

THE ANIMATION FIELD GUIDE

By Michele Bousquet
Presented by Epic Games



The Animation Field Guide v1.2

Contributors

Author: Michele Bousquet

Editor in Chief: Rob Di Figlia

Editors: Colin McLaughlin and Jill Ramsay

Layout and Illustrations: Oliver Morgan and Carys Norfor

Acknowledgments

We would like to thank everyone we interviewed for this guide for generously sharing their time to give us insights into how they use Unreal Engine for real-time animation pipelines, and for their support in helping to make this guide happen. We would also like to thank all of the team at Epic Games for providing additional details about animation workflows with Unreal Engine including: Jeremiah Grant, Kamy Leach, Philippe Rebours, Shaun Comly, Michael Neely, Yafes Sahin, Tony Bowren, Paul Doyle, Shawn Dunn, and Jordan Thistlewood. Special thanks go to Miles Perkins and Karen Dufilho for their careful consideration of the content in this guide to make it as useful as possible, and to Nathalie Bergeron and Annie Belanger for helping us move this guide to the finish line.

Cover credits

Yuki from the *Yuki 7* series, *Yuki 7* is a Chromosphere Production

Andrea from *LEGO Friends*, stills courtesy of WildBrain Studios and LEGO

Blu from *Xanadu*, image courtesy of Cory Strassburger

Allahyar from *Allahyar and the 100 Flowers of God*, image courtesy of 3rd World Studios

Boy and Father from *Wylder*, image courtesy of Engine House

Chase and CATCH from *Chase and CATCH*, image courtesy of Digital Dimension Entertainment Group

Michael, Minnie, and Patty from *Mall Stories*, *Mall Stories* is a Chromosphere / Kikutowne Production

Thunder and Shiny from *Super Giant Robot Brothers!*, *Super Giant Robot Brothers* | Courtesy of Netflix

Igglepiggle and The Zonk from *In the Night Garden*, stills courtesy of WildBrain Studios

Peely from *Fortnite*

Guff from *Fortnite*



Foreword

Some of you may be asking yourself of late, “What’s all this ruckus I hear about real-time animation? Disruptive game changer, you say? The future of collaborative visual development?”

These questions are nothing new in many creative circles. But we haven’t seen much written about the inevitable convergence of the enormous worlds of game engine technology and traditionally produced 2D and 3D animated entertainment in film and TV.

The old phrase “Never the twain shall meet” suddenly comes to mind.

Well, that’s all changing. Quickly. Those “twains” are now both barreling side-by-side down the same “twack.” What was slowly seeping into mainstream entertainment production discourse with an innovative previs effort here, and a feature film made completely in a warehouse there, is now a veritable flash flood.

The COVID-19 pandemic forced studios to figure out remote and cloud-based production workflows, triggering a re-evaluation of their linear pipelines. Suddenly, alternative tools and workflows that used to be considered “too disruptive” to their tried-and-true practices were on the table, with real-time technology front and center.

One thing, however, is for certain—acceptance, integration, and deployment feel like an imperative rather than a dabble.

Faced with a not-nearly-large-enough (and no longer nomadic) talent pool, studios are embracing all manner of production methodologies to handle an unprecedented explosion of animation production, fueled by insatiable streamers demanding a high volume of features, episodics, and shorts with high-quality visuals produced in record time at low cost for every imaginable demographic, genre, language, and culture.

It’s exhausting just to say it, let alone produce it!

For every tentpole show that can afford the luxury of five to six years of development and production, there are hundreds that are being conceived, developed, and put into production much quicker than ever. Fast pre-production, with smaller, remote teams, by necessity drifts towards any toolset that enables quicker development. Previs—used for pitching complex scenes, choreographing stunts, and calculating complex camera placements, in addition to story development—suddenly becomes an incredibly important tool for directors, who can now sit with a small team and visualize an entire show, scene by scene, in real time, with high-resolution imagery, frame composition, and lighting.

Real-time animation workflows are facilitating creative control in unprecedented ways. You can prototype your entire film before needing to hire a caterer!

None of this has happened by accident, or in a vacuum. Real-time, tech-driven production using game engines has fueled all sorts of innovation for years at many well-known studios like ILM and Wētā FX, such as integration of on-set performance capture, or LED wall stages integrating in-camera VFX.



But if you've been paying any attention at all, you'd have seen that Epic Games and Unreal Engine, through initiatives like the Epic MegaGrants program, have been supporting and celebrating the work of thousands of individual artists and small studio teams for years. Epic has been pushing game engine technology and real-time animation pipelines into every conceivable area of content production, not just in entertainment. It's almost like a secret happening in plain sight.

Why is this all important? It's important because artists will, by nature, embrace any tool or means of production that gives them even the slightest edge. Real-time animation tools offer creative teams the luxury of making all manner of changes with immediate effect. No waiting hours, days, or even weeks for rendered results. Final or close-to-final visuals are literally right at hand.

Argue as you will that the traditional, linear production pipeline is still king (and it is), but you can no longer ignore the fact that real-time animation is fast becoming an increasingly important—even imperative—workflow in any serious creative pipeline.

That "twain" sure has kept a-rollin' all night long!

In over 20 years of interviewing hundreds of top VFX supervisors, animation supervisors, CG supervisors, and the like, and asking the question, "Do you worry that production technology exerts too much control over the creative process?" I've always received the same answer: "Production technology should always support the director's vision."

Animation and visual effects productions are incredibly difficult, demanding, and exacting. Complex visuals require tremendous amounts of planning, discipline, talent, and resources to produce. There is never enough time, people, or resources. Ever.

Game engines give teams immediate access to content filled with high-res assets; physics-based simulations and motion; realistic textures and clothing; all manner of particle FX; and cameras and complex lighting—all controlled in real time for fast, iterative, creative decision-making never before achievable. It may be disruptive, but it's certainly not trivial, nor the folly of game-centric pioneers who never learned to respect tried-and-true animation production pipelines.

Real-time animation may not be the cavalry charging down the hill just yet. It's more like having a dependable friend who you know will always show up on short notice, eager to lend a helping hand. And that's the type of help that's always welcome.

Dan Sarto

Co-Founder, Publisher, and Editor-in-Chief
Animation World Network

Dan helms Animation World Network (AWN.com), one of the largest publishers of animation and VFX news and information. Dan has served on juries, delivered keynotes, and moderated panels at numerous industry events such as the Animation Masters Summit and FMX, and is currently Chair of the Computer Animation Festival for SIGGRAPH Asia 2022.

Contents

Foreword	4
Introduction	8
Who Is This Guide For?	9
Unreal Engine and Real-Time Pipelines	9
Chapter 1: Evolution of the Real-Time Pipeline	10
Linear Pipelines	12
Traditional 2D/Cel Animation Pipeline	12
Traditional 3D Pipeline	13
Pipelines and Creativity	13
Real-Time Pipeline	14
How Real-Time Pipelines are Changing Animation	14
Interview: BRON Digital Fables, Gossamer Jason Chen and Aaron L. Gilbert	17
Interview: Spire Animation Studios Trouble David Smith, Cosku Turhan, and Rajesh Sharma	22
Interview: Sony Pictures Imageworks In Vaulted Halls Entombed Doug Oddy	26
Chapter 2: Real-Time Animation Pipeline in Action	30
Working With a Real-Time Pipeline	32
Rendering in Real Time	32
Parallel Workflows	32
Collaboration Across a Single Platform	32
Animation, Live-Action Style	32
Working With Locations, Not Scenes	32
Making the Shift	33
Interview: La Cabane Productions, Shards CGI Mush-Mush & the Mushables Perrine Gauthier and Roman Trimaille	34
Interview: BRON Digital Fables Russ Smith and Alex Filipov	36
Real-Time Pipeline by Production Phase	39
Development	39
Pre-Production	39
Production	39
Post-Production	40
Final Output	40
Interview: Digital Dimension Entertainment Group The Beachbuds, Chase and CATCH Martin Walker and Frederick Faubert	41
Interview: Chromosphere Studio Yuki 7, Mall Stories Kevin Dart and Elizabeth Ito	47



Interview: BRON Digital Fables, Gossamer Monifa Andrew and Andy Bean	52
Technology Behind a Real-Time Pipeline	56
Real-Time Rendering	56
Sequencer	56
Remote Collaboration	58
Animation Tools	58
Other Key Features	59
Getting Started	61
Interview: 3rd World Studios Allahyar and the 100 Flowers of God Uzair Zaheer Khan	62
Interview: Evil Eye Pictures Fortnite Cinematics Yovel Schwartz and Dan Rosen	65
Interview: Cory Strassburger Xanadu	70
Chapter 3: Unreal Engine Pipeline Framework	74
Real-Time Versus Traditional Pipeline Steps	76
Rendering	76
Pipeline Development and Scheduling	77
Storyboards and Concept Art	78
Layout, Look Development, and Visualization	78
Interview: DNEG Animation Mr. Spam Gets a New Hat Taylor Moll and Shelley Smith	80
Interview: Chromosphere Studio Yuki 7, Mall Stories Theresa Latzko and Andrew Wilson	84
Interview: Engine House Wylder Natasha Price and Mike Richter	87
Interview: Sony Pictures Imageworks In Vaulted Halls Entombed Jeremy Sikorski	91
Animation	96
Shot Setup and Review	96
Visual Effects	97
Compositing	97
Editorial	98
Interview: Reel FX Animation Super Giant Robot Brothers! Rey Jarrell	99
Interview: WildBrain Studios LEGO Friends, In the Night Garden Morgana Duque	102
Interview: Sony Pictures Imageworks In Vaulted Halls Entombed Craig McPherson	106
Interview: Reel FX Animation Super Giant Robot Brothers! Paul Fleschner	108
Pipeline Functions	112
Hardware	112
Assets	112
Source Control	112
Importing to the Engine	113
Integratable Tools	113
Technical Support	113
Interview: WildBrain Studios LEGO Friends, In the Night Garden Shazzy Angulo	114
That's All, Folks!	117
Glossary	118
Links	119

Introduction

Animation is, at its heart, about bringing stories to life through characters, worlds, and performance. Artists create stories by following a production pipeline, a series of processes designed to make the workflow smooth and efficient. But even the smoothest pipeline requires persistent oversight and refinement to make it all happen, to get to that moment when it all comes together to produce a work of art.

Working within a pipeline is essential to finishing a project in any reasonable period of time. However, animation pipelines aren't set in stone—studios are always looking for ways to take advantage of new technology, to advance both creative and economic efficiencies.

It stands to reason that any changes to the animation pipelines that have worked for decades must not compromise storytelling, and ideally make it better, easier, and faster. Here at Epic Games, we see the industry poised at the precipice of a new paradigm for animation: a real-time pipeline that gives creators more opportunities to iterate, explore, experiment, and discover, from story development to final render.

In this guide, you'll get a tour of real-time pipelines in use today, and find out how producers, directors, TDs, and artists have eliminated common pain points in the traditional pipeline with this new workflow. These industry professionals, who have made the transition and are producing projects in Unreal Engine, will talk about the creative freedom they've enjoyed from collaborating and iterating in real time; share how production phases like layout, animation, and lighting are no longer linear, but now overlap fluidly; and express how seeing final pixels earlier enables them to tell better stories.

They'll also tell you about intangible benefits like the excitement of seeing the project come together so quickly, and greater artist satisfaction when changes appear instantaneously.



Who Is This Guide For?

The world of animation content is changing: how we make it, how we deliver it, and how we consume it. The linear content we have become accustomed to is being exploited beyond film and TV into new mediums like interactive experiences, virtual and augmented reality, location-based entertainment, and more. From this standpoint, it becomes clear that the worlds of animation production and game development are converging, in a way that gives creators more opportunities to tell their stories. This convergence includes the increasing acceptance of a game engine as a platform for the development of animated content.

Epic Games, the creator of Unreal Engine, has produced *The Animation Field Guide* for anyone interested in learning more about the many facets and benefits of producing animation projects with the real-time workflows that a game engine naturally fosters.

Whether you are a director who wants to streamline the review process for more opportunities to iterate on your vision, an artist who needs instant feedback on lighting or animation changes, or a technical director looking to drive more efficiencies across the entire pipeline, this guide is for you.

And when you are ready to wade in deep for more technical guidance and insights, you can explore the insights, tips, and tricks shared by industry professionals and Epic staff on our new [Epic Developer Community](#), such as the [Technical Guide to Linear Content Creation](#).

Unreal Engine and Real-Time Pipelines

Over the past 25 years, Unreal Engine has established itself as the premier open platform for AAA games, and has also seen wide adoption across multiple industries. In the field of architecture, designers have discovered they can use real-time technology to

bring their visions to customers much faster. In the automotive industry, Unreal Engine drives collaborative design sessions, driving simulators, testing of autonomous vehicles, and HMI (human-machine interface) solutions.

The film and television industry has been no exception—production teams have been leveraging Unreal Engine for years, from previs through post-production. The Disney series *The Mandalorian*, for example, can be credited with proving the viability of real-time technology and in-camera VFX to produce cinema-quality content.

Unreal Engine is experiencing accelerated adoption across the animation industry as a creative platform for development from pre-production through to production and post-production, and not just for real-time rendering. In fact, if your goal in implementing a real-time pipeline is merely to replace your existing rendering workflow and software, you are less likely to succeed, and will miss out on so many benefits.

As a platform, Unreal Engine affects all aspects of your workflow, challenging the familiar norms of a traditional pipeline. So, if you're just here to replace your renderer, this guide is not for you. But if you're ready to jump in and see how a real-time workflow can improve your entire pipeline, then keep reading as we share ideas and user stories on how studios are making the transition.

Animators got into animation to tell stories, and a real-time pipeline makes this more possible than ever. We're excited to be at the forefront of a new way of working that has the potential to move the industry to even greater heights, and we look forward to you joining us.

Let's get started.

CHAPTER 1:

Evolution of the Real-Time Pipeline



Image courtesy of Digital Dimension Entertainment Group



To better understand the benefits of a real-time pipeline, let's start with a quick trip down memory lane to revisit the traditional pipelines developed over the course of animation's production history. Exploring these linear pipelines will give insights into the differences between these approaches and the parallel workflows of a real-time pipeline.



Linear Pipelines

Animation has captured the hearts and imaginations of creators since its inception. Even before Disney and the Nine Old Men revolutionized and pushed animation into mainstream culture in the 1930s, its history was being written.

The 1800s saw a series of developments in stroboscopic illusion, starting with the phenakistoscope in 1833, a simple round card with drawings and slits that the viewer would spin and peer through. This invention led directly to several more illusion-based machines such as the zoetrope and praxinoscope. While all could only play loops, the praxinoscope advanced the science much farther with the capability to show a much longer animation, clever use of mirrors for smoother viewing, and projection of images so multiple people could enjoy the presentation at once.

The advent of film as entertainment in the early 1900s gave the world a new medium for sharing animated stories. After the studio produced the critically acclaimed *Snow White and the Seven Dwarfs*, the 1900s belonged to Disney, who strived to reach ever-higher standards for quality in each new film they produced.

Every project required the services of hundreds of talented, dedicated staff—animators as well as ink and paint artists, photographers, film editors, voiceover artists, sound engineers, and many others.

Disney continually looked to technology to enhance storytelling. Disney's contributions to the evolution of animation include the addition of fully synchronized

sound to animation, as seen for the first time in the short *Steamboat Willie* (1929); figuring out how to combine live action with animation for the film *Song of the South* (1946); and development of xerographic processes to speed up production, used on the film *101 Dalmatians* (1961).

Disney also took advantage of the tools and innovations of the day—the multiplane camera, rotoscoping, and many others—to streamline the production process while maintaining a high standard of visual quality and storytelling.

Traditional 2D/Cel Animation Pipeline

But even with the technological advances created by Disney, the 2D pipelines were very linear, took a long time to produce, and were extremely labor-intensive.

In an assembly line process, an animator would begin by drawing outlines of key poses on celluloid sheets (also known as *cels*), hand the sheets off to “tweeners” who would connect the action, and pass them on for inking, painting, and photography.

Any changes by a director would have an exponential effect on the process. Even a minor change in color would send the process back to be repainted by hand, and more detailed changes could mean starting a sequence from scratch with new drawings. Days, weeks, sometimes months of work could be lost.

Even in the most efficient pipelines of the day, each film took years to come to fruition.

Traditional 2D/Cel Animation Pipeline





Traditional 3D Pipeline

The release of Pixar's *Toy Story* in 1995 put 3D animation on the map for audiences everywhere, and inspired legions of studios to get in the game as the price of computers began to decrease dramatically and out-of-the-box 3D software became available (and affordable) for desktop PCs.

While computers also improved efficiencies in traditional 2D pipelines, the 3D revolution truly unlocked new creative possibilities in storytelling that were arguably second only to Disney's impact on animation in the 1930s. But this innovation also came with its own challenges, and what remained was still a largely linear assembly-line process with some new departments emerging along the way.

3D pipelines still required significant manpower, but with the introduction of the highly computational process of rendering, legions of ink and paint artists were replaced by rooms full of expensive computers. Just as with the 2D process, it took a long time for directors to see the final vision become a reality, and any changes involved expensive combinations of man- and computer-hours to fix. Changes had to go backwards in the pipeline across multiple departments, and were costly and time-consuming.

Year after year, 3D animation technology has continued to push the boundaries of visual fidelity with features like physically based materials, hair and fur, and physics simulators. But despite these advances, the number of man-hours required to finish an animation project remains quite high, causing the time to produce an animated film or series to still be measured in years.

Pipelines and Creativity

While traditional animation workflows have worked for many years, they have a significant drawback: an enormous amount of planning has to happen up front in order to avoid changes down the line, because changes are costly and time-consuming.

Because the director's final creative vision isn't realized until late in the production, directors do not have the freedom to freely iterate and experiment—they must hope their vision is realized, and must be frugal with their changes.

The bottom line is that the inflexibility inherent in a traditional pipeline can be stifling to creativity.

A true improvement to the traditional linear pipeline would disrupt its linear nature altogether, and provide a greater level of creative flexibility through the entire production process. Such a pipeline would put tasks in parallel for greater efficiency, and give artists the opportunity to iterate at any point along the way with minimal loss of production time.

Traditional 3D Pipeline



Real-Time Pipeline

The term "real-time technology" encompasses a number of tools, processes, and workflows, including:

- Real-time rendering, which game engines employ to display the environment and action at full quality as the player moves around;
- Real-time performance capture, where an actor's motions are digitized and transferred directly to an avatar as the actor performs;
- Real-time collaboration, where a group of artists or engineers work together on the same project in a live session, updating models, textures, lighting, and animation on the fly;
- Real-time hair and fur, where artists can see the final version rather than having to work with proxies;
- Real-time visual effects, where the fully rendered results of particle effects, physics simulations, and other memory-heavy post-processes are rendered and visible in fractions of a second, rather than going through a separate compositing step.

Advances in game engines are at the root of these newly available techniques. And when combined, these features and workflows become the foundation of an animation pipeline capable of achieving high visual fidelity while giving artists and technicians

more creative freedom than ever, and also speeding up production through easy collaboration and fast iteration.

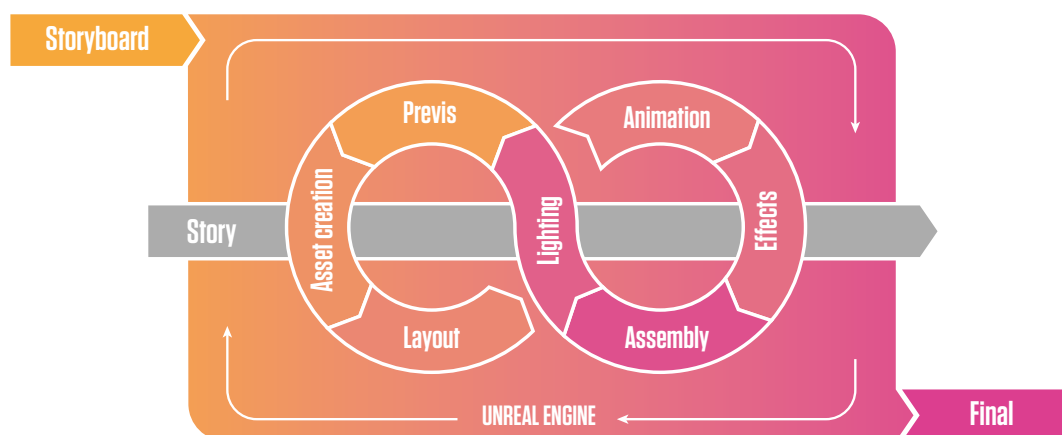
Accessibility, speed, and iteration are the hallmarks of a real-time animation pipeline, ones that set it apart from a traditional CG workflow.

How Real-Time Pipelines Are Changing Animation

The most obvious advantage of using a game engine to produce animation is real-time rendering, which enables rapid iteration and saves enormous amounts of production time simply by removing hours-long or days-long render times from the equation. But along with real-time rendering comes a host of additional benefits that change the way teams approach the process from start to finish, all with the goal of amplifying creativity and storytelling.

In a real-time pipeline, all the jobs and departments in a traditional workflow are still needed—layout, blocking, look development, rigging, animation, lighting, and special effects—but they don't necessarily occur in the order you're accustomed to. What's changed is the timing of these activities, with a great deal of overlap between steps that used to be gated and sequential.

Real-time 3D Pipeline





Another difference is in the way layout and previs flow directly into production. In a traditional pipeline, the products of layout and previs are commonly used only as reference, and all the hard work your team has done is discarded as the production progresses.

With a real-time workflow, your production becomes a first draft as early as story development, and creatives in every department can then fine-tune and polish the piece over the course of the production until completion. Working in this way gives each department a complete view of the project while it moves through development, so they can more effectively contribute to the whole. In effect, it brings post-production into pre-production. Basically, now you're "fixing it in pre" instead of "fixing it in post."

This workflow fundamentally changes how animation teams approach production from start to finish, giving them broad advantages:

- **Speed.** Work faster and smarter.
- **Iteration.** Iterate early and often. Go back, discover, experiment.
- **Visual fidelity.** Work at a high level of visual quality early, and without big render farms and long render times.
- **Collaboration.** Actually explore inside the project together with your team, and work in parallel with other departments.

These advantages translate to creative benefits and production savings:

More opportunities for creativity. When artists have the ability to rapidly address feedback, directors can more freely experiment and explore.

More opportunities for visualization. With real-time technology, you can begin 3D look development sooner, which better informs your artists down the line. Experimentation that would have been too difficult, costly, or time-consuming with a linear pipeline is now possible.

Faster iteration. Need something changed? Want to try a different camera angle? Want to improvise? What would this scene look like during magic hour? Artist changes take seconds rather than days or weeks. Now you can control the edit, which means you can find and enact solutions in the moment without having to go upstream in your pipeline.

Faster turnaround. Real-time rendering removes one of the lengthiest parts of animation production. With render times that can be measured in fractions of a second per frame, several versions can be output for review, cutting down the overall time needed for approvals. Shorter turnaround times have the potential to speed up production and reduce budgets.

Wider range of styles. From hyperstylized to photoreal, now you are only limited by your own imagination, as Unreal Engine has the tools to create just about any look you can dream up.

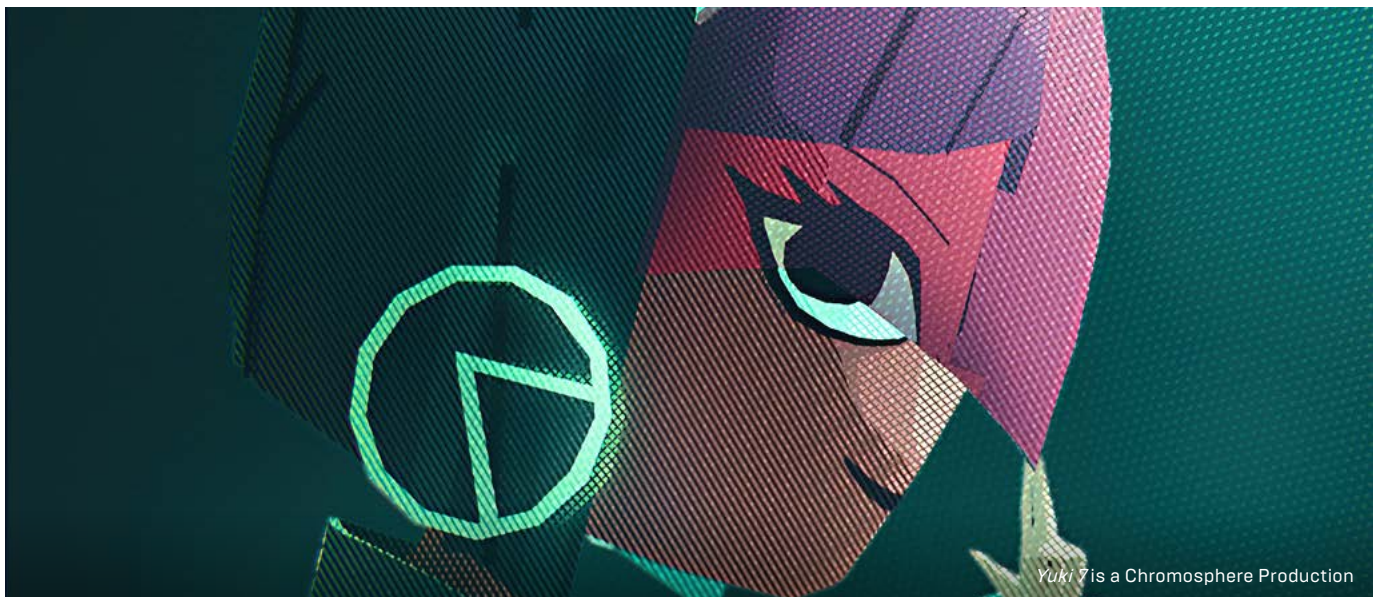
Get closer to the director's vision. When the director can request new mocap takes on the spot, see a dozen lighting options in a few minutes, and explore different camera positions with the slide of a mouse, they (and their entire team) can be in on moments of discovery.

Better storytelling. During production, it's not uncommon for the team to come up with better ways to convey a story beyond the original storyboard or script. With a real-time pipeline, experimenting with new shot compositions, camera angles, and lighting takes just a few minutes, so the story gets the best possible treatment by the end.

Transmedia opportunities. Modern storytellers are creating experiences for their audiences beyond linear storytelling. Real-time technology, as a platform, gives creators more opportunity to engage audiences across all mediums—games, mobile, live streaming, location-based entertainment, and XR, just to name a few.

“What was really new for us, when we started using Unreal Engine for *Yuki 7*, was achieving a final look we were happy with that didn’t involve a compositing tool.”

Kevin Dart
Creative Director and CEO
Chromosphere Studio



Chromosphere achieves a 2D paper look with *Yuki 7* directly out of Unreal Engine.



Engine House went for a soft, illustrative look in short film *Wylder*.



Interview: BRON Digital | Fables, Gossamer | Jason Chen and Aaron L. Gilbert

Jason Chen

Executive Vice President

Aaron L. Gilbert

CEO, BRON Studios

Vancouver-based BRON Studios was founded in 2010 with a “filmmaker first” attitude toward live action and animation, and has gone on to participate in numerous Academy Award-winning projects. BRON Digital was launched as a division of BRON Studios during the pandemic in 2020 with the intent to use real-time pipelines for BRON’s projects.

Fables is a retelling of time-honored tales like *The Tortoise and the Hare*, *Three Little Pigs*, and *The Boy Who Cried Wolf*, with a new spin for contemporary audiences. The studio is also working on *Gossamer*, an animated adaptation of Lois Lowry’s young adult novel about a dream-giving fairy.

BRON Digital has really embraced a real-time pipeline in the past few years. How did this come about?

Aaron: When Jason and I first connected, he was working with me on another show as a VFX supervisor. And one night at 3 AM, he was playing around on his iPad referencing Unreal Engine, showing me quick fixes on shots, showing me how things work. I had never seen it before. I was curious, and I asked Jason to come to BRON and teach us about the engine.

At the time, I liked our traditional CG pipeline. We were producing great stuff that I was super proud of, like *The Willoughbys*. However, I hated that one movie took three-and-a-half, almost four years to produce in our studio. And in that same four years, I was doing 60 to 80 live-action projects.

I wanted to figure out how to get three, four, five, ten animated projects going in that live-action timeframe. So the question became, how do we come up with a live-action approach to animation? I started talking about this 10 years ago. I actually wrote a paper about a live-action approach to animation, with our former head of production at BRON.

Then the global pandemic happened, and that required us to make some decisions. We had multiple live-action shows, a film, and a TV series in prep and production around the world, all shut down. We needed a line of revenue. So I asked Jason if we could get a team going for this real-time pipeline right away, and in April 2020, we started.

What do you mean when you call it a live-action approach to animation?

Aaron: It’s following the same structure as a live-action shoot, using call sheets and so on. It felt like a more efficient way to do things.

The Unreal Engine workflow is much closer to a live-action workflow than the traditional. What I love so much about it is our ability to see so many different things, and how to iterate, and how to change quickly. Just like live action, we can say, “Let’s do this again,” or “Let’s try it another way.” You don’t have that option with animation, if you’re doing it the traditional way.

