The Role of Technical Directors on a 3D Animation Team

Michael D. Hickman

Viewpoint and Professional Insight

OPEN ACCESS

Keywords
3-Dimensional (3D) animation production, 3D technical directors, animation work teams, automation, custom pipeline tools, efficiency, graphics technology, medical animation

Introduction

Over the past decade, there have been significant improvements in the quality and accuracy of medical three-dimensional (3D) animation content, through methods such as incorporating scanned data from patient imaging, working with protein structure data, and utilizing physical-based shading. However, now that the expectation for precision has been well-established in our field, we are transitioning to a new era that is centered around productivity and optimization. When it comes to working in teams, a broad portion of the medical animation field has been slow to address pipeline efficiency, particularly compared with other animation fields. Automation techniques, specialized job positions, and production practices designed to mitigate repetitive processes are currently underutilized. Solving these issues requires a different strategy for how we approach and use technology.

Medical animation teams must balance project commitments and actively follow the latest medical and scientific advances while keeping pace with rapid technological developments. Many professionals in this field have expressed the difficulty of implementing new technologies and workflows. According to the Pricing and Business Practices Reports conducted by the Association of Medical Illustrators (AMI), as far back as 2014, "learning new trends and technology" has consistently been rated as one of the greatest difficulties in the medical illustration field.(Association of Medical Illustrators, 2014) It is demanding to maintain expertise in multiple subjects that continuously evolve. We risk spreading ourselves too thin.

Medical animators often turn to communities like the AMI (Association of Medical Illustrators) or the Scientific/Medical C4D Google Group (Google) to help solve difficult technical problems. In addition, the range of molecular tools released to the public has enabled an entire subset of the field to effectively produce accurate molecular graphics and animations. However, sometimes help from afar can only do so much. There is still a need for direct expertise to contribute to a team's production and solve technical challenges specific to each environment.

As new technologies such as artificial intelligence (AI) continue to emerge, clients may soon expect such advances to lead to increased flexibility and shorter turnaround times. Typically, more drafts or art styles mean additional labor for medical animators. Although many benefits can be gained by using programming to automate the manual work, some aspects of 3D production must first be reduced to core data structures and processes before they can be fully automated. A few of these bottlenecks are addressed in this article. Fusing the latest workflows and technologies into production practices will allow the medical animation field to remain nimble enough to meet this new challenge.

One way that medical animation teams can utilize current technologies and use them to their best advantage is to have a team member who is solely dedicated to the technological aspects of animation production. In the same way that art directors unify the art styles of a large production into a focused vision, technical directors (TDs) unify the production team's mechanics and technology into a cohesive, streamlined workflow. These new capabilities and automation opportunities result in reduced repetitive overhead, lower costs, higher productivity, and greater overall team proficiency.

Relatively few TDs are employed in the medical animation field, and the ratio of TDs to animation staff is significantly lower than in other 3D animation industries. Most medical animators that I have spoken with over the last few years are not aware of how critical the TD role is to 3D productions in other fields. Historically, they have handled all technical aspects themselves. To maximize the efficiency of a medical animation team, our field needs to partner with qualified technology experts with backgrounds in computer science or technology to help streamline our productions.

My goal in this article is to present 3D animation production in a different light and to clarify how a TD can increase the output of an animation team, so that we can benefit our clients and scientific advancement. Medical animation production usually focuses on art, science, and storytelling. However, because automated animation pipelines use a different approach, I invite readers to envision the 3D pipeline through the lens of data, processes, and architecture. Vehicle traffic is safer and more efficient when there is a system of traffic lights and rules to manage it. A similar result can be achieved for an animation team's processes, but the rules and technology must first be designed and applied.

What is a Technical Director?

TDs (sometimes referred to as technical artists in the gaming field) are responsible for overseeing a 3D animation group's long-term technology direction.(Richter) They develop team-based production pipelines, troubleshoot technical problems, and work to permanently prevent the problems a team encounters from reoccurring. Technical directing is both a support role and a proactive leadership role that actively seeks out, evaluates, and implements the latest technology into the team's workflow. When explaining the role of a TD, I sometimes use the analogy of an airplane; if a medical animator is the pilot, a TD is the plane's mechanic. (**Figure 1**)

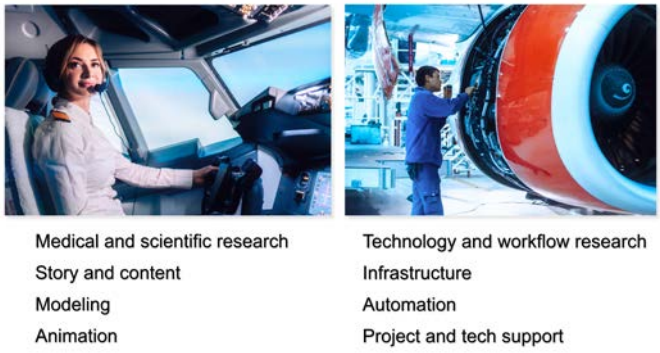


Figure 1. Comparison of the roles of a medical animator (left) with the roles of a 3D technical director (right). If the medical animator is analogous to a pilot, the technical director could be thought of as the plane's mechanic. Used with permission from mrkootv-stock.adobe.com (left) and Gorondenkoff-stock.adobe.com (right).

Medical animators have a range of specialties based on their experience and interests. The same is true of TDs. TD roles encompass an array of specializations:

- General TDs
- Pipeline TDs
- Effects TDs
- Rigging TDs
- Animation TDs
- Compositing TDs
- Rendering and shading TDs
- Modeling TDs

TD work may involve programming custom tools to automate tasks or creating reusable assets, such as procedural effects or textures, that are versatile across multiple projects. (**Figure 2**) These solutions empower team members to be more productive and consistent while also increasing the quality of a team's work.

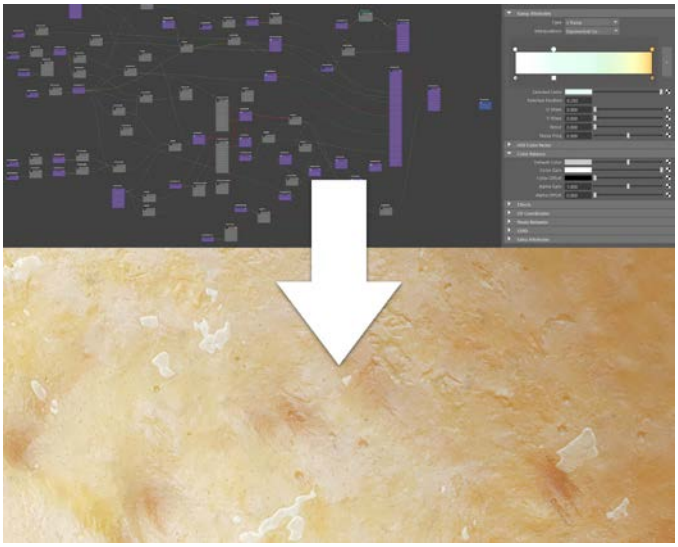


Figure 2. A procedural asset is a series of simple parameters that connect to each other to create complex effects. This example shows a procedural bone texture node network that uses simple mathematical computer algorithms to define the visual properties of bone at close distances. The result is similar to that achieved with vector graphics, where the benefit is that the texture will not pixelate when viewed up close. The resulting asset can potentially apply to multiple models without requiring specific texture coordinates for each model. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

Medical animation production is often conducted in the same way that medical illustration is, with each person working on their own projects and independently solving challenges. Although this method can be effective, it usually does not leverage technology efficiently for the whole 3D animation team. Each person may run into similar technical roadblocks and expend effort to overcome obstacles that others on their team have already encountered. Staff members may not know which version of an asset is the latest, be aware of its location, or have time to save asset updates to the group's common folder. The skills of a TD enable members of an animation team to work a 40-hour week more consistently, while ensuring each member can meet their productivity requirements.

