

MORTEZA RAMEZANALI

3D | LIGHTING | EFFECTS SHOWREEL BREAKDOWN

+ 971 50 745 7405
moto @ cgdna.com
www.cgdna.com

Segment 01 Chevrolet Heartbeat

Software Maya, MEL, Mental ray, Reelflow, Photoshop

Description Look development, Set up lighting, Mental ray shader assembling, Set up render layers and passes including Diffuse - Specular - AO - Reflection - Normal, Texturing, Vehicle rigging including automated Terrain detection - Suspension - Wheels spins using my exclusive Mathematics Nodes for Maya (plug-in), Splash simulation, Writing scripts to generate lights trail from tail lights, Camera rigging and animation

Segment 02 Chevrolet in Motion

Software Maya, Houdini, Reelflow, Python

Description Set up primary splash rigs in Maya, Simulating water splashes, Writing python scripts to generate dripping waters in Reelflow, Adding extra droplets and sprays using Houdini

Segment 03 Procedural Tree Modeling

Software Maya, MEL, Mental ray, Mudbox, Photoshop

Description Modeling the base structure using Maya's built-in Paint Effects and sculpted in Mudbox for extra detail, Writing scripts based on simple LSystem rules to add branches and twigs, Using particles for initial leaf placement, Writing scripts to automate generation of mentalray binary proxies from leaves and branches, Texturing (color/ normal/ translucent/ etc), Optimizing scene and render settings for rendering ~15M unique triangles

Segment 04 Qui Rit

Software Maya, MEL, Mental ray, Fusion, Photoshop

Description Look Development, Indoor and outdoor lighting, Mentalray shader assembling, Setting up render passes and layers, Texture painting and dialing, Vegetation modeling, Writing scripts to modify paint effects strokes attributes based on texture maps, Layers and effects compositing

Segment 05 FLUX: An approximate image based lighting tool for Maya – *Plug in Development*

Software C++, Maya API, MEL, Boost, OpenGL, OpenMP, OpenExr

Main Features:

- Less Noise in Render
- Less Render-time
- Editable and Key able
- Precise Control over the result
- Ideal for lighting Fur, Hair, Fluids and matching lights with Live action
- Combines and interpolates multiple sets of HDRI
- Pipeline friendly; Exports centroids data to Maya's native data structure
- Artists friendly; Customized "Attribute Editor Templates" for easy node connection and traversal
- Fast and responsive; Multi-threaded and accelerated using built-in kdTree
- Interactive modification of lights properties
- Output statistics, verbose, diagnostic data

Flux Pipeline:

Stage (1) - Importing data

- **[fluxLoader]**: Supports all type of openExr files (Tiled, Scanline, Multichannel) with channel shuffling and extra bit info; Built-in support for Latitude-Longitude and Cube environment maps.

Stage (2) - Pre-processing

- **[fluxLuminance]**: Computes luminance energy of input data with compensation at poles
- **[fluxSat]**: Accelerates computation using SAT (Summed Area Tables)

Stage (3) - Centroids Generation

- **[fluxMedianCut]**: Adaptive and fixed importance sampling using MedianCut algorithms with an optional pre-processing Barlet filtering
- **[fluxHammerslayHalton]**: Importance sampling with Hammerslay / Halton sequences and computing energy based on Voronoi patterns
- **[fluxUserInput]**: Generates centroids based on user's input, UVs, Radius, Falloff; Computing luminance/color by using Mean, Normal or Sum operator.

Stage (4) - Visualizing

- **[fluxCentroidsViz]**: Visualizes centroids data in viewport including Color, Luminance scale, Magnitude, IDs, Centroids orientation

Stage (5) - Centroids Editing

- **[fluxCentroidsTransform]**: Centroids transformation including translation, rotation, radius and orientation; Meta information including Location, description and centroids count
- **[fluxCentroidsField]**: Generates centroids field by combining multiple sets using weight balance or union interpolation
- **[fluxCentroidsEdit]**: Clustering and selecting centroids by Luminance, Color, UV ranges, Volume and IDs; Cleanup and deleting centroids; Editing centroids color and luminance using stackable operators with priority feature; including Generic, Set-Range, Normalize, Clamp, Override (Axial) and remapping in RGB or HSV color space

Stage (6) - Exporting data

- **[fluxCentroidsExport]**: Exports centroids data to Maya's native data structure including Matrices, Colors, Energies, UVs; Ideal for transferring data to shaders, lights, etc.
- **[fluxGlobal]**: Global container for flux's IBL settings, lights rigs and flux's Mentalray shader
- **[fluxMentalrayIBL]**: Utilizing centroids data inside Mentalray
- **[fluxLightsGeneration]**: Generates light rigs based on centroids data and global container nodes

