GDC 2018 **Visual Effects Artist** Roundtable Summary

Here are assorted notes from the 7th annual VFX Roundtable at GDC 2018. This year we had the second ever VFX Boot Camp with spectacular speakers from across the industry, an awesome party (complete with drinks, chicken & waffles), our own schwag, a 2017 VFX Reel, and 3 great roundtable sessions in the all new Moscone Building. Look at us taking it up a notch.

Special Thanks!

It takes a ton of volunteer efforts to bring the VFX presence together for GDC, and before we get to the notes I want to recognize those folks.

Gigantic thanks to Boot Camp organizer **Fred Hooper**. Fred, you killed it. The boot camp was incredible. Also behind the scenes helping were **Christina Wun, Keith Guerrette & Jason Keyser**. Speakers **Ryan Woodward**, **Mike Lyndon**, **Mickaelle Ruckert**, **Shen-Ming Spurgeon**, **Wyeth Johnson**, **Bill Kladis**, **Fred Hooper**, **Jamie DeRuyter**, **Ashley Pinnick**, **Martin La Land Romero**. Coordinator **Megan Bundy**. Sponsors for the VFX Meetup: **PopcornFX, Tilt Brush, Beyond FX** and **Undertone FX**. Shoutout to **Mark Teare** for making #RealTimeVFX signage, pins, compiling the 2017 VFX reel and DJ'ing the party! Thanks to the many kind people who hooked me up with notes that I used to flesh this out. And finally thanks to everyone who joined in the fun either in person or via realtimevfx.com. You make this group so damn special.

The biggest complaint of 2018 (Spoiler: it was also the biggest complaint of 2017)

Why is the VFX Boot Camp scheduled on the same day as the Tech Art Boot Camp!?
Last year I thought I had confirmation that these wouldn't overlap, but due to "reasons" which seem mostly centered around moscone construction, it just couldn't happen. We'll push even harder to get these bootcamps on separate days this time.

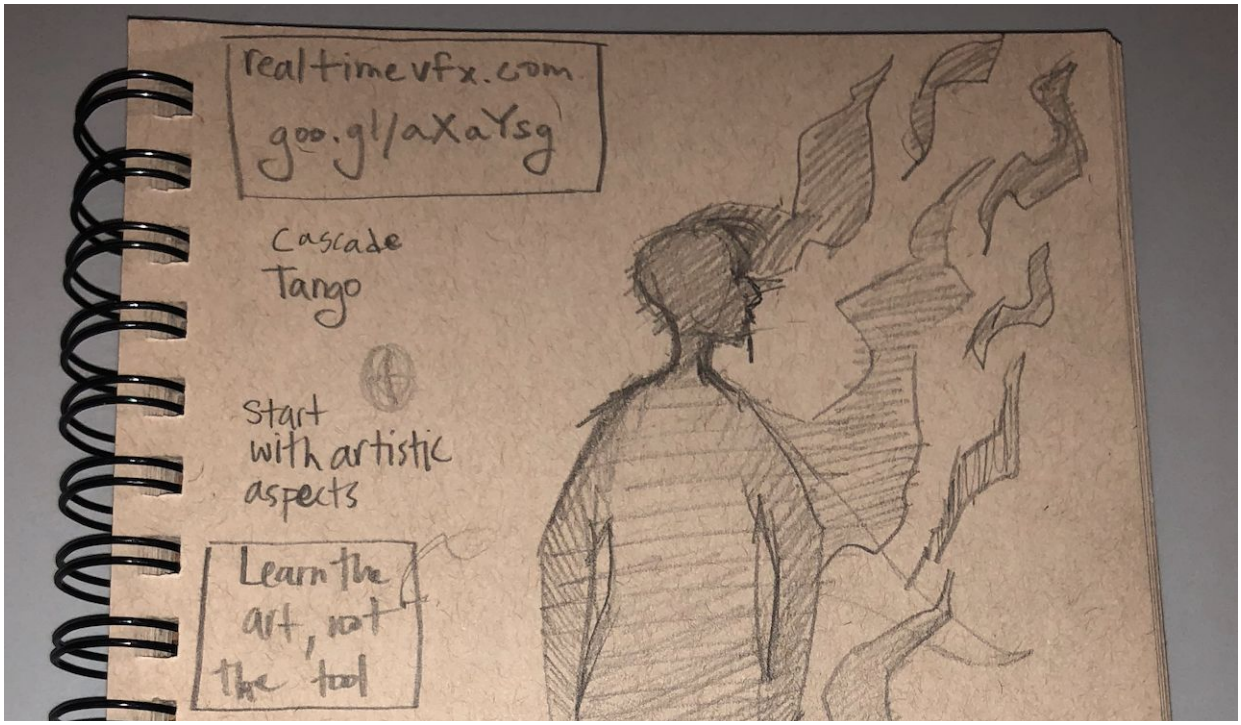
Roundtable Structure

We've landed on a new roundtable structure that seems to be working well, and we'll likely continue it next year:

- Day 1: General Topics largely inspired by the roundtable.
- Day 2: Looking at VFX through an artistic lens (Led by our own Jason Keyser!)
- Day 3: Looking at VFX through a technical lens.

Feedback

The roundtables and the boot camp are all run ad hoc and by volunteers. Because of that, we don't have an official way of collecting feedback (though we do always get the GDC survey responses). Please post on the forum if you have suggestions for how things can be tweaked, changed, or reconfigured for the better!



Attribution: From Alexa Kruckenberg's Sketchbook

DAY 1: General Topics

Where do you start? Advice for beginners.

Luckily we had a larger than normal number of students in this roundtable - hopefully this means schools are starting to focus more on VFX! Our industry continues to be a hard one to break into though. Here were some tips that came out of the discussion:

- You don't need to start with the biggest, beastiest most complicated tool (sorry Houdini). In fact the tools themselves are not the point, you've got to be careful not to get stuck learning a tool when you actually want to be focusing on underlying principles, and the **art** of making an effect.

    - You don't need to know everything when you start! Just a few basic tools in your toolbox is enough to make beautiful effects.
- Look to VFX contests (like those on realtimevfx.com) for useful deadlines to organize your work around. Also consider: CGTalk Challenges, Riot and Blizzard Contests.
- Start simple, use the minimal set of tools / engine features you possibly can.
- Study the masters. Read everything by Joseph Gilland, especially Elemental Magic I & II. Get inspired by Ryan Woodward's VFX Bootcamp talk on the GDC vault (when it's up).
- Reverse engineer other people's FX. For example unity asset store packs. Often shipping games will have assets that you can unpack and dissect (animated sprite textures, etc…)

What's the next big thing in VFX?

i.e. what will we all be talking about next year….

- Real time ray tracing
- Volume fog
- Using realtime fx techniques to accelerate film productions
- Procedural VFX in games where much much more can be interactive. Potentially everything can be destructible.
- All the crazy awesome things that will come out of the new emitter graph systems (see Wyeth Johnson's Niagara talk & Bill Kladis' Fortnight Talk)
- Even more widespread use of Alembic / USD cache workflows

Art Tests. Good? Bad? Discuss.

This was a fun, and heated, topic! Many different perspectives on this one.

- Can be useful to judge how ambitious a candidate is, based on how they interpret open ended tests.
- Can be useful to judge how candidate responds to feedback, criticism and how they ask for clarification.
- Some said you should ask very clear directions similar to what the candidate would experience in production. Others liked the idea of keeping it open ended to see how they request more info.
- Agreement that it's not just about the end result: it's as much about *how* they work.
- Some companies let you use whatever tools, some don't.
- I got called out because I'd forgotten the test I gave Ashley Pinnick was *super duper* open ended: I had simply given her a scene with raw Tilt Brush stroke geometry (no materials or shaders) and asked her to "make it look good" including all shaders, textures, lighting, render settings, etc...
    - Full disclosure: this worked out incredibly well for me and made the hiring decision a no brainer (in this case).
- **The Trap Test**
    - I'd never heard of this, but a trap test is something that is known to be impossible, or at least should not be achievable within the time frame provided.
    - While some argued that this is a fair way of seeing how a candidate will request clarification, others found it… how shall we say… unethical & cruel.
    - Consensus was Trap Tests are probably a bad idea. Use with caution if at all.

Hiring

As always, we take the temperature of who's looking for VFX work and who's hiring. This year, we had more students and new faces! Even though there is still not enough talent to fill the growing need for

Realtime VFX artists, things seem to be moving in a good direction.

- **Students & anyone looking to break into the industry:** be sure to use and participate in the forums on realtimevfx.com. The community is here to help. No questions is a dumb question, we've all been there.