Instead of utilizing art and labor to further a project, TDs contribute technology and programming to production. This effort not only eliminates tasks but also smooths the transition of processes from one stage to the next. Processes become more modular, giving the user the flexibility to reduce the amount of work required when major edits are made. This speed and versatility become especially important as 3D assets accrue and a team's size expands.

Technical blind spots in the early planning stages of a project can later lead to incorrect project estimates and potential overtime. This can delay later projects and may even negatively affect an employee's health or company turnover. While medical animators research the scientific content of a project, TDs can simultaneously scope out the technical requirements and build an efficient solution for the project. When there are fewer unknown factors, more accurate schedules and cost estimations are possible.

The tools and assets developed by a TD can save money on rendering, animation time, rigging, and modeling. For example, software tools such as Molecular Maya,(Digizyme) ePMV,(Johnson et al., 2011; Scripps Research Institute Olson Laboratory) CellPACK,(CellPACK; Johnson et al., 2015) NeuronBuild,(Woolridge) and MolecularNodes(Johnston) can streamline the production of molecular content via technical means. Without tools like these, a content creator might need to model molecules by hand or manually enter x, y, and z coordinates for hundreds of atoms. Because of these tools, ambitious molecular projects are possible that previously may not have been feasible under a deadline.

TDs may also manage the available technical resources, such as render farm capacity and render job priority, and estimate the need for the studio's upgrades. Optimization plays a key role in saving time and money. If rendering a team member's file will take a significant amount of the team's render processing budget, a TD can investigate how to optimize the file to reduce the expense of rendering it. By deploying optimizations to the team via scripts, the TD can then implement those optimizations across entire projects with minimum disruption.

For example, in Disney's *Moana*, the typical scene files that contained the detail of individual pebbles and grains of sand for the beach scenes required about 224 GB of memory to load.(Tamstorf & Pritchett; Walt Disney Animation Studios, 2022) That is a large memory footprint and a lot of data to send over a computer network for each frame to render. However, because of the workflows and optimizations implemented by the film's TDs, the memory requirements were brought down to 88 GB of memory by using new caching techniques.(Li) This significantly lowered rendering and hardware costs without compromising the original artistic vision. This streamlining of production time allows animation team members to stay on schedule, focus on their goal, and move on to other animation tasks.

Medical animators may wait for a technology to mature and be integrated with mainstream software before investing considerable time into learning a new workflow based around it. Once commercial products integrate a new technology and release documentation for it, tutorials are typically created. TDs are often the early adopters who create the first waves of tutorials because they and their teams have been using the technology in production for some time (often under a nondisclosure agreement). TDs frequently beta test new software, submit bug reports, formally request features, serve on technology initiative boards, and engage in technical conversations with developers. This proactive approach helps their team keep their edge in a world of fast-paced technology. When a new software reaches the public market, not only is the team already familiar with using it, but the TD may have custom tools and workflows already built around it for their team.

TDs can stay abreast of technical developments by attending or contributing to industry conferences such as the Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH) or the Game Developers Conference (GDC), two annual conferences on computer graphics that are often the first places these new workflows and technology are shown to the public. At these conferences, TDs can interact with developers and other technology professionals from the moment that the developers are allowed to talk openly about the technology and answer questions. Open-source technologies, such as the Academy Color Encoding System (ACES), Universal Scene Description (USD), and other pipeline initiatives, are often first publicly organized at these conferences.

When a new open-source technology is made available to the public, it is typically released in the form of libraries, raw code, or small command-line utilities. Because coding is usually the primary means for automation, a TD may be able to work with a technology's core functionality before mainstream software can offer official support or create a graphical user interface for it.

New production methods are regularly proven to be effective in the film and gaming industries, but often go unadopted outside of those studios or industries until years later. Such delays can be significantly reduced when there is a TD on staff to locally adopt a new workflow for their team. Instead of merely following the 3D industry, a team with a TD could be involved in driving it.

How Technical Directors Streamline Production

Dividing a project among multiple animators may seem straightforward initially but can be challenging when it comes to revisions, edits, and sharing assets in a team environment. Teamwork and flexibility will inevitably suffer without a clear direction in how technology is used. If there is too much fracturing in a group's pipeline, such as the use of different software or production practices, the team members will find it challenging to work with each other's files. Over time, this tendency can further isolate team members from one another. Conversely, a team's ability to work together in parallel and seamlessly pick up each other's tasks without interference allows for improved flexibility and makes shorter deadlines possible. To achieve this, TDs strive to break apart conventional dependencies that slow production schedules. (**Figure 3**)

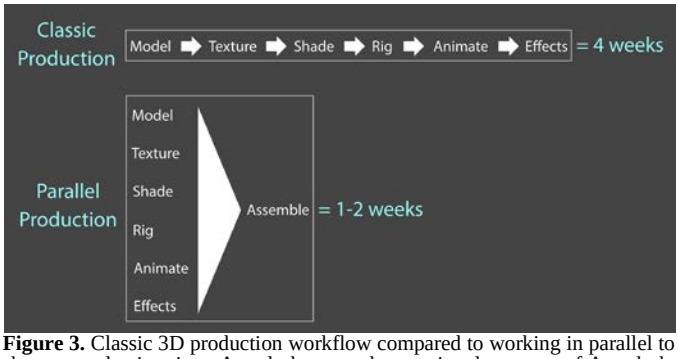


Figure 3. Classic 3D production workflow compared to working in parallel to shorten production time. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

In a classic 3D workflow, a sequence of tasks is performed in which each step is dependent on the previous step. The notion that models must be rigged before they can be animated is an example of this type of classic workflow. A problem with the classic workflow is it segments tasks based on the staff, not the process. If one person completes their task ahead of schedule, it becomes challenging for them to assist another person when the files are separate or when someone employs an unfamiliar workflow. Dividing work tasks between individual workflows means that, if a client has an edit, redoing the work may require a specific team member. Without a way to transfer or reuse previous work, it may fall on that team member to work extra hours if the workflow does not allow for other team members to help.

TDs who specialize in rigging are responsible for more than rigging individual models; they may also write autorigging tools to simplify rigging tasks to only a few mouse clicks. TDs frequently write user interfaces for toolsets that work in concert with rigs to control the 3D models. The advantage of combining a custom toolset with a rig is that the toolset's controls can remain constant to the user and adapt to the different assets or versions of rigs. This abstraction allows proxy models to be rigged and animated before the final model or rig is completed. With this workflow, newer versions of assets can replace older ones with the click of a button to rewire the previous motion data to the newer asset. If a client required an edit to the base model or rig, the change could be trivial to perform. However, when working in traditional production stages, such an edit would necessitate that the user manually redo the modeling, rigging, and animation work due to the dependency-based nature of the workflow.