That’s just not anything in animation you’ve ever experienced. And for me, that’s the live-action approach to animation. Everything that goes into making traditional animation still has to happen, but the steps are different now. You can bring it forward, looking at everything from the moment you start.

Jason: Live action is definitely where the core of my passion for filmmaking comes from. There's nothing like going on set, a singular huge soundstage. You have 75 to 100 crew members there, and then all of a sudden, the AD yells "Roll!" And between "Action!" and "Cut!" you have 100 artisans focused on that small, minute moment, each contributing their own art and skill within that moment.

What Unreal Engine does is replicate that. You're able to have focused energy and involve so many people on remote workflows, multi-user environment sessions, and review sessions.

In the prep phase, we've also adopted the live-action sensibilities of actually hosting live stage walkthroughs so that our directors and our seasoned production designers can virtually walk through our sets before any materials or shaders go on the environments. Our DPs get to look through the set and say things like, "This corner is a little bit dark. Can you punch a hole in this corner so we can have a light that comes through there?" We look at a revision the next day, and everyone signs off on it.

And then it progresses to what we love to call "physical production" in the set stage. The Art Direction team comes up with a swatch of different materials and colors, and they have a live session where they start placing new materials or even colors. We were doing this early on in 2020 with people in New Zealand, London, New York, North Dakota, and me in LA, all in one session. But because we're in Unreal, there's no C-stands, no sandbags, and it's that much more efficient. Our personal record is more than 21 setups in just a two-hour session with one of our DPs in Unreal.

One of the things that made me fall in love with movies is going on the Universal backlot, seeing how things were done. It's funny that it comes full circle. A real-time pipeline applies those same strategies.

Everybody is now contributing. But the really great thing is that from an art standpoint, in our environments, you see all those influences from a global perspective, which is something that we never anticipated. You give an artist a brief for something simple like a handrail, and then all of a sudden, you see these little flares of art nouveau coming through. You say, "That's beautiful. Who did that?" "Oh, that was Bruno." It all gels together in this truly international pipeline and workflow.

Aaron: If you're making a live-action movie, you want everybody talking up front. Inherently, this pipeline brings your entire process, including post and delivery, to the discussion from day one, because you have to be thinking about the end to properly lay out the beginning. For a filmmaker, it's an amazing thing to have, because you get to see things open up in front of you right away.

How has using a real-time workflow affected your speed and output of production?

Aaron: With the Unreal Engine pipeline, after less than two years, we're coming to market with our first show *Fables*, and our second show shortly thereafter, and we're into production on our third show. And there's a fourth, fifth, sixth, seventh, and eighth coming behind that. In a traditional animation pipeline, we'd still be over a year away from even the first show. That's a reality.

You want to be able to get to the end of the day and say, "Wow! Look what I made." In traditional animation, that "Wow" moment takes you a year or two to get to. That's really hard on a director, and on everybody, really, wondering what we're building, where it's going. That's what I love about adopting this Unreal Engine pipeline across the board: you have that "Wow!" moment every single day.



The pipeline that we've built using Unreal is just far more efficient—the timelines, the iteration capabilities, the fact that you get to see things immediately—than a traditional CG pipeline. And the work product that we see is phenomenal. At the end of the day, for me, it's about what we're going to end up with.

We're not doing anything outside of this. This is how we're doing it, period, because it's just so efficient. Why would we spend one dollar elsewhere? Because those other dollars can't do what one dollar can do in Unreal Engine.

Did you train your existing staff on the new pipeline, or did you need to hire new people?

Aaron: There actually was a culture change. We were going all-in with the real-time pipeline, and many folks that had been with BRON for some time decided they wanted to live more inside of a traditional pipeline. And they left us, which we understand, and we wish them well.

But folks who wanted to work in a new pipeline, stayed. We still have a lot of people who have been part of our team for many, many years.

Jason: We also thought about the fact that a lot of our amazing colleagues from our live-action shows are now out of work. How do we reach out to them and involve them in the process? Because one of the big problems and the big disruptions, I feel, that has happened in the past 10, 15 years in visual effects especially, is that the live-action crew shoot the product, they say, "Cut, wrap!" and they don't see it for six months to a year. And by the time they see it in the theaters, it's completely changed because VFX has so much been in a silo.

So we thought, let's bring in our live-action artists, our live-action DPs and production designers, into the real-time mix. They can see real results and not just place one light, come back tomorrow, see what the result is. They can instinctively light a character, light an environment. And there are no C-stands or sandbags to deal with.

Everyone's learning from scratch. We have artists that range in age from 20 to 60, that have come from a vast amount of different experiences. But they're all in the same playing field. It's really cool to see them on our Unreal Engine chat in Slack, sharing random things they find. It's really, really fun.

Within BRON Digital, we have a really great community of open communication. There's an hour set aside at the end of the week for everyone to show everything they've done, whether it's concept art, modeling, rigging, tools, sequences that we show, music, everything. And everyone jumps in, and shares, and gets to be in awe of each other's work, but also share information about how they did it.

Those sessions have been incredibly, incredibly helpful as we progress forward because, eventually, all those employees are going to be working together at one show or another. So it's important that we create that sense of community, especially in this time of the pandemic too.

Aaron: Also, the reality is, our first show was the first time we ever did it. There's a learning curve. The second show is more efficient, and the third show is going to be more. As we continue to refine processes and we can, as Jason is saying, share information across the different team members, that's the way we grow. It's BRON University. That's the goal.

Do you see other opportunities beyond animation with this pipeline?

Aaron: With a traditional pipeline, we would build, we'd rig, we'd animate, but we'd only touch the things that we were ever going to see. You can't throw anything out. This whole Unreal Engine world changes things up completely, because you're not just seeing what's on the screen.

You have to be cognizant of everything, because you're building inside a game engine, so it's literally a 360 of your environment. What that comes with, is just incredible. The director has to look really hard not only at what's directly on this screen, but we have to also project that into the metaverse, and into the online world, and into our games that we're creating, because every one of these shows, every show that we're making at BRON Digital has everything.

The online world that we're creating is cinematic and beautiful. Because of the engine, we can create these incredible worlds that are vibrant and interactive. Everything that we're making, every dollar we're spending here, we get to repurpose into that incredible online world, our metaverse, the BRON metaverse with these properties and then with our games.

We have a very ambitious NFT strategy and digital merch strategy around some of our properties, actually both digital and hard print. Our objective is to get to the plate over and over again with this incredible work. Everybody wants to build a franchise, and every franchise starts with the characters that you're making. Everyone has to fall in love with those characters that are on screen, or there is no franchise. Everything we're building right now is being built with the goal of creating franchises. We believe we're giving ourselves the best chance to do that because of what Unreal allows us to do, and how efficient our dollars can be.

Jason: On the technical side, we were able to do something really exciting with the environment from *Fables*. One of our artists was able to put together a working proof of concept that was pixel-streaming from the engine into a 3rd-generation iPad, and was able to walk around the *Fables* environment that we see on our TV series with minimal decimation. And he put it together in three days.

We could port that over because the assets exist, because the animations are there. As we animate, we also create animation libraries that inherently plug straight into the engine. So it's like we have all the tools available.

I'll always remember that we were sitting in our LA office, and I just handed Aaron and Steven, our CEO and COO, that iPad. And there was just one big green Start button. They pressed it, and they started running around the environment with the same lighting that was set by our DPs for those environments, with the theme music playing.

That's the real dream right there: once you finish watching your episode and you love the content you're watching, pick up your iPad and go explore on your own now. Find little nooks and crannies. We're planning to continue with future versions of Unreal to help tie everything together, to boost the quality, and make sure that we're continuously on the cutting edge of technology, too.



Interview: Spire Animation Studios | Trouble | David Smith, Cosku Turhan, and Rajesh Sharma

David Smith

Senior Vice President of Digital Production and Visual Effects

Cosku Turhan

Real-Time CG Supervisor

Rajesh Sharma

VP of Engineering

Spire Animation Studios' team has an impressive track record, with work on films like *Encanto*; *Spider-Man: Into the Spider-Verse*; and *Storks*. David Smith is the Senior Vice President of Digital Production and Visual Effects at Spire, and here he is joined by Cosku Turhan, Real-Time CG Supervisor, and Rajesh Sharma, the VP of Engineering. Their job is, as David puts it, "breaking pipelines to get things done."

All three are working on the company's current real-time project, the film *Trouble*. The story follows Jax, a teenager who enters a parallel dimension known as the World of Trouble, where he goes on adventures, makes friends, and plots a way back home.

What made you look into real-time animation pipelines?

David: We wanted to get away from the siloed departments you find on traditional animation projects, where an artist goes off into a room and lays stuff out. Instead, we wanted to get a place where we could be a bit more spontaneous and start collaborating in Unreal Engine in the moment as we created a story—anything that moved us away from one artist presenting their work, taking notes, and diving back into an independent file. A real-time workflow made more sense.

How is the real-time process changing how artists do their work?

Cosku: We are in the middle of a huge paradigm shift right now, because Unreal Engine is opening up a lot of collaborative workflows. It's also changing who can do what and when. For instance, our modelers are also set designers and world builders. Artists that never used to light are taking a stab at it, and it's all because a lot of different processes can now happen really quickly in and around the engine. And I think people are enjoying that, because everybody that we've been hiring has been saying things like, "I've been doing the same thing for the last 10 years, and I want to do things differently." They can have a bigger impact in this new paradigm wearing many hats, and they like it.

David: Giving artists the opportunity to wear more hats also lends itself to smaller, more agile teams, which is a goal of ours. We want to bring team sizes down, so we can have tighter communication loops and make sure everything gets considered contextually, instead of separately in bloated departments. We're looking at lighting, for instance, when we're doing the design of the assets, because how the light hits a particular environment is important to the way we're building it. And we're discovering that consideration can be done early in the process instead of too late, when you either have to go back or just live with it because you don't have time to make the change.



Cosku: Doing it in real time leads to a really awesome exploration phase during layout and when you're dialing in cameras, because you have a whole set being built under your hand. And within it is lighting and actual elements, all of which are a lot more developed than what you would have at a traditional layout stage. This allows for this clever workflow where you can put a camera in a place where you wouldn't think to before, because you didn't have the elements in front of you. It allows you to explore things really, really fast and easily. And it makes it really, really fun, too. A lot of people are seeing that spark in the industry right now, and there's a lot of excitement around it.

If people are wearing a lot of hats, how do you bring order to the process?

Rajesh: We organize the teams into pods, which is unique to Spire.

David: We call them pods because the artist teams in each pod are multi-disciplinary. So instead of siloing tasks—for example, having a development designer doing their thing, a modeler doing their thing independently, then a lighter doing the lighting at a later time—at Spire, the whole team is visualizing those disciplines together within Unreal Engine. As the sculpting or modeling process is underway, the lighter is contributing at the same time.

And even when we're doing some of the modeling in Maya, for example, we're immediately bringing that geometry into Unreal Engine, so we can have that multi-disciplinary interaction going as we're thinking about it—as we are realizing the overall design—in the same medium where we will execute the final shot. This is great for asset teams, of course, but it's also helpful for Layout. When we can look at the design through an intended camera, we can see if it's serving the story as the different elements come together.

What state are the assets in when you start collaborating?

David: Pretty rough. Once we have a concept in mind, we try to mock something up in 3D as soon as possible in Unreal, so we can start the process. Sometimes there are some textures on it, but it mostly starts pretty blocky. That's enough to get us started though and get the story beats roughed in.

The next thing the team wants is to get into a mocap volume for a step we call Story Capture; all of us are saying, "Let's just shoot it!" And it doesn't have to be complex. There's lots of very simple ways to capture the story in a virtual production setting, and we don't need great motion capture because we're not intending to use that later on; we will do stylized keyframe animation with feature-quality animators for the final performance in the story.

You get a really cool version of the layout this way. It encourages that spontaneous type of production that feels organic and lets us experiment in the moment with mocap and a Live Link to the Unreal scene. The actors love it, too, because they can see themselves in context. They can turn around and see what they look like as their animated character in the stylized world at any time. It's really helpful to them.

Another advantage is that once you've done those initial shoots, you always have the motion data. If your initial edit has you leaning towards a different angle, you can still use the performance data to refine a rough layout.

Once we have a good edit/layout, then we break the shots up and send them to the animation team currently working in Maya. They use the same environment geometry we were using in Unreal, which we export as USD files. The animation team animates the different shots and performances, and then they send it back to Unreal for further collaboration in context to finish the shot.

Once things start to solidify a bit more, we refine the rest of the details in each shot, for example adding very specific key lighting on the characters. Then we render everything out of Unreal and get these beautiful images. We do a composite pass, a post-process pass in NUKE with our main elements, and anything we can break out. This gives us a little bit of extra flexibility, in that we don't have to solve everything in the engine, while giving us a few extra tools that we can use to play with the look of the picture.

Cosku: Directors also love this process because it's so immediate. Traditional pipelines don't give you the leeway to make changes in the moment, or just have them sit next to you and cycle through lights or colors.

Rajesh: Sometimes, when you leave for the day, you lose the context of the request and why you made it in the first place. When you are making changes side-by-side, in real-time, you can immediately see if an idea works or not.

What was the biggest challenge you faced while implementing this workflow?

David: We're in a unique position because we're starting with this mentality, and everybody we're bringing on is all for it. They've seen the problems with a departmentalized workflow, and they come on wanting to make a change. But even then, you have to figure it out. Most of us don't come from a gaming background. Fortunately, we had the enthusiasm of Cosku, who had been through some AR and VR projects, and had even worked on linear animation with Unreal Engine in the past.

The only barrier was logistics. We had to figure out how to use it and determine what still needed to be done in a traditional way. Then as you learn, you share what you know. We are always talking about what we've learned, what we are working on, and what we still need to figure out.

I think the pipeline will always be evolving, since the technology is changing quickly. Everyone is trying different things, so we try to learn from them and through our experiences, just like we did the first time we hopped on computers 20 or 30 years ago and tried to get it done.

Rajesh: I see another challenge looming in the future, too. We proved it out with a small crew—around 15 to 20 people at the time—but to make a full-length feature animation film, there'll be sequences, and hundreds of shots, and hundreds of people distributed all over the world. Managing that work and still collaborating without stepping on each other: that's going to be the challenge.

Cosku: Definitely scalability, in terms of complexity and the crew. Because if 100 people are in the same shot versus three...of course, that'll make it a lot more complex, but we'll figure it out.

Does moving to a real-time workflow open up any other opportunities for studios?

Cosku: One of the big reasons we chose Unreal was because we want these movies to actually live in the metaverse. Not just distribution to a movie theater or the streaming platforms, but VR, AR, and mobile, so you can experience the characters and their environments on multiple platforms.

Plus, I think a lot of investors are also looking forward to that metaverse world that everybody's interested in. This enables us to be nimble with that. Since everything is built inside Unreal, we can export these worlds and characters pretty easily to different platforms, and then create new experiences much more easily. It's awesome to have a central hub that can serve all these different forms of output.

Interview: Sony Pictures Imageworks | *In Vaulted Halls Entombed* | Doug Oddy

Doug Oddy

Senior VFX Producer

Doug Oddy came to Sony Pictures Imageworks (SPI) from MPC, where he served as a visual effects producer on various films such as *The Mummy*, *Suicide Squad*, *Guardians of the Galaxy*, and *The Amazing Spider-Man 2*. He served as the Producer on *In Vaulted Halls Entombed*, an episode of the Netflix series *Love, Death + Robots* produced by SPI.

In Vaulted Halls Entombed, an episode of Netflix series *Love, Death + Robots*, centers on a squad of soldiers that find more than they bargain for as they venture underground on the trail of runaway terrorists. It was also the cast-off point for SPI's first venture into real-time animation, a move that saw them remaking their traditional pipeline from scratch.

Tell us about building the pipeline for *In Vaulted Halls Entombed*.

Vaulted Halls was the first time we did a real-time project using a game engine at Sony. We believed in the technology—we had seen the quality of work that was coming out of both individuals and companies around the world, and we knew it was capable of doing what we wanted to do.

We also knew that we had a lot to learn, and didn't want to encumber the process by trying to fit it into our existing pipeline. Instead, we decided to embrace the real-time toolsets and use them as they were intended.

We have an extremely stable, robust, and reliable pipeline at Imageworks that has been in development for over 25 years. As you can imagine, the protocols in place would require a great deal of time and effort to integrate a new platform—especially one we intended to rely on for final pixel. There are network configurations, a huge code base, our rigs, our shaders, the way we render. Every little detail we have created to improve the animation process, every tool that our artists employ day-to-day, all of the proprietary tools or modifications to existing DCCs we have written—we literally stepped away from all of it.

We built ourselves a sandbox in which we could use Unreal Engine without modification. We didn't want to spend time trying to make Unreal Engine mimic our workflows, when it was clear the inherent real-time workflows in the engine would let us jump into the creative development right out of the gate. We wanted to focus on creating great imagery immediately, and on creating a compelling visual story that would help propel the narrative.

We had a team of 40-45 people internally working on the piece, which took about nine months to complete, including prep, pre-production, performance captures, and post-production. When you consider that we had almost no experience with real-time, that turnaround is incredibly fast.

It was a big shift to our production paradigm, but the longer you try to push traditional workflows or mentalities into a real-time environment, the more you will struggle. Once we embraced the inherent advantages of real-time, the project really took off.



What were the steps in the pipeline?

We started with what we referred to as the “Nerf” build, which involved building up the piece from start to finish using very rough pieces. This gave us the most basic parameters for what that world might be. World building is a concept that exists in traditional workflows, but you can take a very layered and very measured approach when working in real time.

Then we started blocking out the narrative and dialogue, the key performance aspects. Normally you'd be doing this in individual shots, whereas here you're doing it per sequence and immediately seeing whether the world you built plays out.

Traditional storyboarding is still very effective, but here we actually created storyboards directly in engine. We did a rough layout pass with a version of the entire 15-minute piece that felt like previs—because the cameras are more dynamic, you can continue to add shape to it over time. Much like a painting, you layer in more detail as you go.

We decided to add some of our own performance captures to the piece, as we wanted to provide the actors and stunt team with as much detail as possible prior to the shoot. Our animation team used Xsens suits to work their way through the film, adding detail and playing with the director's ideas for the different beats. We wanted to provide the performers with as much detail as possible given they would be performing on a near-empty sound stage, and we could provide them the ability to see themselves live on screen, in the film, as they performed.

The mocap shoot was incredible. With the entire piece in Unreal Engine, we could not only show the performers the characters they were portraying, but the environments that they were supposed to be in—live on stage. We then recorded their performances with the Take Recorder, which meant we could walk away at the end of each day with all our performances available for playback in our Unreal Engine scene. Over the course of a week, we watched the piece transform from draft animations to a fully fleshed-out piece, complete with the actors' performances and director's selected takes. The captures still required clean-up and facial processing, but what we're able to record live in the engine is tremendous.

What type of work did you do outside the engine?

Aside from the usual modeling and texturing, we did the keyframe animation for non-human creatures outside the engine, the ones we couldn't mocap.

We did some of the effects entirely in engine, and some in Houdini, simply because we needed to art-direct certain aspects with the director. Part of the learning curve also required that we learn to adapt to our clients' established workflows and practices.

We used MetaHumans to create three of the six characters in the film. The team at 3Lateral helped us push the technology to get the designs very close to our actors' looks, without having to get them into a scanning facility. The ability to dial in new character designs in the morning, and then animate those characters in the afternoon, was mind blowing.

Which performance capture systems did you use?

Internally, we used the Xsens suits for the rough blocking, with the animators standing in as our actors.

For the production, we used two different scanning facilities: DI4D did the 4D facial captures, and Echo Studios did our high-resolution face scans. With the facial captures, there was a little bit of machine learning involved to maintain the volume of the face during the performances.

Then we brought the facial data into the engine, and married it to the mocap performances we captured of the actors in the Sony Interactive Entertainment capture volume. The SIE team supports a number of game studios and provided us with a tremendous amount of support, as they had previous Unreal Engine experience.

You've likened working in a real-time pipeline to creating a painting. Can you elaborate?

It is like working on a painting, or maybe sculpting, in so far as you are continually adding layers to the piece, reshaping, evolving, and discovering its form as you proceed.

With traditional pipelines, as you get further into production things become more galvanized, and you're less likely to start opening up the shots or sequences because you might have to go all the way back to the beginning of the pipeline.

With real-time, you don't go backwards. You're just working live, in the moment. Even with fewer crew, you're able to continuously create brand-new ideas all along the way. You can recut the movie in the eleventh hour, no problem. New cameras, lenses, moves, all instantly and at final quality.

The speed and flexibility with which you work in real time enables you to focus on the small details, and in turn step back to see how it plays for the final piece. No waiting for renders to see the impact of an idea. Taking it further, imagine if you could take your painting to final quality, then change and shift perspective instantly, and then do it interactively with a group of people. It's pretty incredible.



Love, Death + Robots | Courtesy of Netflix



Love, Death + Robots | Courtesy of Netflix

How does working in real time change the way you approach a project?

It's a different approach because you're not working on shots—you're building a world. Once the world is created and the animation is built, you can create new shots and re-envision the sequence, the edit, or the cinematography—all at no cost to your schedule. As long as it takes to move that camera or asset, there's your shot, rendered in real time.

It is quite calming to work and create inside the current edit of your film, as opposed to individual shots. Huge changes can be envisioned and completed or discarded without waiting three days to see how it plays.

Even our process for taking notes is impacted—when you are making the adjustments live as you work, you find yourself questioning what purpose the notes serve. We do keep notes, but they're mostly for historical documentation, in case we want to look them over later.

It is said that the beauty of live theater is that you can see a play or performance over and over, and each time simply sitting in a different spot makes for a brand-new experience. Working in real time has that feeling. You can work at final quality and change your point of view, instantly and with only a handful of artists. At that point, it's like Christmas every day.

What do you see for the future of real-time pipelines?

The collaborative nature of the real-time community is going strong, especially with Epic Games' open approach to sharing tools and advancing technology to all their users. They have created an incredible community. There's a sense we have only just begun to see the results of what this community of artists can do with these tools. It is an exciting time.

CHAPTER 2:

Real-Time Animation Pipeline in Action



As you’ve likely gathered by now, the use of a real-time pipeline requires a shift in thinking for those used to working with a linear workflow. In this chapter, we’ll discuss that shift, and also take a look at how a real-time pipeline affects the overall production process. Along the way, you’ll meet some of the creators who have made the leap, and hear about how they did it and why.

Working With a Real-Time Pipeline

What does it mean to produce with a real-time workflow? Here are the main aspects of the pipeline that lead to this new paradigm for animation production.

Rendering in Real Time

One major difference between the two pipelines is render times. Because rendering can happen as fast as a fraction of a second in a real-time pipeline, the long wait between hitting the Render button and seeing results is removed. By reducing the cost of rendering, you effectively disrupt the linear pipeline, and enable meaningful iteration in layout, blocking, lighting, animation, composition, camera placement, and other aspects of production.

Parallel Workflows

Many users of a real-time pipeline mention a parallel workflow as one of the great boons of a real-time pipeline. Lighters can work on top of modeling; animation can work alongside effects. Because all shots are assembled in Unreal Engine, with layout, animation, lighting, surfacing, and effects tools easily accessible by all departments, there’s no longer a need for one department to wait for another to finish and hand off shots for the next step.

Collaboration Across a Single Platform

Unreal Engine provides a central hub for all assets, animation, lighting, cameras, and VFX, so departments can work in parallel across sequences, saving time overall and giving more opportunities for collaborative storytelling.

Every member of your team, from directors to supervisors to artists, can watch the project come together as each piece is added or edited.

In a traditional pipeline, notes are made per shot. With a real-time pipeline, you can work in context across sequences to test variations and changes. A lengthy note on a change to a shot can go from “difficult to explain” and potentially misinterpreted by an artist, to “Let’s work it out while we’re on the call together.”

Such a workflow naturally enhances your team’s creativity and storytelling, as artists are able to contribute directly to the process in real time.

Animation, Live-Action Style

What if your animation process could be more like a live-action shoot where directors can be spontaneous, explore their creativity, and discover happy accidents on set?

- The director and DP can tour the virtual set to plan shots, lighting, and action.
- The Art Department can experiment with different textures, colors, and materials in conjunction with lighters in the environment.
- The production team delivers “footage” instead of shots to Editorial, enabling them to craft a better story.

Just as on a live-action set, these activities can happen at once, with multiple team members present and working together. Instead of a department working on a step and handing it off to the next, teams work in parallel, collaboratively, to bring a project to completion.

Working With Locations, Not Scenes

We mentioned earlier that working in a real-time pipeline resembles a live-action set. One example is the way that each level in the engine is a set or location where you can shoot multiple sequences, in the same way that one might shoot several scenes at one location in a live-action shoot. While you could, theoretically, do the same thing in your DCC app, Unreal Engine, having been designed for



the development of multi-level games, can handle much larger and “heavier” environments than most DCC apps.

BRON Digital has taken full advantage of this engine feature by building out the entire town of “Little Hamilton” in Unreal Engine. The town includes all the sets for their animated series *Fables*, with each episode taking place in a separate location.

“Doing it in real time leads to a really awesome exploration phase during layout because you have a whole set being built under your hand.”

Cosku Turhan

Real-Time CG Supervisor
Spire Animation Studios

Making the Shift

Some creatives come to the real-time table with the idea that they’ll use Unreal Engine for rendering, and nothing else. Why not just bring your completed environment and characters into the engine, and render them there? While there could be some merit to such a scheme, you’d be missing out on a world of benefits by limiting yourself to that alone.

In a real-time pipeline, you will find that scheduling and resource loading differs significantly from the traditional approach. For example, your lighters and VFX artists will likely be able to get to work much sooner than they used to, overlapping with tasks that traditionally come early in the process like layout and blocking. Expect that there will be some adjustment in thinking where scheduling is concerned, especially on your first project.

Overlapping tasks means multiple people working on a shot or sequence at the same time. This will require implementing source control software such as Perforce, an industry standard used by animation teams to keep track of the latest updates and prevent artists from overwriting each other’s work. The use of source control software is uncommon in traditional CG pipelines, and will likely take some getting used to in your real-time pipeline.

Review processes are also likely to change dramatically—for the better. Having a central hub means everyone can see the project progressing and can suggest changes. Balancing this new ability for group input with tight deadlines might take some getting used to. WildBrain Studio reports that during production on LEGO Friends, after the crew realized that everyone could contribute, it wasn’t unusual for a single review session to turn into an hours-long free-for-all. The producers quickly restored order to the sessions by setting new protocols, such as adjusting the list of meeting attendees and setting reasonable timeframes to iterate per shot.

Being open-minded and flexible means teams are able to embrace the changes and benefits a real-time pipeline brings.

“It takes time, personnel, and training to wrap your head around. It’s a paradigm shift, but if you can do that, it’s absolutely worth it.”

Taylor Moll

VFX Supervisor
DNEG

Interview: La Cabane Productions, Shards CGI | Mush-Mush & the Mushables | Perrine Gauthier and Romain Trimaille

Perrine Gauthier

Producer and Founder, La Cabane Productions

Romain Trimaille

CEO and Head of CG, Shards CGI

La Cabane is the France-based producer of Mush-Mush & the Mushables, an award-winning series that portrays the comedic adventures of a community of pint-sized forest guardians. For the second season, La Cabane is partnering with Shards CGI, a studio also based in France, for all CG work outside of rigging and animation. After using a traditional workflow for Season 1, the team has moved to a real-time pipeline for Season 2.

What does your pipeline look like at this point?

Perrine: For the first season, we had used an end-to-end Blender workflow, but now Unreal Engine is guiding layout, lighting, shot review, and final-frame rendering.

By using Unreal Engine for the layout, the CG team is able to anticipate any issues with lighting and rendering during pre-production, improving efficiency and avoiding the need for significant re-rendering of frames when artistic or technical issues arise.

Artists can also push the quality of their shots by observing the position and role of the light before animation begins. The director, Joeri Christiaen, is then able to choose camera angles that already account for the light's direction and shadows, achieving the best photography for a scene while also dodging potential light and shadow-related issues later on.

Romain: The drastically reduced render times are another big advantage. On Season 1, final images were coming in at 50 minutes a frame; we are hardly at three minutes per frame for Season 2, which means we can spend a lot more time on creative quality and introduce more subsurface scattering for added detail.

We're also getting a lot of value out of Unreal Engine's systems, specifically the ones that handle dynamic materials, Blueprints, and LODs. We're finding that the workflows are pretty easy and lend themselves well to quick changes or batch tasks.

In terms of Blueprints, we're using them to instantly modify the grading of multiple assets at one time. We also use Python to automate the ways we create, conform, and export assets and sequences. We hope this API will be extended to more features in the future; it's a powerful tool.

Perrine: This new pipeline also convinced us to refresh old content. For instance, we were inspired to re-render the opening sequence of Season 1 in Unreal Engine for Season 2. We saw a chance to push the quality even further while still preserving the look of the show. Now the render is crisper, the colors are even more vibrant, and the forest is more alive than ever, thanks to Unreal's environment abilities.



Mush-Mush & the Mushables © 2022 – La Cabane Productions – Thuristar – Cake Entertainment – VRT-Ketnet – RTBF Télévision belge

What prompted the shift to a real-time pipeline at La Cabane?

