



Hand Tracked Controls: Design and Implementation for Serious VR

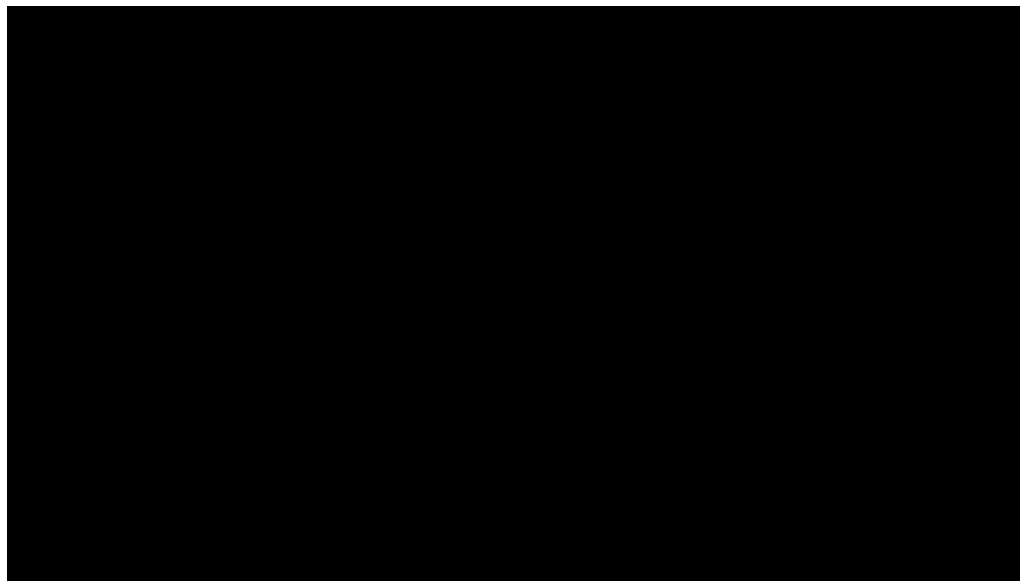
Matt Newport

CTO/Co-founder - Osso VR

A bit about me

- Matt Newport – matt@mattnewport.com
- CTO/Co-founder Osso VR
 - <http://ossovr.com/>
 - VR surgical training
- Started in games industry 2000
 - Software engineer with a focus on rendering
 - EA, Relic (THQ), various smaller studios
- Left games to work on Serious VR 2014

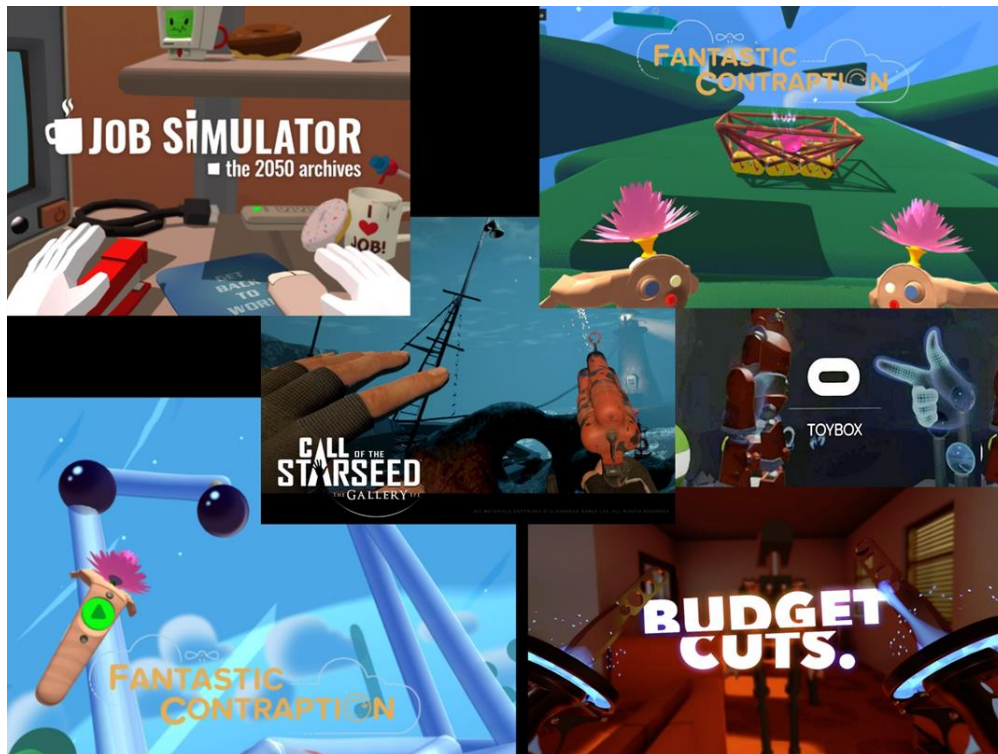
Osso VR



Overview

- Basic design decisions for hand tracked controls
 - Show hands or controllers?
 - Pre-set or flexible grips ('tomato presence')
 - How to handle hands passing through solid objects
- Hand tracking tips and tricks
 - Dropped objects
 - Passing from hand to hand
- The importance of throwing in VR
- Implementing hand tracked interactions

Hands or Controllers?



Picking Stuff Up

- Two main choices for hand or controller:
 - Remains visible
 - Disappears
- This is best illustrated with a GIF!



Disappearing Hands?