It is common for 3D content creators to evaluate software based on how user-friendly it is to access functionality, rather than on technical or programmable merits. For example, certain render engines are popular among individual 3D artists but are rarely used in the film or television industries because they lack deep scripting capabilities or have other technical limitations. Fortunately, user-friendliness can directly be addressed by a TD to bridge that gap.

Consistency is one of the prime objectives of a TD. Without consistency, it is difficult for automation to take place. Incorrectly naming or renaming files and folders can cause problems in a team environment, especially when multiple individuals are collaborating on the same assets. In other animation industries, a TD has a strong influence on the naming and organization of files and folders.

Another limitation that affects production is how quickly the 3D animation field changes. Most aspects of 3D production have a 5-year shelf life before some portion of a technique or workflow becomes obsolete. Sometimes older files will no longer load with newer software. (**Figure 4**) Users may have a window of only a few years to convert older files to a newer format before support for the older files is cut off. It is invaluable for a TD to proactively keep a team's files compatible via batch scripting to prevent unplanned detours during production.

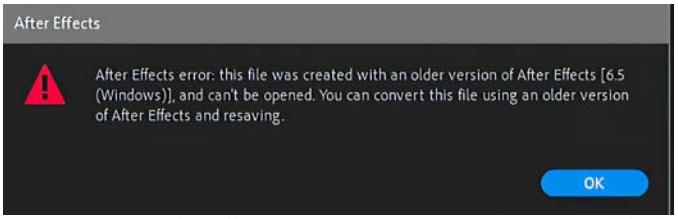


Figure 4. Adobe After Effects error message displayed when an older file is opened in a newer, incompatible version of After Effects. Adobe product screenshot reprinted with permission from Adobe AfterEffects software.

Throughout my time working in the medical animation field, I have heard that students are told to expect that 3D animation jobs will require working longer hours than 2D animation or app development jobs. Animation job postings sometimes will even state that they frequently require the animator to work 50 hours or more per week. Due to large scene files and heavy processing needs, 3D animation production involves delayed feedback, which is not the case for other visual mediums. You bake the cookies and then must wait to see how they turn out. Sometimes they do not fully bake; other times, they catch fire. Regardless, implementing a new plan requires additional time. However, there are numerous ways in which a TD can help to mitigate those delays in feedback without requiring longer hours from the animation staff.

It is demoralizing for an animator to work overtime to ensure a file renders overnight, only to discover the next day that the frames rendered incorrectly. It is especially frustrating when it occurs because of something simple, like forgetting to set a checkbox or selecting the wrong resolution for the final render.

It might be wise to go over a prerender checklist before rendering a project. However, after a long day, when the user is exhausted, they are more likely to miss something—even when using a checklist. Subconsciously, people may want the day to be over with, and may accidently skim past steps. Fortunately, code does not grow tired, cut corners, or betray our best interests after a long day. TDs can help medical animators catch

and correct mistakes before they are aware of the problem, similar to how reflexes help keep our bodies safe from injury.

Production experience and knowledge can be bundled into tools that catch or fix improper settings and optimize scene files before they are sent to render. Before using automation, I occasionally would accidentally render animations set to a preview setting, instead of to the final settings that I intended. My solution was to create a simple warning that was triggered when a file was submitted to the render farm that did not meet a minimum resolution pixel count for production. (**Figure 5**) Other similar automated warnings can help save the production team unnecessary time, expense, and stress.

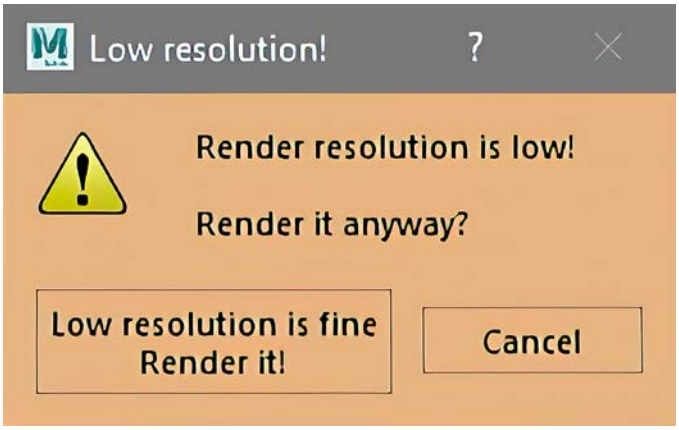


Figure 5. Render warnings. This example shows a customized low-resolution warning message created for Maya. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

A major shift for an animation pipeline is working with granular referenced files instead of large monolithic files. Medical animators are often familiar with Adobe After Effects and how it references external media files. When a media file is modified, the After Effects file reflects that change. This concept can also extend to 3D production and is often considered standard practice in team environments for 3D animation production because of the flexibility it offers.

Monolithic 3D files are limited in that only one person can edit the project at a time, creating a bottleneck in production. These large files take longer to load and save, may consume more system memory, and, due to their size, are not good candidates to save in ASCII plaintext formats for external programmability. When assets and tools are modularly broken down as referenced files, different people can work on the same 3D scenes in parallel during production. The use of referenced files can also prevent accidental deletions of objects. When a user is not aware of an accidental deletion, they may continue to work on a faulty file, leading to redoing work later. With file referencing, if a model revision causes issues, that item can be rolled back, much like an undo button, while the object's animation data and materials remain safe in separate files. Although most 3D software supports this capability, the full processes must be developed by a TD to fully be effective.

Another benefit of dispersing granular files is that an art director can use a custom TD tool to make global changes to large projects. Materials, textures, and lighting can be swapped for an entire project with the click of a button, copying a recipe from one project and applying it to another. This change can even be made without the need to load a 3D file when a lightweight ASCII text file is used as the central layout file that references the external 3D files. By reducing the potential for human error, such a tool may reduce the need for additional test renders.

New technologies like USD are designed around this concept of file granularity and multiple people working on the same scenes at the same time. USD is a new 3D format and framework for 3D assets with numerous capabilities, such as version control, high performance, and universal support. USD is in active development, and it may yet be several years before 3D animation programs support its full capabilities. In the meantime, a TD will need to develop how the team's assets, scripts, and referencing work together as USD continues to evolve.

ACES is a technology that is moderately complex to fully understand and can be difficult to implement across a team environment without a way to centrally enforce coordination between users. Without a TD, an ACES workflow requires that every team member must understand the concept and consistently apply the technology. This is a tall order, even for the most technically competent teams. If a team is not using color spaces correctly, there could be disastrous consequences when team members are working in different color spaces for each file. People will inevitably alter colors and settings, potentially working against each other's efforts and thereby slowing production. A TD, however, could make the process completely invisible to the end user. Scripts can automate many of a team's color management presets upon launching the 3D software or loading a file.

Today's best practice may become tomorrow's bad habit. A TD recognizes when a current practice will become a future liability and helps ensure that the team is on a modern technology trajectory. Just as a skyscraper cannot be built using blueprints intended for a house, a team-oriented foundation is required when the team's goal is to reach the sky.

Examples of Custom Tools

In this section, I describe some real-world examples of tools that I have created to increase our animation team's productivity. After a 3D application is launched, most of the following 3D tools dock on the left side of the interface into different tabs or can tear off to their own window. **Figure 6A** shows the main tools panel collapsed and in its own window, and **Figure 6B** shows the panel expanded while docked. Upon opening a 3D scene file, the tools required for that file are automatically launched.

