

-- PLANNING, STORYBOARD, SHOOT --

The feature acting was written in Final draft and then imported into Stiller Studios SQL database. Locations, scenes and shots were identified. Madcrew 3D team started creating a city plan map of 'Valleby' where the movie was being played out. Henrik Norin (HDR) as a subcontractor, designed and implemented the SQL database and Python scripting together with Olle Westbergh and Tomas Tjernberg from Stiller Studios.

STORY -- 3D MODELLING --

FILM, LOCATION, SCENES, SHOTS in SQL database. 3D MODELLING: Asset based Maya modelling @ Madcrew- each location is a ground level with referenced assets such as buildings and lamp posts. Each asset can further reference assets, geometry in Alembic using Arnold as the primary renderer.

PREVIZ: A full 3D representation of the Valleby city were exported to Stiller's Motionbuilder previz tool where the photographer could place a 3D camera, either static or moving, for each shot imported from draft. The resulting FBX cameras were then imported back to Stillers server and used as base for Mark Roberts FLAIR move programming

MOTION CONTROL PROGRAMMING: CAMERA FBX POSITIONS, LENS (FOV), CHARACTER POSITIONS, NOTES. Using IKTRIX in Maya, Tomas Tjernberg fitted the green screen studio for every shot @ locations and a Cyclops motion control move were programmed satisfying the previz camera FBX and character positions.

SHOOT: The film now has the following structure: film/location/scene/shot/elements. Additional element shoot, the foreground Flair job is loaded back into FLAIR and the Cyclops motion control move is repeated - exact camera pan & tilt by the photographer applied. Could be done long after foreground is shot, for example when a crowd should be added.

OFFLINE RENDER: The raw FBX exported from Flair is automatically baked into a Maya/Nuke compatible FBX in world coordinate space. A DNxHD foreground Quicktime was automatically rendered after ingest and the RUSH render farm renders a matching background proxy for each take using basic Maya software renderer.

OFFLINE RENDER: A Nuke render farm applies a IBK standardized key to the foreground and outputs DNxHD precomposed daily media with corresponding ALE ready for import into AVID. The entire movie is now shot and tons of source material are now transferred to SF and Madcrew facilities using good old disk by courier transfer.

-- OFFLINE EDIT --

EDIT: Editors Mattias Morheden/Elin Hallberg made a rough initial edit using the precomposed DNxHD media output from studio. This edit then evolved into the final edit used for initial production.

3D SCENE GEN: Using the offline edit, the 3D artists @ Madcrew ran through the FBX-es exported from studio and Maya assemblies were automatically generated depending on which asset bounding boxes are visible in camera frustum. The refining of these assets and locations were now initiated and final shading and animation process was taken on.

-- EDL (RE-) IMPORT --

When the shoot were finished at Stiller, the production crew at SF together with Madcrew launched the VFX production. Shotgun was used as the project tracking software and Maya + Nuke were the base apps for 3D and compositing.

Fredrik Pihl @ SF initiated a different approach for structuring the shots and their Nuke projects, instead of having each shot in Shotgun present a clip in the EDL, each camera (take/shot) were used as shots in Shotgun with one Nuke project per camera. This had some new implications on the production: - Each camera could be used several times within the movie, reducing the amount of total shots by almost 66%. - A list of frame ranges is active per EDL and a custom Nuke plugin aided the artist when switching between them. - In those cases two cuts at the same position within a camera not could be composited using the same Nuke script, a new camera (shot) were created with a slightly altered name. Other custom camera/shots was also created covering transitions etcetera. - Overlapping ranges have to be handled with care, taking scene/TOD into account. - A script have to collect the frames and write out a continuous sequence to be used in the grade.

The biggest advantage with this approach is that edit can be refined meanwhile VFX production is going on - when a new EDL is exported no Maya or Nuke scripts needs to be renamed/moved/deleted from their original structure. The only thing that happens is mostly a slight change in frame ranges, a list which is sent out to the responsible 3D and 2D artists using Shotgun.

This entirely camera based workflow had to rely on a solid EDL import script that also could re-import a new EDL merging it with existing Shotgun data.