Show object held in hand

- Requires one or more pre-set grip positions for each object
- Hand position in VR doesn't match user's actual hand position
- User can't freely adjust grip
- Looks more natural on video

Hand disappears

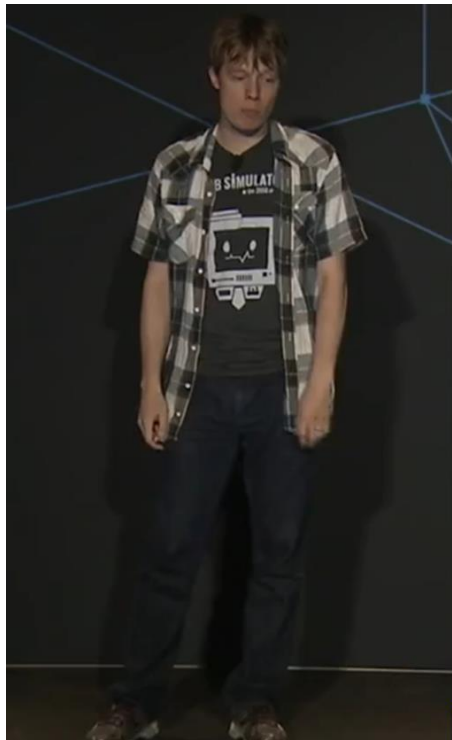
- Doesn't require any custom setup per object
- Users don't seem to mind / notice
- Users can freely adjust their grip
- Can look strange on video

Show Objects in Hand: The Gallery

(Video Review by FunshineX)



Hand Disappears



Tomato Presence (this needs to become a phrase)

- Hand presence stand-in!
- As long as something is being tracked 1:1 with your controller



Hands passing through objects

- Hands in VR should always track 1:1 with real hands
- How to handle passing through solid objects?
- Our solution: allow hands to pass but held objects break out of grip