(A)

(B)

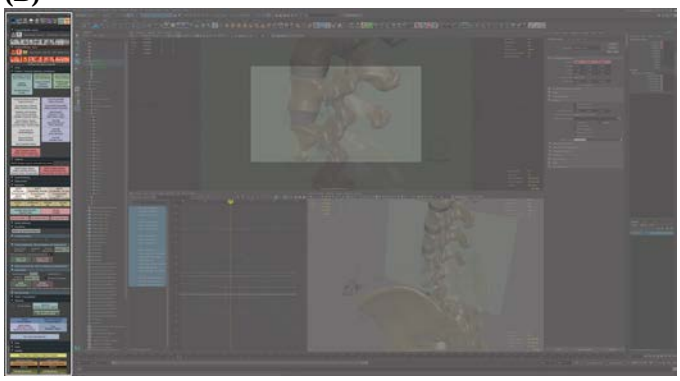


Figure 6. (A) The main toolset panel fully collapsed. (B) View of the Maya workspace, showing the main customized toolset panel expanded and docked on the left. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

To aid usability, tools have detailed tooltips for each item or may include buttons to display visual examples or more in-depth documentation. Most of the tools have a text field at the top that displays important information and warnings as they occur. (**Figure 7**)

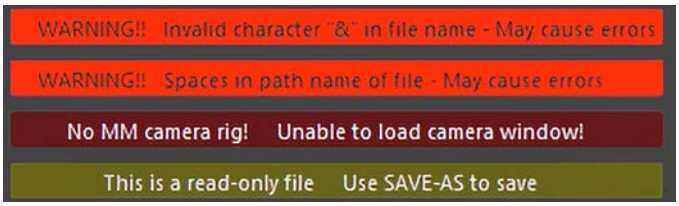


Figure 7. Customized displays of warnings and helpful information appear at the top of each toolset. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

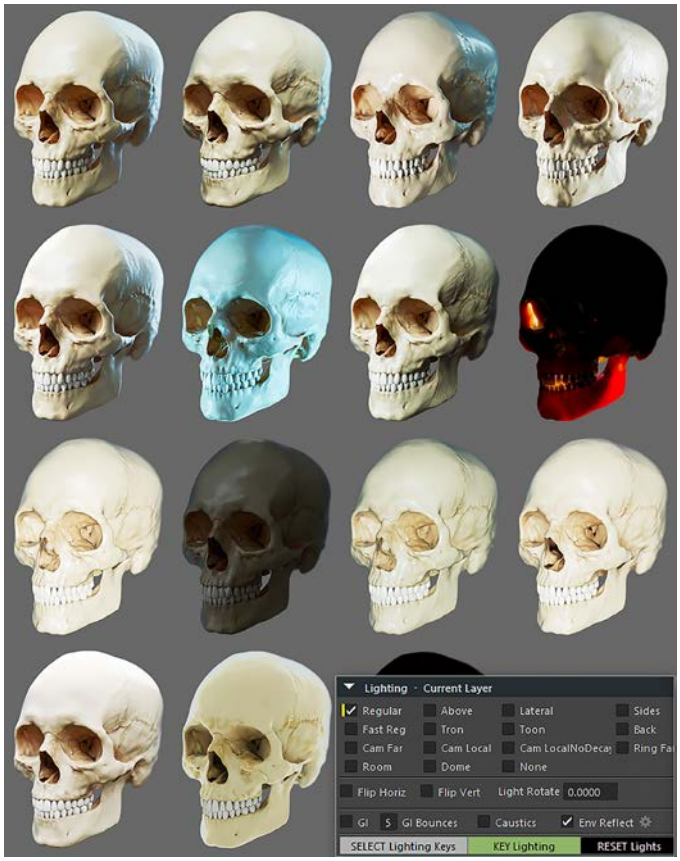


Figure 8. Lighting presets custom interface created for Maya, which the user can select and modify. Inset shows the tool with the currently selected preset (first row, first column). Autodesk screenshot reprinted courtesy of Autodesk, Inc.

Lighting Presets

Although 3D animators can light a scene as they want, it is helpful to have a set of lighting presets with controls. (**Figure 8**) Users may adjust any of the light presets as needed. If a team member creates a new light setup that may be useful for other animations, it can be permanently added as a new preset for the rest of the team to use.

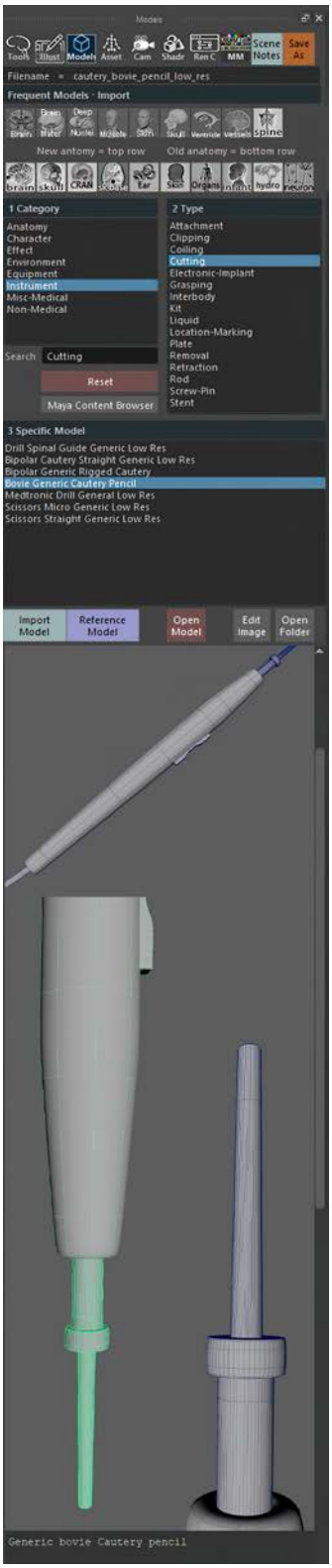
Model Catalog

I wrote a model catalog tool to help inventory our 3D models so that they are searchable and easy to import, reference, or permanently edit. (**Figure 9**) This model catalog tool includes images, renders, photo references, and notes. It is easy for a user to add a model or edit its information or images.

Asset Toolset

To allow the user to easily control 3D assets, I created a custom tool for managing and controlling assets that provides a consistent, user-friendly interface. (**Figure 10**) It lists all related parameters and any relevant information about each asset's controls.

3D assets often evolve over time. An interface that is consistent regardless of the selected asset is important because it eliminates the need to manually investigate how the objects and controls are configured for each asset. The tool simply reads in the list of attributes associated with that version of the asset and provides keyframe and setting controls for each group.