Selected cameras are identified and metadata were extracted from Stiller SQL database into Shotgun. FBX data, surveillance footage, script images located at Stiller servers were transferred using RSYNC+SSH.

The clips within EDL are identified and stored for each camera as a frame range.

Missing foreground/element frames are submitted to the Otis render farm.

METADATA IMPORT

As the camera started rolling prior to Cyclops move started, the most important metadata was the timecode of move start, this was also the timecode when FBX kicked in and when offline media started. Other important metadata included lens, F-stop, studio position, type of camera move.

EDL IMPORT

Henrik Norin (HDR) as a subcontractor, designed and implemented the VFX pipeline @ SF including the EDL importer script. The pipeline, named 'meerkat', was written in 100% Python and designed to run also at Stiller studios during the import stage and in Nuke during the composite stage.

CONFORM

Fredrik Brännbacka @ Madcrew designed a dynamic Nuke script generator that created a Nuke conform script in runtime, besides outputting HD dpx foreground sequences, it also created a lowres foreground sequence to be used within Maya as a videoplane when matching projections and animations. By writing back timecode and other important metadata to source foreground plates, a lot of work were saved later on when constructing the online material.

-- 3D ANIM/RENDER --

3D RENDER

A custom Maya plug-in written by Fredrik Brännbacka @ Madcrew, imported the frame ranges and, in render time, automatically built a render scene and throws it @ Arnold. By extensive use of EXR metadata, the background plates could be related back to its 3D source.

In order to save render time for the 90% non-movement pan&tilt shots, Fredrik Brännbacka constructed an Pan & Tilt projection setup in Arnold outputting a large exr covering the camera frustum. This set of EXR:s were then used in Nuke within a custom projection setup driven by the FBX from shoot.

Madcrew utilized Otis render farm, a job dispatcher written in 100% Python by Max Persson(Looplab). Otis is 100% web based and very scriptable which suited this type of pipeline very well.

Each shot now had the conformed noise reduced foreground plates and background EXR:s it needed in order to be composited in Nuke. Now it's all put together.

-- VFX --

For 2D comp artists to quickly get started and switch between the camera frame ranges, a custom Nuke plug-in had to be constructed. The plug-in talked to Shotgun and collected all information it needed in order to build Nuke scripts - create animation/projection setups, find conformed plates, etc.

Simon Björk @ SF, made a script template with node placeholders that then was used as a blueprint for newly generated scripts. This, together with effective use of Gizmos, allowed for flexible testing without needing to alter the Python pipeline code too frequently.

By utilizing multi-channel EXR, and custom read nodes, the background passes could easily be extracted and used in the composition. Aside from Simon's own gizmos and tools, the comp pipeline relied heavily on the Neatvideo denoise plug-in.

Fredrik Pihl @ SF utilized scrum to achieve the most effective output, all 400+ cameras were given 45min each at most to reach next version/publish. This way, no shot were eating up precious artist time. Instead, difficult compositions were pushed towards the end and the most experienced compositors.s

NUKE SCRIPT GEN

Nuke scripts are generated from the template. Additional foreground elements are added.

COMP

A major + minor version approach were used, each minor version were a save/test render and at publish, the major version was stepped up and Nuke script were closed.

Nuke render @ Otis render farm, 24 high performance dual Xeon 8-core blades were crunching both 3D and 2D frames 24/7.

REVIEW

As a post process for each render job, the DPX sequence(s) were transcoded to 960x540 H264 movies with metadata overlay tracking back to which elements used and versions.

Torbjörn Olsson @ SF were primarily reviewing new shots on traveling foot from Shotgun screening room. The effective notes system within Shotgun made clear which changes to be made and by whom.

As the movie got approved background and comp renders, a complete H264 were concatenated using Henrik and Fredrik scripts into a full length movie which were uploaded onto a cloud web server for easy review.

-- ONLINE --

ONLINE EXPORT

Fredrik Brännbacka @ Madcrew constructed a script that compiled and uploaded the approved final DPX sequences to the grading suite for online process.