DAY 2: VFX through an **Artistic Lens**

Like last year, Jason Keyser led us through this art focused session on everything **ART** & **VFX**.

Shapes

How do you know what shapes to make? (For example healing vs. hurting buffs)

- Check out Julian Love's GDC Talk
- Categorize gameplay into sections. Reach into these categories and mix and match.
- Research shape language
    - Softer rounder, symmetrical are more pleasant & nicer.
    - Jagged, spiky is harsher. Think Aladdin vs. Jafar


vs


How do you use (or not use) art direction when creating FX?

Dealing with situations where there aren't concepts, or a producer/designer just says: "Make it cool."

- Discuss the game vision with the designer, at the early stage
- Ask for concept art up front. Take ownership.
- Create a mood board. Sit down with the art directors and have them pick from the mood board.
- 75% of the time, a designer is just referencing something from a film or game. MAKE them tell you what it is (you may have to pull it out of them).
- Most work requested by a designer can be really fast to prototype using existing assets (30 minutes). When in doubt, build something as fast as possible and see how it resonates when you show it to them. Then iterate.
- Communicate that VFX isn't magic - it's an art and it takes time like all the other disciplines.
- Don't be afraid to doodle and use thumbnails. Ask: "Is this what you want?"
- Start communication with something more tangible than words
- Show images. You can't expect art references like "art deco" to be meaningful to everyone in a production.
- There was a request to start a new forum thread: **Process. Sharing the early work that went**

into final fx.

- Generally speaking, it would be helpful to share more early process work on the forum.

Reference Material

How do people collect, organize, share their VFX references across the team and across projects?

- Check out Raindrop.io
- Check out Kuadro.
    - Interesting tool built for viewing and collecting reference.
- Make a clip montage in premiere. Have this montage loop in the background on another screen at the office.
- One individual (sorry didn't get your name) has reference material for sale on Gumroad. Said he will post to the forum. Actually, I might have found it here.
- Someone brought up Confluence. I'm not familiar with it, but looks cool.
- Consider Youtube playlists
- Look for novel sources off the beaten path:
    - Demoscene
    - French comics
    - Chemistry experiments on youtube
    - Traditional art / painting / sculpture
    - Sky phenomenon
    - All fantasy effects are based on real phenomenon.
    - Documentaries like "ice finger of death"
    - Psychedelic stuff like oil in water.
    - **Mother nature will own you**

What are examples of how you use traditional VFX techniques in your work?

- Ink drops on paper. Then scan and use as sprites.
- Taking footage of dye in water as flipbooks & reference material.
- Play with fake blood, and other real world substances. See how it looks first hand.
- Using hand painted noise in place of the photoshop cloud operation and other preset operations.
- PBR material scanning from an LED dome can get normal map, depth map, albedo, etc.…
    - Used for debris. Scanned materials + height map displacement (two sided) on top of traditional sprites. Looks physical but still very fast, and can be switched to standard sprite rendering at low LODs.

Exploring workflows that include paintovers or sketches in VFX

- This is a common practice in film, but really hasn't been very prevalent in games.
- You can take a complex 3D simulation, render it out as a flip book and put in engine. Then modify that flipbook (using any means necessary) until the visuals match what's need. Finally push those changes back to the original simulation.
- Jason mentioned that he has used sketches to provide feedback on VFX.

What are some examples of how VFX influenced, or started, a new project?

- See Insanely **Twisted Shadow Planet**, created by the legend himself: Michel Gagne.
- **Dead island 2** - some examples were shared on how the VFX drove core game mechanics
- Though not a game, **Tilt Brush** is another example of a project that was heavily driven by VFX

Rapid prototyping approaches

- Keep it simple. Use big shapes and broad strokes.
- Rapid prototyping can help blank page syndrome. It's hard to get started if you're trying to make something perfect.
- As part of this process, try and teach people how to look at unfinished work.
- Get designers an empty effect / test particle to put in the engine. Often this is the most time consuming aspect of a large production. Just stub something out and then polish it up later.

Watch out for the swoop and poop!

- Watch out for Art Directors swooping in, taking a dump on your work, and flying away. A close relative of the seagull manager. Speak up - it's not good for anyone!

DAY 3: VFX through a **Technical Lens**

Digging into the nitty gritty!