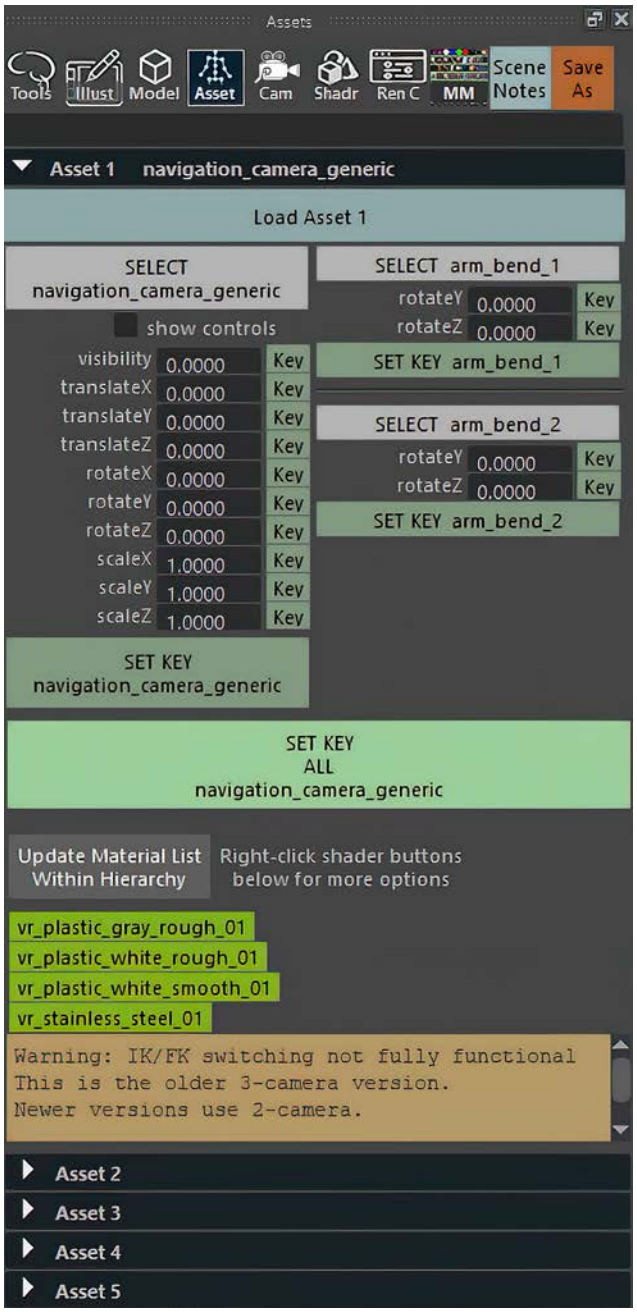


Figure 10. Customized asset toolset to consistently access an object's controls in Maya. It lets the user pin several rigs at once. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

Figure 9. Customized model catalog toolset created for Maya to help inventory 3D models so that they are searchable. It includes viewport screenshots, reference photographs, renders, and notes. Models can quickly be edited, imported, or referenced. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

Shader Management Toolset

Sorting through hundreds of shaders in a software's default material window is exhausting, especially when complex layered materials or rigging relationships are involved. When there are many materials in a scene file, it can be overwhelming for another user to understand which materials are currently in use or how to quickly alter shading assignments to a different set of materials. I wrote this shading assignment tool to manage our ongoing material library, (**Figure 11**) which is organized in groups. When the user is especially in a hurry, there is a button that automatically assigns recommended shaders to all objects by using an internal identification system for objects.

Camera Rig Toolset

Predictable camera motion requires a camera rig with precise control. However, camera rigs can become complicated because they are often composed of multiple nested objects and complex relationships. A camera toolset (**Figure 12**) was created to allow the user to quickly edit the camera as well as noncamera objects. Our toolset allows for camera keyframe data to easily be transferred to other animation files with a button click.

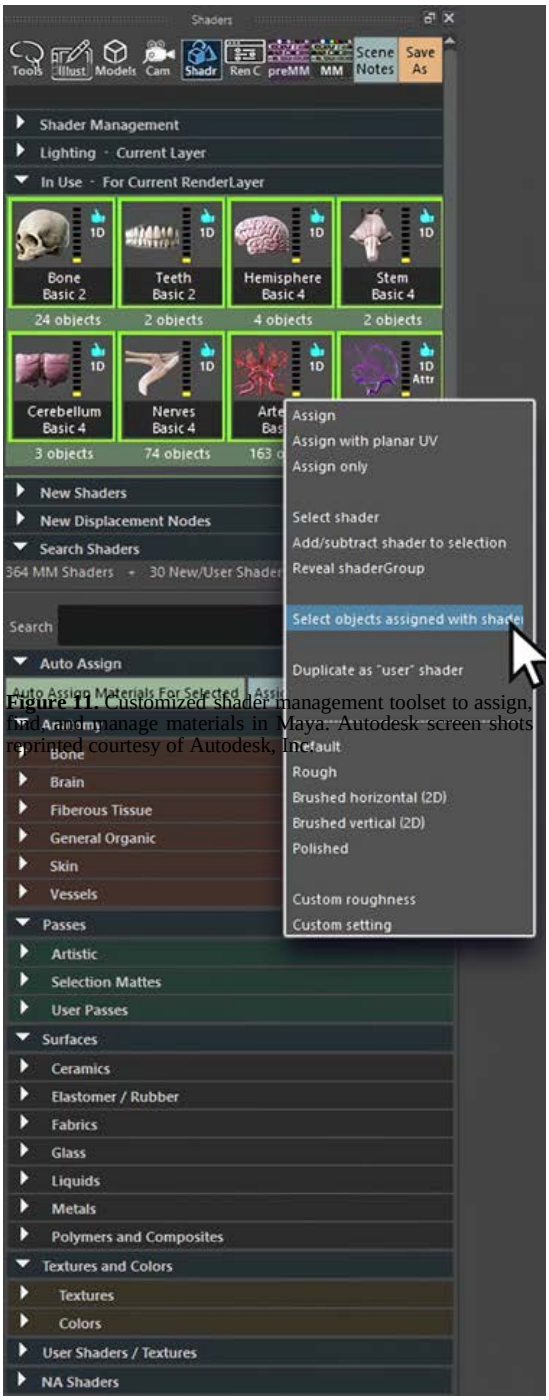


Figure 11. Customized shader management toolset to assign, find, and manage materials in Maya. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

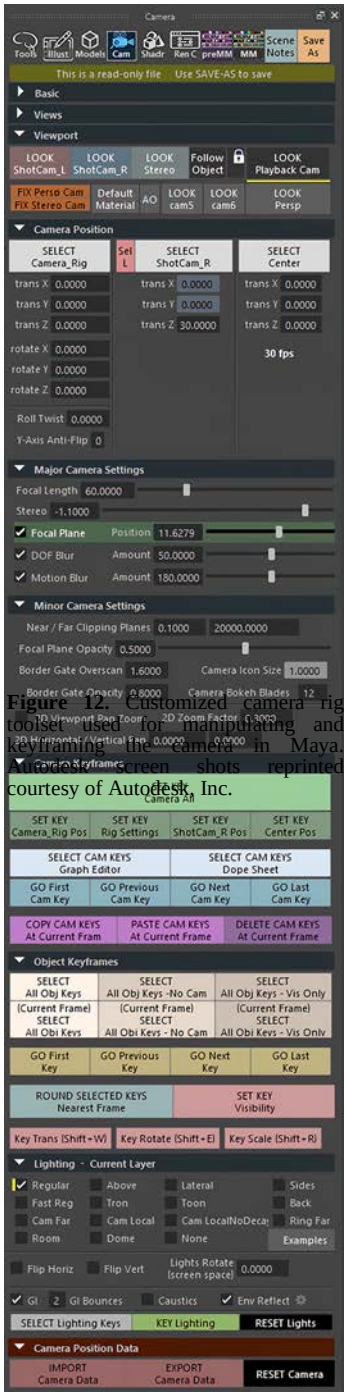


Figure 12. Customized camera rig toolset used for manipulating and keyframing the camera in Maya. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