Perrine: One of the big reasons is that we wanted to reduce our carbon footprint, and the change is already bearing fruit. Since incorporating Unreal Engine into our *Mush-Mush & the Mushables* workflow, we've seen a 90 percent drop in rendering time between Seasons 1 and 2, and a drastic reduction in electric consumption, render farm volume, and the need for air-conditioned rooms. For the render time alone, we are logging 11,000 fewer hours on a workload of 14,300 frames. We're using Unreal Engine from the layout stage, which means we can anticipate a lot of the lighting and avoid re-rendering further down the line.

What's the value of real-time animation? How do you see it progressing?

Romain: The main advantage, at this stage, is the ability to iterate very rapidly, and of course not having to wait for hours to see your render. This gives us a lot of flexibility and creates more comfortable work conditions for our team.

We're very curious to see the next developments, and to see if real-time animation will catch up with more traditional technologies—especially when it comes to specific elements like glass, vegetation, and hair.

Perrine: When we greenlit the production, *Mush-Mush* Season 2 was one of the very first long-form animated series to be produced using Unreal Engine. This is clearly evolving, as several new shows are now in the works that use this technology, which is great for the animation industry. New pipelines bring fresh perspectives on projects and workflows, and interesting things happen when you get out of your comfort zone.

Plus, there's the environmental aspect. Although it's not typical to change a CG pipeline between seasons, we saw this as a great opportunity to implement a workflow that was not only innovative, but more eco-friendly. There are clear artistic, technical, and production benefits to choosing real-time animation, but these technologies can also trigger more sustainability in the industry.

Innovation certainly comes with additional risks and costs, but we believe that choices like this will contribute to reducing the carbon footprint of our sector. For instance, we're currently working with a sustainability consultant to carefully evaluate our emissions after the switch, and to see how it will help us reach the objectives of our decarbonization plan. We feel like we're headed in a good direction.

Interview: BRON Digital | Fables | Russ Smith and Alex Filipov

Russ Smith

VFX Supervisor

Alex Filipov

Animation Supervisor

Russ has worked for various studios, including Walt Disney Feature Animation and DreamWorks Animation, on projects such as *Meet the Robinsons*, *Bolt*, *Abraham Lincoln Vampire Hunter*, and *Night at the Museum*. **Alex** was Senior Animator on a variety of projects such as *The Suicide Squad*, *Wonder Woman 1984*, *Men in Black: International*, and *The Greatest Showman*. Both came on board at BRON Digital to work on real-time animation and VFX for the animated series *Fables*.

How has it been switching from a traditional animation pipeline to a real-time one?

Alex: One day, the “new thing” just takes on a forward momentum, and you get pulled along with it. It reminds me of when I started to move from classical animation to 3D. Originally, I didn’t even want to consider it. I could already draw anything I wanted, so I didn’t see the need. But then *Tarzan* came out and it convinced me that 3D was something I should look into, and maybe there was something to all those benefits I had been hearing about. I felt the same moving our latest project into Unreal. You can feel that something new is coming, and it’s worth taking advantage of. And already, we are fixing some legacy processes that used to have us sending things back to the 3D team. It’s refreshing.

Russ: Right now, we create almost 100 percent of our environments in Unreal Engine; use props from a variety of sources, including what we can get for free from the Unreal Marketplace; and bring in the other purpose-built assets from our DCC apps. So it’s a mix, not a complete replacement. You still have to make models, texture them, rig them, and animate. But what you get with Unreal Engine is an incredibly collaborative area to test and finalize ideas that is so much more rapid and engaging for everyone.

How do new animators feel about this workflow?

Alex: All of my animators are ready to jump into Unreal—if they haven’t already—especially the ones that just got out of school. For them, there is no long history of the traditional pipeline. Whatever you show them, it’s their “first time.” They are just like, “Okay, this is the way we do it.”

Starting in real-time also lets them sidestep some of the back-and-forth most teams have to endure. Communication is essential, but there’s a big difference between sending something back up the pipeline, and finalizing edits together in the moment without shifting tools. It does take some choreography and thinking about how to do it, but the tools in Unreal help us connect the VFX supervisor, director, and artists into a live discussion. And that cuts communication problems down a ton, which is so important because what we do is so visual and sometimes you need to see a change happening in front of you to figure out if it’s the right one.

Are you working with many big scenes with lots of elements?

Alex: We are, and that’s one of my key arguments for moving traditionally laborious processes out of our DCC apps. I remember this really heavy scene we had going, which had something like 70 to 80 full-res characters in it. In a DCC app, you’d have to do that in sections. This is a problem because compositions are changing all the time, the camera is changing, and ultimately, you just need to see all the elements in one place. If we were trying to render this all in a DCC app, we’d still be working on that shot nine months later.



Russ: All the cheats that you would have had to do...the significant polygon reductions, turning off deformer stacks, and all the rest of it would have been brutal, to the point where you aren't seeing what you hope to see anyway. But with Unreal, we're using fully realized, full-res assets during the look dev and lighting process and making judgments about composition and pacing. It's night and day.

And a lot of times, what we are seeing is running at five or six seconds a frame, which is perfectly acceptable. We don't require things to run in real time; we just want the best result we can get, and to make decisions faster.

Alex: What I love about Unreal is that no matter how many characters you have, or environmental elements, or lights, if you have a powerful enough machine, you can move them around pretty easily. What you see is what you get.

Do you always run the project at final settings, or do your settings depend on where you are in the process?

Russ: When you're driving the live sessions with the artists, you tend to turn the quality down so it reacts in real time. But then for final renders—to quote *Spinal Tap*—we go “up to 11.” It doesn't run in real time anymore, but who cares? It's seconds per frame, which is totally fine. We have our settings that we do for interactive, and they're more than informative enough. And then we have our settings for high-glossy, shiny, happy renders at the end. Both are extremely helpful.

Alex: You just never want to get that surprise after 11 hours or 24 hours of rendering during compositing. That, “Oh my God, I forgot that pass, or it's the wrong one” feeling. Lowering the settings—which is common practice with offline renders—makes sure we never get there, leaving us with this great assembly point for our artists and supervisors to build from.

Are you using grooms for all the fur in *Fables*, or are you handling that with textures?

Russ: We are doing both! There's a ZBrush sculpt on the assets that mimics the look of fur, and then for close-ups, we turn on the dense hair grooms. The grooms, working with the sculpt as a sort of undercoat, become a normal map so we don't have 100 million fur hairs. It's more like 10,000. I didn't think of it until we started dealing with Unreal and were about to lose a groomer. I imagine you could do the exact same thing with a DCC app, but the difficulty with attempting something like that is, you wouldn't see the final result until you render. You have to go through a lot of hoops to show displacements and you wouldn't see every part. It's just better in Unreal.



Fables image courtesy of BRON Studios

Real-Time Pipeline by Production Phase

A real-time animation pipeline hits almost all the same beats as a traditional pipeline, with a new twist: all steps can be performed in parallel, on the same platform. Let's explore the phases of production in the context of how familiar workflows translate to a real-time pipeline.

Development

During the Development phase, the story team needs tools that make it easy to build and arrange storyboards, and to experiment with different ways to tell the story. At the same time, the Art Department needs to start working on character and environment art, and on look development and visualization.

In a traditional pipeline, whatever work is done for storyboarding, look development, and visualization needs to be re-created later in the production environment.

With a real-time pipeline, the story team can begin the visualization process earlier in Unreal Engine, using rough representations of their 3D props, sets and characters, or even by kitbashing assets from resources like [Unreal Engine Marketplace](#) and [Quixel](#) or by using [MetaHumans](#). This can all happen in parallel as the Art Department finalizes their designs, enabling them to accelerate their 3D visualization.

As scenes are assembled and assets are finalized, the Look Dev team can experiment with materials, shaders, effects, and lighting to arrive at the unique look of the project, a look that can easily be applied to all scenes in the sequence.

Pre-production

In an animation project, pre-production includes preparatory steps like layout and asset building, which encompasses modeling, surfacing, and rigging of both environments and characters.

With a real-time pipeline, the steps of the pre-production phase remain largely unchanged. What does change is *when* pre-production starts—planning up front is absolutely key in a real-time pipeline, and that means pre-production work should start earlier than with a traditional pipeline. In fact, it commences before development is finished, and runs in parallel with it. In the same way, pre-production flows into production with parallel processes.

In a real-time workflow, once models are created, shaded, and rigged in a DCC application, the models can be imported into Unreal Engine for review by the entire team. Or they can be rigged directly in Unreal Engine with [Control Rig](#), ready for either keyframe animation or mocap within the engine. As the assets evolve and creative changes are made, they can flow directly into the Engine for iterative review. Pre-production would also include setting up versioning and source control, so the team can confidently work on scenes collaboratively.

“We’re seeing these benefits in pre-production, in production, and in post-production, where things that would usually be hard for us, all of a sudden become a lot easier.”

Andy Wood

Director of Real-Time Media
Disney Media & Entertainment Distribution



Production

Production is where the action is, where the shots and sequences are put together in Sequencer. But like pre-production, this process now runs in parallel—characters can be placed in their environments even before they are finalized, and the director of photography [DP] can start setting up camera angles while the lighter lights the scenes. The Art Department can continue to use these budding scenes to develop the project's visual style. And as models are updated and imported as part of the review process, any work done to date on materials and filters will be automatically applied to the new version.

Quite late into production on Treehouse Digital's *The Well*, a short horror film created in Unreal Engine with MetaHumans, Co-Writer and Director Peter Stanley-Ward felt that there was something missing with Trish, the lead character. The team added paint to Trish's face, and they instantly knew that was just what was needed.

"That's a great example of how you're able to constantly evolve because of that real-time pipeline," Peter says. "You're constantly able to make decisions creatively as you go, and the characters evolve and the story evolves."

If motion capture is used to animate the characters, a [Live Link](#) can be set up between the mocap session and the rigged character in the engine for immediate feedback on how well the action is working for the scene. The director can ask for more takes while the mocap actors are still suited up, and iterate until the action is just right. If remote collaboration is set up, the actors don't even need to be in the same city as the director—everything can run over the internet.

You can also now keyframe animation over motion capture data, or use your motion capture as reference and keyframe directly in engine with [Control Rig](#). Similarly, keyframe animation from DCC apps like Blender and Maya can be imported into Unreal Engine in multiple formats.

Post-production

In a traditional pipeline, post-production is what happens after the shots and sequences are set up and animated. However, in a real-time pipeline, even while the DP is still experimenting with camera angles and animators are working on their performances, the lighting and VFX artists can get to work. They can start placing lighting and effects even while look development is still in progress, with the effects informing the Art Department's decisions. Physics simulations can begin on placeholder assets, and be ready for further testing as final assets are dropped in.

Final output

When you're ready to render final pixels, it's time to turn up all the settings and turn on the post-process effects, and let it rip. With all the bells and whistles turned on, it's not unusual for rendering to take several seconds per frame. At that pace, you can provide Editorial with "footage"—several options for each shot—so they can pick and choose to put together the best version of your story. And if they need changes or additional shots, you can turn them around on the same day. You no longer have to wait hours, days, or sometimes weeks to get shots or wait for changes.

*"For final renders—to quote **Spinal Tap**—we go 'up to 11.' It doesn't run in real time anymore, but who cares? It's seconds per frame, which is totally fine. We have our settings that we do for interactive, and they're more than informative enough. And then we have our settings for high-glossy, shiny, happy renders at the end."*

Russ Smith
VFX Supervisor
BRON Digital



Interview: Digital Dimension Entertainment Group | The Beachbuds, Chase and CATCH | Martin Walker and Frederick Faubert

Martin Walker

Chief Technology Officer

Fred Faubert

Chief Creative Officer

Martin and Fred came to Digital Dimension with strong backgrounds in game development, and joined the team that developed one of the first Unreal Engine animation pipelines for the *Zafari* series in 2017. They have now gone fully real-time with projects for the company's Saturday Animation division.

As the animation wing of Digital Dimension, Saturday Animation Studio has been focused on creating children's content using innovative methods. In addition to the groundbreaking *Zafari*, the company has developed *The Beachbuds*, which follows a family of birds who run a resort, and *Chase and CATCH*, an upcoming show about a teenage cop who takes on a diabolical syndicate with the help of his by-the-book robot partner, CHASE.

Why did you decide to move to an Unreal pipeline?

Martin: It's all about real-time production for us and the ability to do very fast iterations to see results and get the feedback quickly, and then react to the feedback almost instantaneously.

With instantaneous feedback, you are actually sitting down with the Art Director and making changes on the fly, as opposed to the typical cycle of, render something, give feedback to the artist so he can change it, and then we'll render it again. That cycle takes three or four days. With Unreal Engine, we can do it in a matter of minutes.

Fred: That's the reason on the production side. I'm also overseeing all the IP development, and one of the goals is to build brands, not only animated series. By working directly in a game engine, we can fast-track products to market in other areas than the series itself.

This is the metaverse strategy behind building IP. It's important for us to make sure that our IP will have longevity on the market, and can reach and engage with a larger audience on different platforms in media.

How has your real-time pipeline changed since *Zafari*?

Martin: The first iteration of our pipeline, we were using Unreal for VFX, lighting, and rendering. All the artists were still working in traditional DCC apps, and then we would do a massive import of data to Unreal.

For the second project, we started doing more fancy special effects and shaders in the engine. This last iteration of the pipeline, we brought 60% of the artists into Unreal. The biggest challenge there is training, getting the artists to understand Unreal enough so that they can do their work in it.

For our latest iteration of the pipeline, we're doing all the layout inside Unreal. We're even using Control Rig for animation. We've developed our own custom shading model that we're using inside of it. The list goes on and on and on. Right now, all we're doing outside of Unreal is modeling, rigging, and skinning, and some animation if we need to. Everything else is done in Unreal.

What was the biggest challenge in switching to a real-time pipeline?

Martin: One, training the staff in Unreal, and two, understanding Unreal enough so it influences our actual pipeline and workflow internally.

We've got some artists that have been working in the same traditional animation pipeline for 20 years, and we're trying to convince them that it's much more flexible if you do it this way.

The traditional animation pipeline is more of a waterfall type of planning and production, whereas in game development, Fred and I are used to having lots of iteration. Getting them to understand how to do iterations to get stuff done, and even flipping some of the workflows backwards, required a shift in their mentality—in a nutshell, we had to educate the staff on understanding what's possible in Unreal in real time, and then adapting the workflows so that we could take full advantage of real-time features of Unreal.

What would be an example of flipping the workflow backwards?

Martin: Let's talk about environments, which they call set dressing or backgrounds in animation. In a traditional pipeline, they take the storyboard or the animatic with the concept art, and they draw up a list of the assets that they're going to need. Then they build all the assets and hand them off to an environment artist to build the set for a particular shot or sequence. So the steps are, build all the pieces, and then assemble the pieces, but before you can do this, you have to have a preliminary understanding of what all the pieces that you really need actually are. In the end, you build assets you don't need, and discover that you missed some of the required assets in the preliminary analysis.

What we're doing now is that we're getting the layout to do a first preliminary environment and camera setup in Unreal, even with just square boxes if they need to, just to get the lay of the land. And then we give that to an environment artist, who can start looking at Megascans and some of the foliage features in Unreal, or whatever they need to start assembling the set as seen by the camera before even having assets. At some point the environment artist says, "Okay, all I need now is these handful of assets to be able to finish the job." This is much more efficient and there is no waste.

That's a way of flipping it around, where we're starting to work on environments before we even have the assets.

What do you like best about a real-time pipeline?

Martin: What I really like is the quick iterations, and everybody working together towards the same goal in an iterative fashion. I find that it creates a lot more synergy between the departments. It's really interesting to see the transformation.

Fred: You can invest more in crafting the storytelling and the universe that will expand later, and engage your audience with diverse entertainment experiences. That's what you're doing in the end—you're crafting the storytelling in a more iterative way. That's a win-win.



Image courtesy of Digital Dimension Entertainment Group

Martin: It gives us even more flexibility than live filming. In a live shoot, the crew goes out and shoots a lot of film, then comes back to the studio to do post-production and the assembly of the film. Then if they decide they need a new shot, they have to go back and reshoot it, or they'll figure something out with what they have. But with Unreal, we have the opportunity to just go and change it. We can move the sun, move the camera, get whatever shots we need.

We did that once with a client, a quick demo for them. We have a stylized shader in Unreal Engine that makes 3D look hand-drawn, so we produced a video in that style and sent it to the client, and the client said it would be cool if the camera moved this or that way. We sent them a new video within an hour, and they were blown away.

What are some of the custom controls you created for your projects?

Martin: A studio in Bali hired us to set up the real-time pipeline on *The Beachbuds*, which as a series is 52 episodes, each one 11 minutes long. That's a lot of animation to produce, over 600 minutes! And our production team pulled it off in about 18 months, from start of project to end.

The biggest challenge for *Beachbuds* was automating the asset integration into ShotGrid for asset tracking. Also, the birds in *Beachbuds* mostly express their emotions through their eyes, so the animation is heavily focused on the eyes. One of the big wins we had was producing dynamic rigs for eye, specular, and lid tracking inside of Unreal using Control Rig. The animators had a field day creating expressions.

The other big win was the custom shader that we built. For *Chase and CATCH*, some of those images look hand-drawn, but they're actually 3D renders out of Unreal with our shading model. We modified the Unreal Engine source code to support our custom shader.

Fred: We're exploring and are doing a lot of look development with our custom toon shader, where we can play with the way the light bounces off the surface to create really unique looks. With our custom toon shader, we can colorize the line art by geometry so it gives us a look in between 3D and the classic 2D look of Disney's *Pocahontas*, which also had a lot of colorful line art. We're exploring a way to automate blending different art styles, where you can set a value to the degree you want and generate a never-before-seen visual signature.

We're trying to define our look as a studio, and get our IP out as well, so the audience knows by the aesthetic that it's part of the same studio, the same storytellers.

It sounds like Unreal Engine is helping you define your style.

Fred: It is. A lot of Unreal users are going for the PBR [physically based rendering] look, the more realistic way. By playing with these shaders, I think this brings opportunities to differentiate ourselves from the masses.

How have your clients responded to your shift to real-time pipelines?

Fred: It's a fun journey, but as we are beginning this new adventure, as an industry we constantly need to educate our clients that are new to real-time technology because they're used to running in a certain way, at a certain pace. It seems to them like we're doing the end at the beginning, and the milestones can sometimes be different from what they're used to. For example, as Martin mentioned earlier, we prefer to approach the layout before starting to build assets for the environment because we know we're going to save costs and maximize the use of the toolset and library provided by Unreal, but they're used to approving assets before doing layout.

It's hard to get a comprehension of the creative freedom a real-time pipeline can bring without experiencing it. Or sometimes, on the other hand, they think that because we are in Unreal Engine, everything is possible and there is no limit. They see a behind-the-scenes video on the web, and they see that it's an open sandbox, and everybody has fun and it's so productive. It's all true, but at some point we still need to draw a line in the sand. We need to establish clear milestones in order to do the entire show in a productive way. It's opened a box where some clients think they can iterate forever on things.

Martin: Here's the funny part, and we're even facing it internally as well: if we're talking to the creative people, they're all about the idea that the sky's the limit. And then we talk to project managers and producers, and they're saying they want the steps in this particular order, and otherwise we're not going to get paid. So there's a contradiction between those two worlds, right now, with regard to the way we do projects in real-time.

In a traditional animation pipeline, everything is so planned and set and predictable, to a certain extent. People are afraid of iteration because there isn't the same level of prediction for when you're going to hit your end game.

It's the same kind of comparison in software development, between a waterfall model and an agile model. Most managers like the waterfall model because it's somewhat predictable, but agile is all about iteration, two- or three-week sprints, and you just chug along and attack stuff as you need to attack it, and eventually you call it quits and you say "Okay, that's it." That's where we're at—you get to the point where you say, "We're going to release this now." We're facing the same type of scenario right now in animation, the difference between waterfall and agile. Waterfall is the traditional pipeline, and real-time is all about being agile.



“The main difference is instead of shots, we build places now. Everything is laid out in a way that resembles live action, like you’re on location.”

Theresa Latzko
Technical Art Director
Chromosphere





Interview: Chromosphere Studio | Yuki 7, Mall Stories | Kevin Dart and Elizabeth Ito

Kevin Dart

Creative Director and CEO

Elizabeth Ito

Director

Chromosphere is a boutique design & animation studio behind films such as *June*, and the visual direction of shows such as *Carmen Sandiego* and Elizabeth Ito's *City of Ghosts*. Kevin has previously worked as an art director and designer for *Steven Universe*, *Big Hero Six*, *Inside Out*, *Cosmos*, *Her*, and *The Powerpuff Girls*.

Chromosphere began exploring real-time pipelines for the first time on the third episode of *Yuki 7*, an animated web series that embraces the aesthetic of 1970s spy films. They're now putting the same pipeline to good use on *Mall Stories*, an Elizabeth Ito series centered on the day-to-day thoughts of people who work at a mall.

Tell us about your journey from traditional animation to a real-time workflow.

Kevin: With *Yuki*, we really just dove right in because we had already done two episodes of it using our regular pipeline. We already had the characters and the environments in Maya. We refer to it as our traditional pipeline, but it's definitely not like ink and paint days. It's still pretty modern, using all the digital tools.

What was the biggest challenge in changing the way you worked?

Kevin: What was really new for us, when we started using Unreal Engine for *Yuki 7*, was achieving a final look we were happy with that didn't involve a compositing tool. This was pretty scary for us because we always used the same consistent tool across all our projects.

In our previous pipeline, it's a long, drawn-out, shot-by-shot thing, and every week you get some new shots and plug them in. But in Unreal Engine, it's like you're working on one shot and continuing to improve it the entire time, and everything's happening simultaneously. So the look just kept improving uniformly across our entire episode as we were working.

Whether we were doing 2D or 3D, we were always very, very dependent on compositing to get the signature look of our projects, where they all have a unique, stylized look to them. Stepping into Unreal was abandoning that aspect that we've relied on pretty much since I started doing animated projects in 2008.

There were a huge number of unknowns about whether we could even get that amount of control over the images that we're so used to having. In the new Unreal Engine pipeline, getting the look to work was like we'd entered a whole new world, and now we're building our home where we can do the look the way we want.

How does using a real-time workflow impact your work on *Yuki*??

Kevin: We weren't thinking about it a lot while we were doing it, but once we finished that third episode for *Yuki* 7, we kept talking about how different it was, that everybody who was working on the project in Unreal was always able to see the entire film in there, in Sequencer.

It's actually a pretty seismic shift for everybody. Now, they always have access to the latest thing. They don't need to wait for it to go through lighting and rendering and compositing, and then get it to me to plug into an edit. It's all just happening right in there, all at the same time. And I feel like it gives everybody a lot more context the entire time, much more than they're used to having.

As a director, it's nice that I'm not the only one who always has an overview of the entire project. Everybody can see the whole thing put together all the time, and less things tend to fall through the cracks.

There are so many small things when you're making a film. I kept thinking about the spark effects we did in *Yuki*. On the other two episodes, there were so many little things like that, where I thought, "Oh, man, we forgot to make this one splash here. I've got to let somebody know."

In our old pipeline, we send it to another person, and that person animates it, and then it gets sent back through this whole process. But with the real-time pipeline, because everything has to go through less hands, we can just assign all the effects to one person, and they can go through and see where all the effects need to be.

One of the changes is us simply needing to recognize that it's really different in a good way. And now that we know it's like that, we can plan for how to utilize the fact that everyone can see everything all the time on the following projects. It's more of an advantage.

Part of the motivation behind it is knowing that, once this is all done, that we could take this and get all these other uses out of it, because we could do more films in the same space because we've built it out in this way. Or, we could do other interactive experiences.

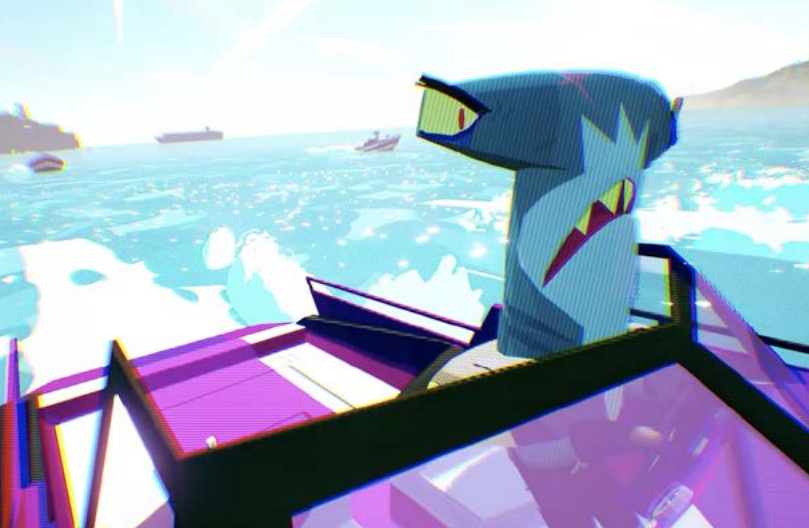
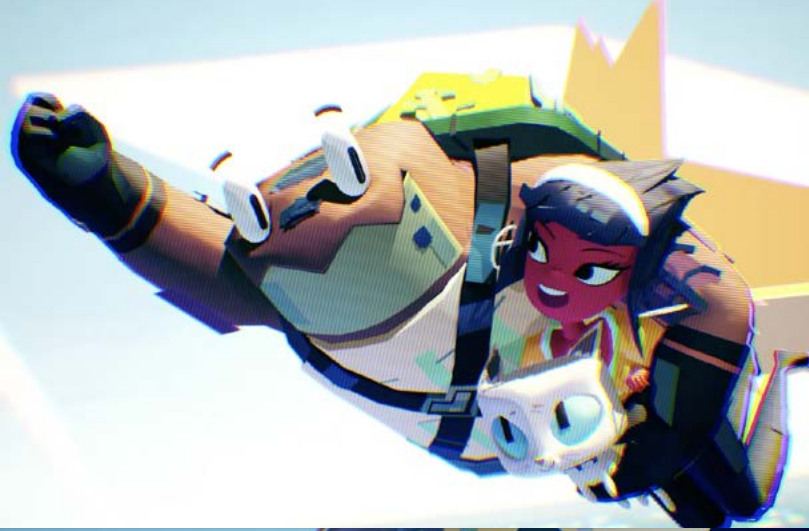
Tell me about the real-time pipeline on *Mall Stories*, in comparison to the pipeline for your earlier project, *City of Ghosts*.

Elizabeth: With *City of Ghosts*, we used an outside studio to create the environments, so we had to have specific ideas of what shots we were going to do before we took renderings of the space to use as backgrounds.

Unreal Engine gives us the possibility of moving around in the environment and figuring out our backgrounds that way, as opposed to having to draw storyboards. For *Mall Stories*, it was really nice to be able to just walk around and see it and ask myself, "Is this the space that I had in mind? Is the size of this space good?"

Another example is with one of the characters in *Mall Stories* who works at the Mongolian grill. We asked him, "How did you end up working here?" and he said, "I just came into the mall for a cheesesteak and I saw a 'Help Wanted' sign at this restaurant, so I came over and applied for a job."

So I told Kevin and his designers, "There's got to be a cheesesteak restaurant somewhere, where we can pan over to it behind this Mongolian barbecue." The difference is, I didn't have to know that for us to get started. We can add things in as we need them.



Another difference is being able to see some changes right away, and not have to wait for someone to do them and render them out. For example, in the first passes that came back for *City of Ghosts*, there were a lot of scaling issues, where we had to send characters back to the CG studio to have them made a certain percent bigger. Then the characters would come back, and it still didn't look exactly right, and we had to figure out why. With Unreal, it's more immediate—you can see somebody scale something right then and there, in context.

On *Mall Stories*, we had an issue with the width of an escalator. We knew it had to be wider than it was originally designed, but we weren't sure exactly how much, and also whether we needed to scale some of the characters to fit on the escalator. It would have taken weeks to have figured out something like that with our previous way of working. For *Mall Stories*, we could figure it out really quickly in Unreal.

How did working in Unreal change the way you approach problem-solving?