Hands passing through objects

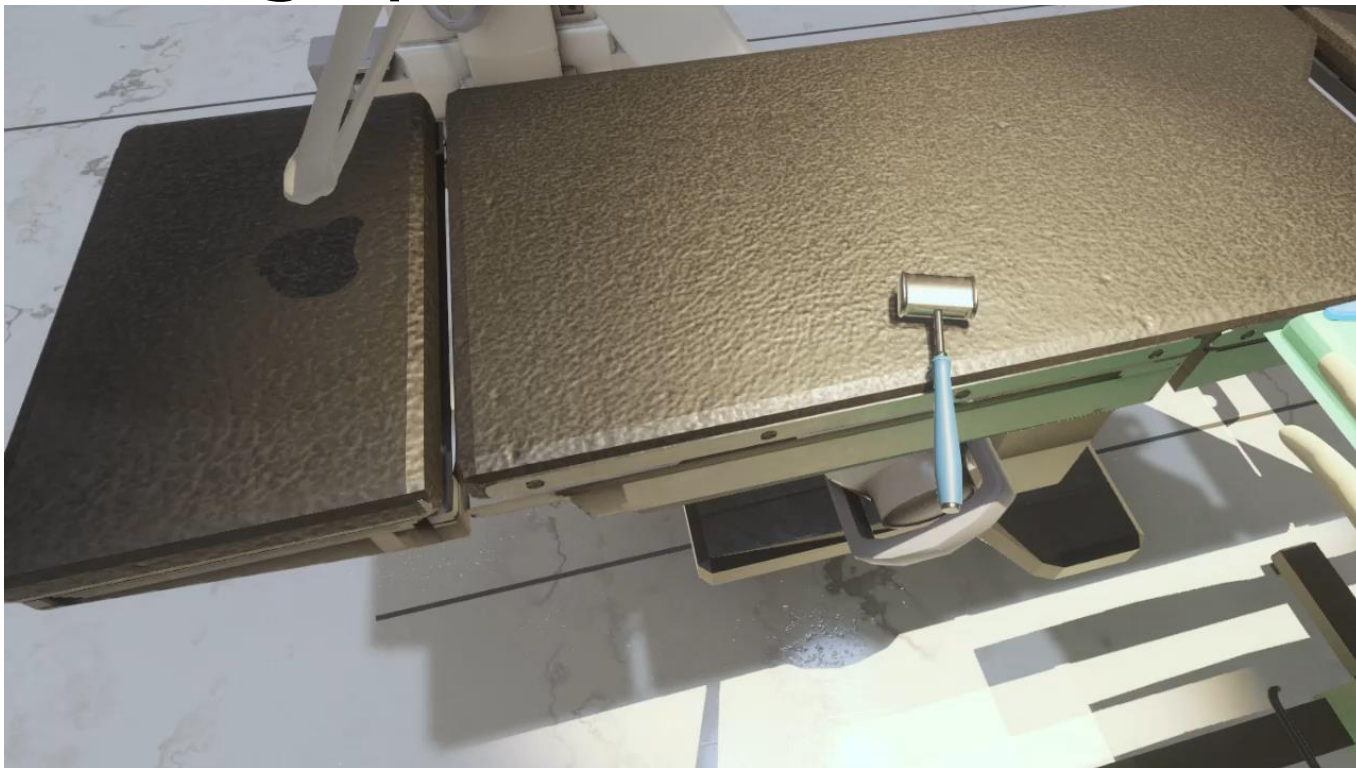




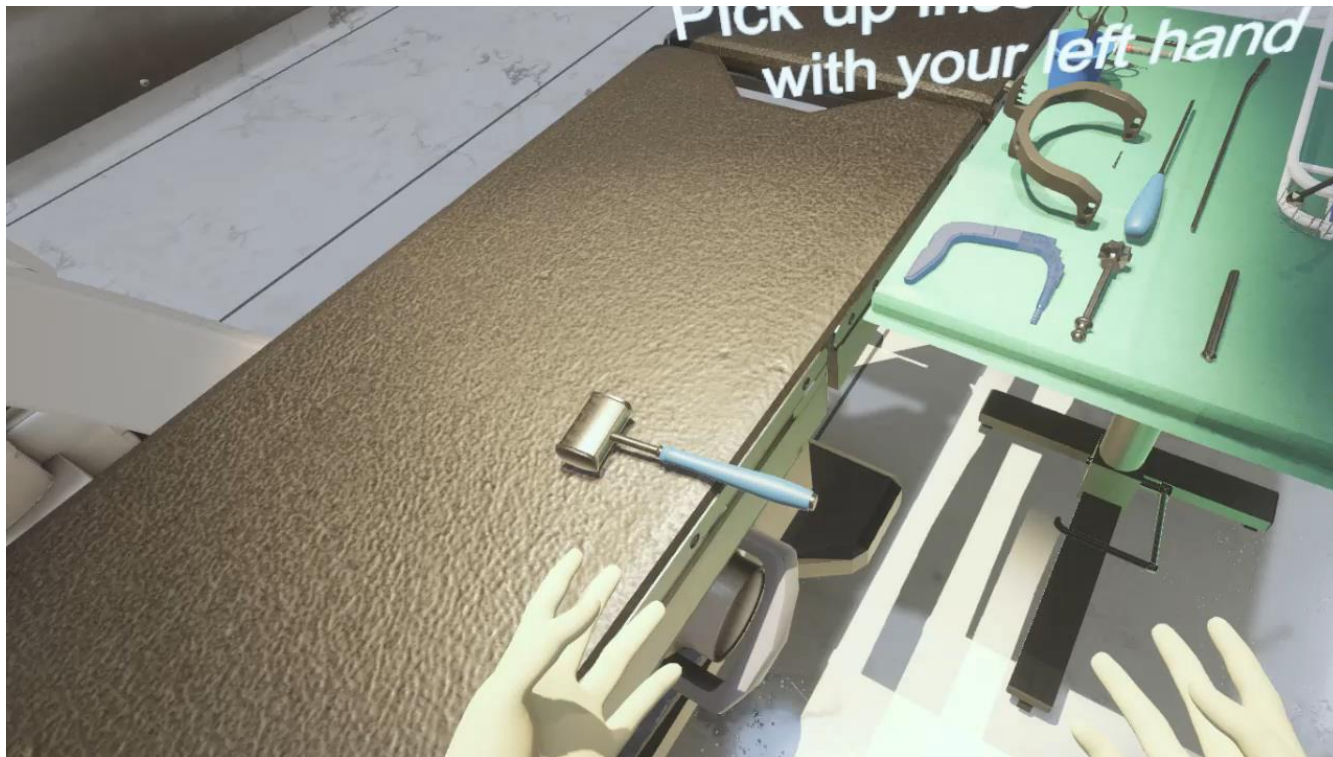
Downside of showing hands

There can be multiple different natural grips for an object depending on the user's intent (pour, carry, use as a weapon)

