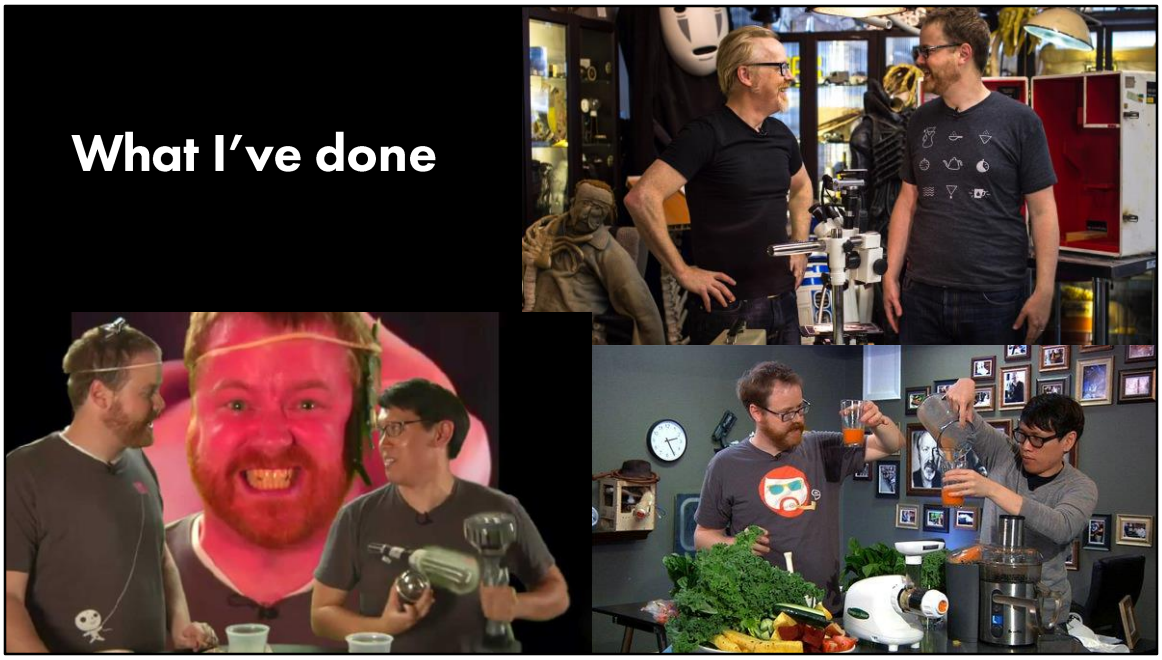
Building the  ture

Will Smith, Andre Infante

I'm Will Smith

I'm Andre Infante

We're roughly 2/3rds of FOO VR. For the last year and a half, we've been working with Sindre Skaare and some other folks to build FOO