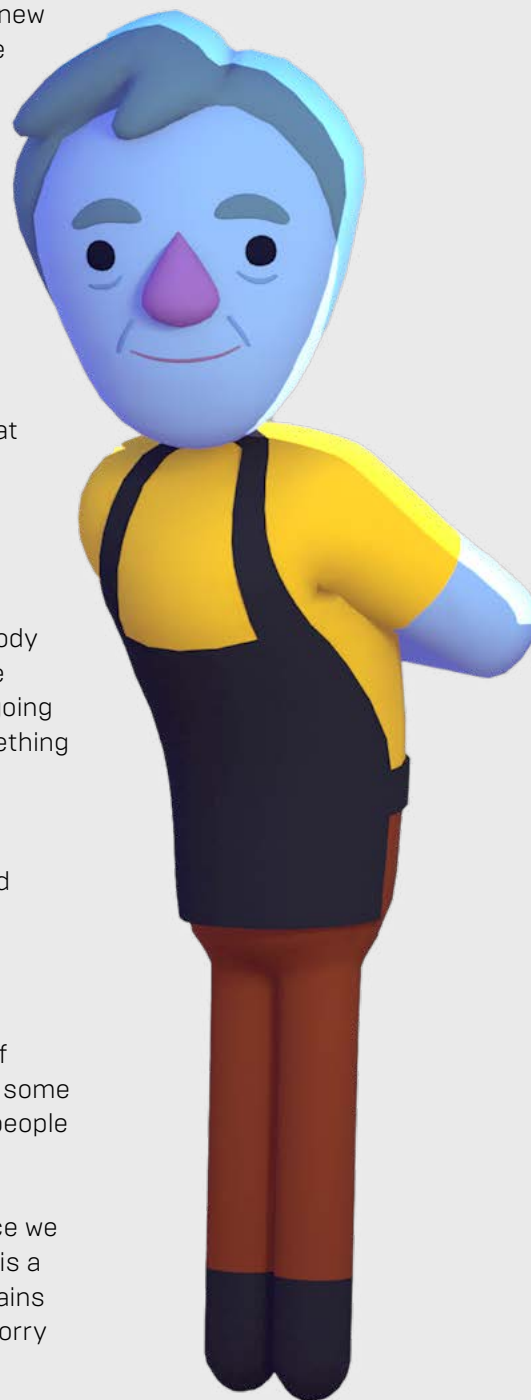
Kevin: Despite all the things we might adapt specifically for that pipeline, we're still the same 2D-minded people, and we always look at everything and ask, "What's the easiest way we could possibly solve this?" For all those things that could be huge technical headaches, we look at whether we can do something really easily just by moving things around in Unreal.

The noodles in *Mall Stories* are a really good example of this. Everybody was stressed out at first, knowing there are all these shots of people cooking and moving noodles and food around. We thought we were going to have to do noodle simulations with physics and collisions, or something complicated like that.

But Theresa [Latzko, Technical Art Director at Chromosphere] came up with a really great rig where we just modeled a pile of noodles, and she gave the animators some great tools to jiggle them around a bit. And it looks completely realistic. It looks like we did something really technically complex.

Elizabeth: For *Mall Stories*, we're playing around with that concept of background actors. You have each of the background characters on some kind of loop, where they're eating and looking at their phone, or two people just strolling around and around the mall.

Kevin: We're trying to set it up so that all the environmental ambience we essentially get for free any time. Once it's all set up, it's like the mall is a living thing. There's people walking in it, and they're shopping. Fountains are going. There's people sitting and eating. And you don't have to worry





about, per shot, what is actually happening in the background because it's just always something. It's like you have a live set that you're shooting on.

We're hoping we'll be getting happy accidents, like people wandering in front of the camera.

Elizabeth: I'm excited to see that happening with animated characters, kind of like out-takes but for animation. And that's definitely not how *City of Ghosts* worked, where we had to plan out exactly what the characters would do in each shot.

Kevin: Elizabeth keeps saying she just wants a tool where she can sit there and watch stuff happen in the mall—which we could totally do.

Elizabeth: We can solve really simple problems earlier than we might be able to, if I wasn't involved in seeing everything about it.

Kevin: It's so different to actually be in the space and know it as a 3D object that you can look around yourself, or to see a character and be able to understand it the way you do when you're actually able to manipulate it in some way. I feel like that's a huge difference.



Interview: BRON Digital | Fables, Gossamer | Monifa Andrew and Andy Bean

Monifa Andrew

Line Producer

Andy Bean

Head of Real-Time Finishing

Monifa came to BRON Digital after working in VFX on projects like *Deadpool 2*, *Blade Runner 2049*, and *The Hunger Games: Mockingjay - Part I*. **Andy** Bean was a contract VFX artist for various studios including DNEG, Electronic Arts, and Method Studios. Both Monifa and Andy had some real-time experience before they came to BRON Digital—Monifa with virtual production, Andy with virtual reality—and they jumped at the chance to make a real-time pipeline work for the animated series *Fables*.

What has been the biggest challenge of switching to a real-time pipeline?

Monifa: For me, one of the big challenges was the terminology. People would ask me about environments, and I would explain that in Unreal Engine, each level is a different environment.

There was also gearing everyone else's minds to a real-time pipeline. As a line producer, you're in charge of explaining certain things to people, like schedules and dates. Talking about that with confidence while learning a new system has been an adventure.

Andy: On my side, the biggest struggle is getting people to forget that traditional mentality, where they just expect everything to work, and we'll just fire off a render at the end. It's a constant battle of re-educating people.

A lot of people coming in from VFX are perhaps a little bit institutionalized after doing the same thing for 20+ years. Being an old dog and learning new tricks can be tricky at this stage in life, but at the same time, it's massively rewarding, and keeps you on your toes.

Monifa: It does. You could be proficient in so many software packages, but you have to be able to adapt your mind and restructure in terms of everything that you've learned. And it's not necessarily scrapping it all, but it's mixing it up a little bit, and then being able to say, "Okay, now this is our pipeline."

Andy: It's still an ongoing fight when I say, "Look, we don't have a comp stage."

Monifa: Right. I haven't said the words "render passes" in 18 months!

What makes a real-time pipeline so interesting to you?

Andy: I was working at some other studios, and I was getting bored with traditional VFX. I got into animation and VFX to tell stories, and I felt like I was getting removed further and further away from that. I always seemed to be concentrating on some small detail, like a dinosaur's toenail, or something like that. And there would be six or seven people between you and the director, making decisions, each one trying to say their own thing and being a little bit nervous about making bold changes to try and tell a better story.

I just sat there bored one day, and I thought I'd check in on this new virtual reality technology, and I was just completely blown away. The software being used to create these amazing experiences was Unreal Engine, so now I wanted to get into that.



There's a lot more spontaneity and thinking on your feet, making sure all the different parts play together nicely. At the same time, you've got to be able to cheat things and hack things. It feels almost like the old days of VFX, where you'd have to hack things and stick them together with virtual duct tape. When you're invigorated, it encourages you to find solutions. I'm definitely not bored!

I was surprised at how easy it was to pick up Unreal Engine, and I found a lot more controls than with a traditional renderer, a lot more knobs and settings to dial stuff in and get it exactly how you want it to be.

Has the pipeline affected how you present works in progress?

Monifa: You have to adapt the way that you present the work because, within Unreal Engine, you can take your director around the environment multiple times. The person who made the environment is taking you on a guided tour, a 3D tour. In the old way of doing things, they'd get a cyclorama as a QuickTime video.

But with the artist giving the tour in real time, it means that you're heavily reliant on your artist. There needs to be that strong bond of communication and collaboration. We're always in Zoom, which means that we need to be very clear and communicative with our teams, even more so when you're working within Unreal Engine than with a traditional pipeline.

Which Unreal Engine tools have helped you the most?

Andy: I've always shied away from the programming side of things—I'm not into Python or C++—but with the Blueprint system in Unreal, it's like building with LEGO blocks. You can just put these colored blocks together and streamline the whole workflow process, and it was very easy to put that together. It's a very intuitive, easy-to-understand way of doing things.

What kinds of utilities did you build with Blueprints?

Andy: First up is the lighting utilities. A lot of the rendering and lighting in UE is controlled by Console Variables [Cvars] which are a little hidden under the hood. We've created a set of lighting utilities to give easy access to these, like a button to turn ray tracing on or off.

These are just the needs we run into on a daily basis. There's a button to open Sequencer, and another to open the environment map. We can turn the glass on and off, and deal with the skylight. If we want to look at the exposure values, we can bring all that up. Dealing with real-world physical light values and physical camera values means we can copy and paste light settings and camera settings to meet the requirements of our DPs.

The *Fables* project has 345 levels. We're constantly having to manage those and turn them on and off, so we have tools for that.

In my previous VFX projects, quite often you'd just be working on a per-shot basis. But with this workflow, we can grab multiple shots together and scrub across the whole timeline, and check that the lighting looks consistent, that it's telling the story, and that everything's behaving properly.

These utilities are just so simple to put together. There's a little bit of a headache here and there, but what I find with the LEGO approach is that I can almost work backwards, reverse-engineer things, and see what plugs into what and what doesn't. It's like doing Wordle, but in Blueprints.





What would you tell an artist or a studio looking to convert to a real-time pipeline?

Monifa: Reach out to people in the industry who you know are working on something that is Unreal-based, and ask them, “What have you found that’s simple, going through it? What have you found that are pitfalls? How would you say you’re dealing with that?” Definitely, reaching out to people and seeing exactly what they’re doing is the best way to get started.

And then on top of that, diving in and just having a little play around. Look at how you can take your existing experience, whether you’re into lighting or modeling or animation or VFX, and just dig into it, full-blown, and see what it can do.

Andy: I totally agree with Monifa on that last point. And there’s so many educational resources—there’s the documentation, YouTube videos, forums, and so on. Some of the talks from Epic people on YouTube are quite long, sometimes two or three hours, but you get to learn more about the mental processes behind Unreal Engine and a lot of its features.

Working in real time does take a bit of getting used to, but there’s no excuse. It’s free, so just download the thing, crack it open, and get your hands dirty.

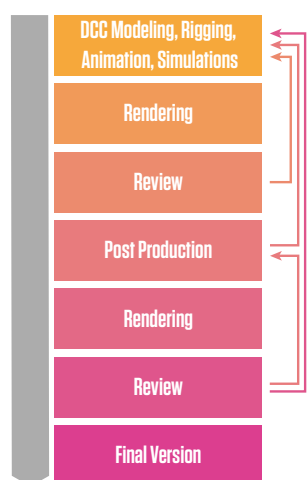


Gossamer image courtesy of BRON Studios

Technology Behind a Real-Time Pipeline

Unreal Engine started its life as a tool for game development, but has since proven itself as a viable production tool for linear animated content.

Traditional Linear Pipeline



Real-Time Pipeline

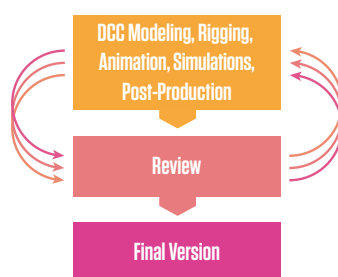


Figure 2: Comparison of pipelines with traditional rendering and real-time rendering

As the central hub for pre-production, production, and post-production tasks, Unreal Engine can render finished shots in real time or near-real time. It also provides a single space for all assets, animation, lighting, cameras, and VFX, so departments can work in parallel, saving time overall and giving more opportunities for collaborative storytelling.

Unreal Engine has also found its way into other non-games industries such as in-camera visual effects; architectural visualization; automotive design and photography; medical and military training simulation; virtual reality experiences; and many others.

Here are some of the key features Unreal Engine offers for creating a real-time animation pipeline.

Real-Time Rendering

Real-time rendering means what you see is what you get as you work. This technology has been at the core of game engines from their earliest days. Nowadays, the quality of game-engine rendering has evolved to rival that of offline renderers.

With features like dynamic lighting, particles, physics, hair/fur, and real-time post-processes, it's now possible to get cinematic quality right out of Unreal Engine. With the introduction of Lumen, Unreal Engine 5's fully dynamic global illumination and reflections system, the gap between offline renderers is effectively closed.

Also introduced in Unreal Engine 5 was Nanite, a virtualized geometry system, which renders high-poly detail and high object counts in real time. Nanite gives VFX and animation studios more flexibility to migrate their high-density polygonal meshes into real-time production workflows.

When you need higher image fidelity than what is delivered with the real-time renderer—for example, if you want better anti-aliasing and motion blur—you can use in-engine features like Movie Render Queue and Path Tracer. These features give you more options for asset optimization—you can dial down levels of detail for real-time reviews and turn them back up for final output, while still rendering in seconds or minutes and not hours or days.



Unreal Engine Sequencer



Sequencer

Unreal Engine's **Sequencer** offers the power of a nonlinear editor for creating and previewing cinematic sequences in real time. A real-time pipeline often begins with blocking and layout in Sequencer with low-resolution objects and characters, then each asset can be replaced with a more final version as work progresses. Sequencer brings nonlinear editing tools to your 3D environment so you can have control of your characters, visual effects, objects, cameras, and sound effects across a multi-track timeline.

In Sequencer, you have the option of making changes across the full sequence, in addition to making a

change to affect only a specific shot or shots. Each character, camera, light, and object can be edited across an entire scene, giving you the ability to make global changes, and then play them back instantly for review. You can also create, modify, reorder, or delete shots from a sequence in real time.

With this workflow, Sequencer is an ideal tool for experimentation and iteration on shots and sequences, with any changes automatically propagated throughout the entire scene. Sequencer is accessible to all artists on the production so they know they are always working in context.

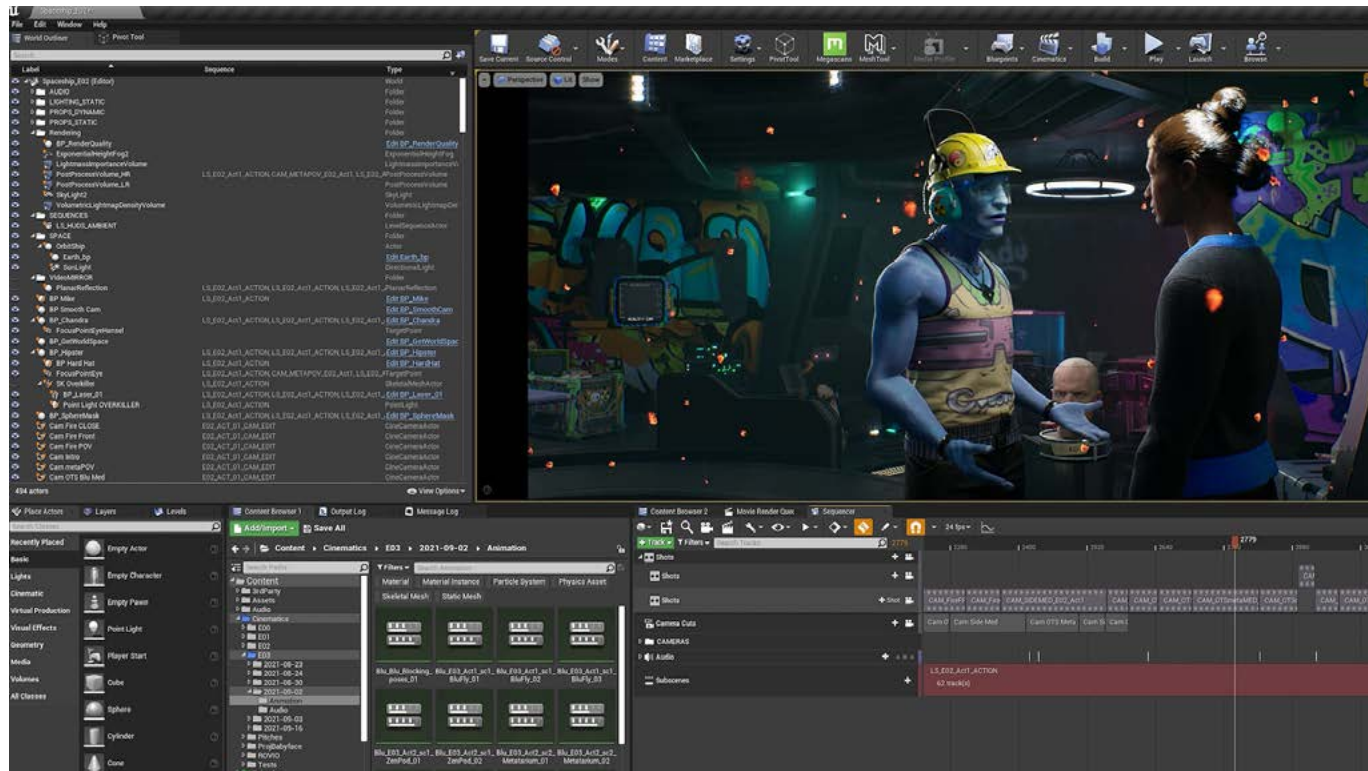


Image courtesy of Cory Strassburger

Remote Collaboration

Even before the COVID-19 pandemic caused office workers to set up shop at home, many animation studios had already started to explore the possibilities of remote collaboration. With the advent of remote conferencing systems like Zoom, distance is no longer a barrier to group brainstorming and story sessions.

Remote collaboration has also given studios more options for hiring the best talent for the project. It no longer matters whether an artist lives down the street, down the hall, or halfway around the world—aside from time zone differences, there’s no longer a reason to limit your team to locals, or ask artists to relocate.

Animation teams also use remote collaboration tools to enable virtual location scouting and live motion capture import.

Multi-User Editing. Unreal Engine’s Multi-User Editing system enables users on the same LAN or VPN to connect to a single session inside the engine. With the entire project in one central (virtual) location, teammates around the world can log in and have live review sessions within the project through Unreal Engine’s Multi-User Editor, regardless of location or time zone. As each user works on the scene, other users can see the changes in real time. When coupled with remote conferencing for live conversation, multi-user editing makes it possible for team members to discuss shots and sequences while updating them on the fly, making for a truly collaborative experience for all concerned.

Pixel Streaming. The ability for users to configure Pixel Streaming in Unreal Engine makes it possible to broadcast your project to the internet-connected device(s) of your choice. This means that a team member can watch the latest version of the project on a tablet or smartphone, from any location, without having to have Unreal Engine loaded on the local machine. The project remains on the server, with all the heavy lifting done server-side—the only thing that gets pushed out to the device is the visuals themselves.

Suppose the director is at the airport, waiting for her flight to be called for boarding, and you have a new version of the animation to show her. To see it, she could pop open her laptop, wait for Unreal Engine to load, join a multi-user session, and click Play. But the laptop might not be powerful enough to play back in real time, making it impossible for her to give feedback on timing. Instead, you could text the director a link, and she taps it to watch the playback on her phone. Simple as that.

The earliest uses of Pixel Streaming were for distribution of interactive consumer content—games, car configurators, and the like. With the advent of virtual production techniques came a new use for this real-time technology: the delivery of content for development team review.

Animation Tools

Animate in engine. As of Unreal Engine 5, animators can now keyframe within the engine. TDs can quickly and easily create artist-friendly controls for facial and body rigs using Control Rig, and share them across multiple characters; artists can then pose characters in Sequencer, save and apply the poses with the Pose Browser; and easily create natural movement with the Full Body IK solver. And with Motion Warping, you can dynamically adjust a character’s root motion to align to different targets—for example, vaulting over walls of different heights—with a single animation.

Motion capture tools. Whether you use motion capture for final animation or as a reference for keyframe animation, Unreal Engine’s Live Link and Live Link Face app can stream the data onto your character in real time, enabling mocap actors to create more meaningful performances and ensuring you nail the best take on the spot. With the ability to keyframe on top of the mocap data in Sequencer, you can refine the captured data into polished animations. Moreover, Unreal Engine’s robust retargeting toolset enables you to reuse animation between characters, even if they have different skeletons and proportions.



Other Key Features

Movie Render Queue. This tool was designed to give greater control and higher-fidelity renders for linear content when you can't compromise on areas like anti-aliasing and motion blur. In addition, it offers you the flexibility to render specific render passes when you absolutely need to leverage traditional compositing workflows and leverage plugins like The Foundry's Nuke integration with Unreal Engine.

Import/export. Unreal Engine supports file formats commonly used in animation pipelines including FBX, USD, and Alembic.

Customization via Blueprints. The Blueprint visual scripting system offers a straightforward method for customizing animation events. Common uses for Blueprints in animation include changes to materials based on a character's proximity to an object, and triggers for particle systems to create water splashes or fire.

VFX. Unreal Engine helps make effects more accessible with numerous tools that run in real time, so you can see what's happening immediately. These tools include:

- **Chaos Physics.** If your project calls for breakage and destruction, cloth, fluids, or ragdoll animation, real-time physics can be a game-changer. By seeing the effects during the production cycle rather than in post, animators and environment artists can tweak models and placement for the best possible effects.
- **Niagara particles.** Create fire, water splashes, smoke, and other particle-based effects, render in real time, and see exactly how they'll look in context.
- **Hair and fur.** Real-time rendering of hair and fur means you can instantly see how lighting and motion affect them, without having to wait. This workflow greatly speeds up the process of tweaking hair properties like color and shininess to get the look you want, and gives instant feedback on how secondary motion affects the hair and fur.

"The confidence in knowing that when we hit the Export button it is going to be right the first time, just means we move faster."

Andy Wood

Director of Real-Time Media
Disney Media & Entertainment Distribution

Post-process volumes. With real-time rendering, the intent is that you can achieve final pixels directly out of Unreal Engine and avoid the need for compositing. Unreal Engine offers flexible and powerful post-processing effects to tweak and fine-tune your scenes. Take advantage of built-in tools for color grading and camera emulation techniques like lens distortion, chromatic aberration, depth of field, bloom, and more.



Image courtesy of Engine House

Python integration. Unreal Engine now supports Python 3, giving you the ability to build custom tools to meet your specific production’s needs. Combining Blueprint-based Editor Utility Widgets and the power of Python, you can build new tools and interfaces to refine and automate your pipeline.

MetaHumans. Epic’s MetaHuman framework enables you to create photorealistic human characters, complete with hair and clothing, in a fraction of the time it would take to build them from scratch. Leveraging scan data from multiple real humans, the MetaHuman Creator cloud-based application produces

characters with complete body and facial rigs, ready to animate in Unreal Engine, in minutes. And with the MetaHuman Plugin for Unreal Engine, you can easily turn your own custom facial mesh into a MetaHuman.

Quixel Megascans. Quixel’s vast and stunning library of vegetation, organic and man-made assets, and high-resolution textures is free for all use in Unreal Engine. With Quixel Bridge built in to Unreal Engine 5, assets come ready to drag and drop into your scene, giving you the ability to quickly create a rough layout or even a full, photoreal environment in minutes, all of it rendering in real time.



Image courtesy of Cory Strassburger



Getting Started

The idea of switching to a real-time pipeline can be daunting. Where do you begin? Here are a few tips that our featured teams have given us for getting started with this new paradigm in animation production.

Start with a modest, asset-ready project. If you have assets from an existing project, that will save a step for your test project. Reel FX, for example, tested the waters by reproducing a single scene from an animated sequence they had already created for another project. They had all the assets already, and knew the look they were going for, so they were able to concentrate on learning Unreal Engine's toolset to reproduce the shot.

Adopt a “try it in the engine first” mentality. There are many areas Unreal Engine excels at today for animation pipelines, and also areas Epic Games is in the process of improving. When you dive into your first test project, we recommend you take an “engine first” attitude: try doing as much as you can in Unreal Engine, even if you suspect it might be faster or easier in a more familiar pipeline. This approach will lead you to discover where a real-time pipeline can contribute the most to your goals for the project, and how it can best be merged into your existing pipeline. Plus, when they dive in with this mentality, studios tend to discover exciting uses and benefits they hadn't expected or imagined.

Read up on relevant features. Which Unreal Engine tools can you make the most of? For example, if your project calls for high-poly assets such as trees and foliage, you can take advantage of Unreal Engine's built-in procedural foliage, removing the need to build and optimize assets in your DCC application.

Consider quick asset options. The Unreal Engine Marketplace is a great place to find all manner of engine-ready assets, whether for use as placeholders or final. One-man band Cory Strassburger makes liberal use of the Marketplace to find everyday items for his Blu character's spaceship home Xanadu.

Check out existing projects. Choose from dozens of Unreal Engine sample projects, accessible from the Samples tab of the Epic Games launcher, to see the tools and settings the creators used. Here's just one example, created by Weta Digital, which showcases the use of hair and fur.

Pick a team that wants to do it. Our users mention it so often that it bears repeating: you'll be best set up for success if you work with producers, directors, and artists who are excited about real-time technology, and who are eager and open to tackle the challenges of switching pipelines with creative problem-solving. For BRON Digital, this meant gracefully letting some talented artists go in favor of new hires who couldn't wait to try it out; for Sony Pictures Imageworks, it meant asking for volunteers from their extensive pool of artists and technical personnel.

Learn from others. The Epic Developer Community is where you'll find free video courses on just about every topic you can think of, from the basics of lighting and materials to more specific workflows for animation. It also hosts the Unreal Engine forums, which are a goldmine of information from users themselves.

“The power of a real-time rendering is that you have the ability to create, craft, and see that story in a new way early on, so that you can iterate and make better decisions, and ultimately create better films earlier.”

Jeremy Sikorski
Asset Supervisor
Sony Pictures Imageworks

Interview: 3rd World Studios | Allahyar and the 100 Flowers of God | Uzair Zaheer Khan

Uzair Zaheer Khan

Executive Producer

Uzair Zaheer Khan is the founder of 3rd World Studios, a Pakistani animation studio known for *Allahyar and the Legend of Markhor*, the first animated feature film built entirely within Unreal Engine. He also serves as director, screenwriter, and artist on his projects.

Khan's latest project, *Allahyar and the 100 Flowers of God*, is the first sequel in the *Allahyar* series, and will also be Pakistan's first stereoscopic 3D film. The story takes the heroes into the realm of action/sci-fi as they grapple with climate change and deforestation.

Tell us about your decision to use Unreal Engine to produce your latest film, Allahyar and the 100 Flowers of God. What was it about your previous experience that made you go in that direction?

Unreal Engine has come a long way since we made our first film, *Allahyar and the Legend of Markhor*. There were way too many benefits for us to not continue with Unreal Engine for our next film, the most obvious ones being the real-time lighting and rendering, its ability to handle massive amounts of data, the totally nondestructive workflow, and many more.



**What is your look development process for achieving the distinct aesthetic of the *Allahyar* films?**

Unreal Engine plays a very important role in our look development process. The visual workflow of the Material Editor is very empowering for our artists, who can create complex shaders with real-time feedback.

Overall, the real-time shading and lighting workflow and tools in Unreal Engine are a game changer. The fact that we can modify nodes using material instances makes the entire process very quick and easy, giving artists the ability to match and enhance the approved concepts without compromise.

We use Blueprints to populate individual scenes for lighting, which stores all the material information and eliminates redundancy. Our lighting artists are still able to tweak the material instances in real time for each scene or shot to achieve the desired look, and the look can be reused in other sequences if required.

For the first *Allahyar* film, you used Unreal Engine for environments and lighting, but did all animation outside the engine. Did you follow the same workflow for *Flowers of God*, or are you now doing more in the engine?

We are still using outside tools for the bulk of the animation, but are now using Unreal Engine for more and more animation-related requirements. For example, we deal with cameras almost exclusively within Unreal Engine as opposed to our previous film. We've since streamlined our workflow through automation tools for importing of assets, animation, and scene setup in Sequencer, and so on. However, we're very excited about the character animation workflow in Unreal Engine, and we're in the process of developing a brand-new animation pipeline that relies only on Unreal Engine for our animation needs.



Image courtesy of 3rd World Studios

Back when you made the first *Allahyar* film, you mentioned that there were naysayers in the industry who said you couldn't make an animated feature film with a game engine. Has that attitude changed at all in the industry, since then?

The attitude of the industry has completely changed in favor of Unreal Engine. It is fast becoming an industry standard in Pakistan, with multiple mainstream studios switching entirely to Unreal Engine-based pipelines for all sorts of CG content, from animated films, series, and commercials to architectural visualization.

We see that you are passionate about showing the world that Pakistani artists can produce great content. How does Unreal Engine help you achieve this goal?

Unreal Engine is a very robust and versatile tool that can help artists and engineers achieve objectives faster and with ease. For example, we have developed a complete stereoscopic solution within Unreal Engine, and are proud of the fact that *Allahyar and the 100 Flowers of God* will be the first stereoscopic 3D Pakistani film, all thanks to Unreal Engine.

But an extremely important aspect of working in Unreal Engine is the type of help and education that has been made available to the users from the Unreal Engine teams at Epic Games. This has been extremely useful for us in training and transitioning artists to an Unreal Engine pipeline. Despite being a very small studio, we have been very fortunate and feel grateful for the full spectrum of help and support.

What is the best thing about using a real-time workflow, when compared to a traditional pipeline? Conversely, what are some of the biggest challenges?

We laugh when younger artists complain about render times using real-time workflow. For people coming from conventional pipelines, who had to wait days to see the outcome of their hard work, Unreal Engine is a most beautiful dream come true. The challenges, if any, are being effectively dealt with by the lightning-fast development work being done on the engine. I suppose the challenge could be keeping up with the ever-evolving, almost revolutionary technology of Unreal Engine.

