	27	Y	N	0.5	400	1	-	
42	01	closeup wolf feet	0	33	Y	N	0.5	200	1	-	
43	05	she turns away and runs	97	148	Y	N	0.5	200	1	-	
44	11	she runs and jumps over a gap	0	40	Y	N	0.5	400	1	-	
45	04	she looks behind while she runs	0	34	Y	N	0.5	200	1	-	
46	05	she sees the exit	0	44	Y	N	0.5	600	1	-	
50	01	wolf runs	0	50	Y	N	0.5	300	1	-	
51	30	wolf runs toward the hood	0	74	Y	N	0.15	200	1	-	
53	19	the girl tries to lift the branch without success	0	50	Y	N	0.25	64	4	-	
		Total number of frames		**3120**						-	

Color Legend: Ready to Import | Ready to Render | Rendering | Done | Broken | Re-render

Making Fetch – Conclusion

- Special developments
  - Efficient handling of massive number of alpha cutouts
  - Dropbox-based render farm tools
  - Vast improvements to Maya-to-appleseed exporter (mayaseed)
- Everything has been released

Making Fetch – Conclusion

- appleseed one of the most reliable component of the pipeline
- Did not have to worry about:
  - Flickering
  - Glitches in the middle of a shot
  - Unpredictable catastrophic slowdown

Making Fetch – Conclusion

- Only two questions:
  - What render settings?
  - How long will it take?

Making Fetch – Conclusion

- What would we do differently today?
  - Export Alembic files from 3ds Max
  - Lookdev in Gaffer
  - Real hair?
  - OSL shaders?

Making Fetch – Conclusion

- Published on Vimeo
- Picked up by many big animation channels, ended up on YouTube
- Great reception on the web
- Some really nice articles written about the project

Making Fetch – Conclusion

- Official TIFF Kids 2015 selection!


Thank you!


Questions?

Extras

There’s never enough!

Additional References

Direct Ray Tracing of Full-Featured Subdivision Surfaces with Bezier Clipping
http://jcgt.org/published/0004/01/04/

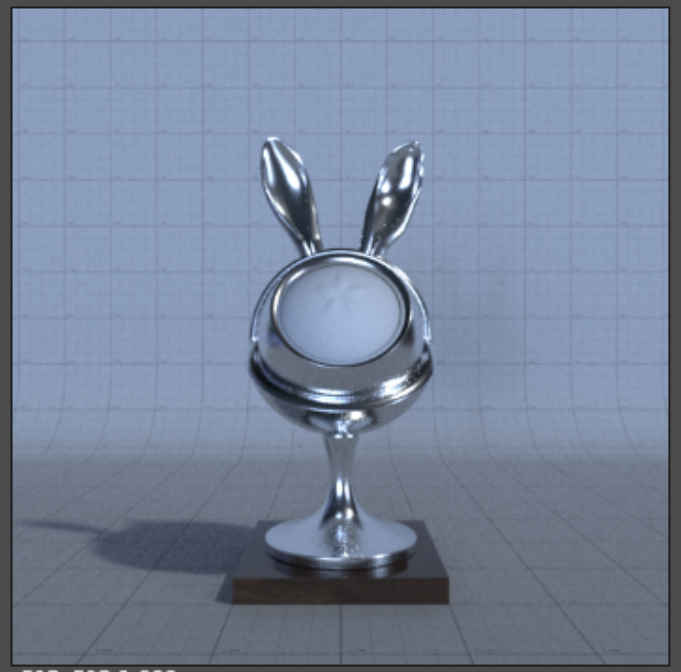
appleseed

- Many important features still missing
  - Volume rendering
  - Subsurface scattering
  - Subdivision surfaces
  - Displacement
  - Robust, complete, performant Maya integration
  - Documentation

Gaffer : material_tester_metal2.gfr * - /home/est/gaffer/projects/default/scripts