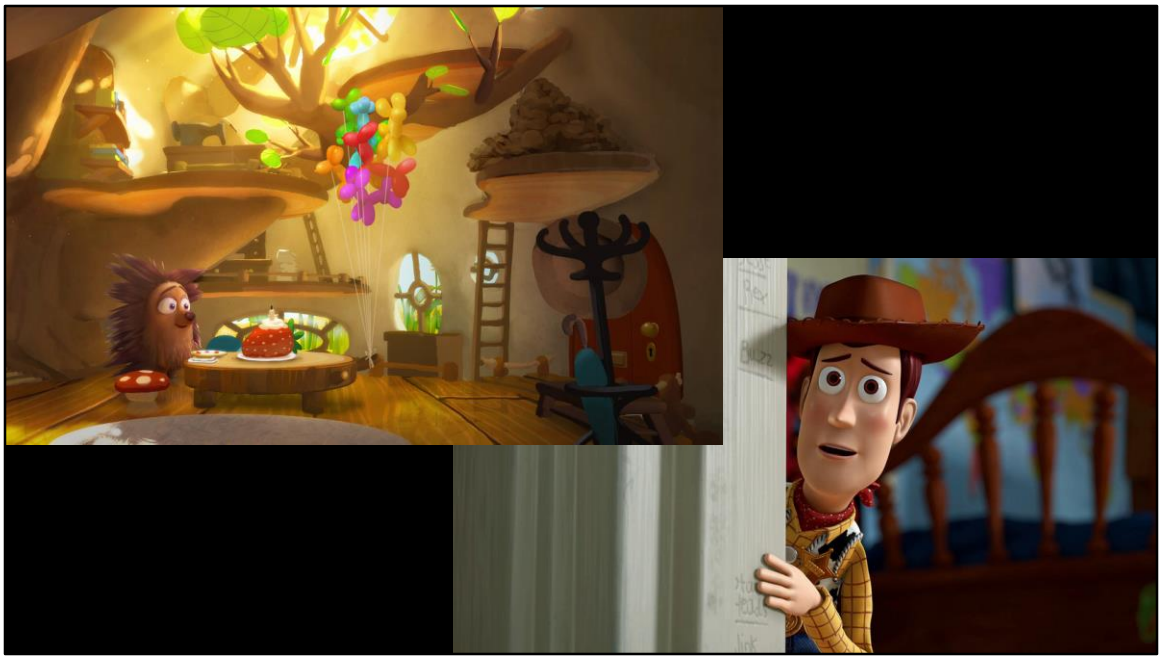
Before we started FOO, I worked on a website and Youtube channel called Tested. We had a really small staff, just two on-camera people and one producer, but we wanted to produce a LOT of long videos. Over the 6ish years I was at Tested, we made something like 2400 videos, or roughly eight videos a week.

We were able to do this much work with such a small staff by shooting lots of our long videos live-to-tape. Rather than add chyrons and insert images in post-production, we prepared them in advance and live mixed them in while we were recording. After the shoot was done, we usually had videos that were ready to upload to YouTube. And yes, the results were occasionally janky, but the audience didn't mind, as long as whatever we showed them was interesting.

Then, I got kind of into VR.

And then, I got into VR.

I started looking at what it would take to make the kind of shows we made at Tested for VR audiences. At first, 360 video seemed like an obvious choice—the content pipeline is more or less the same as normal video and all I'd have to do was get some decent 360 cameras and learn how to use them. But after I used the tech, I wasn't impressed. Looking toward the future, 360 is clearly a bridge technology—it will get video production houses on VR platforms, but the experience for users isn't good. When you ask viewers to let you take over their two favorite senses, you need to fully engage them. Putting them inside a non-interactive video bubble wasn't going to cut it.



That left more traditional 3D animation. It's interesting, because as Oculus's Story Studio, Penrose, WeVR and others have shown, you can tell powerful, compelling stories in amazing interactive worlds using traditional 3D animation. But I don't need to tell you all the problems with using 3D animation for a daily or weekly show. It's expensive and incredibly time consuming to make animated shorts the same way Pixar or Oculus Story Studio does. If you want to run a daily or weekly VR show, traditional 3D animation isn't the right tool.

What is FOO?

- FOO brings the live TV studio to VR by using procedural animation
- Creators front-load art creation, then produce regular content quickly using off-the-shelf hardware
- Cost structure is closer to 360 video than 3D animation
- Using the same hardware as users offers interesting opportunities

We designed FOO for episodic shows. We have essentially built a live television studio in VR. We use procedural animation and the least common denominator hardware, the Vive or Oculus Touch, to capture performances and distribute them as fully-interactive 3D-rendered experiences for the audience to explore with us.

When you watch a show that was recorded in FOO, you don't feel like you're watching a TV show. Instead, you feel like you were in the room when the recording happened. You can follow along with the conversation, but you also have agency. If you want, you can go explore an entirely different part of the scene.

The goal is to be able to record a show, spend a little time cleaning it up and doing light edits, and then publish an episode shortly after it's recorded for viewers to experience in VR.