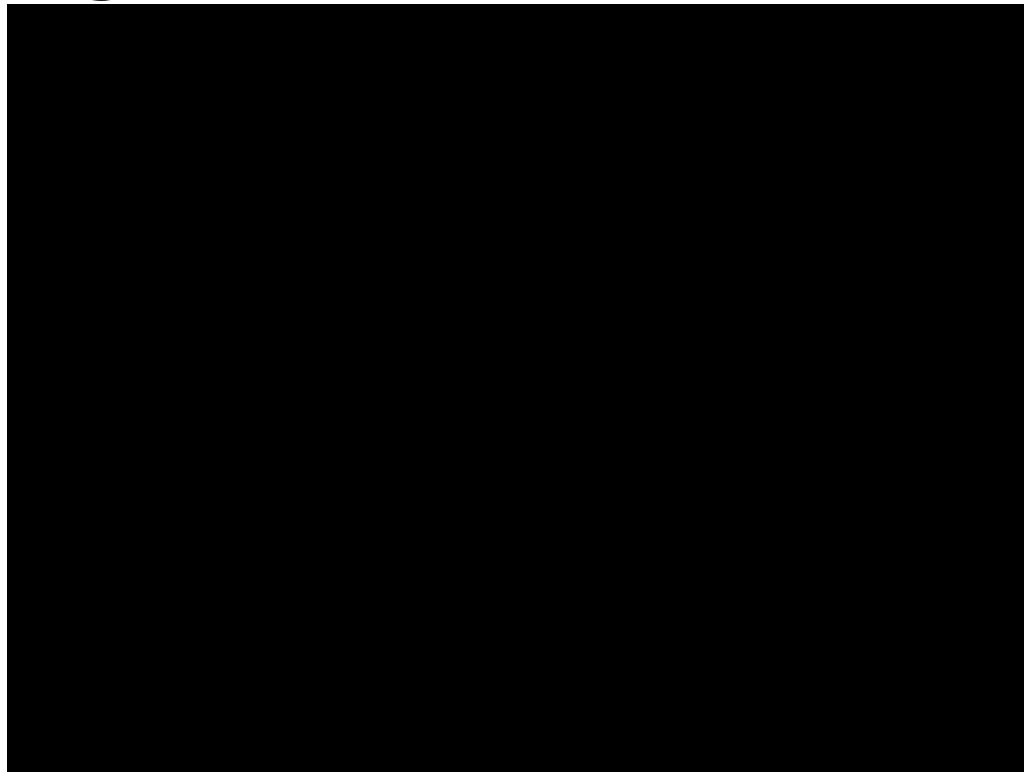
Flexible grip choice



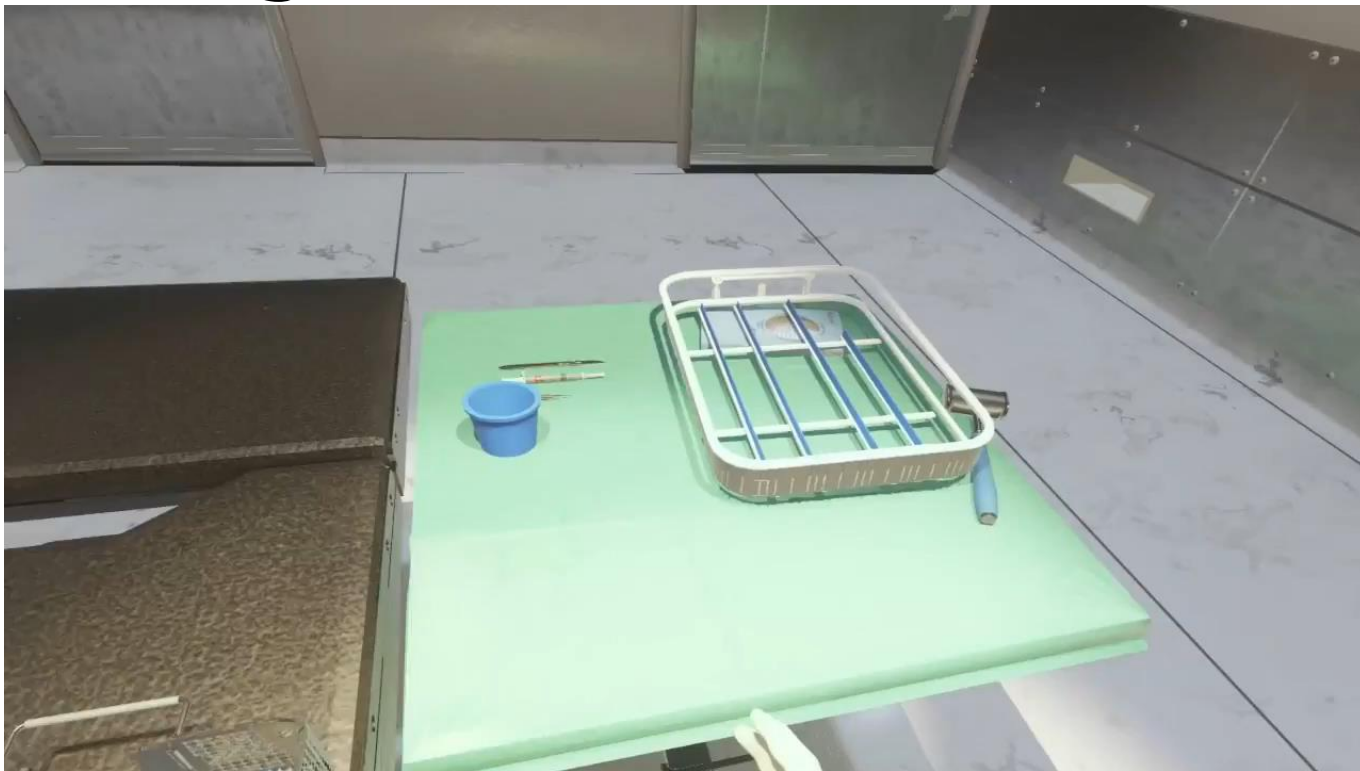
Passing from hand to hand



Throwing



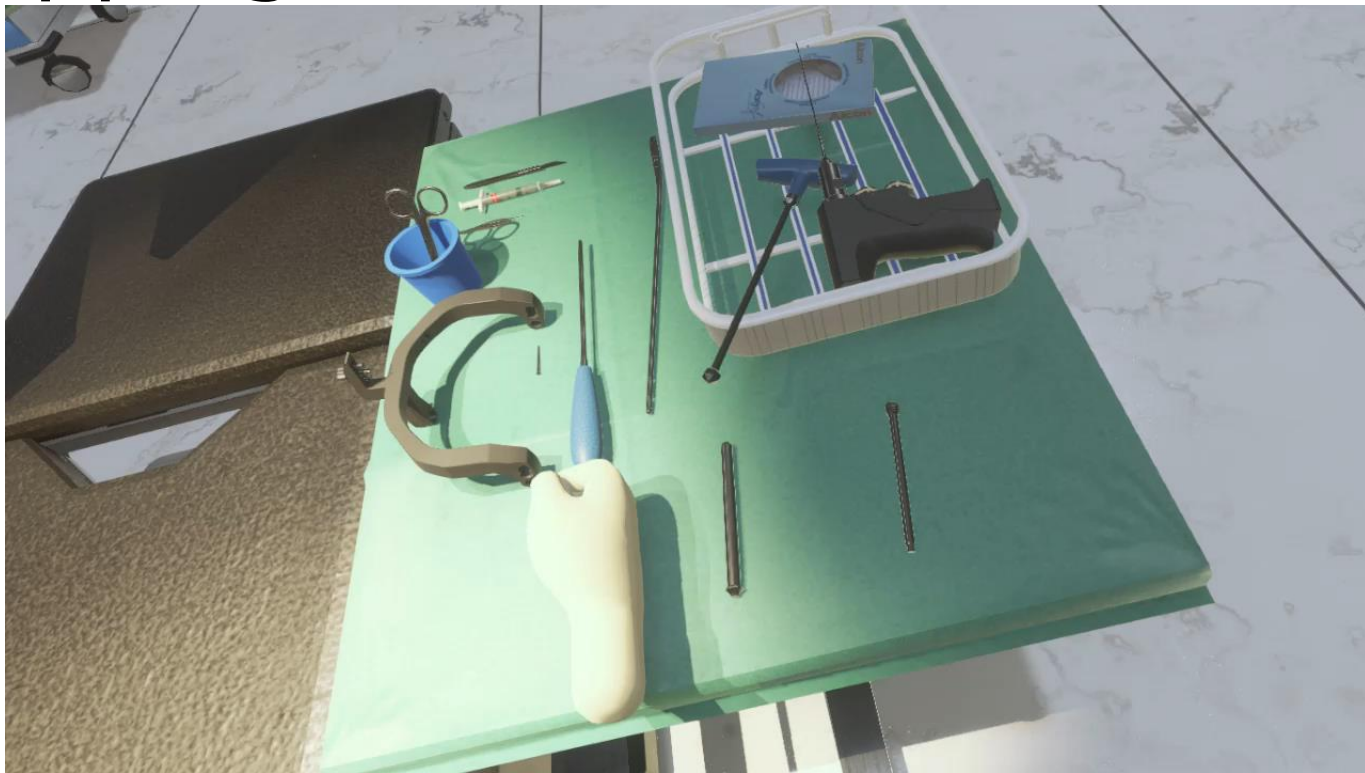
Throwing



Dropping stuff on the floor