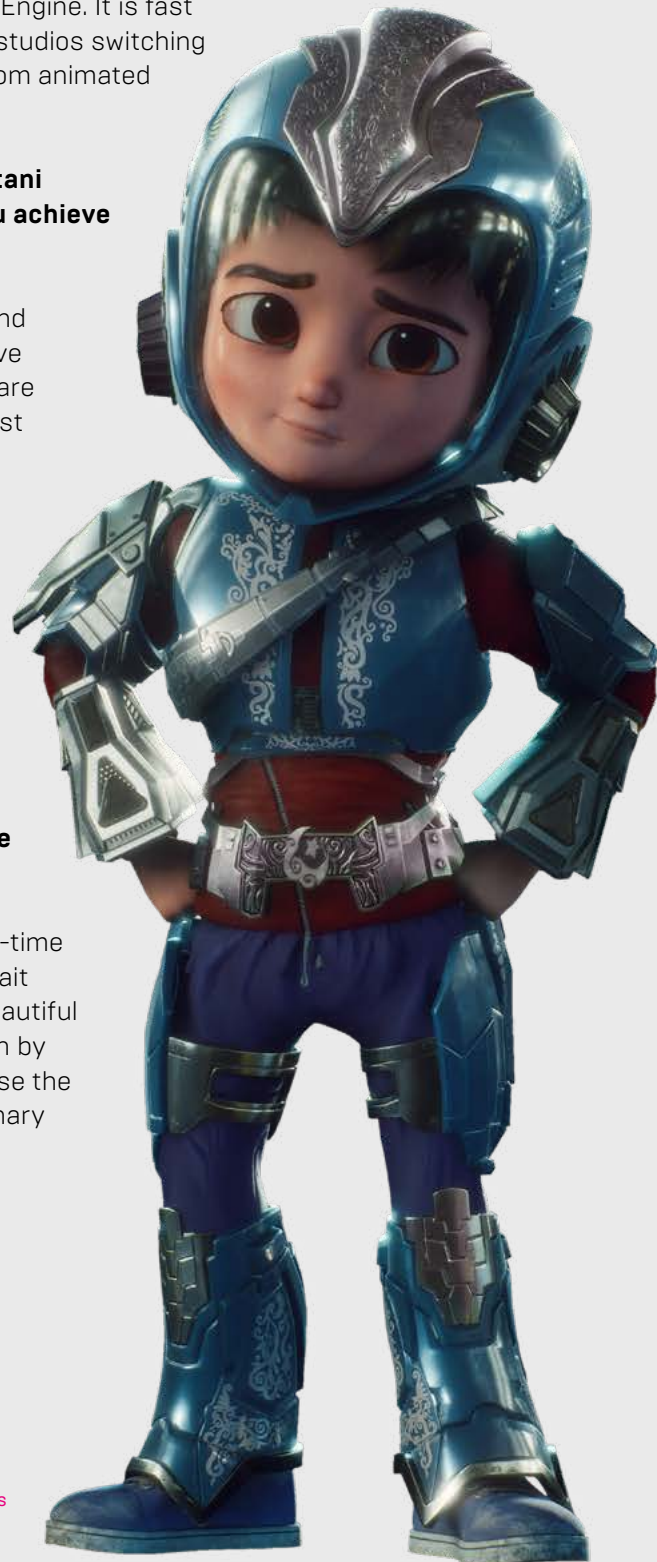


Image courtesy of 3rd World Studios



Interview: Evil Eye Pictures | Fortnite Cinematics | Yovel Schwartz and Dan Rosen

Dan Rosen

Co-Founder and CEO

Yovel Schwartz

Head of Production

Dan has contributed to award-winning films and special projects for over 25 years. Working as a compositor on Academy Award-winning films like *The Matrix* and *What Dreams May Come*, Dan moved into supervision on *The Matrix* sequels and other Warner Brothers projects. After founding Evil Eye Pictures in 2004, Dan led key visual effects on over 30 films, and contributed standards in VFX workflows to the Visual Effects Society Handbook.

Yovel has been working in production for 25 years on award-winning feature films, television shows, and commercials, and has a knack for applying new technologies to the film and real-time media space. As a producer, his clients have included Google, Apple, YouTube, Bot & Dolly, Audi, Virgin Records, Psyonix, and Epic Games.

Evil Eye is responsible for some of the earliest real-time animated pieces, like cinematics for *Fortnite* and *Rocket League*, a ray-traced piece commissioned by Epic Games for SIGGRAPH 2019, Google's famous VR project *Pearl*, and the recent *The Matrix Awakens* project that helped launch Unreal Engine 5.

How did you get started with using a real-time pipeline for animation?

Dan: My first experience with real-time technology was while we were working for Google Spotlight Stories, making animated short films in VR. They had developed a custom engine and brought us on to produce a few of their projects.

We would host people in our studio, and show 2D output in our screening room, and then have them put on the headset and experience the same film in 360° VR. It was always fascinating to discuss how storytelling could use the different mediums.

We continued experimenting with other real-time demos to generate new business focused on using Unreal Engine. We made a conscious decision to pivot from being a traditional visual effects vendor to a high-level experimentation studio focused on art and tech in real time.

The bet on using Unreal Engine and moving into real-time tech has definitely worked out for us. In 2021 we developed FUTURE.STAGE, a remote live system that captures actors' performances from anywhere in the world and streams them in sync into Unreal Engine. We're currently talking to some exciting partners who want to use it for real-time live animation as well as a playback system to seamlessly move into post-production.



How many artists are at Evil Eye?

Dan: We've always been a relatively small studio, scaling for projects. We fluctuate around 20-40 people. We've always maintained an intense attention to detail and a certain studio culture that would change if it became much bigger. Over the last few years, we've seen a lot of success in tapping into the veterans from feature film visual effects and animation, and training and inspiring them to use this new technology.

Yovel: That's how we hire now, because it's the path we followed. We get more smart, amazing people from our backgrounds, and they approach the engine in a unique way.

Many of your staff artists come from a traditional VFX background. How has this impacted your adoption of Unreal Engine for your pipeline?

Dan: A lot of us are re-invigorated by experimenting with computer graphics again. It reminds me of the early days in visual effects in the 1990s. I've always found joy in trying to crack the code of getting to the visual that the director is asking for while maintaining a sensible, technical pipeline. Sometimes that means we have to bend Unreal Engine to our will, since the engine was originally built to make games and not necessarily linear content. That's all changed now since Unreal has made supersonic leaps on many of its tools. It's happening all over again; we're using most of our old feature film tricks, but now we're developing them to use Unreal Engine.

Yovel: The artists that we bring in often have a more traditional VFX background. When they dive into the tool and start to figure out how to use it, whether it's how to set up a Blueprint or a post-process material, they're approaching the learning process by trying to achieve a look that they used to arrive at in a different way.

Having that approach, from that traditional background, ends up pushing us into less common areas of the engine; it makes us ask for things that other types of users aren't asking about yet, because people haven't historically been using the engine in that way. What ends up happening is we might expose a lack of functionality, or sometimes just a different functionality during the image creation process.

Like Dan said, we're trying to bend it to our will. That mentality is consistently creating new avenues for development. We're developing a lot of our own tools, but Epic is constantly iterating, and we'll often see things that we're developing appearing as new features in the engine. Because of that collaborative space, and the fact that the code is open and available to everybody, it makes room for that type of workflow and philosophy.



What are some of the challenges of using a real-time engine for linear animation content?

Yovel: As we got deeper into using Unreal, we started asking Epic questions about certain things. A lot of the time, the response we would get is, “Nobody’s tried to use it like that.” That’s all changing now, as more and more studios are starting to adopt the engine, but early on, it felt like trailblazing.

Dan: The biggest challenge is preparing assets for Unreal Engine, or any game engine, versus other 3D packages. The modeling, rigging, skinning, and weighting all have to be built to work specifically for real-time playback. The upside is that the engine can have “smart assets” that can be programmed to have physics and move in certain ways procedurally. That’s a big payoff.

In the early days, I tried to use Unreal Engine as more of an offline renderer, a headless renderer. Traditionally, we’re used to looking at a frame and discussing motion blur or the effects of that output and image. Sometimes you have to rethink how you’re imagining the effect. We also used compositing for many years to address some of these issues, but nowadays we push ourselves to get everything rendered out of the engine.

It becomes a challenge, but one that makes us more creative.

Yovel: All the incredible advancements outweigh any of the confusion that comes with rethinking how you have to approach something. Anybody who wants to rethink their pipeline in this way, for animation or otherwise, needs to get over the fear of it being different.

If you want to make use of one of the most powerful tools that’s available to you now, then you need to just integrate it and see how you can work with it, or around it. We’ve taken a very specific approach to that process. It’s based on how we like to work, and our visual effects background, and everything that we’ve got in our history.

When we first started, we were basically doing everything outside the engine, in a DCC application, and then bringing it into Unreal and rendering it out. But the more you learn, the more you integrate and stay in the engine, you just don’t want to go back. You start to figure out how that works: Can you take this part of your pipeline and shove it inside the engine? The answer, for the most part, is yes.



Many of your works are stylized or have a 2D look to them. What is your approach to achieving this look in Unreal Engine?

Yovel: We had a client that wanted to make a cool VR animated series. They asked us if we could give it the look of these old-style background paintings, these Gouache paintings.

We quickly put together a scene with a street that we already had, and some trees we painted in with Quill. And then we built an Unreal shader that gave it this Gouache look with those characteristic outlines and a little bit of hand drawn waviness. It took about a week of messing around. You can just iterate and experiment incredibly fast.

Dan: When we showed Google how we could use Unreal Engine for Spotlight Stories, we took one of our existing projects, *Age of Sail*, and recreated one of its sequences in Unreal Engine.

Age of Sail is built in 3D, but has a hand-drawn, 2D animation look, achieved through shaders and lighting. There is a lot of photoreal rendering done with Unreal Engine, but it was an accomplishment working with our TDs to recreate the hand-drawn style from *Age of Sail* in Unreal.

There's a real excitement working in the engine. You're flying around and creating, feeling a little bit like you're playing a game while viewing your scene. From my years of compositing, I'm used to getting baked, rendered images to work with maybe once a day, but now I can work interactively in the engine to make changes in real time rather than waiting. Different disciplines are stepping on each other, but in a way that promotes more collaboration. You get to see animation, FX, lighting all at once to really judge how the shots are coming together faster than ever.





What advice would you give someone on how to get started with a real-time workflow?

Dan: It depends on what your end goals are. There is such an enormous wealth of materials and tutorials that are available directly from the Epic launcher or other sources online. Keeping in mind that we happen to be very cinematic-focused, I think learning Sequencer is a decent start. It's a way to visualize a scene cinematically and set parameters on objects and dig into what makes up elements in the engine.

Yovel: As Dan said, it's incredibly overwhelming to say, "I have to go learn this new software." It's too open-ended. I would say instead to decide to make something with it. If you want to start, then start. Make something small. Make it internal. Make it a project that you have coming up for someone else. Once you've decided what to make, your approach to actually learning how to do it will come naturally.

Doing it in Sequencer will teach you a great deal about the flow of creating a linear piece of content in Unreal Engine. Sequencer is a great place because it's so central. You have so many aspects of the engine that can funnel into it, and you can choose to put as much of that or as little of that in there as you need for the project.

Dan: Take a shot or something you did somewhere else, outside of Unreal and try to go through the steps and remake it in the engine. And either you really want the target to look the same at the end, or you make something super cool that's different.

But if that's not who you are, you could go to the Marketplace and grab something. There are so many incredible assets that you can learn from.

Materials and Blueprints are two examples of things that could intimidate you, but they're node-based, so you can follow the logic since they're visual. Say you grab something that has a Blueprint, like a mannequin puppeted by your game controller. You can open that Blueprint and pick it apart node by node.

If you're good at burrowing, you'll learn a lot. It might take you a few hours on your own, but that's part of the fun. That's how a lot of us spend our time these days.

Interview: Cory Strassburger | Xanadu

Cory started out as a VFX artist, working on such projects as *The X-Files* series and the film *Minority Report*. He eventually moved into virtual reality, creating VR projects for clients like GE, Universal Studios, and Lionsgate. His journey eventually led him to explore real-time performance capture with Unreal Engine. As part of the Kite & Lightning team, Cory went on to win the SIGGRAPH 2018 *Real-Time Live!* competition with a demonstration featuring the baby characters from the VR game *Bebylon*.

Xanadu is a one-man YouTube show starring Cory Strassburger's alien alter ego, Blu—who happens to be building a “virtual empire” from his corner of the metaverse. *Xanadu* is a unique fusion of virtual character, vlog, and comedy skit, with some Unreal Engine tips and tricks thrown in for good measure.

Besides Bebylon, what were some of your early projects in Unreal Engine?

My first motion graphics project using Unreal was in 2017, producing some animations for Hulu's promotion of the TV series *The Handmaid's Tale*.

Hulu had a campaign where the iconic red dress was blowing and burning, and that was the VFX task. Normally I would just do the model and cloth simulation in Maya and render it out, but instead, I did a cloth simulation with Marvelous Designer and exported it in Alembic format. Then I imported that into Unreal Engine, where I did all the texturing, lighting, and rendering, then exported the animation to After Effects to add the burning effect.

The huge benefit, obviously, was faster rendering. If I had done it the other way, Hulu would have gotten one 2K shot, because that's all that was within budget. But with Unreal Engine, I was able to give them a lot of variations, and I could give them 4K renders, and even 6K and close to 8K renders so they could crop in for additional shots. The animations ended up in promotional videos for the show, and some of the images ended up on billboards.

The fun really started to come in with some of the later ones I did for Hulu. They did a big *Game of Thrones* campaign with the map and banners for all the houses. They were looking for stills, or very minimal animation. Because I did it all in Unreal, I was able to get Marketplace assets for snow and ice, and did a full flythrough of all the banners with cloth sims. I did it super fast, and was able to give them these massive animations, all kinds of flythroughs with VFX and full depth of field.

One of the last things I did for Hulu was a few cinemagraphs for *The Handmaid's Tale*, where the main character has these huge fabric wings, and the final animation was shown on billboards in Times Square. I modeled the wings in Maya, and then brought them into Unreal Engine and did a cloth simulation there.

I was shocked at how fast the simulation worked, because those wings were pretty dense. Unreal handled it incredibly well.

And then came *Bebylon* and *Xanadu*.

I was actually already working on *Bebylon* when I did the motion graphics work for Hulu. Motion graphics was a fun thing to do on the side, but my real focus at the time was VR. My Unreal chops evolved through those early VR projects, and then *Bebylon* itself was a huge learning experience for working with characters and performance capture in Unreal Engine. And all this led right to *Xanadu*.



Image courtesy of Cory Strassburger

We would expect to see a series like *Xanadu* come out of a studio or a small team, not a single individual. That alone is pretty amazing.

I don't think *Xanadu* would be possible without Unreal Engine, not at the level of quality I'm going for, and doing it all myself.

The ecosystem around Unreal is a huge component of what makes *Xanadu* possible, because I can't model everything in the timeframes I've set for myself. The Unreal Marketplace is a huge asset for that part of the process. You need a new scene in the desert? Boom, download a desert scene, already lit, ready to go. You drop in your character, and you've got your scene.

There's a tropical scene in one episode of *Xanadu*, where Blu visits the dinosaurs. If I opened that huge environment in a DCC application...I'm not even sure I could open it there. And then if I had to light it and then render it? No way. So beyond the real-time rendering, it's just amazing that with Unreal Engine, you get access to these sets and worlds that you can drop your character into.

And that's another big advantage of real-time: with how fast things are, all I've got to do is just change it and then it changes across everything, and then I hit Play, and now it's there.

How long does *Xanadu* take to render per frame when you're finally going out to final?

Four or five frames per second. It's pretty fast. I could definitely jack up a lot more. I'm still using the old-school antialiasing, which is holding me back from getting the most quality out of the renders. But I like how fast it renders, and I'm fine with the quality. Obviously I want more, but I have to pick my battles.

My PC is a Falcon Northwest Talon, with an NVIDIA 3090 RTX for the graphics card.

I have an Xsens suit for body motion capture, and for facial capture I just use an iPhone and stream the capture

What would you tell someone who wants to create something like *Xanadu*?

Once you're ready to invest in a mocap suit, you have a few options. There's obviously Rokoko suits and

1. *Journal of Management Education*, 2000, 24(1), 1-10.

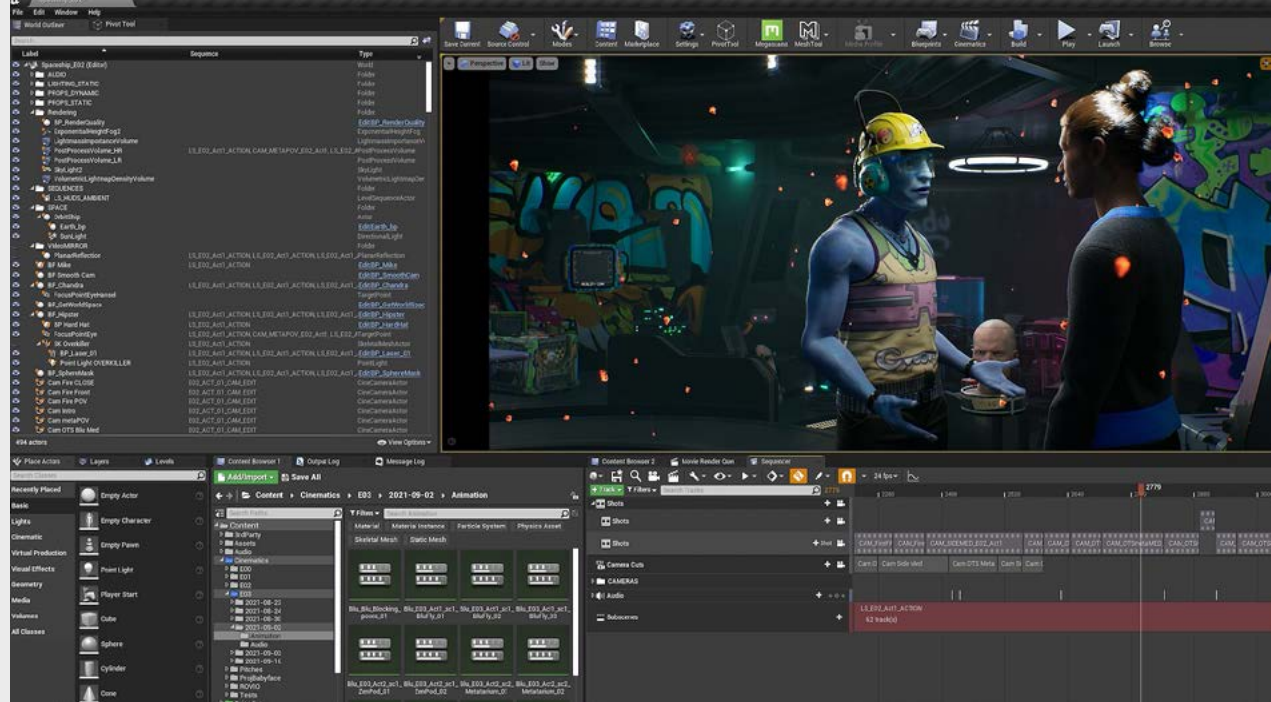


Image courtesy of Cory Strassburger



I'm a firm believer that if you have a good character and a good story, you don't need to hit a super high bar for motion quality. Your goal should be just to make people connect to whatever you're doing. I don't think you need all the super-expensive stuff to do that. I think you just need your imagination, and just enough Unreal chops to get the job done.



Image courtesy of Cory Strassburger

CHAPTER 3:

Unreal Engine Pipeline Framework





Throughout this guide, we have exposed the familiar elements of a real-time pipeline—layout, blocking, lighting, texturing, and special effects, for example—and discussed how they don't necessarily occur in the order you're accustomed to.

With a real-time workflow, the animation production process bears more resemblance to a filmmaking shoot with live actors. In a live shoot, the director sees the footage as it's happening, and can order new takes on the spot until it's right. The process with Unreal Engine is similar, but with the added benefit of rehearsing actors on the set while the Art Department is still tinkering with color schemes and the DP is zooming around the set to try new angles.

In animation, the “actors” are tireless, and always available. Your characters will do take after take all day, every day, without a break. You can “shoot” them with or without an environment, with variations in lighting and camera angles, with infinite combinations of wardrobe and props. And while you're shooting, you can review it and see if it works in real time. And unlike a shoot with live actors, you can come back to a shot several days later to find your set just as you left it and your actors still ready to perform, and you can try something new.

With a real-time workflow, instead of a department working on a step and handing it off to the next team down the line, teams can work in parallel, creating a new paradigm for bringing a project to final. Instead of each department finalizing its contribution and passing it on—and perhaps never seeing it again—the entire project is first created as a rough draft, and then polished over and over by all departments until it's done. Working in this way gives each department a complete view of the project while it's under development so they can more effectively contribute to the whole.

In a traditional workflow, real-time collaboration between departments is the exception rather than the rule. Suppose that in a traditional pipeline, the lighter is handed an animated shot, and the lighting doesn't quite work with the action. She could send the shot back to the animator with notes, causing a delay to the work in progress. Or she can try to make the lighting work somehow, which will likely give a less-than-best result. Neither one of these scenarios is ideal.

In a real-time pipeline where lighting starts upfront, an animator can finally create her shot in context to ensure a better performance. For example, if the action is driven by somber and moody lighting, or if they need to adjust a pose because a shadow is obscuring a critical expression, both the animator and lighter can be in the shot together, making tweaks until the shot works. And with video conferencing now commonplace among today's distributed workforce, collaboration and communication can take place regardless of where the artists are physically located.

In short, most of the jobs and departments needed in a traditional animation pipeline are still needed for a real-time workflow, and a few new ones as well. But the most significant change is the timing of their activities, with a great deal of overlap between steps that used to be sequential.

Real-Time versus Traditional Pipeline Steps

Using a real-time pipeline doesn't just mean adding real-time rendering to your existing linear pipeline—it means a shift in thinking about the pipeline as a whole. The steps of a traditional pipeline are all there, but they're now performed as stacked or circular tasks, where each department iterates during production, from beginning to end.

Let's consider some familiar pipeline steps, and look at how these steps translate to a real-time workflow.

Rendering

One major difference between real-time and traditional pipelines is render times. Because rendering can happen in as fast as a fraction of a second in a real-time pipeline, the long wait between hitting the Render button and seeing results is removed. You can now benefit from seeing final renders from as early as the story development phase of your production, not just at the end.



While real-time rendering is at the root of a real-time pipeline, it is by far not the only aspect of production affected by such a workflow. By reducing the cost of rendering, you effectively disrupt the entire linear pipeline, and enable meaningful iteration in layout, blocking, lighting, animation, composition, camera placement, and other aspects of production.

Pipeline Development and Scheduling

Just as with a traditional workflow, you need to invest in your pipeline to ensure smooth sailing through the project—or, at the very least, a not-too-choppy ride.

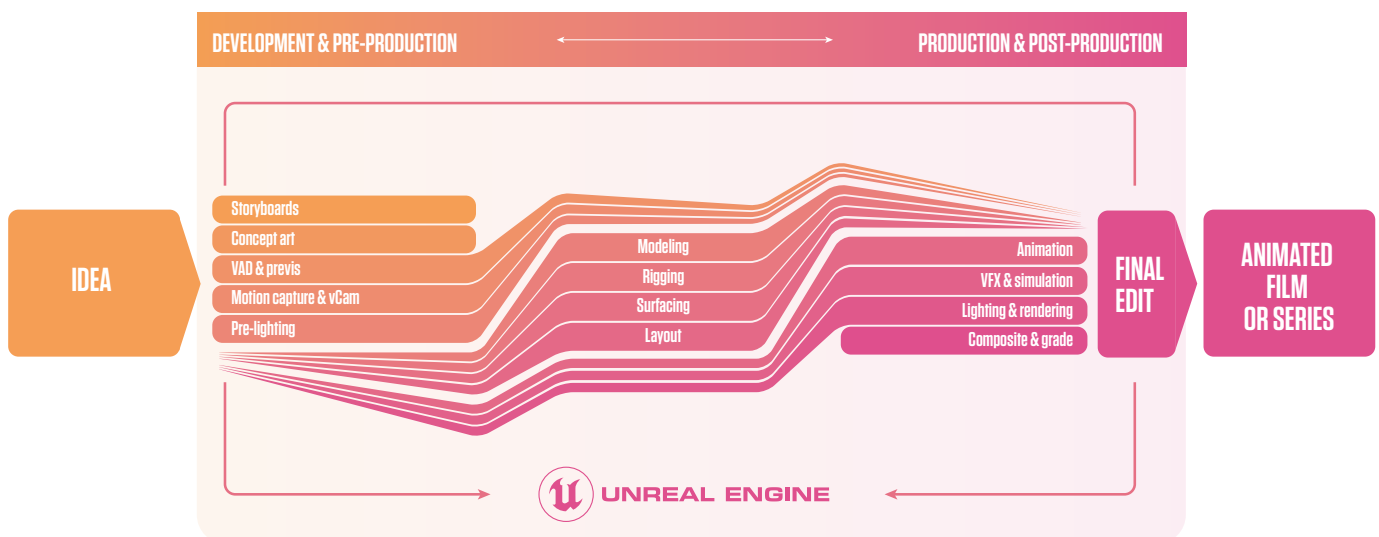
Unreal Engine’s open source architecture was designed to give you the flexibility to build your pipeline on top of, and integrated into, your real-time pipeline. As with all pipelines, it will continue to grow and change as each department comes to understand the pipeline and what they can get out of it.

The Pipeline team will help determine how shots will flow across departments, and work with producers to define how assets and shots will be tracked and reviewed. To help with production management, you

can find integrated plugins for industry standard tools like ShotGrid, which helps you schedule, track, and review projects.

The parallel workflows that result from a real-time pipeline naturally lead to a scheduling process that is very different from the traditional approach. Teams that are using a real-time pipeline are rethinking the entire scheduling and review process, and are becoming more flexible and agile. But in the process of accommodating overlapping tasks from different departments, the schedule still needs to account for familiar milestones for both clients and artists alike.

For example, DNEG Animation, the team behind Willam Joyce’s short film *Mr. Spam Gets a New Hat*, ran into an unforeseen issue with their real-time pipeline—their animators, unaccustomed to the real-time review process, were confused and even upset by the lack of definitive sign-off on their work. In response, DNEG reimaged their approval and feedback process and schedule just for the animation team, which helped the animators ease into this new way of working—they were able to enjoy the benefits of rapid iteration and feedback, but without the stress of wondering whether they were going to be continuously asked for changes.



The team member in charge of scheduling is often the first to really get how a real-time workflow affects nearly every aspect of the pipeline, and will need to work closely with each department to make sure they're on board with this new type of schedule.

Because everyone is collaborating on your project simultaneously, it is critical to ensure all files and versions are being managed properly. Source control has always been essential for game development, but is often unfamiliar for animation studios. We highly recommended that studios adopt one of the many version and source control products that integrate with Unreal Engine, such as Perforce and Subversion (SVN), into their pipeline.

Storyboards and Concept Art

In a traditional pipeline, the storyboard and concept art are 2D processes that use tools ranging from sticky notes and physical paintings to software like Toon Boom and Adobe Photoshop. While these traditional tools and processes can be powerful and expressive, there is an inevitable gap between the early 2D representation and how it will translate when recreated in 3D.

With a real-time pipeline, studios are augmenting these traditional workflows by merging 2D tools with Unreal Engine, which is creating more impactful visualization and opportunities for creativity and iteration at the earliest stages of story development.

One approach, for example, is to merge drawn characters with a rough 3D layout pass for environments, all in Unreal Engine. With this approach, backgrounds can be quickly kitbashed with Quixel Megascans and assets from the Unreal Engine Marketplace. With this basic version of the 3D world assembled, the director gets a true idea of the spaces and environments they have to work with, and has the freedom to experiment with camera angles, add new environmental elements, and consider composition. Then Story artists can use the plugin EPOS by Praxinos to combine hand-drawn story panels of character poses and action directly in engine.

“We did a rough layout pass with a version of the entire 15-minute piece that felt like previs—because the cameras are more dynamic, you can continue to add shape to it over time. Much like a painting, you layer in more detail as you go.”

Doug Oddy

Senior VFX Producer
Sony Pictures Imageworks

Layout, Look Development, and Visualization

In a traditional pipeline, layout, look development, and visualization happen separately. The teams might share assets, but each one follows its own track toward completion. In addition, the assets and animation, developed through a great deal of time and effort, are used only as visual reference, and are otherwise discarded.

With a real-time workflow, all three steps can happen in parallel in the engine, with no waste. Layout can start with rough assets to give the director a world to explore, and the rest of the team a space to work with. While Set Design, Lighting, and Animation add and refine their own elements, Layout replaces assets as they're developed, eventually resulting in a fully realized world. In essence, layout starts when the project starts and finishes when the project is nearly complete, and facilitates a parallel workflow for nearly every department.



When developing the look for animated content, it's important to be able to try different materials, lighting, and filters until the details look just right. Playing with colors and textures, trying out plugins, testing out hair/fur settings, foliage generation, and particle systems—it's all part of the process.

When you do look development with Unreal Engine's comprehensive Material Editor, lights, and other tools, all the steps you used to get the look can flow seamlessly into previs and production, where the look can be automatically applied to all the applicable sequences.

Instead of a typical previs process where all the hard work is thrown out as it moves into production, this new visualization workflow in Unreal Engine means productions can continue to build on top and update and iterate. The scenes can move seamlessly into later stages like layout, animation, and lighting, with assets and performances being updated as they are completed.

Working in this way also gives the lighting, surfacing, and VFX teams the opportunity to give feedback on how well the assets are working out, which the asset team can then incorporate into the next round of refining the assets. The end result is final assets that are in perfect harmony with these other aspects of production.

“We have a stylized shader in Unreal Engine that makes 3D look hand-drawn, so we produced a video in that style and sent it to the client, and the client said it would be cool if the camera moved this or that way. We sent them a new video within an hour, and they were blown away.”