The cost structure for what we're doing is much closer to 360 video than traditional 3D animation. If you think about a show like Friends or South Park, most of the action takes place in relatively few sets. Most episodes feature the same characters. So, if you're recording on the same sets, with the same characters and props, your per-episode art costs amortize to zero. It turns out that the more you use your art, the

cheaper it gets.

Andre will talk about this more in a moment, but because we're interpolating from relatively few data points, we avoid some of the problems of using traditional mocap—like jitter—as well. Naturally, we introduce our own set of nightmarish errors to counteract that

Why use the same hardware as users? It offers some really interesting opportunities. Anyone with a VR headset can make shows with FOO. Least common denominator hardware gives creators a ton of flexibility, I gave a talk in Stockholm from my office in San Francisco few weeks ago.

The FOO Approach To Procedural Animation

Now that we've talked about boring stuff like cost structures, Andre's going to explain a bit about how we animate full body characters using only the Vive

Procedural Animation Is Really Hard

- Tracked data is sparse
- Humans are sensitive to incorrect human kinematics
- The obvious cheats don't work well
- Authenticity > accuracy

Hopefully we've convinced you that procedural animation is a useful tool.

Here's why it's tricky:
HMDs track eighteen degrees of freedom (six for each limb and the head) - the human body has hundreds. Therefore, most of the data we need to present to the user needs to be generated programmatically

But generating all this data can be tricky. Humans are very sensitive to how other humans move. If you get it wrong, the results can be very creepy.

Because this is so challenging, people use a couple of straightforward cheats to side-step the problem. Stuff like rigidly mounting the torso onto the user's head, or only rendering the head and hands, or not giving the user legs.

Unfortunately, these cheats have their own problems. Rigid torsos fail in some common use cases. Head-and-hands avatar are tough to parse, especially in busy scenes. Leg and body-less avatars don't feel physically grounded in the scene.

In order to solve this problem properly, we need avatars that both look and move like

people.

Challenging Cases

- Touching the back of the head
- Jogging / Jumping
- Most torso systems fail at one of the following:
  - Head gestures
  - Leaning
  - Walking

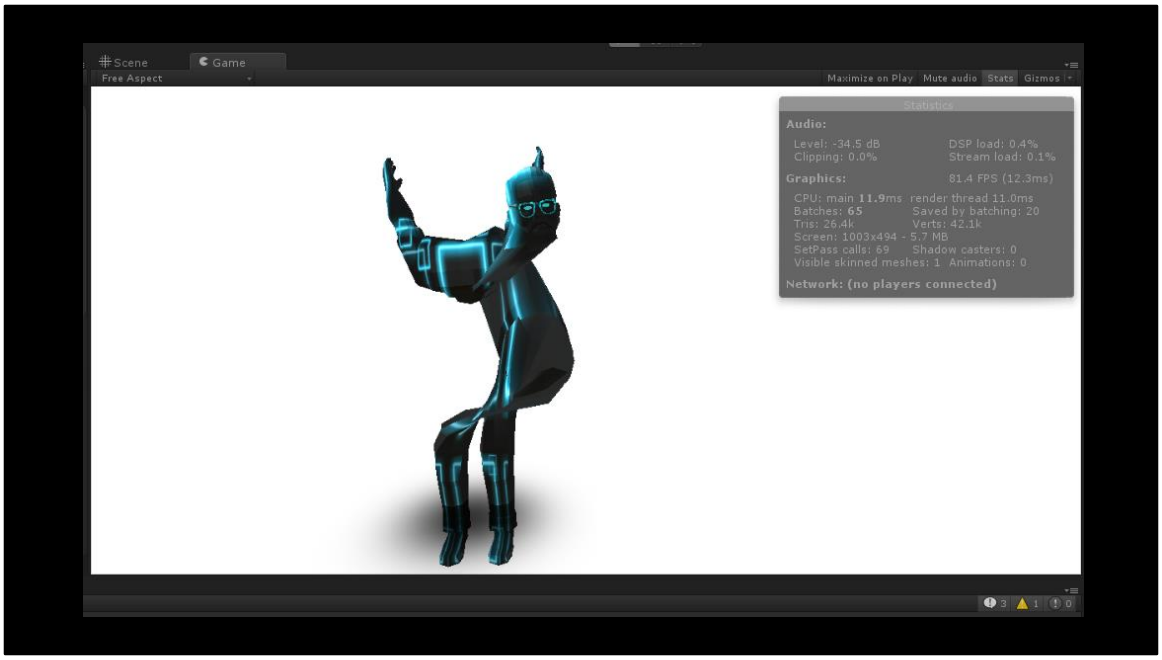
It can helpful to think about cases that are difficult to handle well, then design for them specifically.