Dropping stuff on the floor



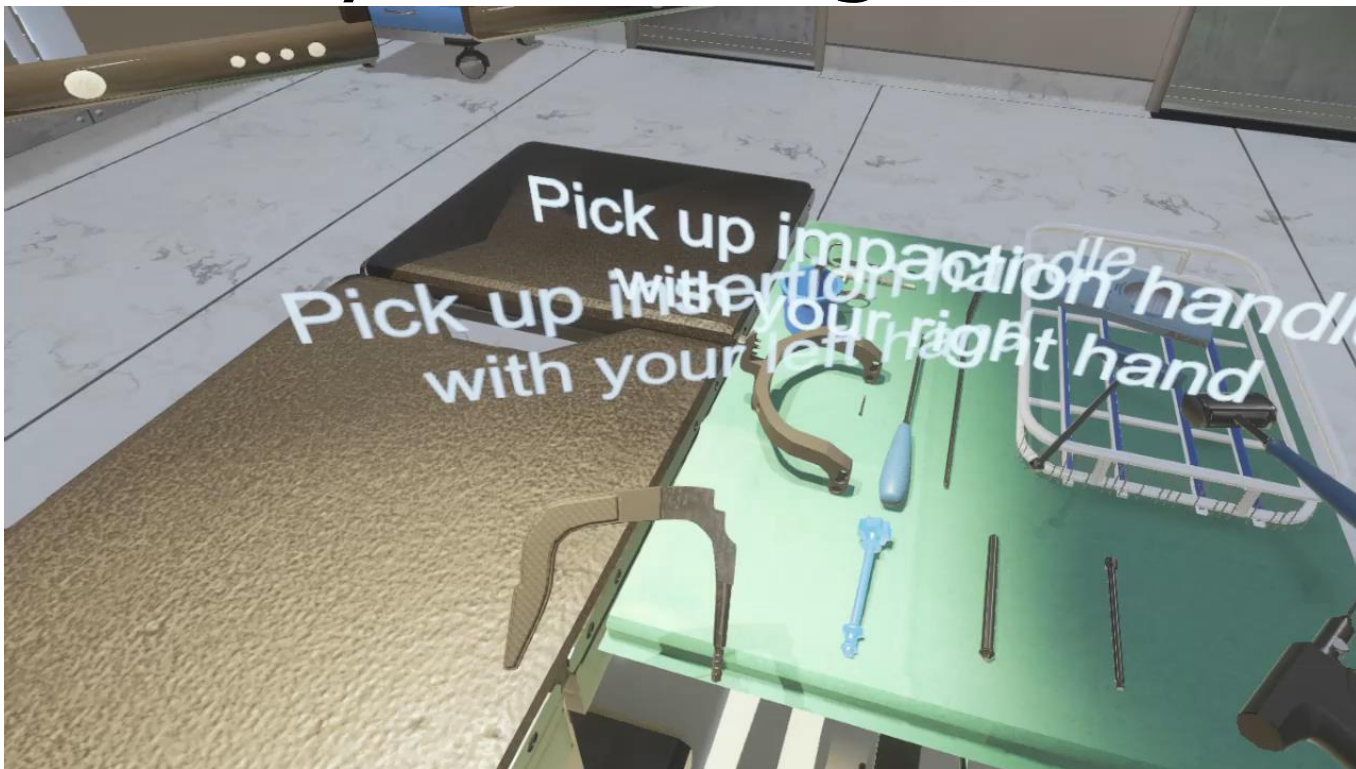
More complex interactions

- Attaching and detaching parts
- Screwing
- Drilling
- Hammering
- Positioning the C-arm (X-ray machine)
- And more...