What are examples tech we should be using more more often in VFX

- Interesting methods of normal approximation for fun lighting tricks.
    - e.g. for clouds of particles, find vectors between center of the system and each particle, use it to for the normal as a volumetric approximation.
- Shadow buffer, use it! (So particles look different in/out of sunlight)
- Check out Doom 2016 tech.

What are some basic tricks for optimizing effects for VR?

- Definitely check out Shen & Wyeth's boot camp talk.
- Be sure to use **Profiling tools** and check for common gotchas
- **Overdraw control**. Lower particle count when closer (cull/LODs).
- Use two layers, one where it cuts out earlier, jittered fadeout.
- Avoid overdraw by minimizing sprites with a ton of detail. Sprites with lots of details (like many small dots) won't work well in VR.
- **Do more with meshes & shaders**. Model shapes w/scrolling, distorted textures on them.
    - Something that looks complicated with 3 layers rather than a whole bunch. Explore Vertex noise that wobbles around.
- **To help find problem spots:** Record off of video from play through, track states from the GPU/CPU spikes. You can diff between the video and profiling tools to understand exactly where and how VFX messes with framerate.

Anyone using PBR on their effect materials?

- One team was using PBR on particles, and it effectively translated to less crunch at the end of the project (by actually using the lighting *in* the scene if a the scene gets relit).
- **Show of hands**: only a few people were using PBR in their particles (but this is still an improvement from previous years)
- It was difficult to get effects artists to turn on lighting at first, but it saves a ton of time. It was important to convince VFX artists to *not* fix lighting behind the scenes (in the effect) and instead get the actual scene lighting fixed.

Who's using GPU emitter graph effect systems like Niagara?

- Bungie uses node-based emitter system. They were already familiar with node graphs for their shaders. Had good stuff w/black boxing, inheriting, templating.
- Bungie, Epic (Unreal) and DICE seem to be 3 of the big ones tackling this at the moment.

If GPU Emitter graphs are the future… how do VFX Artists feel about it?

Is it scary having to know all the *maths??*

- One artist was currently using Cascade in their project and are wondering whether it's a complete redo to upgrade.
    - The Battlefront 2 talk suggested that, at least in their case, the legacy FX system and new emitter graph system co-existed. This is likely what would happen in production for Cacase vs. Niagara as well.
- One artist had a version of this idea at Avalanche without tooling. It's hard to give up that level of control (using states, variables, modularity).
- Many folks were quite excited for Niagara, having seen how much is possible. However, not excited about losing the easy things you get for free in Cascade, and having to re-invent the wheel to do them.
- Lots of enthusiasm around conditional effects, reactive particles, cross-talking, etc…
- Some concern was expressed that these changes would push VFX more technical, and even less art driven (which is already a problem by most accounts).

Ways to move away from data entry towards true creative expression

At the very end of Epic's talk discussed Niagra 2.0: "The future is how do we get away from an effects artist's job being just typing in float fields/data entry as a form of art?" What's a more visual way for artists to author VFX?

- Something SideFX has been thinking on — how do we help artists do more complex stuff? Widgets in a viewport, for example.
- I don't want a number from my velocity, I want an effector in the world I can move in the world and see the effects in realtime
- Better ways to use primitives and other ways to represent data and that connection so we don't have to always just use numbers
- Design multiple input systems: precise, large scale systems. It's really easy to show how data gets input in different ways (Vicarious Visions). If you could show some kind of debug lines, bounce parameters, etc.

Thoughts about Machine Learning in RTVFX?

Are we just going to not have jobs anymore?!

- Did everyone see the Siggraph real time live video, driven by ML?
- ML with physics trees — give it a bunch of videos and say it'll give you a realtime visual effect. Training data isn't well generalized though, so it's specific to one thing.
- Video analysis for vector field production. Could you take a pic of an explosion, take a couple frames to do an optical flow vector map/vector field?

What are some techniques for visually balancing lit particles with other PBR surfaces?

Many folks hesitate to use true PBR on their particles for performance reasons. In that case, how do you make the particles appear visually correct next to true PBR materials?

- Suggestion: maybe try slightly emissive, use a fraction of the realtime light?