Render Control Toolset

The render control toolset (**Figure 13**) optimizes rendering and helps prevent errors. Additionally, this toolset provides an overview of all the render information on a single page. The render tool automatically adjusts specific render settings based on detections within the scene file. Upon submitting the scene file to render, if any parameters are outside the specifications of a typical render, the tool will trigger a warning to notify the user before the render proceeds. (**Figure 5**)

Figure 13. Customized render control toolset for managing the file's render setup in Maya. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

Rather than sending the full 3D scene file to the render farm to render, this tool automatically converts each render to an optimized standalone render format and submits each render job to render. This optimization results in render speeds that are multiple times faster. All relevant scene data are captured for each job and listed in the render manager, which is useful when troubleshooting render issues. In addition, this tool autogenerates a starting composite and a postproduction edit file with the team's settings. The initial composite file automatically loads all rendered footage and sets up render passes with appropriate effects commonly used with each pass.

Texture Management Spreadsheet

The texture spreadsheet toolset highlights texture map problems. (**Figure 14**) It can automatically fix multiple texture settings and copy all texture files to the correct folder on the render farm and update the file path.

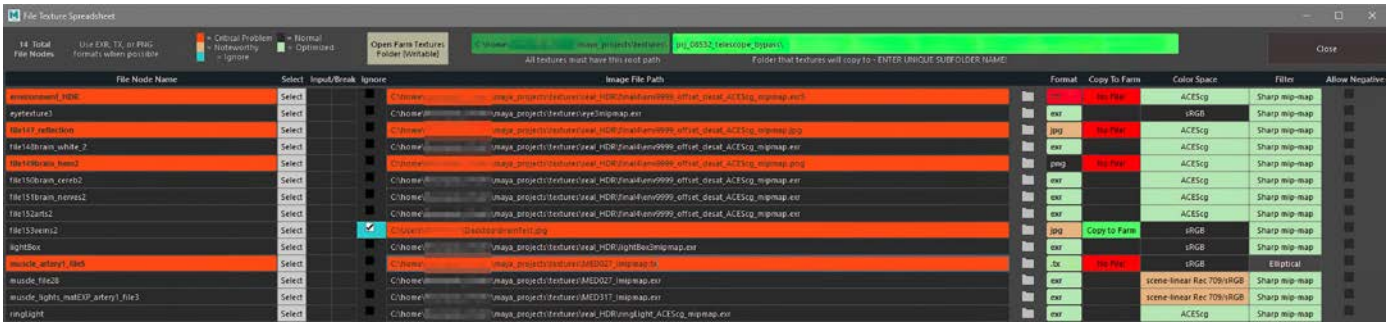


Figure 14. Customized texture management spreadsheet that calls attention to texture map problems and can fix common problems associated with texture map files in Maya. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

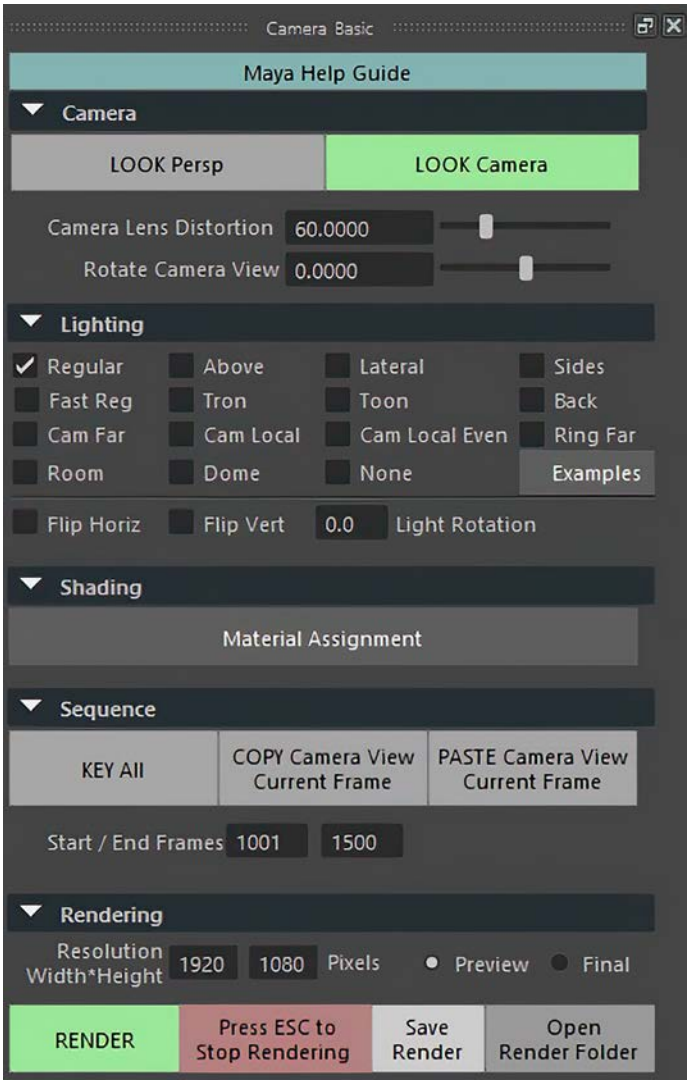


Figure 15. Customized 2D illustration toolset for Maya, allowing 2D illustrators to utilize and manipulate 3D models. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

2D Illustration Toolset

The 2D illustration toolset allows 2D medical illustrators to work with 3D models as reference material for creating 2D illustrations. (**Figure 15**) This toolset consolidates the most relevant tools for manipulating the models or viewport so that creating reference images for 2D illustration purposes is fast and easy. The tool interface remains consistent for the user regardless of when the location of the tools changes with each new version of the software.

Basic 3D Render Toolset

This toolset is designed for non-3D users (**Figure 16**) with no experience using 3D graphics software, so that anyone on a team can create and render 3D visual content with a few minutes of training. This simplified toolset leverages all the models, shading, textures, and lighting that our 3D team has created, ensuring accuracy and quality.