Segment 06 Formula One

Software Maya, Fusion, MEL, Photoshop, Mokey

Description Modeling highly detailed F1 cars from photos, Texturing (color/reflection/decals/etc), Object tracking, Vehicle rigging, Setup lighting rigs for outdoor lighting, Shader assembling, Setup render passes and buffers including color / shadow / occlusion / velocity / id / reflection / mattes / clear coat / etc, Cleanup plates using Fusion and Mokey, Layer compositing, Developing fusion color operator node using Fusion API for diagnostic ranges in 32Bit workflow

MORTEZA RAMEZANALI

3D | LIGHTING | EFFECTS SHOWREEL BREAKDOWN

+ 971 50 745 7405
moto @ cgdna.com
www.cgdna.com

Segment 07 ZU

Software Maya, Mental ray, Fusion, Image Modeler, Photoshop

Description Image based modeling, Texturing, Lighting and shading, Setup render passes, Dynamics and particle simulation including palm tree movement, sand explosion and dust, Layer compositing

Segment 08 Horse

Software Maya, MEL

Description Particles and rigid bodies simulation including horse and chips trail

Segment 09 Trex

Software Maya, Boujou, PFTrack, Fusion

Description Dynamics simulation including crushing and foot prints, Camera tracking, Marker-less object tracking, Rig removal and Compositing

Segment 10 Tornado: Case study - Plug in Development

Software C++, Maya API, MEL

[**dnaTransferFluidAttributes**]: Transfers particle's scalar and vector including customPP attributes to fluid containers within Add, Replace and Multiple modes. Interpolates data using linstep, smoothstep and hermite functions.

[**dnaTransformInterp**]: Interpolates transformation matrix between arrays of matrices

Description [**dnaRampInfo**]: Maps array of attributes using curves

In this example **dnaTransferFluidAttributes** was used to transfer particles' attributes from tornado rig to fluid container; **dnaTransformInterp** was used to interpolate surface between main controllers (Stretchy Setup) and **dnaRampInfo** was used to add extra details to the surface of tornado.

Segment 11 Gillette

Software Reelflow, Python

Description Water and splash simulation, Python scripting to generate droplets

Segment 12 Open Water

Software Maya, Mental ray

Description Look development and Shader assembling

MORTEZA RAMEZANALI

3D | LIGHTING | EFFECTS SHOWREEL BREAKDOWN

+971 50 745 7405
moto @ cgdna.com
www.cgdna.com

Segment 13 Pinky

Software Maya, MEL, Fusion, Photoshop

Description Look development, Set lighting, Fur and Hair dynamics and grooming, Texturing, Layer compositing and finishing

Segment 14 Qui Rit (Concept)

Software Maya, PFTrack, Fusion, Photoshop

Description Camera tracking, Lighting, Shader assembling, Vegetation replacement including setting up grass interaction

Segment 15 The Silver Goblet

Software Maya, Mental ray, Mudbox, Photoshop,

Description Modeling, Lighting, Shading, Texturing and Rendering

Segment 16 Mathematic and Utility nodes for Maya - *Plug in Development*

Software C++, Maya API, MEL, Boost

It is composed of a set of mathematics and utility nodes (over 60 functions) for Maya to control shading networks, Key frame and procedural animation, character or vehicles rigs without writing expression. This will result in a faster evaluation of DG network; Also supports Maya's native renderer and Mental ray.

- Description
- **[Angular]**: Computes value to radian/ degree/ minutes/ seconds
 - **[AnimcurveSpeed]**: Computes instant and constant speed of animcurve based on given input time
 - **[Cardinal]**: Computes Cardinal and Hyperbolic Cardinal
 - **[ConditionRange]**: Tests the relation between input value and specified range (Inclusive and exclusive)
 - **[Constant]**: Pre-defined constant including pi, e, golden ration, etc.
 - **[Curvefunc]**: Smooth, incrementing transition between values (Linstep, Smoothstep, Polynomial hermite)
 - **[Exponential]**: Exponential functions including exp, log, log10, cubed root, square root, etc.
 - **[Factorial]**: Computes rising and falling factorial of base input
 - **[Limit]**: Imposes limit on values including abs, ceil, floor, sign, trunc, fraction, round
 - **[LimitAnimCurve]**: Returns min or max value of animcurve keyframes
 - **[LimitMinMax]**: Finds minimum and maximum of value(s)
 - **[Parabolic]**: Solves general form of 2nd degree Polynomial equation including root, delta, vertex
 - **[TLGamma]**: Computes true, log and log derivative gamma of input base
 - **[Trigonometric]**: Computes trigonometric value(s) of input, including sine, sine hyperbolic, arc sine, cosine, cosine hyperbolic, arc cosine, tangent, tangent hyperbolic, arc tangent and hypot.