Gaffer File Edit Layout Execute Help

Viewer : Display



512x512 1.000

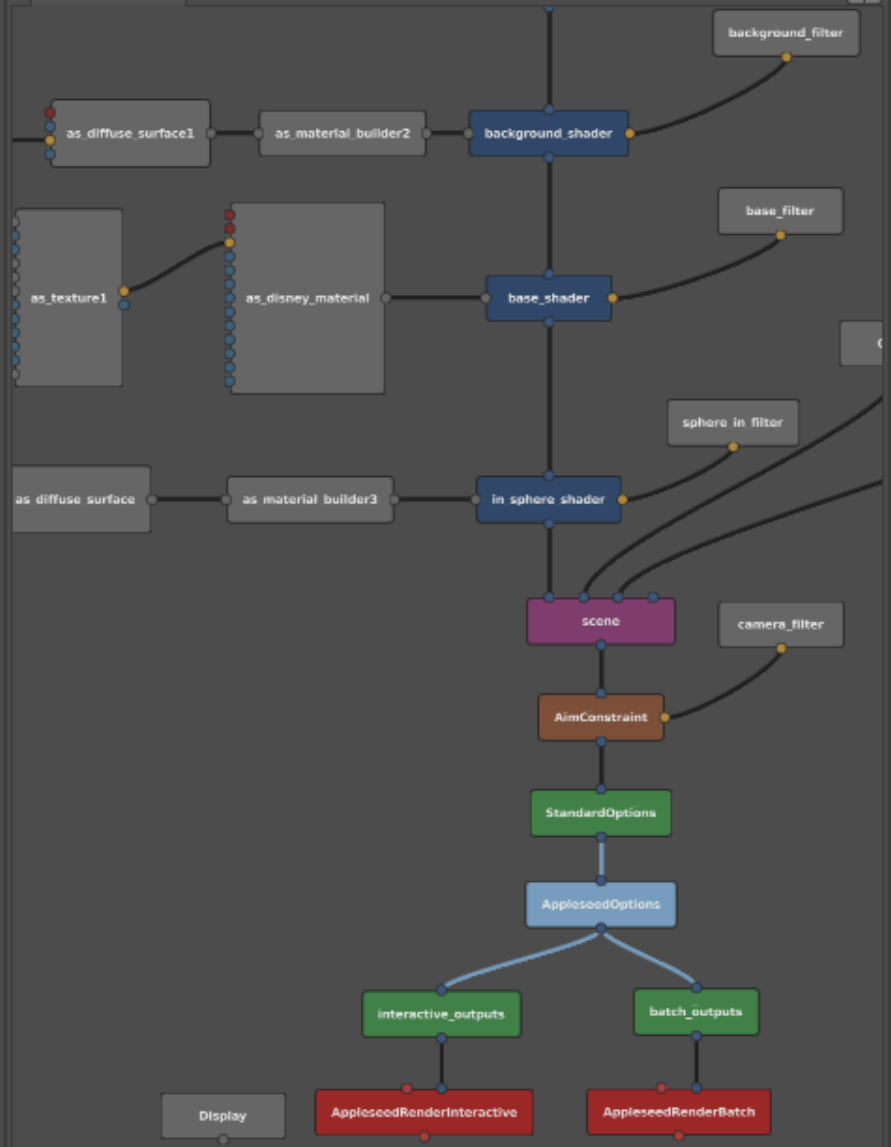
XY: 566, 163 RGBA: 0.000, 0.000, 0.000, 0.000 HSV: 0

Scene Hierarchy : AppleseedOptions

Name

scene

Node Graph



Node Editor : AppleseedOptions | Scene Inspector : AppleseedOptions

Node Name: AppleseedOptions — AppleseedOptions

Settings | Node | User

Main (AA Samples 64, Lighting Engine pt)

Passes	1
AA Samples	64
Force Antialiasing	✓
Decorrelate Pixels	✓
Lighting Engine	Unidirectional Path Tracer
Mesh File Format	BinaryMesh

Environment (Environment /scene/lights/env_light)

Distribution Ray Tracer

Unidirectional Path Tracer (Max Bounces 10, Min Bounces 3, Next Event Estimation, IBL Samples 8)

Direct Lighting	✓
Image Based Lighting	✓
Caustics	
Max Bounces	10
RR Start Bounce	3
Next Event Estimation	✓
Direct Lighing Samples	1
IBL Samples	8
Max Ray Intensity	1

SPPM

System

1 1 1 100 100