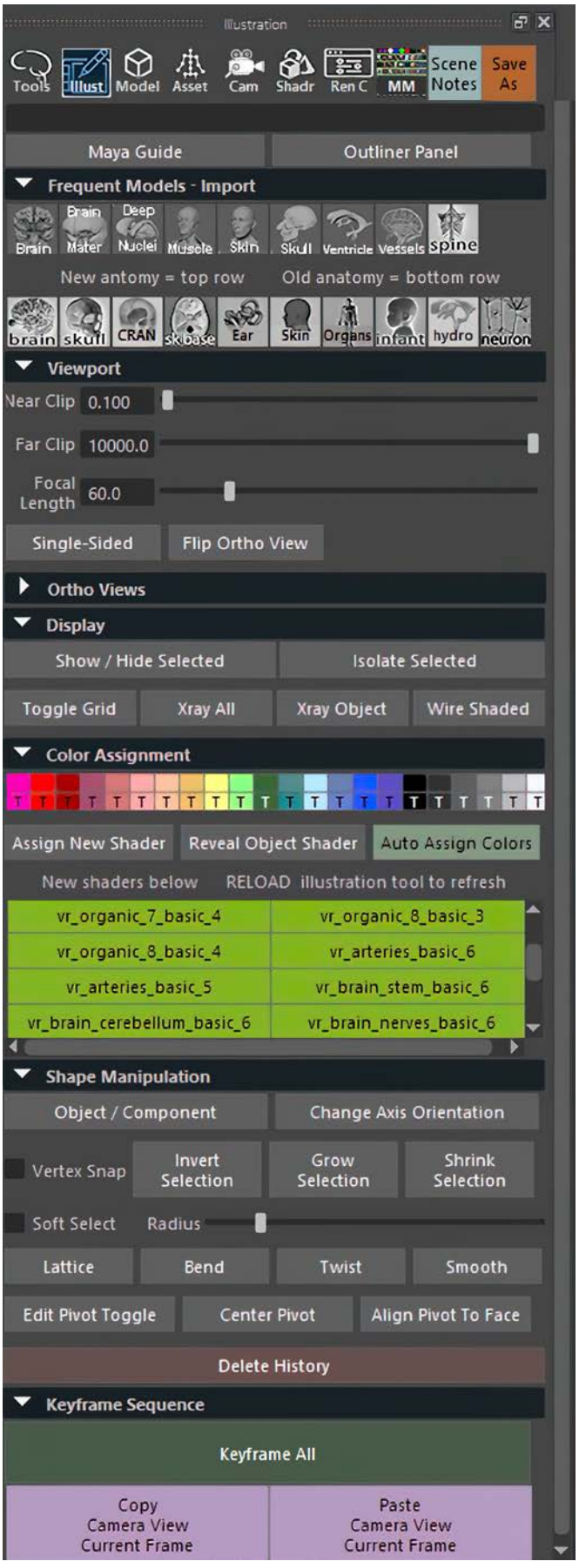


Figure 16. Customized basic 3D render toolset for creating basic 3D renders in Maya. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

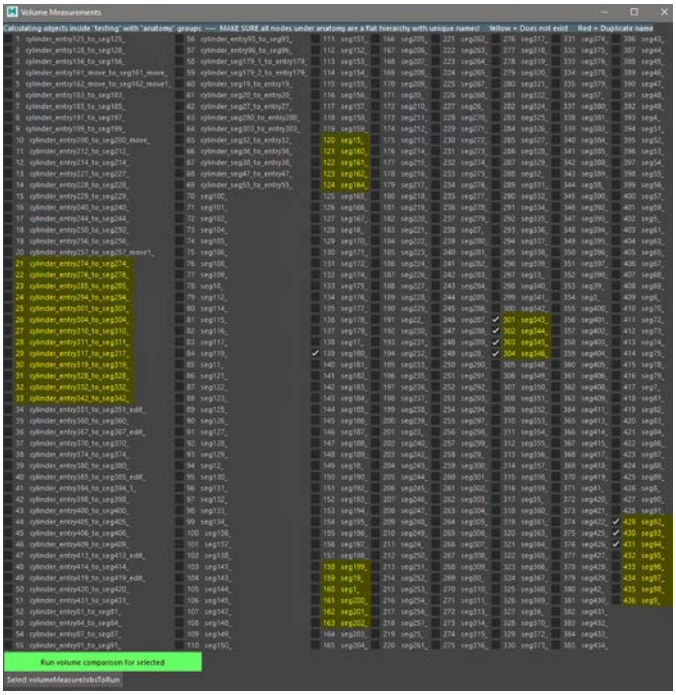


Figure 17. Customized batch calculation tool used to process intersection volumes of 3D models in Maya. Autodesk screenshot reprinted courtesy of Autodesk, Inc.

Real-time Scripts

Scripts can be embedded directly into a 3D scene file that will execute each time that file is opened. In a pipeline designed by a TD, these scripts can detect in real-time when specific parameters change or when particular objects are present. Embedded scripts can also automatically open tools required for an asset or launch external processes to connect data to.

General Batch Processing Tools

As a group's projects accumulate over the years, solutions for manipulating a large amount of files become increasingly important. To address this, utilities can be written by a TD to streamline general tasks; create statistics and reports, such as tracking productivity via file modifications; or ensure proper file and folder naming. Altering thousands of images using Python or third-party utilities like ImageMagick (ImageMagick) is an order of magnitude faster than using larger graphics programs such as Adobe Photoshop (including its Actions batch tools). Additionally, custom tools may include logic or image detection capabilities to further reduce manual inspection that are not necessarily available with larger graphics software.

Scientific projects often deal with mass-scale data. For one such project, I wrote a tool to batch-process 3D volumetric data. (**Figure 17**) Over 237,000 calculations were processed for each data set, requiring many computers to complete the calculations in a short time period. An interface was made to help track which computations had yet to complete (highlighted in yellow). The resulting data were then used to create 3D visualizations and underwent machine learning analysis for publication.

For any situation in which it is not viable to manually process large-scale data, custom batch processing tools created specifically for one task can be expanded on for later projects, raising the team's capability and performance over time.

Evaluating Candidates for a Technical Director Position

Just as a medical background is crucial for producing accurate medical media content, a technology or computer science background is crucial for a TD. Utilizing new AI tools like ChatGPT to write code is useful for simple scripts or refactoring code but not suitable for creating and managing a team's complex production pipeline needs. I recommend hiring or contracting a TD rather than transitioning a medical animator to handle full TD responsibilities.

There are many paths that can lead someone to pursue TD work. 3D production experience is an advantage as it provides an understanding of the artistic thought process involved in producing animations. Strong artistic skills may indicate that a TD will be effective at automating the visual quality of effects. A computer science background may imply that the person is more likely to write clean and scalable code. Information technology skills may mean that the person has experience with large-scale computer administration and infrastructure technologies.

If a TD includes screenshots of graphical interfaces in their portfolio, those interfaces can be evaluated on how intuitive they appear. You can ask the TD candidate how they would improve those tools if given the chance from a performance or coding management perspective. If they do not have a visual portfolio or screenshots showing user interfaces, that is not necessarily a bad sign; many studios that employ TDs may not allow their work to publicly be shown, or the TD may have previously been responsible only for back-end coding work. Regardless, TDs ideally are receptive to user feedback and strive to create effective workflows for production staff.

When examining tools that a TD has written, it can sometimes be challenging to know which programming tasks are difficult to accomplish. More lines of code is not necessarily better; it could easily mean the opposite. Some data structures perform faster than others for different scenarios, whereas others may be easier to understand or edit. The interviewer can always ask a TD to explain how their code works by describing it in small steps, why they chose to organize it the way they did, and how they would improve on it.

Some TDs do not write code from scratch and may only modify existing code. Their focus may be on creating reusable assets or troubleshooting issues for their team. Procedural node network

workflows are often referred to as visual programming and utilize many of the same principles for controlling complex relationships through user-friendly controls.

In the film, video, and games industry, TDs generally take live coding tests during job interviews so their coding skill and thought process can be evaluated. During these tests, they are encouraged to speak aloud what they are thinking, explain what they need to review, or write pseudocode when there is not enough time for specific code syntax details.