One hard cases is touching the back of the head. By default, IK solvers want to keep the elbow below and behind the user's hands. And, most of the time, that's the correct solution to the problem. But if you scratch the back of your neck, suddenly your elbow is in totally the wrong place.

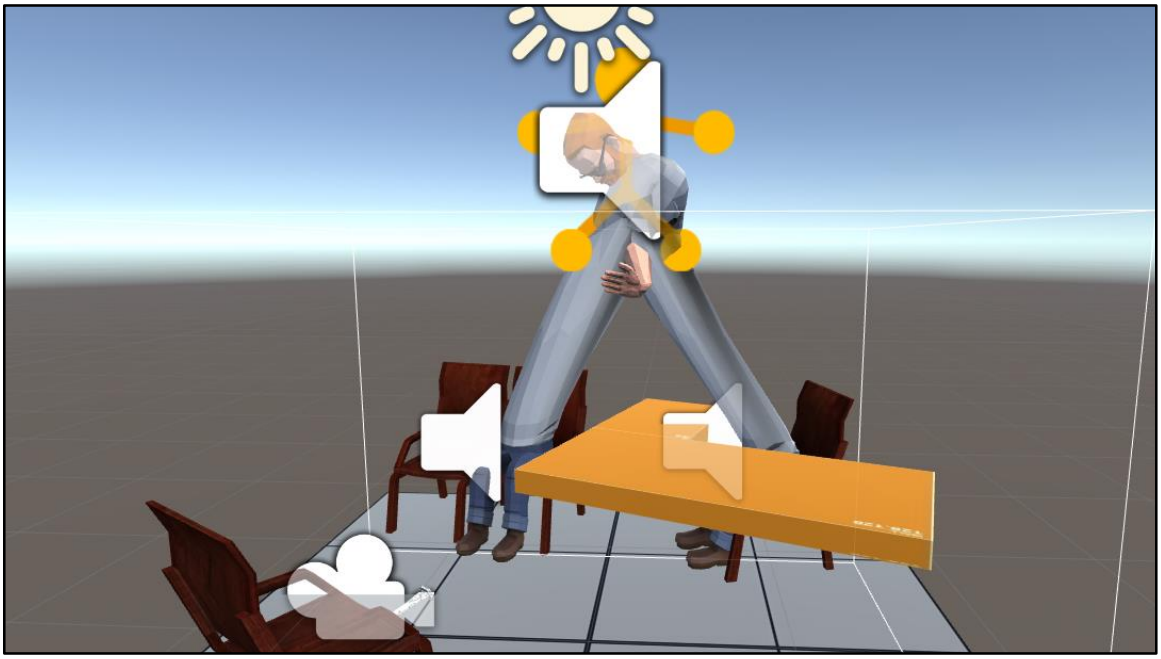
Moving quickly is another challenging case: providing plausible body movements get a lot harder if they body is rapidly shifting its balance around.

There's also a set of three things that very few of the standard approaches to the torso problem do well. If you choose to mount the torso rigidly to the head, nodding and shaking your head causes the whole body to shake. If you hang the body from the head like a pendulum, you lose the ability to bow or lean. If you fix the hips in space, you lose the ability to walk around and have the body make sense.