Martin Walker

Chief Technology Officer
Digital Dimension Entertainment Group



Image courtesy of Digital Dimension Entertainment Group

Interview: DNEG Animation | Mr. Spam Gets a New Hat | Taylor Moll and Shelley Smith

Taylor Moll

VFX Supervisor

Shelley Smith

Producer

Taylor Moll has over a decade of animation experience, working on projects like *Ron's Gone Wrong*, *Happy Feet*, and *Kung Fu Panda 2*. Shelley Smith has worked on various projects such as *Tales from the Radiator Springs*—a mini-series featuring Pixar's famous *Cars* characters—to DNEG's first foray into real-time animation.

Mr. Spam Gets a New Hat is a short film written and directed by William Joyce, the same artist who brought us the Academy Award-winning short *The Fantastic Flying Books of Mr. Morris Lessmore*. DNEG collaborated with Joyce to produce *Mr. Spam* as its first Unreal Engine project.

What are some of the biggest challenges of using a real-time pipeline?

Taylor: Figuring out how to manage newly overlapping departments and how that works with a schedule. Once we did, everyone was in the scenes together making decisions; we actually got pretty savvy by the end of it. We were so much more informed, too. We knew what questions to ask the director much earlier in the process.

Shelley: Real-time really changes the whole dynamic. Editorial had to work much closer with Layout and Animation on a daily basis. Typically, Editorial receives everything at the end, but they had to be in there to know what they were going to receive because things could sometimes change hourly at every review. The impact of everything is so immediate that everyone always has to be in the loop; it's not a hands-off pipeline anymore, we're all on the same team together now. But that immediacy, especially around changes, was great because we could move things forward much quicker, and the collaborative aspects were really exciting.

How did you help Editorial adjust to the new workflow?

Taylor: One of the things that we're working on is an EDL [edit decision list] interface and workflow in and out of Unreal Engine. That'll be huge going forward because we want Editorial to be able to work seamlessly alongside Layout. Changing shot order, cutting shots, adding shots midway—these are typically editorial decisions. Due to the nature of working in the package, many shot updates happen within Layout now, so when Layout makes an adjustment, or an alternate, this needs to be fed back to your editor to control the final cut. Edit and Layout have to be in the same discussions. And when the process is no longer linear, you need tools to accommodate all the collaboration and feedback. So we're writing some of those tools now.

How did the new workflow change your approval process with the director?

Taylor: "What do we show, and when?" was an ongoing topic for Shelley and me, because while everyone might be in the review, every team isn't always at the same place in their design. So you might have inconsistent lighting in some places—because it isn't done—and yet, your animation and final camera is there.



So that's the conundrum. Is the director going to be confused that each shot is lit differently? Sometimes we'd have to caveat a shot that might be too distracting, and sometimes we'd show the same shots with two different contexts asking the director to just talk about the camera, or the surfacing relationship between a character and the environment.

Shelley: This probably depends on the director, their familiarity with the technology, and your ability to provide context into what they're going to see, but the animation was the spot where we noticed the most distractions. He wasn't sure what he should be focusing on. We solved that by pulling that part out in Maya for review, and bringing it back into Unreal again when we got to lighting reviews.

How did the artists adapt to the review process?

Shelley: The ability to always be changing things, or do that together, can be a little confusing for some, so we still had to make milestones about when things are done and when things are approved.

Taylor: The one thing that became really apparent to us is the importance of that to an artist. That artist needs to know when a task is done.

Shelley: The animation team came to us at one point and said, "We don't know where our shots are at with the director." That was a big reality check for us. We might be saying, "it's ready for lighting," but for animators, that doesn't mean it's done. And that makes them unsure about whether everything is going to come back for a redo. They need those milestones so they don't feel a massive panic coming on.

Taylor: That was our mistake, not recognizing that in the teams, so Shelley and Dave Lowry, our animation director, just swooped in and said, "Ok, we'll get Bill to buy off on placement and staging. Anything that comes back at this point should be an expression. If that happens, we'd classify it as a redo." This really spoke to the animators and made a huge difference in morale. You need those small victories, those things you can check off. You can't just wait for the project to be finished.

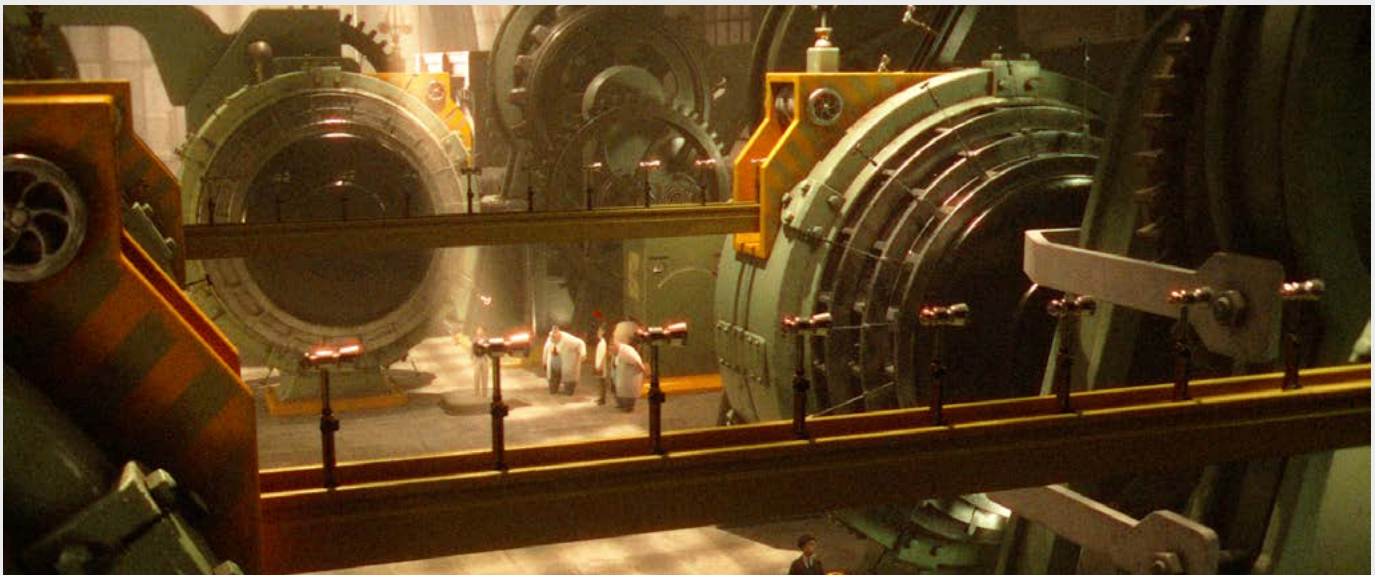


Image courtesy of DNEG Animation and Howdybot Productions



What does real-time rendering lend to the storytelling process?

Shelley: Many things, but one that stands out is that you can start location scouting really early on with the camera and layout, which lets you start informing the story way earlier. Usually you're handcuffed at a certain point, but with real-time, Bill could say, "You know what? I want this type of shot," and we'd send it to Story and they'd draw up the boards.

Taylor: We had really interesting feedback from the story artists. Some of them thought this approach to storyboarding was brilliant and amazing; others felt chained down, and like they had to design boards to match a camera. They wanted the freedom to pitch their story ideas with the same freedom we felt scouting the area. Which is fair, and part of what you learn when you work through a new process.

Shelley: One artist just loved it, though. They said it felt like they had their perimeters and could just focus on making the acting work with the boards. Ultimately, Unreal Engine is such a useful tool if you want to use it. So while everyone might not need to use it, it certainly opens up processes that were never available before.

So if you can just keep going and going with your edits, how do you decide when to stop?

Taylor: Unreal is a boundary-free zone. You can do almost anything in it if you wanted. But at some point, you have to draw the line. So that became a challenge of, how do we define that in an application that defies that?

In a traditional pipeline, you'd say, "That one asset looks great. Let's put it up in front of the director." Or you might do that with the lighting for a particular sequence. But with everything in Unreal, we're showing the director multiple things simultaneously.

It was a paradigm shift. We're not showing an asset or the sequence lighting—we're showing the *sequence*. We're not showing key shots—we're showing the animation with the lighting *and* the new camera.

Our director said that it's amazing how quickly you get used to the new things and want more. We reached a point where he wanted to see all the shots around the shot, and we had to tell him, "We don't have that right now; we just wanted you to sign off on this one thing." But now he wants to see the whole sequence to see how it plays. You get used to this dark art, this black magic, just working. Ultimately, we had to change the expectations of what the next set of reviews would be. "Organic process" doesn't even begin to describe what Unreal Engine does to your brain.

So how can studios tell if Unreal Engine is right for them?

Taylor: When this project started, I came at game engines like a skeptic. I had worked at a video game company a long while back and thought, "I don't like this, I'll never do it again." My CG Supervisor, on the other hand, thought it would be great. And ultimately, it was. At this point, I'm very pro real-time as long as you're using it for a real need. Don't do it because it's the cool new thing. You have to invest in this, and get on board with it. It takes time, personnel, and training to wrap your head around. It's a paradigm shift, but if you can do that, it's absolutely worth it.

Interview: Chromosphere Studio | Yuki 7, Mall Stories | Theresa Latzko and Andrew Wilson

Theresa Latzko

Technical Art Director

Andrew Wilson

3D Artist

Theresa Latzko is a technical jack-of-all-trades who helps Chromosphere come up with stylized real-time looks and solve difficult problems for the studio's more technically challenging projects. This has included time on the Netflix show *City of Ghosts*, Google Spotlight's VR shot *Sonaria*, and Lyft's *June*. Andrew Wilson is a 3D generalist who excels at asset creation, texturing, lighting, and compositing for commercial and television productions.

Chromosphere has traditionally used a lot of post-processing to achieve a stylized look. How did you convert those processes to Unreal Engine?

Theresa: We started with a little research. We got a pretty detailed breakdown from our compositing artist covering his thoughts going into a scene and how different effects typically get applied to different parts of the picture. Algorithmic breakdowns are always the easiest thing to emulate when you're working with computers, so I just tried to emulate his steps in the tools I was creating.

For example, the compositing process uses a lot of these kinds of retro patterns—such as stripes across the screen, which are applied differently based on the lighting information. Something that's lit requires a different pattern than the shadows. So what I ended up doing is rebuilding part of our compositing process within the post-processing materials in Unreal Engine.

We also had a really good time just moving effects in wholesale. Splashes, particles, they just worked in Unreal, which wasn't our experience with other real-time engines. We ended up doing everything in Niagara for these projects. It's just been very flexible, and a lot of fun to learn.

How did the development of the *Yuki 7* pipeline differ from more traditional ones you've worked on?

Theresa: We really had to figure out what worked best in engine and what should stay outside it. Part of it is also figuring out how things are going to work. For instance, we ended up doing layout in Unreal Engine. We were planning out this big boat chase, and we felt pretty early on that it would be good to take advantage of a connected setup where we could just lay it all out in Sequencer, sort of like a theme park ride that's all in one piece.

Andrew: It's pretty different from a traditional pipeline, because lighting and compositing are happening in real time; it's definitely a lot more engaging. You don't have to wait for frames to render, and having instant feedback makes it more fun to work with.

The only issue is that in order to get the compositing working properly, you have to make the tools yourself. But the fact that you can composite it right there in Unreal Engine on top of your lighting helps you iterate fast—it's definitely helpful.



Theresa: It helps with feedback, too, because we could have review sessions where the art director gives instructions in real time. “Can you make this a little bit brighter?” And you can, and they can see it and let you know if it works. As opposed to, “I’ll get back to you in a week and we’ll see if you like it then.”

Andrew: In other pipelines, just waiting for renders would destroy my focus, because I have to do something else while I wait. Then you come back and sometimes can’t remember what you were trying to do in the first place.

Theresa: Upside, no more render breaks. Downside, no more render breaks.

Andrew: And at this point, the project has totally moved to 3D. In the first two episodes, a lot of the backgrounds were still 2D, which means you can’t change the camera. But working in Unreal Engine gave us more compositional freedom to make changes, even late in the process. It’s no longer like a video game, where if you go outside the bounds of the map, things start to fall apart.

Theresa: We can push things around pretty easily now. It used to be that if you were rendering out a shot and had the compositor comp it, changing the camera angle just 10 degrees would be a no-go. But now, we have finished lighting, finished effects, finished everything. And if you want to adjust the camera a little bit—it’s just fine.



Part of this project has been about tool creation and tweaking things to fit your needs. Can you take us through the process?

Theresa: Here's one example off the bat: the water work. The water surfaces you see, especially in the third episode, are based off a single shader. We decided early on we weren't going to attempt complex movement for the water geometry because it's a pretty simple style of animation. The shader itself ended up being pretty robust, though, and helped us take various elements we found in the material library and assign them to different layers of water depth. It almost created a particle effect, and it's happening in a single shader. That instant feedback was so helpful, too. You don't usually get that, and it was very satisfying to see everything develop right in front of me as I changed and tweaked parameters. It felt like I was creating art in real time.

On the rigging side, we had to come up with a solution because our rigs weren't set up for real-time. We got around it by converting different deformer into joint-based solutions. Some of them were efficient, and some of them weren't. We had an issue because our character's eyes are like 2D disks, and in our previous process there was a special deformer that made sure it conformed to the shape of the eye. There was no other way of really recreating that without just rigging it completely differently. We just took the deformation, made a joint for every vertex, and then baked that.

Andrew: One day, we'd love to fully rig our characters in Unreal, and we might once Epic supports adding joints and paint weighting. That'd make a big difference because that's most of what our rigging is.

Theresa: In terms of the future, something I'm personally interested in is the kind of modular way of asset creation. We didn't really get a chance to do much R&D on this, but right now we're building this big mall and there's a lot of opportunity to assist with objects like storefronts and kiosks, all of which are similar but slightly different. With Blueprints, you have very simple tools for automating these kinds of variations between objects, and I think that's pretty interesting. We might explore that for future projects.

When you think about what's different now, what really stands out to you?

Theresa: We build places instead of shots now. You do that and just make sure it all fits together, that all the scales make sense. Then we find the shots within that. That's a pretty different approach from what we are used to.

It's almost like you are creating this one coherent unit that, in our case, happens to be in the shape of a mall. You can go and look at the various parts of it, and they look like you'd expect that part of the mall to look. In the past, we'd design the one area of the mall we were trying to make the film about, and then we might have a second set for a bunch of background stories destined for another shot. Everything is laid out in a way that resembles live action, like you're on location.



Interview: Engine House | Wylder | Natasha Price and Mike Richter

Natasha Price

Producer

Mike Richter

Studio Director

Engine House is a UK-based, award-winning animation studio: a team of animation directors and producers with a shared vision of telling meaningful stories with artistic flair. Having moved their production pipeline to real-time, the team was awarded an Epic MegaGrant in 2020 for its slate of film and TV projects being developed in Unreal Engine.

We profiled Engine House on our blog for the team's work on *Wylder*, an animated short based on the children's book of the same name. Natasha "Tash" Price, the team's producer, kept a production diary of Engine House's work on *Wylder* to offer insights into their workflow. Here we talk with Tash and Mike Richter about *Wylder* and other real-time projects Engine House has taken on.

Have you done any more real-time projects since *Wylder*?

Tash: We've moved almost all of our work over to Unreal in the last year or so. What we did with *Wylder* is, we made it look different from a lot of things that you would expect to come out of a game engine. We went for something that's quite illustrated-looking.

We've also started a project called *Petropolis*, a metaverse project with pets. We're making the 60-second teaser for that, showing you the world you can live in and all the different things you can do.

Wylder led to some of this work, in a way. In general, the hair that artists render in Unreal Engine is more realistic, but with *Wylder* we went with a more illustrated look. So now we're chucking loads of fluffy fur on cats and dogs for *Petropolis*. The two projects have entirely different styles, in that *Wylder* looks 3D, but *Petropolis* is intentionally made to look flat and hand-drawn, with a focus on interactivity.

So now you're making interactive content, too?

Mike: Working on film and animation projects means we focus on art first and interactivity second. It's novel to be able to do this in a game engine where the focus is typically the other way around; clients love that we can concentrate on getting things looking amazing, and then build interactivity around that. A games company might make the interactivity really well, but then it doesn't look good. This is why we got contacted to work on interactive content—they wanted to start with people who can make the artwork first.

Going the other way, there's a client that we do work for every now and then, making trailers for computer games. They've developed a game, they've got a huge amount of assets, and they want us to pull assets out of the build and do cinematics.

Tash: We know the engine. We know what we can make in it. We know what we can get through in the given time. We tell them to go with a real-time workflow because it's a better decision. Knowing Unreal Engine and knowing what's possible, that helps us with a really quick turnaround, like six weeks in some cases.



Image courtesy of Engine House



Mike: Part of it is that the pipeline is really easy to mix and match per project. At the moment, we're doing a character a day, which could be just finding something online that we can buy that's at least a good starting point, tweak it, add stuff to it, make it look good. Then do the groom, rig it, and get it over to the animators.

How does using a real-time pipeline affect schedule and budget?

Tash: It's way more efficient, and quicker. But you can make the decision about whether you want to use Unreal to save money or to make it better, because it does both.

On *Wylder* there was a lot of look dev, and I was sending Mike lots of thoughts and new ideas at once, each one quite different. That process was so quick. We managed to turn the project around quickly, and we were really pleased with how it looked.

It was quicker, and therefore cheaper, but also we managed to do so much more with it. We kept saying, "Can you imagine if we had to render this out?" It probably would have been impossible.

How are the production roles split up on your team?

Mike: We're all somewhat responsible for the whole thing, as opposed to, "So and so's doing the modeling, and then it goes to rigging, and then it goes to light." It doesn't really go through that process. It's all happening at once. In that way, it's very much like game development.

With the pipeline in Unreal, there's a lot more of an initial load of work to be done—good practices like making sure everything's named and it's the right scale and that the materials are there, because it's not staying in Maya or 3ds Max where you can name it later.

It's more like, this is getting exported to the engine now, and it needs to be good. We'd get error messages on import to Unreal because the model wasn't to scale or there were split vertices, things that don't matter when you stay in 3ds Max or Maya. And there's a limitation on how many bones you can skin to. It makes you really tidy up everything.

What inspired you to try out a real-time pipeline?

Mike: That *Fortnite* cinematic trailer that came out a few years ago with loads of Ryan Brucks's cloud effects and smoke monsters. It was completely bonkers. We wondered, how is that real-time? We just have to do that.

Every now and then, we'll be looking at something and say, "How is this real-time?" You just get in and do it, and it's done.

Have you been using Unreal Engine for other types of projects?

Mike: There was one job that was these big robots with eight main characters, and the client wanted each one in eight different poses, 64 images. If I had rendered this in a DCC app, I'd have 64 files, each with its own iteration. Then if they said, "Can we change the lighting?" it would have been a nightmare.

Instead, we did it in Unreal Engine as one sequence, 64 frames long. When it came time to render, half a second later, it was done.

I realized that it's the organization at the beginning that takes longer. But it means that it maybe only takes twice as long for that beginning bit, but then the middle and end sections are done in a ridiculous fraction of the time. And that makes us happy because those are the bits that are so painful and boring.

Tash: Unreal Engine makes the creative process more accessible. It's a lot more like a live-action movie shoot.

I did a film production degree, and I don't work hands-on on the animation. The DCC app we were using feels quite scientific, whereas Unreal feels like you're in a scene. You see where the camera is. For a director or an art director who's not necessarily hands-on, it makes so much more sense, and it's more readable as well.

What are some of the other differences between a real-time pipeline and the traditional?

Mike: We've not quite shaken off the prior feeling of when we get towards the end of a project and the client says, "The curtains, can they be blue instead?" Our first reaction is still something like, "You signed off this room two months ago!" and getting annoyed with them. And then a minute later, you say, "Oh, actually, it's fine. We can do that." It is fine, but I'm not used to it being fine.

Tash: With the traditional way, quite often there would be some feedback right at the end, and we'd say, "Well, we can do it, but it will be another week." We don't have to say that anymore as you can make the changes directly into the final render.

I think we probably could have done *Wylder* without Unreal, but it would have suffered from that lack of being able to be creative the whole way through. And it also would have taken us longer.

Where do you see Engine House going next?

Mike: Besides all the projects we get in, we really hope more studios start using this pipeline, and we hope we can help some of them get there. We'd love to consult with other studios on best practices, and how to really take advantage of the pipeline to be more creative and save time.

We spoke to this university student recently, and he said his animation course and game development course are overlapping more and more. At some point we have to ask, "Why are there two separate courses?" It should be one. The question will really be, "How do we bridge both worlds?"

Tash: A new consideration is bringing interactivity to film, and it's making film and also games more cinematic, bringing both closer together.



Interview: Sony Pictures Imageworks | In Vaulted Halls Entombed | Jeremy Sikorski

Jeremy Sikorski

Asset Supervisor

Jeremy Sikorski's credits as a CG and Asset Supervisor include *Spider-Man: Far From Home*; *Men in Black: International*; *Lucky 13*; *The Angry Birds Movie 2*; *Storks*; *Hotel Transylvania*; and of course, *In Vaulted Halls Entombed* for the Netflix series *Love, Death + Robots*. Jeremy specializes in producing photoreal CG assets, and is never happier than when he can fool a supervisor into thinking it's the real thing.

In Vaulted Halls Entombed, an episode of Netflix series *Love, Death + Robots*, centers on a squad of soldiers that find more than they bargain for as they venture underground on the trail of runaway terrorists. It also was the cast-off point for Sony Pictures Imageworks' first venture into real-time animation, a move that saw them remaking their traditional pipeline from scratch.

What is Sony Pictures Imageworks' traditional pipeline like?

Our work runs the gamut from visual effects blockbusters like *Spider-Man: Far From Home* and *Jumanji* to more stylized projects like *Spider-Man: Into the Spider-Verse*. Because of this, we've had to build a really robust pipeline that can support both of these types of projects in the same space, which is really unique for our industry.

What we've built can run five to 10 projects concurrently, with timelines ranging from three-year CG-feature animation projects to six-month, 100-shot endeavors for a visual effects film. So it's a well-oiled machine.

What prompted the jump into real-time animation?

We were asked to submit a short for the new season of *Love, Death + Robots*, and we decided to use it as a testing ground for real-time rendering and Unreal Engine. We also wanted to do as much as possible in engine to really put it through its paces and learn as much as we could.

How did you choose your crew?

Our crew was assembled based on people's interest. We sent out a poll to 900 SPI artists and said, "Who wants to work on a real-time project?"

It's interesting when you build teams based on how passionate people are about a topic. When people are genuinely interested in the technology and the process, and bring a wealth of animation visual effects experience to it, you can totally flip the script on how outside entities or even other studios say you should do things. It was a very exciting group to work with.

On my end, I was the Asset Supervisor and Art Director on *In Vaulted Halls Entombed*, so I oversaw modeling, texturing, shading, rigging, and groom, and provided concept art, as well. Towards the end, I even jumped in as a shot lighter.

The fact that one person can do all that is a real testament to Unreal Engine and its toolset. In our traditional pipeline, we are divided into departments and have a very strict hand-off process between departments. It's very regimented, and it's the reason we can work on 10 animated CG features across 900 artists at any given time.

But the exciting thing about Unreal is it promotes on-the-fly, artist-driven creativity. With Unreal, a high-fidelity story can come together in a matter of days or weeks. We are used to waiting months or years to see our first visual, so this is a huge shift in the way movies, animation, commercials, basically everything, will be produced in the future. It's very exciting, and you can see how much enjoyment artists get out of the process.

When they see their story play out in front of them so fast, and still have the ability to change a camera, change a light, turn a character around, make all these small adjustments, which normally need to get sent to another artist via a note system...it's exhilarating.

How did you start incorporating Unreal Engine into your pipeline?

When we start a new film project, animated feature, or hire a new crew, we do something at Sony called "running the pipeline," where we select a one-shot or sequence of shots—usually throwaway work—and go step-by-step through how it would be touched by different departments and key artists. I was in charge of this for *In Vaulted Halls Entombed*.

This was a great discovery period for me. Before this project, I had a basic grasp of the differences between asset-building for real-time versus offline rendering, but to actually take something to the final quality that we were aiming for—that's a very different experience. Through that, you learn all of the little fine details that you need to be paying attention to, where things work, where they don't.

There were portions of the pipeline that remained largely unchanged, like modeling. We still use ZBrush, Mudbox, and Maya to build our models. Our texturing process, which is based on Substance Painter and Substance Designer, was largely consistent as well. It just became a process of figuring out how we get those into the engine.

The big challenges were in rigging. In Unreal Engine 4.25, which is what we used for the project, we were pretty limited in how we could move rigs back and forth.

A lot of us are patiently waiting for, and trying to figure out, the best way to animate in real time. This is totally worth pursuing because it opens up a whole slew of opportunities like animating with artificial intelligence, or using complicated deformers with AI or on the GPU. There's just so many interesting avenues to explore that are helping us to think differently about how we create performances.

How did you handle motion capture?

At its most basic level, we are doing mocap into the engine, sending it out to Maya, then bringing the Alembic cache back into Unreal.

Getting there was interesting, though. When we started this project, it was April 2020—the middle of the pandemic, during lockdown. We'd watch mocap sessions over Zoom, trying to direct and get information to the limited number of people that were allowed on the stage.



Most times, we didn't have our shots completely defined before we went into a shoot, but we knew the general action that needed to take place. This was a great opportunity to experiment with real-time mocap—you can capture sequence-level animation and then tell your story through the cameras and lenses after the fact.

We captured a lot of lens information while we were capturing the mocap. We'd clean up the results in Maya and add or change whatever we needed to, then push the FBX rig data back to the engine.

The only major challenge was the faces. Our faces were captured using DI4D, and we built the whole rig system around that outside of Unreal, and then pushed Alembic caches into it for our faces. There was a complication in making sure that our bodies, heads, and necklines all lined up, matched, and stayed together, and we definitely ran into our fair share of problems with that. But at the time, that was our only way forward if we wanted to hit the quality levels we were looking for.

We also tried MetaHumans on three background characters. These were animated with the mocap going directly onto the MetaHuman rig.

Since real-time mocap sessions are a little more free-flowing, did you have any “happy accidents?”

Definitely, and a lot of it comes down to having more cameras available at any given time. In our traditional animation pipeline, we're always changing cameras. When you do this late in production, though, those changes have a cost.

The same is not true for real-time mocap. When you have a plethora of cameras shooting the action from a variety of angles as the sequence plays out, you get those happy accidents. You also get the freedom to just go into the shoot with a script, a shot list, and a general idea of what you want, and see what happens. Sometimes what you didn't intend to happen works way better than what you originally intended. With real-time, you can make those on-the-fly decisions in a way that can't be replicated by traditional toolsets.

The big thing for us is moving that faster decision-making process forward as much as possible, so we can create a better story. Unreal Engine lets us visualize our ideas in a more complete form—it's really hard to get the big picture of your story when you're looking at play-blasted gray renders, with things popping around and not looking right.

The power of a real-time rendering is that you have the ability to create, craft, and see that story in a new way early on, so that you can iterate and make better decisions, and ultimately create better films earlier.

Since you have a background in texturing and shaders, how did Unreal Engine apply to that part of the process?

I was personally responsible for creating our master primary shaders and defining that whole setup for *In Vaulted Halls Entombed*, and I think between myself and one other texture painter, we had everything covered—which is crazy. That's unheard of. And that's just a testament to the kind of speed that you're able to work and get feedback on everything.

Our texture painter, Jessie, was texturing all of the outfits and clothes for our characters and some of the props, and then I would take them and bring them right into the engine. Neither of us had much experience with Unreal Engine, so this was a great learning process for us both.





If we sacrificed a few frames per second during the process, we didn't care because we were rendering final pixels at the end of the day. You want to make sure that you're working as fast as possible, but when we flipped on all our complex shading, and SSS profiles, and multiple layers of shaders, as long as we were happy with what we were seeing visually, the playback speed didn't matter.

We definitely had shots where we couldn't hit 24 frames per second. So there was a bit of working backwards, too—finding the problem areas, putting some of those beefier features of the shaders and rendering behind some switches and toggles. Setting that up took a little bit of time because of the learning curve. The results were worth it, though.

How many people were on your team?

Our modeling team was two character artists, two environment artists, one texture painter, and myself as the shading artist. We had four to five lighters who also helped create character Blueprints. There was a lot of setup work required, and our lighting artists really liked working with that kind of stuff, so they really dove in and got all nitty-gritty in some of the Blueprints

Our effects team was four. We probably have six animators at any given point.

We had a really big ramp-up in the beginning, but our crew definitely slowly started to slim down as we got to the final look. After we got the bulk of that asset work and animation cleanup done, it was a very small group that took it over the finish line. And we were doing everything—camera tweaks, lighting tweaks, character position tweaks.

The only department we created for this project was what we called Shot Finaling. Those artists were responsible for ingesting our caches and the FBX state of our animation, snapping our grooms to our humanoids, compiling all the finalized data, animation simulations, effects simulations, and all our caches. Basically, they made sure everything worked. Toward the end of the project, we hired four or five people to come in and do that.

What do you think about the future of real-time pipelines?

Using Unreal Engine for *In Vaulted Halls Entombed* has made us rethink our pipeline on a micro and macro level, and we're sure it will be a big part of wherever we go. There's no looking back.