Python is one of the main programming languages typically used for TD work. It is commonly supported by numerous 3D graphics and compositing software programs and ships with most Linux distributions.(Python.org) Graphics software programs often have their own scripting language, such as Maya's MEL, 3ds Max's MAXScript, or C4D's COFFEE. Compared to these native scripting languages, Python is a general-purpose programming language with wide universal support and modular libraries to extend its capabilities.(Lutz, 2013) A software's native scripting language may sometimes have specific advantages over Python, such as better performance or special commands. As the future of programming evolves, newer programming languages may replace current ones. Mojo, for instance, aims to eventually replace Python by offering significantly better performance while supporting native Python code syntax.(Yegulalp, 2023).

Writing plug-ins for 3D software is considered to be an advanced skill for a TD, blurring into software developer territory. Plug-ins offer better performance, such as when manipulating millions of polygon vertices to change the shape of a 3D model, and they can even tap into a graphics processing unit (GPU) or neural processing unit (NPU) for higher-performance functions. Plug-ins are usually written in a language such as C or C++, which have finer control over performance and hardware compared to "higher-level" languages, like Python, that utilize more levels of abstraction from the hardware. Scripting is usually preferred over writing plug-ins for tools that do not require maximum performance because scripting has better long-term compatibility, requires less specialty knowledge, and often is faster to initially write.

Evaluating the skills of a TD may be challenging when a team has not previously worked with one, but regardless, a good TD will be able to explain how they would help solve a team's challenges and how they may benefit a group's production process.

Conclusion

The TD role has been a standard part of the visual effects industry since the mid-1990s, but the role is comparatively rare in medical animation contexts. Animation TDs reduce mistakes, streamline production processes, augment capabilities, and improve the consistency of project schedules, which significantly increases long-term productivity. They help support animation teams with the latest workflows and new technologies that are deployed in a user-friendly way for the team to utilize. TDs solve tasks that involve vast quantities of files, ensure performance optimization, and create the data management pipeline that run behind the scenes for a production team.

Most 3D animators understand the value of using a render farm. It is not the computers themselves or the render manager software by itself that matters as much as it is the combination of the two that make it so powerful. Likewise, augmenting an animation team with a TD unites a team's skills and enables the team to become greater than the sum of its individual members. Because they enhance a 3D animation team's output, a TD can help the group avoid excessive hours and burnout, improving work-life balance. TDs help everyone to row in the same direction, while still allowing individuals to influence the needs of the group as they evolve. When implementing an animation pipeline, some compromises may need to be made, but as in any good relationship, the results are worth it.

Paul Krugman—an economist, author, professor, and columnist—once said, "Productivity isn't everything, but in the long run it is almost everything. A country's ability to improve its standard of living over time depends almost entirely on its ability to raise its output per worker."(Krugman, 1997) I imagine that similar metrics determine a medical animation team's ability to function efficiently and define its long-term success.

If a production group is large enough to justify the job roles of an art director or creative director, a TD role should also be considered for addressing technical long-term productivity. The results of having a TD extend beyond animation production. The role is a key foundation for modern teamwork in a technology-driven environment, especially as we enter a new age of combining graphics production with automation.

Additional Information

The TD role is discussed and additional information is provided in online videos and articles by Alexander Richter and Allan McKay.(American Association of Community Theatre; McKay, 2021; Mediamatch, 2024; Richter; Richter, 2018, 2023)

Acknowledgements

Thanks to everyone in the medical animation community who has supported and encouraged me to share these topics. In particular, I would like to thank Eric Small, Gary Welch, Kristen Larson Keil, Danielle VanBrabant, and Amy Hickman for their contributions to this article. I thank the staff of Neuroscience Publications at Barrow Neurological Institute for their assistance with manuscript preparation.

References

American Association of Community Theatre. The Technical Director's Job. Retrieved January 20, 2024 from https://aact.org/technical-director

Association of Medical Illustrators. Association of Medical Illustrators home page. Retrieved January 20, 2024 from https://www.ami.org

Association of Medical Illustrators. (2014). Pricing and Business Practices Report. https://hub.ami.org/business-resources/survey-results

CellPACK. Retrieved November 11, 2022 from http://www.cellpack.org/home

Digizyme. Clarifi Tools: Molecular Maya. Retrieved November 11, 2022 from https://clarafi.com/tools/mmaya/

Google. Scientific/Medical C4D Google Group. Retrieved January 20, 2024 from https://groups.google.com/g/sci-med-c4d

ImageMagick. Retrieved November 8, 2022 from https://imagemagick.org/

Johnson, G. T., Autin, L., Al-Alusi, M., Goodsell, D. S., Sanner, M. F., and Olson, A. J. (2015). cellPACK: a virtual mesoscope to model and visualize structural systems biology. Nat Methods, 12(1), 85-91. https://doi.org/10.1038/nmeth.3204

Johnson, G. T., Autin, L., Goodsell, D. S., Sanner, M. F., and Olson, A. J. (2011). ePMV embeds molecular modeling into professional animation software environments. Structure, 19(3), 293-303. https://doi.org/10.1016/j.str.2010.12.023

Johnston, B. A. Molecular Nodes. Retrieved November 11, 2022 from https://github.com/BradyAJohnston/MolecularNodes

Krugman, P. (1997). The Age of Diminished Expectations. MIT Press.

Li, Y. K. Twitter post, July 4, 2018. https://twitter.com/yiningkarlli/status/1014418038567796738

Lutz, M. (2013). Learning Python (5th ed.). O'Reilly.

McKay, A. (2021). Technical Artist + Technical Director Job for VFX Explained (Allan McKay).

Mediamatch. (2024). Technical Director. Retrieved January 20, 2024 from https://www.media-match.com/usa/jobtypes/technical-director-jobs-402798.php

Python.org. Using Python on Unix platforms. Retrieved January 20, 2024 from https://docs.python.org/3/using/unix.html

Richter, A. The role of a technical director. Retrieved November 11, 2022 from https://www.alexanderrichtertd.com/post/the-role-of-a-technical-director

Richter, A. (2018). Technical Director in Visual Effects, Animation and Games? https://www.youtube.com/watch?v=vpGNlv6j00Y

Richter, A. (2023). From Artist to Technical Director: A Guide to Success in Visual Effects.

Scripps Research Institute Olson Laboratory. ePMV: embedded Python Molecular Viewer. Retrieved November 11, 2022 from http://epmv.scripps.edu/about

Tamstorf, R., and Pritchett, H. The challenges of releasing the Moana Island Scene. Eurographics Symposium on Rendering - Industry Track. Eurographics DL. 2019; https://diglib.eg.org/handle/10.2312/sr20191223.

Walt Disney Animation Studios. (2022). Technology data sets. Retrieved October 28, 2022 from https://www.disneyanimation.com/data-sets/?drawer=/resources/moana-island-scene/

Woolridge, N. NeuronBuild. https://github.com/NWoolridge/NeuronBuild

Yegulalp, S. (2023, June 7, 2023). A first look at the Mojo language. InfoWorld. https://www.infoworld.com/article/3697739/a-first-look-at-the-mojo-language.html

Author

Michael D. Hickman, BA is a Senior Technical 3D Artist at the Barrow Neurological Institute, St. Joseph's Hospital and Medical Center in Phoenix, Arizona. Neuropub@barrowneuro.org

Licensing