So what you're seeing here is a bunch of different mechanisms and heuristics we tried for generating animation. Most of them didn't work. But, because we were able to crank out a prototype in under a day in many cases, we were able to quickly discovered which avenues were promising and which were not.

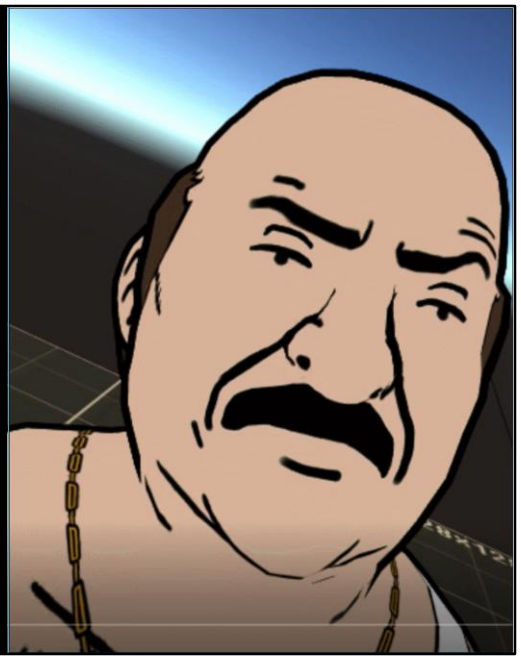

Scene
Game
Free Aspect
Maximize on Play
Mute audio
Stats
Gizmos
Statistics
Audio:
Level: -34.5 dB
Clipping: 0.0%
DSP load: 0.4%
Stream load: 0.1%
Graphics:
81.4 FPS (12.3ms)
CPU: main 11.9ms render thread 11.0ms
Batches: 65
Saved by batching: 20
Tris: 26.4k
Verts: 42.1k
Screen: 1003x494 - 5.7 MB
SetPass calls: 69
Shadow casters: 0
Visible skinned meshes: 1
Animations: 0
Network: (no players connected)


Persp



Our Approach

- Plausibility is our goal
- Fail gracefully
- Recover from mistakes quickly
- Users will forgive weirdness
- Art-style sets expectations
- Cartoons can get away with more than photorealistic humans



The approach we've settled on is to try to make reasonable assumptions about how the user moves. This doesn't always give the right answers, but it fails in reasonable ways. The important thing is not to get it right 100% of the time, but to capture the kinds of motions that we care about.

That said, our avatars do sometimes glitch and make unphysical motions. To help mitigate this, we use stylized avatars to cue audiences that these are cartoon characters, not photorealistic humans. This makes it less upsetting when the illusion is briefly broken.

Expectations from Art: Normal VR

Normal VR does a great job of using art to set up expectations for their avatars
Even though the puppet only roughly approximates the motion of the user, the art allows us to accept the fiction that the character is alive and inhabited.

What Works Well

- Heuristics for torso pose
- Facial animation driven by audio
- Machine learning for elbow position
- Balance analysis and IK for foot position
- Dynamic eyes based on points of interest

For torso pose, we use heuristics that try to capture lean / squat dynamics by analyzing the motion of the head to determine if it's best explained by rotation or translation.

Our faces are animated off of raw audio intensity, using the mouth and the eyebrows. Our faces don't have expressions, but they make it clear who is talking, and how loudly they're speaking. The subtle motions also break up the face and make it feel more alive.

Elbows fail in several cases for standard IK. To fix this, I made a mocap suit out of Vive controllers, recorded thousands of examples of hand-elbow relationships, and used a BEAM search to fit a high-order polynomial to the data. This works surprisingly well

For balance, we have an ideal placement for each foot, we adjust that placement based on velocity, and figure out which foot to move to minimize our total error. Then we use IK to move the leg to its new target.

For our eyes, we use a system that's aware of points of interest in the environment and automatically switches between plausible gaze targets, while simulating saccades

and blinking. This is a really simple approach, but works so well that we've had people ask how our eye tracking works.