Going forward, it's about figuring out how we button up the workflows to make sure they're rock-solid. Our traditional offline pipeline is 25 years in the making; it's definitely rock-solid, but it's not taking advantage of what's available now.

Technology, hardware, and software have moved so rapidly in the last 5-10 years, and everyone should be looking at what's available. It's only a matter of time before our GPUs and computers can deliver super-high image quality and a super-high frame rate operating concurrently.

For now, there are things within the real-time space that we can start adopting: the front end, shot creation and layout, and preliminary visdev of those shots—basic lighting, basic materials. It's already more than we can do right now in our normal traditional pipeline, which means it's absolutely worth taking advantage of.

Animation

Prior to Unreal Engine 5, many teams chose to animate in a DCC application rather than in the engine, and then import the animation to Unreal Engine via the FBX or Alembic formats. With the production-readiness of Control Rig in Unreal Engine 5, animators now have the option to keyframe directly in the engine for faster iteration during review, or do nondestructive animation fixes on imported keyframe animation.

The field of virtual production has made great use of real-time motion capture via the Live Link plugin for Unreal Engine and the Live Link Face app for iOS. With this technology, mocap data can be retargeted to the 3D character in real time, giving the director the full context of the performance in the scene. Better yet, seeing the mocap data live often sparks an idea for experimenting with action and camera angles on the fly. In addition, when an actor can see themselves as the 3D character they embody in the 3D environment in real-time, it naturally leads to a better performance. You now have the opportunity to bring the spontaneity and creativity found on a live-action shoot to an animated production.

Neither workflow is mutually exclusive; you can explore and find what the right combination of keyframe and motion capture workflows are best suited for your production in order to create the best performances possible to tell your story.

Shot Setup and Review

In a traditional pipeline, each shot is saved in a separate DCC file. When a shot is reviewed, notes are given for that shot, and go back to the team responsible for whatever aspect needs to be changed—assets, materials, lighting, animation, and so on.

In a real-time pipeline, shot setup and review aren't limited to a single shot at a time. In a game engine, each level is a set or location and becomes the

foundation of your sequences and shots, which now live in a single project. You can now set up, edit, and review multiple sequences in the same way that one might shoot several scenes at one location in a live-action shoot. With Unreal Engine's collaborative environment, everyone—the director, set designers, DPs, lighters, animators, VFX artists—can “get on set” all at once, and work together to polish and shine the sequence until it's perfect.

With this workflow, teams can work in context across sequences to test variations and changes. A lengthy note on a change to a shot can go from “difficult to explain” and potentially misinterpreted by an artist, to “Let's work it out while we're on this call together.”

“It's a different approach because you're not working on shots. You're building a world.”

Doug Oddy

Senior VFX Producer,
Sony Pictures Imageworks

By using Unreal Engine as the central hub for all assets, animation, lighting, cameras, and VFX, every member of your team, from directors to supervisors to artists, can watch the project come together as each piece is added or edited. Such a workflow naturally enhances your team's creativity and storytelling, as artists are able to contribute directly to the process in real time.

And because each virtual set is used for all sequences in that environment, any changes to the set or character automatically propagate through all the shots, and appear in the rendered output. Conversely, in a shot-based workflow, a significant amount of effort is required to ensure all shots across a sequence are updated with the most relevant assets and positioning in a shot.



Visual Effects

In a traditional pipeline, 2D visual effects are added during comp, and 3D ones after final animation, with things like lens flares, water splashes, dust, fire, smoke, sparks, halos, and other effects layered into the process. Because effects happen so late—and the VFX department is often siloed from previous steps in the pipeline—as a director, you often have to hope your vision is executed and the style is right, but it’s difficult to know for sure until you get there. And if an effect isn’t working due to the way an element was set up in a previous step, fixing the problem means a costly trip back down to an earlier part of the pipeline.

With a real-time workflow, visual effects can be created right in the engine alongside the sequences and shots, and VFX artists can start testing effects as early as layout. Once the DP sets camera angles for a sequence, the VFX team can check, in real time, whether an effect is working and what it will look like, and can communicate with other departments to tweak the scene for the best representation.

“To be able to see exactly how a lens flare is going to animate while you’re dialing in a camera, or to be able to swap in updated surfacing of an environment as you’re working out the blocking of a scene...it’s a real game changer.”

Jason Greenblum

VFX Supervisor
Sony Pictures Imageworks

Compositing

In a traditional pipeline, a common workflow is to output render passes, a series of images for each frame that contain channel values for that frame. Render passes are commonly used to control a particular visual aspect of the scene, such as the darkness of shadows or the brightness of a reflection, or to aid in production of visual effects by providing depth cues or transparency information. Once all the desired changes have been made, the passes are composited (layered) to create the final image.

A real-time pipeline doesn’t really have an equivalent for compositing, because the problems compositing was designed to solve are already taken care of by the nature of the workflow. Because visual effects can be integrated seamlessly in engine along with all other scene elements—assets, lighting, cameras, and so on—there is little to no need to create render passes for the VFX team for a traditional assembly process. As for tweaks to lighting, reflections, and other aspects of the visuals that might need attention, you can see and tweak these values directly within Unreal Engine and use the native post-processing and material tools to ensure they’re just right before going out to final.

There may still be circumstances where traditional compositing is desired—if you’re easing into a real-time pipeline and don’t want to replace all your traditional steps just yet, for example, or if you have specialized visual requirements. For these situations, the Movie Render Queue in Unreal Engine can isolate specific render passes and mattes for your compositing stage.

“It’s still an ongoing fight when I say, ‘Look, we don’t have a comp stage.’ ”

Andy Bean

Head of Real-Time Finishing
BRON Digital

Editorial

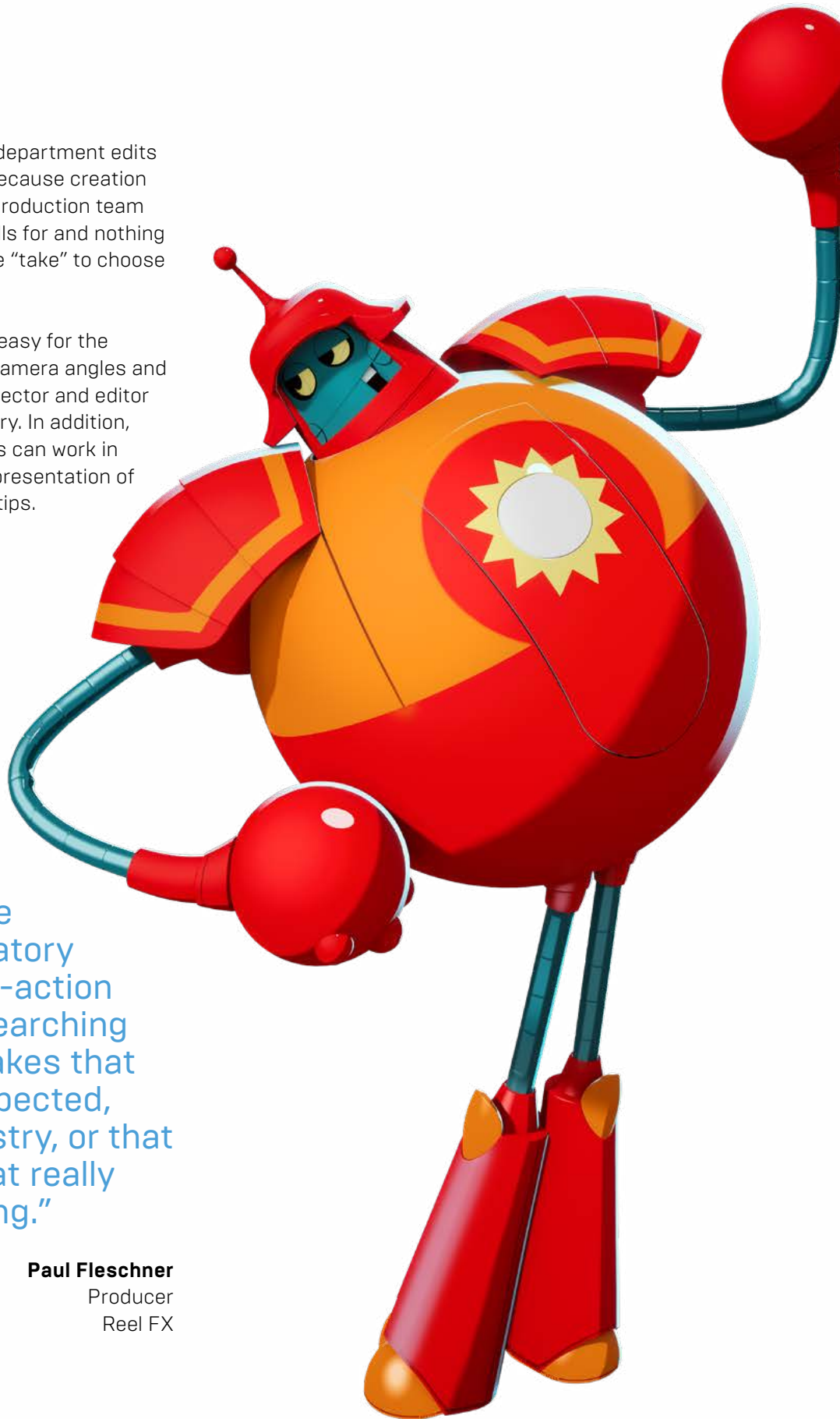
In a traditional pipeline, the Editorial department edits the finished shots into a story. But because creation of shots is so time-consuming, the production team delivers only what the storyboard calls for and nothing more, meaning Editorial gets just one “take” to choose from for each shot.

In a real-time pipeline, it’s quick and easy for the production team to deliver multiple camera angles and shot types for Editorial, giving the director and editor a variety of options for telling the story. In addition, through the use of Sequencer, artists can work in context and always have a closer representation of the final edit in Engine at their fingertips.

“The editing process for a real-time pipeline has that same exploratory quality you find in live-action productions. You’re searching for combinations of takes that will give you the unexpected, some surprise chemistry, or that special something that really makes a sequence sing.”

Paul Fleschner
Producer
Reel FX

Super Giant Robot Brothers | Courtesy of Netflix





Interview: Reel FX Animation | Super Giant Robot Brothers! | Rey Jarrell

Rey Jarrell

Virtual Production Technician

Rey Jarrell acts as the Unreal Engine operator for the mocap and virtual camera needs of *Super Giant Robot Brothers!* Rey's background in interactive AR theater is now helping them unite computer vision, movement, and creative coding into a seamless real-time workflow for animation productions.

***Super Giant Robot Brothers!* is a 3D action-comedy series on Netflix that follows two robot brothers who not only fight off invading hordes of Kaiju, but battle their own sibling rivalry all the way. It is the first series of its kind to be completely rendered in a game engine.**

As someone who was using real-time technology before you worked in animation, how do you think it changes the dynamics of the workflow?

The beautiful thing is that it brings more people into a shared discussion; everyone is collaborating and working together in real time, instead of waiting for the project to hit their part of the pipeline. But this is also disruptive to the traditional way you do things, which is a very linear pipeline. Now, it's more like, "If I do this, you can do that, and then you can do this." You have to work and play together in tandem a bit more.

That's why it's important to have a tool that works as fast as you can. Our team iterates quickly and is very playful, so when you have something like Unreal Engine, it quickly becomes another member of the team. It really helps us run, which is so energizing.

Unreal also reminds you that everybody is going to come at this from a different direction. Nobody learns it in a classroom. Someone might have always wanted to make a movie. Another might have been really into games. And when you all come together, you end up looking across the campfire and seeing that everybody knows something you don't, or is working from a different perspective—and that adds up.

It also helps when you hit the challenges, because more perspectives means more ideas about how to take what the engine does well and apply it to your problems. Sometimes you'll see people being really clever with the shaders they're writing, or breaking out of comp with a post-process material. It's fun to just watch people be brilliant and know you're part of that team. It's inspirational.

You spoke of playfulness. Do you see that as being an essential part of the process?

Yes! So much of what is fun about this process is that you can be so playful and inspire each other with "Oh, I thought of this! I thought of that!" But beyond just being playful, it helps on so many levels. One example is in terms of the communication between departments during the environmental scouting process. A lot of times we would set up an environment in a volume that had just been modeled, and Mark [Andrews, Director of *Super Giant Robot Brothers!*] would walk around with the camera and figure out where things should go. Mark would just say, "I want this mushroom over there. I want this to be slightly larger." Because of how Unreal Engine works, I can do that right away.

We have this saying that you kind of have to be an octopus, because everything is going through Unreal and everybody who needs anything changed in terms of the set—which is like the entire shoot—needs it from that box. And everybody needs it right now. The characters have to be in position, the camera has to be at the correct point and the correct size...and keep in mind, some of these characters are over 40 times the size of a six-foot human being. So you've got to figure that all out and make everything flow—which is fun for me, because I have 25 threads going at one time in my brain. And this way they all get to be productive, which is fantastic.

We had scenes where the actors were improvising on the stage, and that would give Mark a completely different idea, and he'd say, "Let's change this flight path." And I'd move something around so he could get it in a different way.

There's been a lot of experimenting and kind of inventing right there on the spot. That is much harder to do in a traditional fashion, because you have to do one thing at a time and check it out. Here, we can play. And as soon as we got it working, we just said, "This is going to be it now." That's huge. You can figure things out there, versus trying to figure them out ahead of time and then have it go completely wrong once you get there on the day.

Every episode, we added more and more to our box of tricks. By the tenth episode, we felt like we could do anything we wanted.

What is your mocap setup?

We used OptiTrack on the stage because we needed full-body characters and props as well as a camera all together. Once mocap was done and we were just shooting VCam, and all we needed to track was the camera, we moved to a simpler setup, a VIVE volume with just a single puck.

How does scripting come into play?

As our process began to evolve, we took note of all the things we do over and over again, and decided to make a custom tool to do it for us. For instance, every time I want to put a character on a puppet, I need to make their hips go in a specific spot really fast. We need a tool for that, so we script that.

Unreal Engine features like Editor Utility Widgets and custom editor scripting were so helpful, even necessary, to help us evolve our processes with these custom tools. One used a lot on stage was the "magic scaler tool," a widget I created for attaching one actor to another, and making the parent actor larger while keeping the child actor in the same world position. That was great for when you get everything lined up and attached to the puppet in the right place, but then you discover the puppet needs to be bigger.

I also made a custom stage Blueprint that had an approximate model of the mocap stage floor attached to it, which grew and shrank with its parent. That was really helpful for lining things up and figuring out the right scale to cover the area we needed. We've also developed tools to automate the editing of sequences, set a Time Dilation Track value back to 1.0, and so on.

It was a really fun challenge from a programming perspective to think, "Okay, how do I do this in a way that is fast enough that I can do it on the stage, and be intelligent, and elegant about it from a Blueprint perspective?" But also from a workflow standpoint, it was so fun to see the team evolve and start to speak in the same language and work from the same techniques.



Interview: WildBrain Studios | LEGO Friends, In the Night Garden | Morgana Duque

Morgana Duque

VP of Production

Morgana Duque is a producer with over a decade of experience in the film and television industry, including time on projects like *Slugterra*; *Endangered Species*; *Mega Man: Fully Charged*; and most recently as the Producer on *Go, Dog. Go!*, which became a Top 10 series on Netflix in over 45 countries.

WildBrain's latest projects include the television series *LEGO Friends* and *In the Night Garden*. *LEGO Friends* follows the everyday adventures of five girls in Heartlake City, while *In the Night Garden* is a magical interpretation of a nursery rhyme picture book.

What was your first foray into a real-time animation pipeline?

It was *LEGO Friends*, WildBrain's second project in a real-time engine. There was some knowledge of Unreal Engine in the studio, but we hadn't really even delivered the first episode of that first project yet and we started this second project.

It was a good kind of transitional space where we knew some things, but we didn't know all of it. The process was shifting from, "How do we make this work right now for delivery?" to thinking, "How do we make this work better, for a better process and a better outcome?"

Tell me about your day-to-day in this pipeline.

The first two months, a lot of it is just wrapping your head around what you can and can't do. There are moments of "Mind blown! Wow, can we do this?" or "Oh, we can't do this?" or "It doesn't work the same way as..."

In terms of touching the pipeline as a producer, the majority of it is in the review space, when we're having modeling reviews, surfacing reviews, lighting reviews, and all of that.

It's having that real-time review and not having all the notes go into ShotGrid because some of the notes were addressed as we were speaking about them. There's also the client-facing education of what this pipeline means to them.

As a producer, the go-to, it was up to me to explain that they were going to receive their lighting pass that much earlier than they're used to. And what is that going to look like? And why is that better?

Did clients embrace the new type of review process?

The client always wants to review the animation and make sure it's perfect. We can do lighting while animators are animating, so we can have them review the animation together with lighting, and still have six weeks to fix it instead of just one.

At first they don't understand what's new. So I say, "Let's talk about it again." It's a shift in your brain. Education is key here—seeing the client's journey from, "I don't understand this," to "This is amazing! I can't wait to see it in the next project." It's unlearning certain patterns, and then embracing what the new ones can do for you.



LEGO was really excited about it, so that was one win right away. I can see there's going to be another year or year-and-a-half of educating folks around it. And we're just going to get better.

It's exciting because it feels like it's starting up. It feels like everyone's in it together. So, having a client that's excited about it makes it that much easier.

Tell us about your internal review process with this new pipeline.

We went through a few different ways of doing the reviews. It started as a free-for-all, but then you start thinking in dollars. Suppose we wanted a change to a model. The modeler can make the change right then and there during the review session, and everyone wants to see it. So there are all these expensive people—a director, a CG supervisor, a modeling supervisor, two coordinators—all watching it being done, instead of making a quick note for the modeler and moving on. It's all about prioritizing which fixes to do live and which not to—some are worth it, some can be done later.

That's a different way of doing things. Now that we can do it all instantly, everybody's got an idea. But that's the thing that's amazing, too. You have people that would usually be quiet in meetings, and now they're chiming in, asking, "What if we did such-and-such?"

It took a while to find the right rhythm. You have to walk that line of wanting it to be a group effort, but there's limited time. And there's a hierarchy for a reason. Eventually, we set a rule that we will only allow so much time for each thing to be reviewed. And at some point, you've got to give your notes and it has to go to an artist, and you're going to see it tomorrow.



In the Night Garden - Stills courtesy of WildBrain Studios

What are the types of issues you've been able to handle in review sessions?

The first thing that comes to mind is skies. We used to spend months, sometimes more than six months, on things like how the sky should look, how many clouds, where the sun should be. You'd give notes to Design, they'd come back with something, it wouldn't be quite right, and we'd send it back with more notes.

When we set up a sky in Unreal, we played around with it for an hour, and there it was. The sun's moving and the clouds are moving, and we just saved six months of back-and-forth with Design.

For WildBrain's series *In the Night Garden*, we saw it with approvals on little things like grass and tree bark. Sometimes those approvals can take a long time, but when you can tweak it right then and there, the approvals go really fast.

How has the real-time pipeline impacted the way you communicate internally?

Now lighting and animation are all happening at the same time, so we're forced to communicate much more often, which is great. And producers help drive that communication. So yes, there are more meetings, more check-ins.

It's something we still work on. A lot of this happened during COVID, working remotely. I'd love to see a parallel universe where we could have started all these projects with people sitting beside each other and being able to walk by and just talk.

With everything happening at once, how does this affect artists working in their own departments?

In a traditional pipeline, where one thing comes after the next, each production coordinator and the supervisor would think their department is the most important. That's normal, and it makes for a good-quality production. Because it's one after the other, after the other—every department has a time to shine and a time to be especially appreciated.

But now, with all the departments stacked and everything happening at the same time, we hear things like, "Animation is doing this thing, and Modeling is doing this other thing, and I'm just sitting here in Rigging with nothing to do. Please pay attention to me!"

And then between the producer and the CG supervisor, we need to manage this. We just need to have more conversations.

It's going to take a while, but we do need to undo our "one thing at a time" thought process.

How has the new pipeline affected scheduling?

One obvious thing is that, in the past, we've always scheduled to finish surfacing before shot production. That's always been our schedule. We finish surfacing, and then we start layout.

So, bringing lighting all the way to layout shouldn't be a problem, right? But then when I saw that, as a producer, I went over to Robin [Robin Shea, Vice President of Production Technology at WildBrain] and said, "How can we finish surfacing in time for lighting?" He replied by pointing back at our old schedules with surfacing wrapping up before layout. Turns out we used to push surfacing all the way to after animation and just before lighting in order to manage some delays in asset production.



LEGO Friends - Stills courtesy of WildBrain Studios and LEGO

I realized that was a crutch we no longer needed. Going to a real-time pipeline has exposed a lot of inefficiencies. The new way of working is, in a very refreshing way, pushing us to do better and not to just settle.

To me, that's the big nut to crack right now: how to make sure assets are in good shape for shot production, because shot production now is not one thing after the other anymore.

What is your personal opinion of working with a real-time pipeline?

I love problem-solving and I love a challenge, so it's exciting. I have to admit, there were moments when I wanted to throw it all away and just go back to the traditional way of doing things, but then there'd be another moment that made it all worth it.

One thing we'd really like to improve is our asset library. So many of the benefits I see around modeling and putting things together in Unreal is having a library and arranging things. But so far, having the time to build up a library first and then do the environment build—we're not there yet. I'm looking forward to having a full asset library and utilizing it to its full potential.

And in terms of our direction, sometimes seeing everything together makes you change what you want to do. It's exciting. I'm looking forward to seeing where it's going to go.

Interview: Sony Pictures Imageworks | In Vaulted Halls Entombed | Craig McPherson

Craig McPherson

Animation Supervisor

Craig McPherson's credits include *Thor: Love and Thunder*, *Jumanji: The Next Level*, *Men in Black: International*, *The Meg*, and *Edge of Tomorrow*. Prior to joining SPI, he was also an animator on *District 9* and *Happy Feet*.

You've worked on game projects in the past, so you already had some real-time experience going in. Were there any surprises when you started applying it to an animation context?

I think I didn't really understand what the workflow was going to be when we started. When I worked in games—which included cinematics—all the work was done offline, then compiled in the game engine and played back in real time. What I didn't understand here was how much of the actual production would be *done* in real time.

I sat in on the layout reviews and watched how the scenes were created with our director Jerome Chen, and I was amazed at how much work could be done in engine. That made the process so much more collaborative, since Jerome didn't have to wait for us to go away, make changes, re-render, and then show him what we did. So much just happened right there. For instance, as we were doing the layout reviews, the lighting and environments were being updated, and those changes were always a moment away.

I was honestly shocked to find out how iterative and collaborative the process could be. Moving things back and forth is one of the slowest things we do in our offline pipeline, so we tend to over-engineer things in one department to reduce issues down the lines. But when every department is working in a unified sandbox, you can throw things into Unreal Engine and have a look without worrying too much about it. That helps us be more creative. I can try something radical and see how it might affect the shot. In our offline pipeline, I might need several people in varying departments to incorporate my change before we see what the result might be. This way makes more sense.

Can you take us through the new pipeline? How much of the legacy one remained?

Our animation was still mostly created offline and compiled in Unreal Engine. That said, we did all our camera animation within Unreal, which was a giant leap forward. Now we can design our camera moves with lens flares, shadows, reflections, depth of field, and motion blur in mind.

In general, though, Unreal was used as our presentation medium, as well as our final renderer, which means all our animation reviews were done in engine.

This gave us two big wins. One was that we could view our animation with a near-final look, including lighting, FX, motion blur, depth of field, and so on. This was great because normally this comes much later in the process for us, and we don't get as much of a chance to iterate after having seen the final look.

The other big win is that I could look at a rendered shot from any angle I wanted. I can orbit the camera, look from a witness angle, scrub through slowly—I can even see surrounding shots and change our edit on the fly to see if we can find better ways to present the performance. This helped me fully understand the performance in a holistic way and give much better notes.



What was the best part of the switch? Was it worth it?

Absolutely worth it! In a real-time workflow, you are ultimately more iterative, more collaborative, and more creative. This means better results on screen, almost every time. We had some trade-offs, though. For instance, we couldn't have assembled all of our different asset types in engine without some brute force. We also had some limitations in terms of character effects like cloth and hair, muscles, fat, and so on.

It's definitely a project-based decision, but I think the possibilities are quite endless.



Interview: Reel FX Animation | Super Giant Robot Brothers! | Paul Fleschner

Paul Fleschner

Producer

Paul Fleschner was a freshman in high school when he had a “Eureka!” moment about becoming a filmmaker. He made Super VHS films with his friends, learned about editing from John Hughes’ post supervisor while still in high school, then worked his way into the production world. This took him from Fox VFX Lab to ILM, and then to Reel FX working as a producer on *Super Giant Robot Brothers!*, with a focus on editorial.

What made Reel FX decide to use a real-time workflow for *Super Giant Robot Brothers*?

We’ve been really lucky to have internal advocates. Manu Laurent—the head of our division—has been interested in real-time for years. He was entrenched with all the departments that were versed in traditional animation, which helped him make a case. He was able to help us sniff around and start figuring out who wanted to try something new. We then handpicked a special team to lead the effort and began our practical investigations. Since then, it has really been about experimentation and making sure that the right minds are linked up and building off one another.

The last year and a half has just been an insane sprint. We’re building the pipeline as we’re making our show, with a relatively small team. We’ve had to build bespoke code, and integrate with our internal IT and engineering team. It sounds like a lot, but it’s been a blast.

This team is bridging the gap between the traditional animation pipeline and virtual production, and working to bring all the things live-action directors love—like virtual cameras and motion capture—into the creative process.

What are some things teams have to do differently to make a real-time pipeline work?

You need to commit to building the assets up front. There’s a domino effect that carries all the way through to animation if you’re behind on those assets. If you can commit, and I mean really commit, you’re going to reap so many rewards.

If you’re using this real-time pipeline for final pixels, you have trust that you’re going to come up with ideas, maybe even some of your best ideas, in a live-action context on the virtual set. And that’s very different from traditional animation, where you’ve got storyboards, you iterate, you plan and execute on that.

With real-time animation production, you’re playing with elements live, which has a different feel. It’s almost like a new art form just came out, or at the very least a synthesis of the past and present. When you smash live action and animation together, you get this new process.

I think that’s one of the biggest challenges for most teams, and I get why it’s hard—you’re asking companies to not only take a leap of faith, but also to be willing to invest in something they haven’t done before. Suddenly you’re telling them they have to figure out the story on the stage with assets, not primarily with a storyboard.

But there’s a real freedom to working in this way. I’ve always been enamored with any process that naturally generates sparks of creativity. Committing to this process is about trusting that you can be creative and come up with things in the moment.



How does editing shots for a real-time pipeline differ from the traditional?

The editing process for a real-time pipeline has that same exploratory quality you find in live-action productions. You're searching for combinations of takes that will give you the unexpected, some surprise chemistry, or that special something that really makes a sequence sing.

For example, suppose an editor's got a wide, a medium, and a close-up from two angles. Who knows how they're going to put it together, or if something magical is going to happen because he or she noticed the right head glance, at the right time?

That same magic in the editing suite is hard to get otherwise, because you have such a limited number of shots to work with. And the reason behind that limitation is that shots are so time-consuming to produce, so you tend to produce only what's predetermined, and not a whole lot more.

With real-time, we can produce hundreds of shots a day and give editorial a whole lot to choose from, to pick and choose and make the story as interesting and engaging as possible.



Super Giant Robot Brothers | Courtesy of Netflix



Don't get me wrong, there's nothing wrong with the traditional pipeline and there's absolutely stunning, beautifully animated films made in the traditional way. This real-time hybrid just gives a different feel to the process and content. The magic an editor finds through cutting multiple angles and takes brings a different kind of emotional energy, the kind you ordinarily get with a live-action shoot, when you get a shot through a person's unique eye or whims in the moment. In real-time, in the same way, you get options you might not even expect, and can't even plan for. That openness to a spirit of discovery is what drives this editorial process.

Please tell us about your review process.

We always start with mocap selects. The VCam can be so generative that you'll quickly wind up with too many shots for the editor if you don't whittle them down. We tend to run a VCam volume in a studio. If you've got your performance selects from mocap and your Master Layout stitched together, then you've got your 3D in the round. You choose your best takes to shoot, then you can press Replay, and you've got your best performance running over and over again in your environment.

Then you can shoot as much as you want. It's crazy. We can do hundreds of shots in a day when we want to, which ultimately begs the question, is that too much? You have to ask yourself, why are we shooting this much? I think we're still trying to find our footing there, but it's also what our director wants. So we shoot a lot, do selects, and send them to editorial.

The editorial process is something we try to keep very sacred and intimate. We don't want too many cooks in the kitchen. I really designed it to include just the editor, the director, and a coordinator—that's it. You want the right people to have their time to creatively cook on this process.

They bring out an animatic cut very, very quickly for feedback. This is reviewed in a typical fashion, internally at the studio, and then sent on to the streamer. The editorial process is a lot like the process for live-action projects.

How has this mash-up of virtual production and animation personnel worked out for Reel FX?

That's been a fun challenge for us: getting our virtual production team, who have that set mentality, intersecting with an animation studio. That's a collision of mindsets. If you want to be successful, you've really got to get people to come to the table and hear what the other side is saying, how they're approaching it, and what they need in order to work together. So energy is important, but it's also about communication and getting people with different backgrounds to share a common vocabulary.