- Bungie does lighting per particle, per vertex. Sample the probe then apply the color to a given vertex.
- Suggestion: take the average color of the screen to use as the 'environment color' for the particle.
- **Production tested suggestion:** Tessellate sprites to get more lighting in the verts, then do per vert PBR lighting.
- **Production tested suggestion:** Try using voxel lighting at very low fidelity as a pre-pass to get consistent lighting across the entire effect (so individual cards wouldn't be distinct).
    - General rule of thumb: perform expensive lighting calculations at the most granular level possible, saving only the cheapest operations for the final per-pixel computation.
- Avoid LODs in tessellated particles because the pop is awful. 4x4 grid of verts usually works and is OK for performance..
- Try converting color spaces and only do expensive operations on the luminance. Spend your cycles where they matter. Separate low freq from high freq as much as possible.
- Suggestion: In AR people are doing more and more interesting screen space approximation light light estimation / sampling the color of different areas. Might be an interesting way to get "cheap" particle shading that still feels integrated in various lighting scenarios.

Let's talk more about this voxel solution for lighting...

- It's been doable for a while. Assassin's Creed black flag used voxel lighting w/sampling clouds/fog, and voxel fog concepts applied to particle systems.
- Implementation suggestion:
    - Compute job that's got a fixed function generic lighting solution
    - Set the bounds of your particle system, then write into an RGB buffer with data like color, light energy, density, etc.
- Once you have a voxel structure, you can raymarch a shadow map as a voxel, not in the pixel shader
- Multi-sample shadow map in a low frequency voxel solution and vertex shader (not pixel shader)
- For papers/reference just look for "voxel fog."

How to make a dust cloud fast

Question: "When I want to make a dust cloud I draw smoke as an animated atlas and put many of these smoke sprites together, on particles, to create a dust cloud. This ends up creating a ton of overdraw. Is that hilariously naive?"

- Instead of using particles, pan your dust in the shader rather than multiple overdraw passes
- Trim out the unused space in your sprite geometry.
    - You can trim out the edges in Cascade as well.
- Try changing the particle size so that frames which have smaller drawn elements in them actually correspond to smaller particles. This means that your atlas will need to look different though, or you'll need to modulate the UV scale along with the sprite animation in script somehow.
- Instead of drawing on a sprite, have the atlas play on a mesh that approximates the end shape. Draw the frames as if they're moving, but then scale the mesh to get the solution you want.

Perf Tricks

Perf and optimization analysis. Any cool ways you're doing it?

- Bring up a debug console and show the emitter list. We had a field in the effects editor that would filter any effects you wanted so you could see the performance load of any subset of effects. Makes it really easy to narrow down on problem areas.

- Engine profilers should list both the CPU cost of an emitter system, and also the GPU cost of any given particle draw call. One of the columns records those things, sort by the columns to find the worst offenders. Will steer you in the right direction.
- In Tilt Brush: we batch load a bunch of different sketches, then generate perf graphs which provides insight into the cost of brushes on different platforms. Helpful when dealing with UGC content.
- Show of hands: “Who has visual filters for GPU traces etc in the live view?” Lots of folks!

The Lightning Round

Like last year we did the lightning round at the end of the final roundtable. Many questions! Singular answers!

Question: Has anyone been playing with raymarching and mesh flipbooks? I’d like to hear more.
Answer: Bill (Kladis) did a flipbook talk last year. Check it out: Art Direction VFX for Stylized Games.

Question: I want to iterate on delta time in a shader and want it delta time to be relative to particle birth. How do I get that in my shader, in Unity?
Answer: In Unity hijack color.a (or another channel) to pass through the particle start time. Anyone know what the right solution is to this using custom per particle attributes?

Question: I’m curious about solutions for sorting in a stylized way 2D sprites into 3D spaces without weird billboarding issues where sprites intersect with walls?
Answer: There’s a talk at 1:30 (Volition) that’s literally about how to handle rendering. A big part of it is about lighting and transparency. Should be able to help point you in the right direction. See Rendering Technology in ‘Agents of Mayhem.’

Question: In Unreal in VR there’s a checkbox to cancel out HMD display so the particles don’t rotate around when you’re tilting your head. Is there a way to do that in Unity so the rotation doesn’t happen?
Answer: Use meshes/a plane instead of actual sprites because you can have more control over orientation. Bias certain axes in the shader? Again, similar for the Volition talk.

Question: Any and all techniques for shader driven stylization in effects?
Answer: Out of time: A thread about that would be fun! Style transfer from paintings to rendering of a particle???

Craziness! Til next year.