The Heuristic / Constraint Model

- Rules of thumb can fail catastrophically
- Averaging several rules improves performance
- Adding common-sense constraints dramatically improves performance

Heuristics work well most of the time, especially if you average several that work in different ways.

However, sometimes they get wacky. So it's helpful to set rules about what they should never do, and use those rules to limit the output of the heuristics. This is the heuristic-constraint model., and is a good way to think about many of these problems.

Where Our Approach Breaks Down

- Torso model breaks down in edge cases
- Model assumes rigid spine
- Foot placement is crude
- Face and hands aren't very expressive
- Body freaks out when head is close to ground

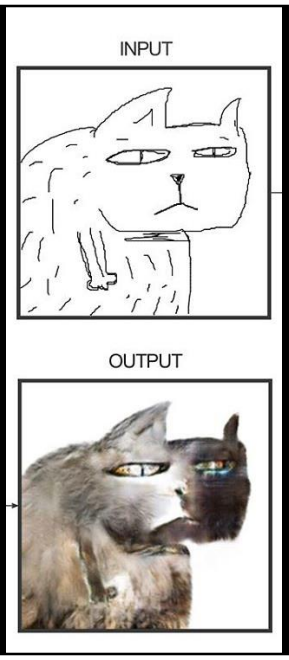
The torso logic makes some broad-strokes assumptions about the way people tend to move. Moving in unusual ways can lead to a breakdown of those assumptions, which leads to the body doing weird things.

The model also treats the spine as essentially a rod. This simplifies a lot of the math, but can make the body feel rather stiff. We'd like to improve on this in the future.

We also have a problem that some of the rules we use break down as the head gets very close to the ground. If you think about it, a body that's fully extended is much more constrained than a body that's neat the ground. There's only one way to stand up straight, but there are many ways to be folded. You can see what happens if the user puts the HMD on the ground in this video.

Future Refinements

- Neural network for leg pose
- Predictive finger poses / eliminating paddle-hands
- Audio analysis for facial expression
- Flexible spine model
- More data from additional optional trackers (Call us HTC?)



We're interested in using machine learning to predict facial expressions, hand gestures, and other features of the performance that we can't directly track and aren't critical to capture perfectly. We'd also like to use neural networks to learn to predict which foot positions will be most useful, to get better balance behavior without awkward corrections. There are also some specific issues with our model that we can patch by adding new behaviors.

Takeaways

- Don't be afraid to try things that are impossible
- Do fast, cheap experiments to validate approaches
- Don't let perfect be the enemy of good enough
- Test often, collect feedback
- We made too many demos of the wrong product
- THIS BULLET INTENTIONALLY LEFT BLANK
- Source control is a necessary evil

People told us using 3 data points to animate human using IK was impossible.

Fast, cheap experiments let us know when to dig deeper and when to bail on potential implementation

We could spend infinite time tuning animation. Our approach won't work if you want 1:1 tracking of human bodies, but if you can tolerate imperfection, we offer a unique way to reach an audience

Testing new implementations and collecting both active and passive feedback was incredibly useful. Adding analytics to the first episode of the show completely changed our approach going forward.

When we started out, we made a bunch of demos that were essentially proof of concepts to show to potential investors, venture capitalists and the like. Those demos were great for us, as they gave us a good idea of what challenges we'd face as we moved toward making real products, but they weren't what the investors wanted to see and made us look like a studio instead of a technology company. This wasn't good for our chances with VCs.

After the first episode was out, we got a lot of interest from a ton of big-name media companies. But we probably weren't ready with our product or as a company for that kind of project.. Additionally, if we'd spent the time we spent making demos working on the product, we would have been ready to release the first season of the show in the fall instead of the spring.

When we were

Thanks!

Find us online at www.foovr.com

Twitter:
Will Smith - @willsmith
Andre Infante - @AndreTi