We're lucky to be working with a process that feels like a game. That's what virtual production filmmaking is: filming inside a game where the goal is to connect with your audience. And all of these people around me, we all get together to collaborate on this game, using everyone's different skill sets to make it work. If you tell a good story, you win the game.

Pipeline Functions

Shifting from a traditional workflow to a real-time one requires some changes in the way your pipeline works. Here, we outline some of the major areas of difference. For more technical details about this pipeline shift, see the [Technical Guide to Linear Content Creation](#).

Hardware

While you can reduce your dependence on large, expensive render farms in a real-time pipeline, the individual workstations that run Unreal Engine need to be fitted with more powerful hardware. In general, the same hardware setup used to develop games works for animation production, but will vary from artist to artist depending on the work they are doing. See the [Hardware and Software Specifications](#) page in the Unreal Engine documentation for details.

Assets

Game and film productions create, organize, and utilize assets very differently.

In game development, for example, entire worlds are created, whereas in film productions, sets are often incomplete and created “for camera”. When you transition to a real-time pipeline, you begin to see the value in creating your worlds and environments in their entirety: the ability to explore whole sets, similar to shooting on location, while still doing set dressing as needed on a shot-by-shot basis.

An additional consideration is optimization; the way you build your assets is important to ensuring real-time playback. New features like [Nanite](#) are making that process more flexible, but we also encourage using [established practices for asset optimization](#).

Source Control

Source control, or versioning, is the practice of managing updates to files that are likely to change over the course of project development, such as 3D assets, textures, scripts, and engine source code.

Source control is a must when multiple users will be modifying a project at the same time. Experienced animation studios have one or more systems for keeping track of assets and other files, but many of these systems are prone to inaccuracies due to human error.

One type of system might use a spreadsheet or other tracking tool, with an Asset Manager assigned to manually update the list. A studio might also take a step in the direction of automation by setting up DCC app-based scripts that search for new files based on strict filename conventions. A studio might use a “versionless master” system, where the loading of any scene automatically pulls in certain externally referenced files.

However, such systems rely heavily on the team members themselves to name each file correctly, or to provide information manually. A single mistake on the part of an artist—who was hired for his or her creative ability, not attention to administrative detail—can cost the project days or weeks of work.

Unreal Engine takes care of this risk with native support for source control tools such as Perforce and Subversion (SVN). [Perforce](#) is a reliable paid source control solution that many of our customers prefer. As an alternative, [SVN is available through Github as a free open source solution](#).

Both Perforce and SVN provide the ability to “check out” assets—when an asset is checked out, no one else can edit it. After an artist or technician has finished updating the asset, the new version is “checked in” (committed) and is now available for others to access.

These tools can also provide a history of asset updates, and the ability to roll back to a previous version of any given asset.

The source code for Unreal Engine is available to all users, and some studios also like to customize the



code for their own purposes. UnrealGameSync (UGS), included with Unreal Engine, provides synchronization and dissemination tools for any code changes to the engine.

UGS has an additional purpose—it acts as a wrapper for Perforce to provide a seamless experience in managing data for UnrealEngine.

Importing to the Engine

To utilize a real-time workflow, you will still need to use DCC applications like Maya, 3ds Max, Cinema 4D, or Blender to create your assets. But once the assets are in engine, you can apply transforms, apply materials with your chosen textures, and control familiar attributes such as glossiness and translucency. You can also apply full-body IK animation, and edit assets with built-in modeling tools.

Unreal Engine includes tools for importing FBX, USD, and Alembic data, and also plugins to export from specific packages such as 3ds Max into UE's native

format, *.udatasmith*. Many studios take advantage of the Unreal Engine Blueprint system, or Python or C++ integration, to customize and streamline their import processes to suit their own pipelines.

Integratable Tools

In addition to importing data from common DCC apps, there are plugins that enable live integrations between Unreal Engine and [ShotGrid](#), [Houdini](#), [Ozone Rig Studio](#), and [Maya](#).

Technical Support

In this guide, we have provided an overview of the concepts involved in creating a real-time animation pipeline, but we've only scratched the surface. For more detailed tips and tricks to meet your specific needs, and to connect with other creators, join our [Unreal Developer Community](#), where you can find more information and resources to help you build your Unreal Engine animation pipeline, such as the [Technical Guide to Linear Content Creation](#).



Image courtesy of Engine House Films

Interview: WildBrain Studios | LEGO Friends, In the Night Garden | Shazzy Angulo

Shazzy Angulo

Unreal Development Supervisor

Shazzy Angulo is the Unreal Development Supervisor at WildBrain, a studio that will have delivered seven TV shows with final pixels out of Unreal Engine by next year. Shazzy has been working with real-time technologies and children’s animation for six years, and has spearheaded the development of the Unreal Engine pipeline at WildBrain for the past two years.

Tell us about your role in getting the real-time pipeline going at WildBrain.

I was brought in to help go from completing a 22-minute pilot to building an entire Unreal pipeline. We were building the pipeline during our first real-time project, *Fireman Sam* season 13, which consisted of 26 Episodes, each 22 minutes long. I helped develop best practices and workflows while working with the CG supervisor.

My team consists of myself and two Unreal developers who helped build out the tools in Python and Unreal Engine Blueprint. Aside from building the tools, a lot of my time is spent training and teaching artists, supervisors, and other production team members on how things are done in Unreal Engine.

One of the funny things is that I’m one of the few people here who never actually worked in a traditional pipeline. I’ve always worked in game engines, so when I came to WildBrain I had to learn about the traditional pipeline to help me understand where everyone was coming from and the best way to guide them to a real-time pipeline.

At the start, much of my time was spent with people who had been in the industry for 10+ years in traditional pipelines, introducing them to something new outside of their comfort zone. A lot of concepts at the core are the same, and seeing each person have that “Aha!” moment is so exciting and rewarding, because I know I’ve converted them over to a real-time workflow.

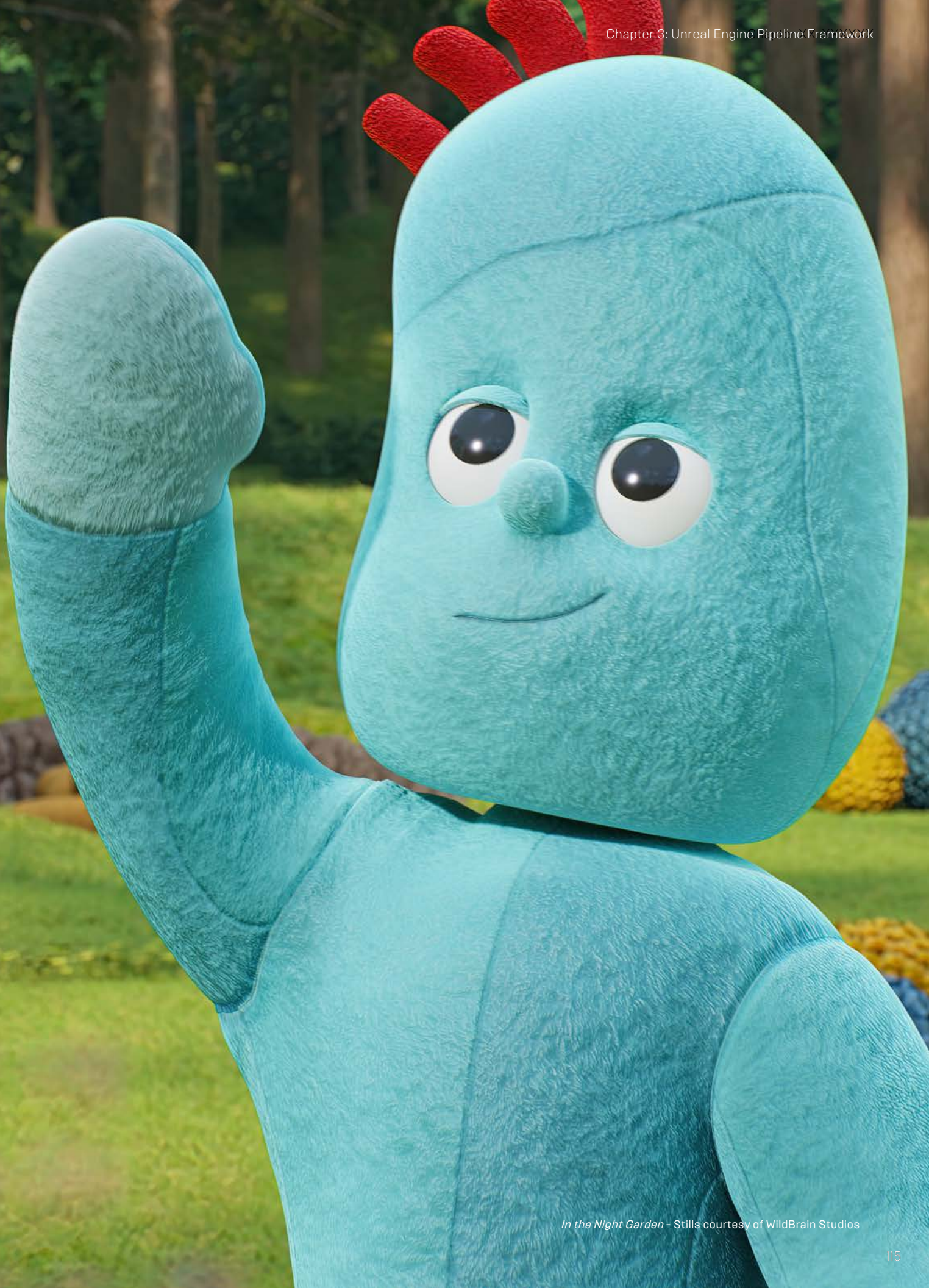
When we were in animation for Episode 5 of *Fireman Sam*, *LEGO Friends* started production and the process started all over—training, educating, and winning everyone over.

What are some of the challenges you face in training people on a real-time pipeline?

To excel in Unreal, it’s important to be open to embracing the engine’s benefits, while understanding it’s not a magic button that fixes everything. There are those who want to go real-time, and there are those who are comfortable staying where they are, and that’s key—getting the right people to be in engine.

When you have deadlines, learning a new technology can be daunting, but I try to spend as much time as needed with each person until they feel comfortable.

As the engine continues to develop, there are things we run into that aren’t built out or don’t function as we expect. I have the mentality of, “We’ll find ways to work around it and will provide options.” Embracing this new pipeline is about accepting where it is, but with each release of the engine, we see previous pain points resolved.



Having to pitch the promise of the future of real-time takes patience and is a prerequisite for a position like this!

Another challenge is building the pipeline as we go, but being able to iterate quickly and keep improving is key. Figuring out what is best when no one else is doing it, knowing you can improve on the next show, and continuing to do so, is a great feeling.

All pipelines are constantly working through issues and bugs to improve, and the Unreal pipeline is no different. The updates to the engine, and new feature releases, push us to constantly be moving forward and move more and more departments into the engine.

What are some of the new things you can do with the new pipeline?

Once you have all your assets in engine, it starts to open up so many possibilities. We've made executables of our sets for our storyboard teams so they can get background plates without needing a powerful machine to run a ray-traced project. We can do real-time reviews with directors for material changes, lighting, and effects.

My favorite moments are when you have directors and art directors who start asking if they can get a new machine so they can be in engine. It's inspiring, and it lets them see things in a new way. For us, we have removed comp completely—final pixels come right out of Unreal. Being able to see your shots in engine, in real time, is super exciting. You get feedback faster, and adjustments and notes are easy to make.

What do you like best about the real-time pipeline?

What I enjoy the most is how much a small team, or even one person, can do by themselves. Previously, each step in the pipeline was so dependent on the previous. Now you can have a modeler set up their asset with light rigs, post-process volumes, and effects all at once. You can have lighters adding crowds and working with animation. It's made animation accessible, for big studios and independent ones. It's empowered artists to go beyond just their discipline.

I also love the sense of community around it, within our own studio and externally. People are open and sharing what they've built and how they solved issues, and with open source tools. We have all of our supervisors mentoring new supervisors who haven't used Unreal before. It's not about just being on your show anymore—it's about improving the pipeline for the next show and passing along the best practices.



That's All, Folks!

We trust you have found *The Animation Field Guide* useful in helping you understand how a real-time pipeline can help you increase creativity while achieving your goals for your animation projects.

As for your journey into the world of real-time animation, we'll leave you with this: *There's nothing to it but to do it*. You can research real-time pipelines all day, but there is no better way to learn than roll up your sleeves and try it for yourself on your own project.

In this guide, we have aimed to demystify real-time animation workflows, and to encourage you to [download Unreal Engine](#) so you can start building your own real-time animation pipeline.

Epic Games remains committed to the animation community, and we hope you'll continue the journey with us. You can find more resources at our [Animation hub](#), and also at the page for our 2022 [Future of Animation event](#).

Glossary

Blueprint

Script created from the Blueprint visual scripting language in Unreal Engine—defines how an asset interacts.

Cinematic

A pre-rendered, noninteractive sequence in an otherwise interactive experience. Also called "cutscene".

Cutscene

A pre-rendered, noninteractive sequence in an otherwise interactive experience.

DCC application

Digital content creation software such as Maya, Blender, 3ds Max, or Cinema 4D.

Final pixel

Final image quality without the need for additional post-production.

Game engine

A software development environment designed for the creation of real-time interactive content, initially for video games, but now used in many other applications.

Graphics processing unit (GPU)

A specialized type of microprocessor optimized to display graphics and do very specific computational tasks. Modern real-time engines rely heavily on GPUs for performance.

In-camera visual effects

Special effects rendered in the camera at the time of recording. Example: a scene filmed with an actor standing in front of an LED screen displaying a virtual background.

Motion capture (mocap)

The process of recording actions of human actors, and using that information to animate digital character models.

Performance capture

An advanced form of motion capture that includes faces and/or fingers, and captures subtle expressions.

Previsualization (previs)

The process in pre-production of visually mapping out scenes of a project before principal photography.

Ray tracing

A rendering technique for generating an image by tracing the path of light as pixels in an image plane and simulating the effects of its encounters with virtual objects.

Real-time rendering

The translation of a scene into display pixels fast enough for instantaneous playback at real-time (live) speeds. In contrast, traditional offline rendering may take minutes or even hours to produce each frame, with 24 frames required to display a second's worth of animation.

Retargeting

The application of mocap or animation data to a CG character of a different body geometry.

Source control

The practice of managing and tracking updates to engine source code.

Stage (mocap)

An area or volume where motion capture takes place with actors.

Time Dilation Track

A type of track in Unreal Engine Sequencer that speeds up or slows down a sequence.

Versioning

The practice of tracking and controlling updates to files that are likely to change over the course of project development, such as 3D assets, textures, and scripts.

Virtual camera (VCam)

A camera in the game engine that can be driven using a device such as an iPad.

Virtual production (VP)

The cross section between physical and digital worlds. VP lets directors work on a physical stage but use real-time technology to view and interact with virtual environments and characters.

Virtual reality (VR)

An immersive experience using headsets to generate the realistic sounds, images, and other sensations that replicate a real environment or create an imaginary world.

Virtual scouting

Use of VR technology to share a model of a set for shot planning and production design purpose.

Volume

The physical space in which performance capture is recorded.



Links

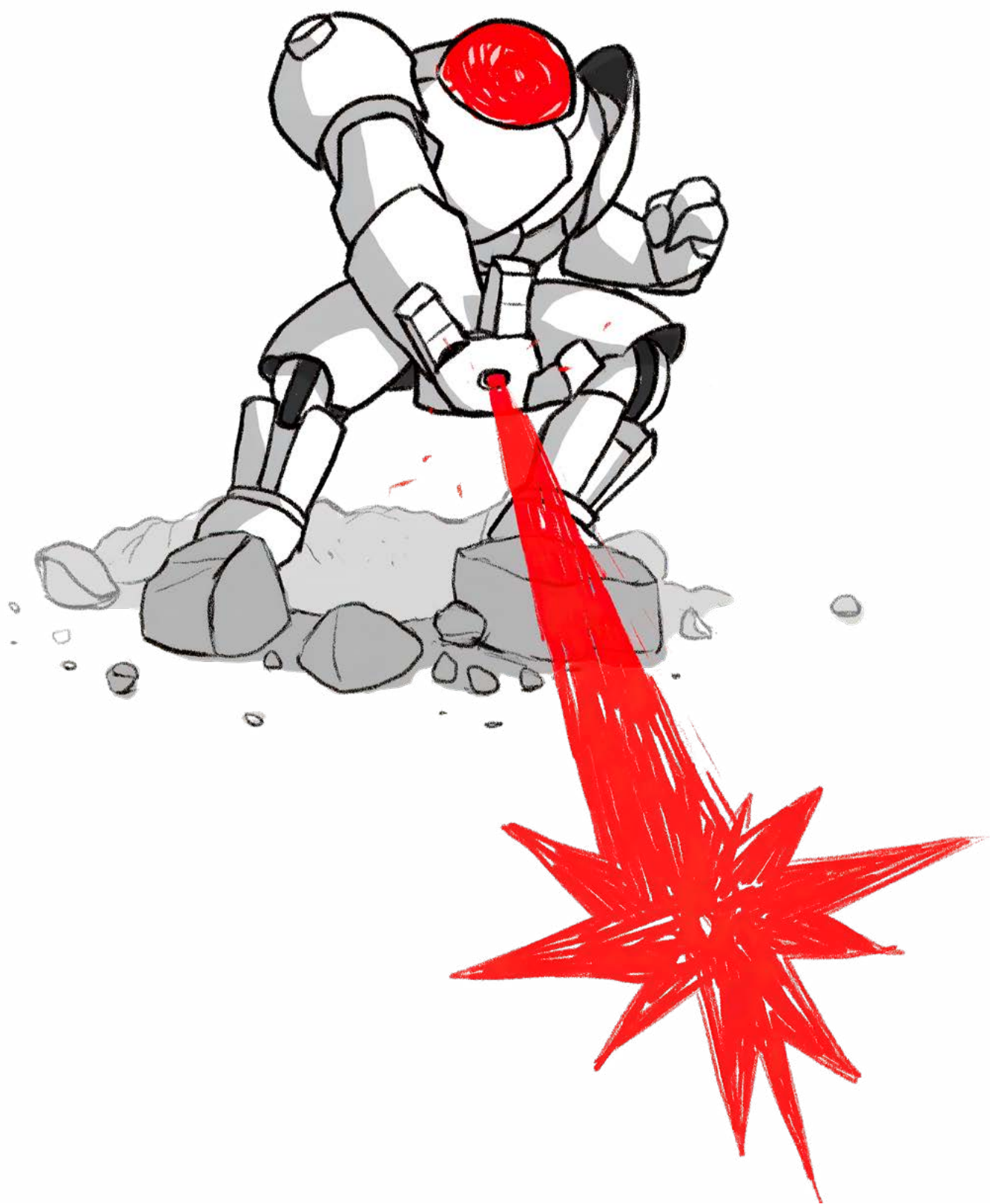
9	Epic Developer Community	https://dev.epicgames.com/community/
9	Technical Guide to Linear Content Creation	https://dev.epicgames.com/community/learning/paths/YJ/technical-guide-to-linear-content-creation
17	BRON Digital	https://bronstudios.com/divisions/digital
17	Fables	https://www.unrealengine.com/spotlights/bron-digital-s-fables-finds-new-levels-of-creativity-with-real-time-animation
22	Trouble	https://deadline.com/2021/03/danny-mcbride-spire-animation-studios-trouble-feature-brad-lewis-pj-gunsagar-1234722164/
22	Spire Animation Studios	https://www.spirestudios.com/
26	Sony Pictures Imageworks	https://www.imageworks.com/
34	La Cabane Productions	https://lacabaneproductions.com/
34	Mush-Mush and the Mushables	https://www.youtube.com/channel/UCmPk_TSQUsKbj5aQi_WikLQ
36	BRON Digital	https://bronstudios.com/divisions/digital
38	Control Rig	https://docs.unrealengine.com/control-rig-in-unreal-engine/
38	Engine Marketplace	https://unrealengine.com/marketplace/store
38	Unreal	https://unrealengine.com/marketplace/store
38	Quixel Megascans	https://www.unrealengine.com/marketplace/content-cat/assets/megascans
38	MetaHumans	https://www.unrealengine.com/metahuman
40	Live Link data	https://docs.unrealengine.com/using-live-link-data-in-unreal-engine/
40	Control Rig	https://docs.unrealengine.com/control-rig-in-unreal-engine/
41	Digital Dimension Entertainment Group	https://digitaldimension.com/
41	The Beachbuds	https://saturdayanimation.com/beachbuds
41	Chase and CATCH	https://www.awn.com/news/saturday-animation-develop-chase-and-catch-unreal-pipeline
41	Zafari	https://www.unrealengine.com/spotlights/animated-children-s-series-zafari-springs-to-life-with-unreal-engine
47	Chromosphere Studio	https://chromosphere-la.com/
47	Mall Stories	https://www.youtube.com/watch?v=DPjAV9jlzDM
47	Third episode of Yuki 7	https://youtu.be/mx6h-pErhgg
47	Yuki 7	https://yuki-7.com/
52	BRON Digital	https://bronstudios.com/divisions/digital
56	Path Tracer	https://docs.unrealengine.com/path-tracer-in-unreal-engine/
56	Movie Render Queue	https://docs.unrealengine.com/AnimatingObjects/Sequencer/Workflow/RenderAndExport/HighQualityMediaExport/

56	Lumen	https://www.unrealengine.com/blog/exploring-lumen-unreal-engine-5-s-dynamic-global-illumination-and-reflections-system
56	Nanite	https://www.unrealengine.com/blog/understanding-nanite---unreal-engine-5-s-new-virtualized-geometry-system
57	Sequencer	https://docs.unrealengine.com/AnimatingObjects/Sequencer/Overview/
58	Pixel Streaming	https://cdn2.unrealengine.com/Unreal+Engine%2FPixelStreamingWhitepaper%2FPixelStreamingWhitepaper-V1.9C-ecl985a0f32c53c6650b2d02c11a67c32fa4e176.pdf
58	Live Link	https://docs.unrealengine.com/live-link-in-unreal-engine/
58	Full Body IK solver	https://docs.unrealengine.com/ik-rig-solvers-in-unreal-engine/#fullbodyik
58	Motion Warping	https://docs.unrealengine.com/motion-warping-in-unreal-engine/
58	Pose Browser	https://docs.unrealengine.com/AnimatingObjects/SkeletalMeshAnimation/AnimHowTo/CreatePoseAsset/
58	Control Rig	https://docs.unrealengine.com/control-rig-in-unreal-engine/
58	Live Link Face app	https://apps.apple.com/us/app/live-link-face/id1495370836
58	Multi-User Editor	https://docs.unrealengine.com/ProductionPipelines/MultiUserEditing/
59	Post-processing effects	https://docs.unrealengine.com/post-process-effects-in-unreal-engine/
59	Niagara particles	https://docs.unrealengine.com/overview-of-niagara-effects-for-unreal-engine/
59	Movie Render Queue	https://docs.unrealengine.com/AnimatingObjects/Sequencer/Workflow/RenderAndExport/HighQualityMediaExport/
59	Blueprint visual scripting system	https://docs.unrealengine.com/ProgrammingAndScripting/Blueprints/
59	Chaos Physics	https://docs.unrealengine.com/physics-in-unreal-engine/
59	Hair and fur	https://docs.unrealengine.com/WorkingWithContent/Hair/
59	The Foundry's Nuke integration with Unreal Engine	https://learn.foundry.com/nuke/content/comp_environment/unrealreader/unreal-intro.html
60	MetaHuman framework	https://www.unrealengine.com/metahuman
60	MetaHuman Plugin for Unreal Engine	https://www.unrealengine.com/marketplace/product/metahuman-plugin
60	Quixel Megascans	https://www.unrealengine.com/marketplace/content-cat/assets/megascans
61	Epic Developer Community	https://dev.epicgames.com/community
61	built-in procedural foliage	https://docs.unrealengine.com/BuildingWorlds/OpenWorldTools/ProceduralFoliage/QuickStart/
61	Meerkat demo	https://www.unrealengine.com/marketplace/product/meerkat-demo
61	Unreal Engine Marketplace	https://www.unrealengine.com/marketplace/store
62	3rd World Studios	https://3rdworldstudios.com/
62	Allahyar and the Legend of Markhor	https://www.unrealengine.com/developer-interviews/visually-stunning-animated-feature-allahyar-and-the-legend-of-markhor-is-the-first-produced-entirely-in-unreal-engine



65	Evil Eye Pictures	https://www.evileyepictures.com/
65	Ray-traced piece at SIGGRAPH 2019	https://www.unrealengine.com/spotlights/evil-eye-pictures-uses-latest-ray-tracing-features-to-create-new-branded-visuals-for-unreal-engine
65	The Matrix Awakens	https://www.unrealengine.com/wakeup
65	Pearl	https://www.youtube.com/watch?v=WqCH4DNQBUA
70	Xanadu	https://www.unrealengine.com/spotlights/youtube-show-xanadu-carves-out-a-corner-of-the-metaverse-with-performance-driven-cg
70	Hulu promotional video for The Handmaid's Tale	https://youtu.be/8tZt3BjX0lw
77	ShotGrid	https://www.shotgridsoftware.com/
78	Subversion (SVN)	https://docs.github.com/en/get-started/importing-your-projects-to-github/working-with-subversion-on-github
78	EPOS by Praxinos	https://praxinos.coop/epos.php
78	Unreal Engine Marketplace	https://unrealengine.com/marketplace/store
78	Perforce	https://www.perforce.com/blog/vcs/unreal-virtual-production
78	Quixel Megascans	https://www.unrealengine.com/marketplace/content-cat/assets/megascans
79	Material Editor	https://docs.unrealengine.com/placing-material-expressions-and-functions-in-unreal-engine/
80	DNEG Animation	https://www.dneg.com/dneg-animation/
80	its first Unreal Engine project	https://www.unrealengine.com/spotlights/dneg-animation-explores-a-real-time-pipeline-with-mr-spam-gets-a-new-hat
80	Mr. Spam Gets a New Hat	https://www.youtube.com/watch?v=SWCDeE3B7es
84	Chromosphere Studio	https://chromosphere-la.com/
87	Wylder short film	https://vimeo.com/580353884
87	Engine House	https://www.engine-house.co.uk/
87	Engine House production diary	https://www.engine-house.co.uk/blog/wylder-production-diary-01
87	Engine House on Unreal Engine blog	https://www.unrealengine.com/spotlights/engine-house-tames-animated-short-wylder-with-real-time-workflows
91	Sony Pictures Imageworks	https://www.imageworks.com/
96	Live Link Face app	https://apps.apple.com/us/app/live-link-face/id1495370836
96	Live Link plugin for Unreal Engine	https://docs.unrealengine.com/live-link-in-unreal-engine/
96	Control Rig	https://docs.unrealengine.com/control-rig-in-unreal-engine/
99	Reel FX Animation	https://www.reelfx.com/
102	WildBrain Studios	https://www.wildbrain.com/
106	Sony Pictures Imageworks	https://www.imageworks.com/
108	Reel FX Animation	https://www.reelfx.com/
112	Technical Guide to Linear Content Creation	https://dev.epicgames.com/community/learning/paths/YJ/technical-guide-to-linear-content-creation

112	SVN available through Github	https://docs.github.com/en/get-started/importing-your-projects-to-github/working-with-subversion-on-github
112	Established practices for asset optimization	https://docs.unrealengine.com/testing-and-optimizing-your-content/
112	Hardware and Software Specifications	https://docs.unrealengine.com/hardware-and-software-specifications-for-unreal-engine/
112	Perforce	https://www.perforce.com/blog/vcs/unreal-virtual-production
112	Nanite	https://www.unrealengine.com/blog/understanding-nanite---unreal-engine-5-s-new-virtualized-geometry-system
113	Unreal Engine integration with Maya	https://apps.autodesk.com/MAYA/en/Detail/Index?id=3726213941804942083&appLang=en&os=Win64
113	Unreal Developer Community	https://dev.epicgames.com/community/
113	Technical Guide to Linear Content Creation	https://dev.epicgames.com/community/learning/paths/YJ/technical-guide-to-linear-content-creation
113	Ozone Rig Studio	https://www.ozonerigstudio.com/
113	ShotGrid	https://www.shotgridsoftware.com/
113	Unreal Engine integration with Houdini	https://www.sidefx.com/products/houdini-engine/plugin-ins/unreal-plugin-in/
114	WildBrain Studios	https://www.wildbrain.com/
114	Fireman Sam season 13	https://www.youtube.com/watch?v=LJIWz7D0Vfo
117	Animation hub	https://www.unrealengine.com/animation
117	Download Unreal Engine	https://www.unrealengine.com/download
117	Future of Animation event	https://www.unrealengine.com/events/future-of-animation
120	Animation hub	https://www.unrealengine.com/animation
120	Download Unreal Engine	https://www.unrealengine.com/download



What if you could slash the time it takes to
create high-quality animated content, and at
the same time, gain the creative freedom to
find the perfect way to tell your story?
With Unreal Engine, you can.

Learn more at our [Animation hub](#), and start
your journey into real-time workflows by
[downloading Unreal Engine today](#).