Interactions: physics based or not?

- I'm not a physics engine expert!
- We're using PhysX in Unity
 - Your results may vary with other engines
- When it works, physics based interactions can be a time saver and produce good results
- Unfortunately it often doesn't work well
 - Particularly for two handed interactions
- Experiment and decide on a case by case basis

Case study: screwing



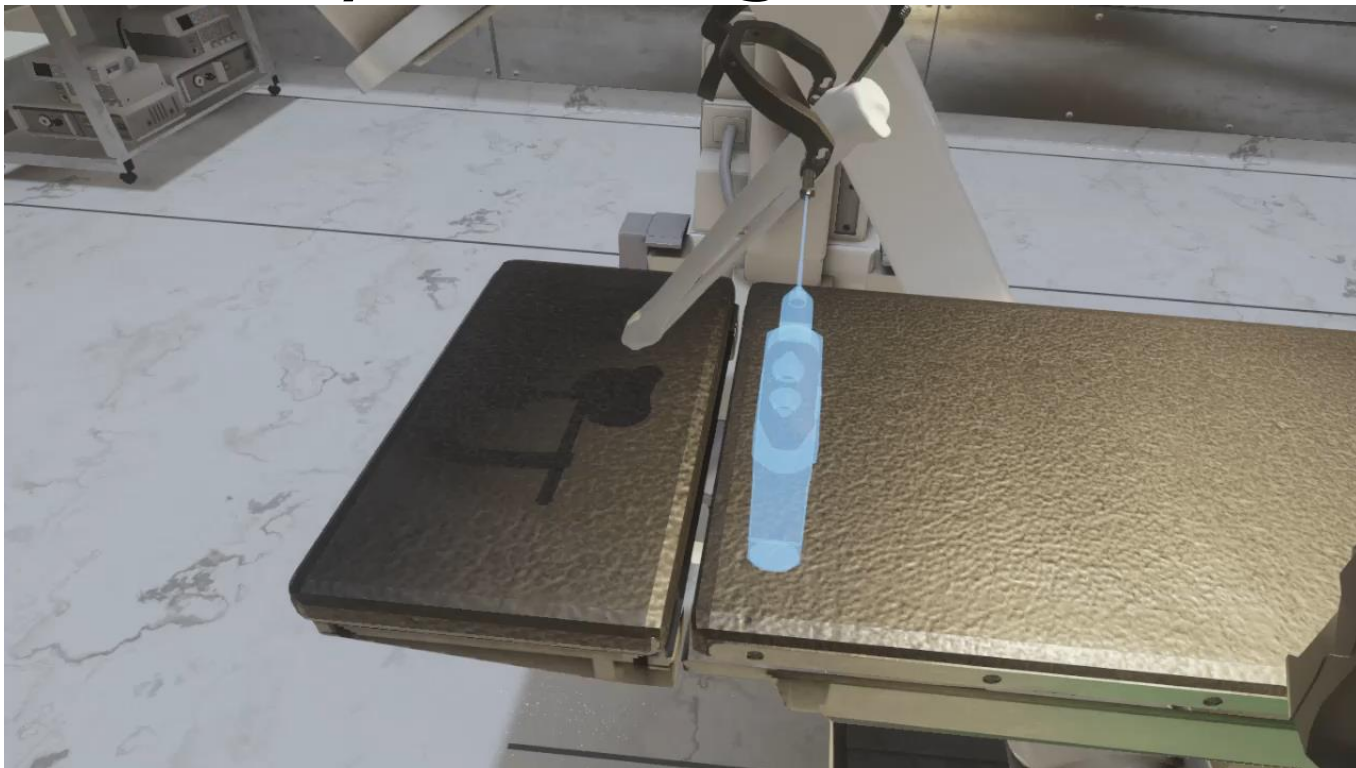
Physics (joint based) didn't work



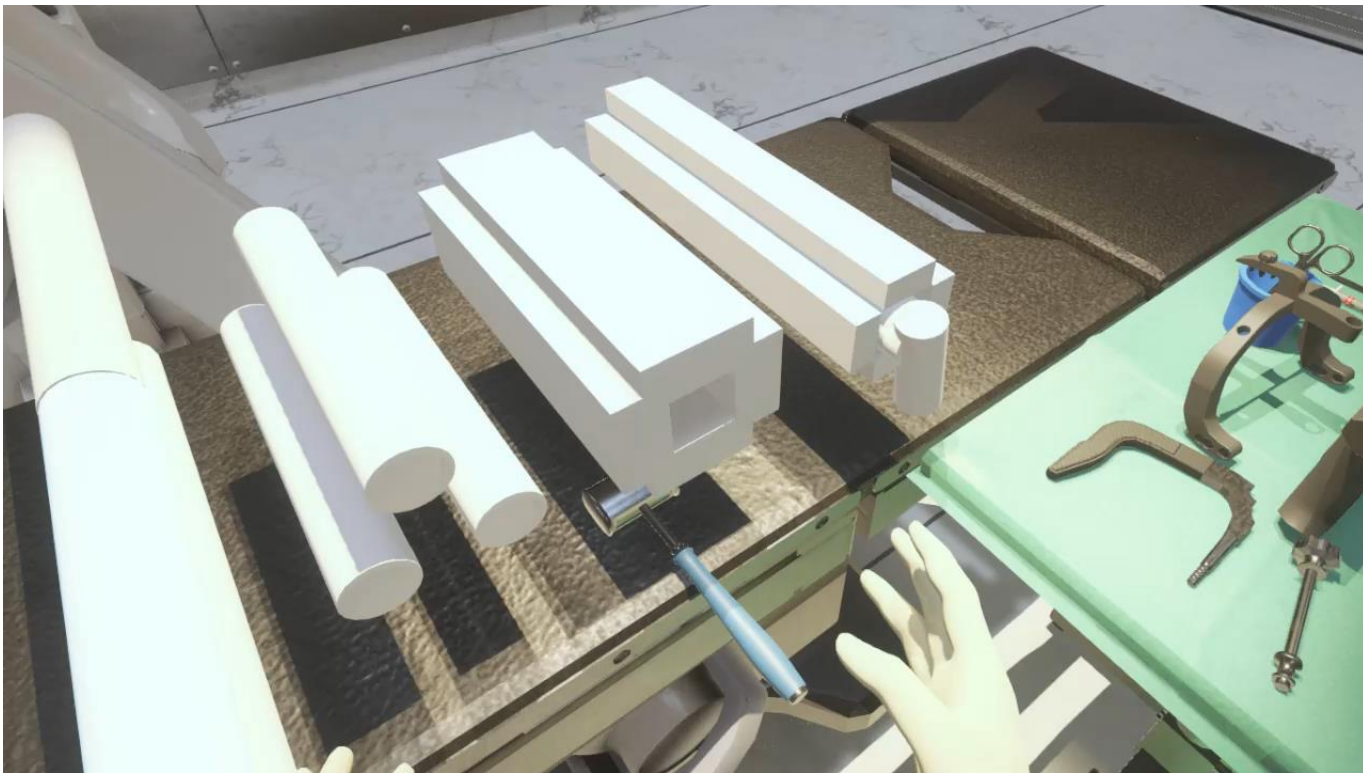
Our custom solution

- Use parenting in Unity
- Custom script does a bunch of quaternion math to try and produce a plausible rotation based on hand movements while enforcing constraints we care about
- Trigger haptics proportional to the local space rotation to give a feeling of threads

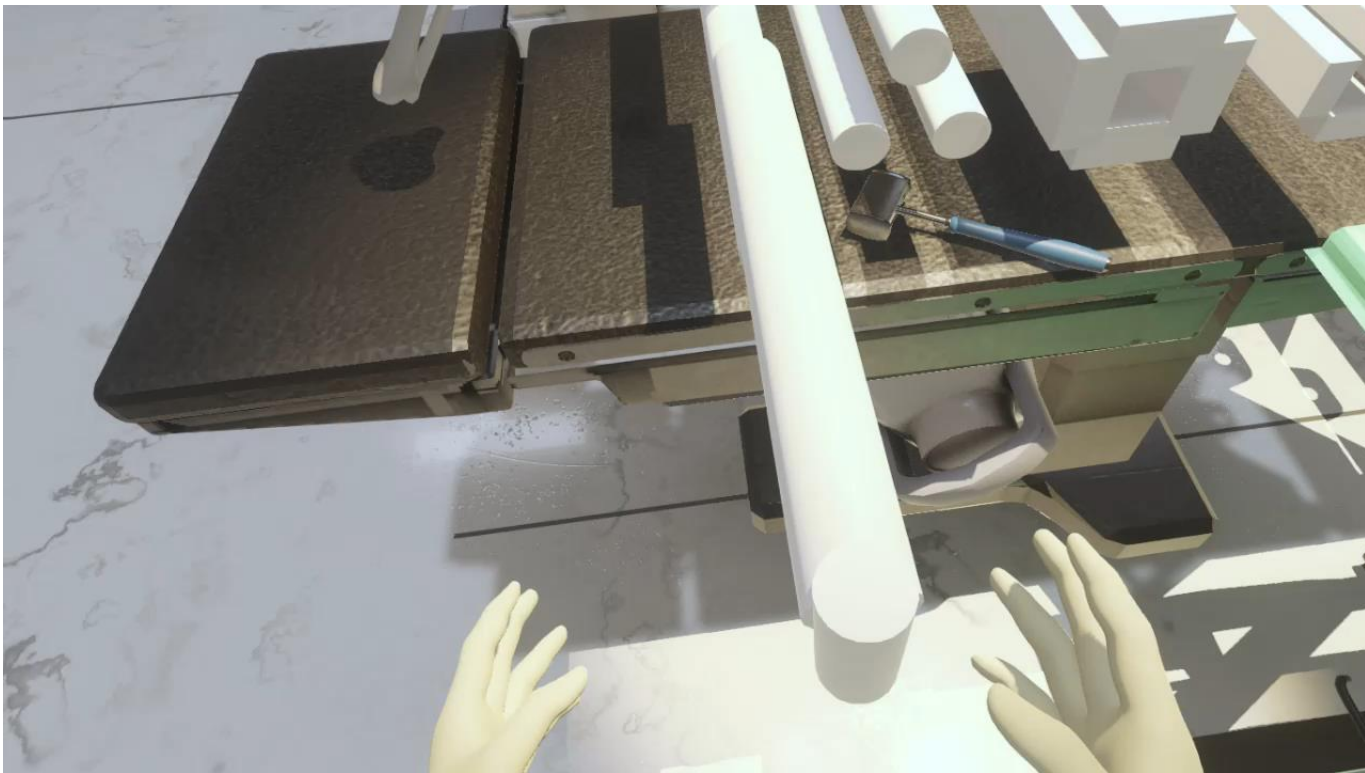
Case study: drilling



Physics (collision based) didn't work



Physics (joint based) worked better



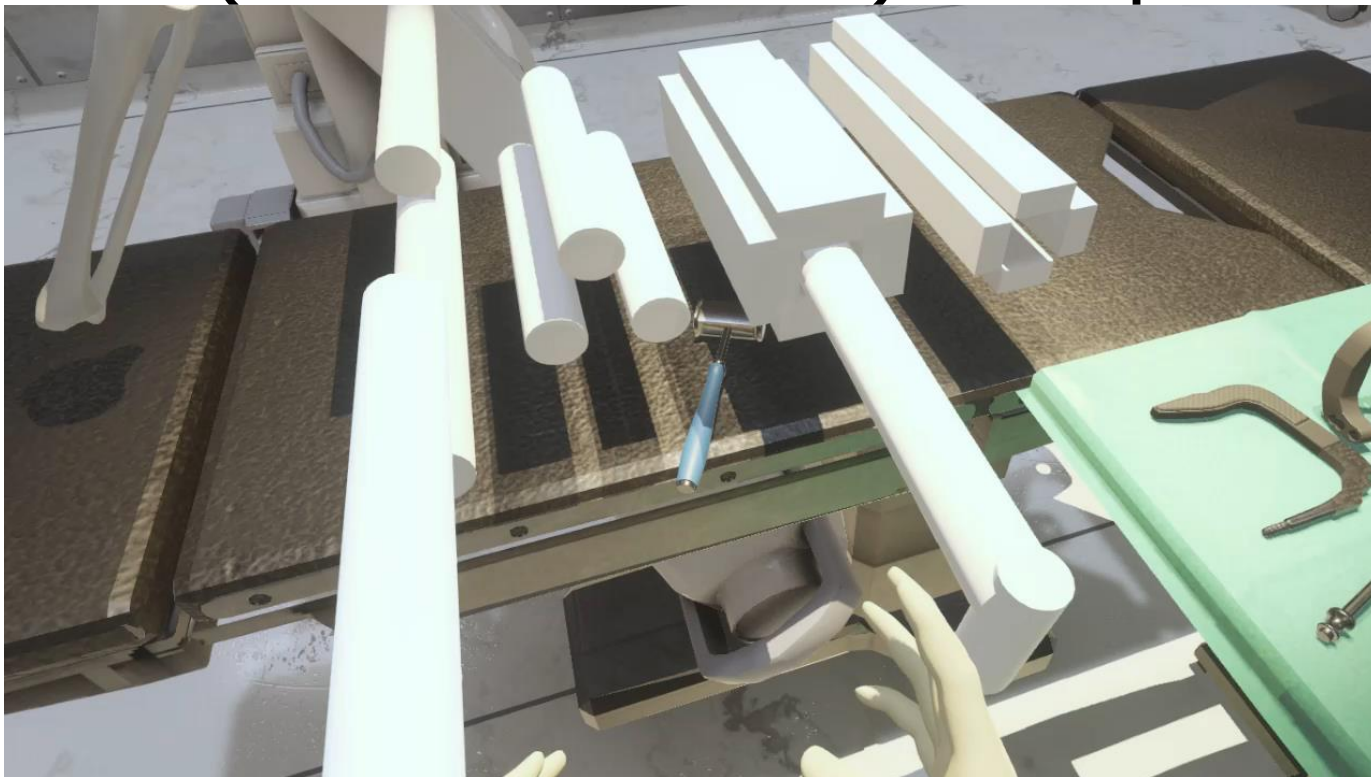
Our physics based solution

- Drill is locked to a configurable joint constraining motion to one linear axis and rotation around that axis
- Script adjusts the range of motion of the joint according to the time the drill is active
- When the drilling is complete, make the joint breakable to allow drill to be removed

Case study: hammering



Physics (collision based) had promise



Our custom solution

- We opted not to use physics in this case
 - May revisit this
 - Physics based solution might give us displacement proportional to impact force 'for free'
- Custom script detects hammer blows and sets position of nail

Haptics and audio

- Haptics helps sell interactions
- When haptics and audio are used well together the whole is greater than the sum of the parts
- Haptics that are driven by real world position and rotation changes work well

NewtonVR

- If you're interested in experimenting with physics driven interactions check out NewtonVR - open source Unity plugin
- Developed in part by two of our employees when they were at Tomorrow Today Labs
- We currently have custom interaction code but may switch to NewtonVR in future

Conclusions

- Hand interactions are a powerful tool for training applications
- Make sure to get the basics right and think about the right choices for your application
- Custom interactions can be time intensive to develop but the payoff is usually worth it
- Physics based interactions are great if you can get them to work reliably

Resources

- Owlchemy Labs Oculus Connect 2 talk: *Being There: Designing Standing VR Experiences with Tracked Controllers* - <https://youtu.be/hjc7AJwZ4DI>
- Collection of useful links: <http://www.uxofvr.com/>
- My blog: <http://blog.mattnewport.com/>
- Email me: matt@mattnewport.com
- Osso VR: <http://ossovr.com/>
- NewtonVR – Physics based interaction for Unity: <https://github.com/TomorrowTodayLabs/NewtonVR>

Video Credits

- *"Three Bs on the B" - The Gallery: Call of the Starseed* - HTC Vive VR Gameplay – FunshineX
 - <https://youtu.be/8MJm1EWWHi0>
- *Baby Beth throwing toys out of the cot* – jennymcdiarmid
 - <https://youtu.be/8tTRDmdi18g>

Questions?

