



timoni west

Follow

Nov 10, 2015 · 13 min read · Listen



Save



## UX pointers for VR design

This article is tailored for VR designers & devs who are creating full-room VR setups, but may not have the opportunity to do lots of user demos.

We have a full-room Vive set up with controllers at Unity's San Francisco office, and give demos to people every week. That means over the last few months I've been in the privileged position of seeing over 100 people try out their first room-sized experience in VR. It is amazing fun to see how people react, and what they like best. As I've given demos, I've noticed some consistent behaviors, pain points, and user confusion, and written them out here.

### TL;DR

Too busy to read this whole thing? I feel you. Here's the big takeaways.

- **Don't force users to do anything until they've oriented themselves.** Give them time to look around, just like in real life.
- **Always explain unusual button mappings.** If you don't, users will think your experience is broken — they will not always mash buttons like they do on game controllers.
- **Use animated text or audio cues to explain interactions,** button mappings, and gameplay. Don't use static text unless it's huge. HUGE.
- **People don't have a perfect sense of their physical self, and VR amplifies this.** It's very hard to draw a straight line in 2D, but impossible in 3D; knowing where your arm really ends is also difficult.
- **Give users feedback that they are being tracked,** and everything is still working. Haptic feedback is helpful here.
- **Try out all the new crazy stuff you can think of.** VR is a blank slate, intuitive interfaces aren't obvious, and now's the time to think big.

### The demos I'll be referencing

Here are the apps I most often demo, roughly in order of frequency. I reference these a lot in the post below, so if you're not familiar, you might want to take a second to check out the mechanics.

- Tilt Brush: A 3D drawing app. (Skillman & Hackett)
- Job Simulator: A game in which you perform menial tasks (and throw things). (Owlchemy Labs)
- TheBlu: An underwater simulation. (Wevr)
- Fantastic Contraption: The VR version of the classic 2D building game. (Northway/Radial games)
- Aperture Robot Repair: A structured cinematic experience with game elements, set in the Portal universe. (Valve)
- The Gallery: Six Elements: A RPG puzzle/adventure game. (Cloudhead Games)
- Final Approach: Another VR remake, this time of the classic air traffic control game. (Phaser Lock Interactive)
- Chunks: A Minecraft variation with a roller coaster. (Facepunch)

## Caveats

Here's a few points to keep in mind before you start reading.

- In the interests of time and space, I'm only detailing **anecdotes** from a UX perspective: this is not hard data.
- I appreciate that some of the demos are games and designed to be a bit tricky, and I've accounted for that in these findings. If I mention it here, it is not a failure of overall game design, but simply hardware or affordances.
- **I'm not going to go over the basics of VR UX in this article.** User comfort, avoiding sickness, presence, frame rates, proprioception and space are the basic building blocks of VR UX, and if you are unfamiliar with them, here are some useful resources to read first: [Oculus Best Practices](#), [Unreal Best Practices](#), [The VR Book](#) (excellent resource), and [LeapMotion's blog](#).

## Key observations.

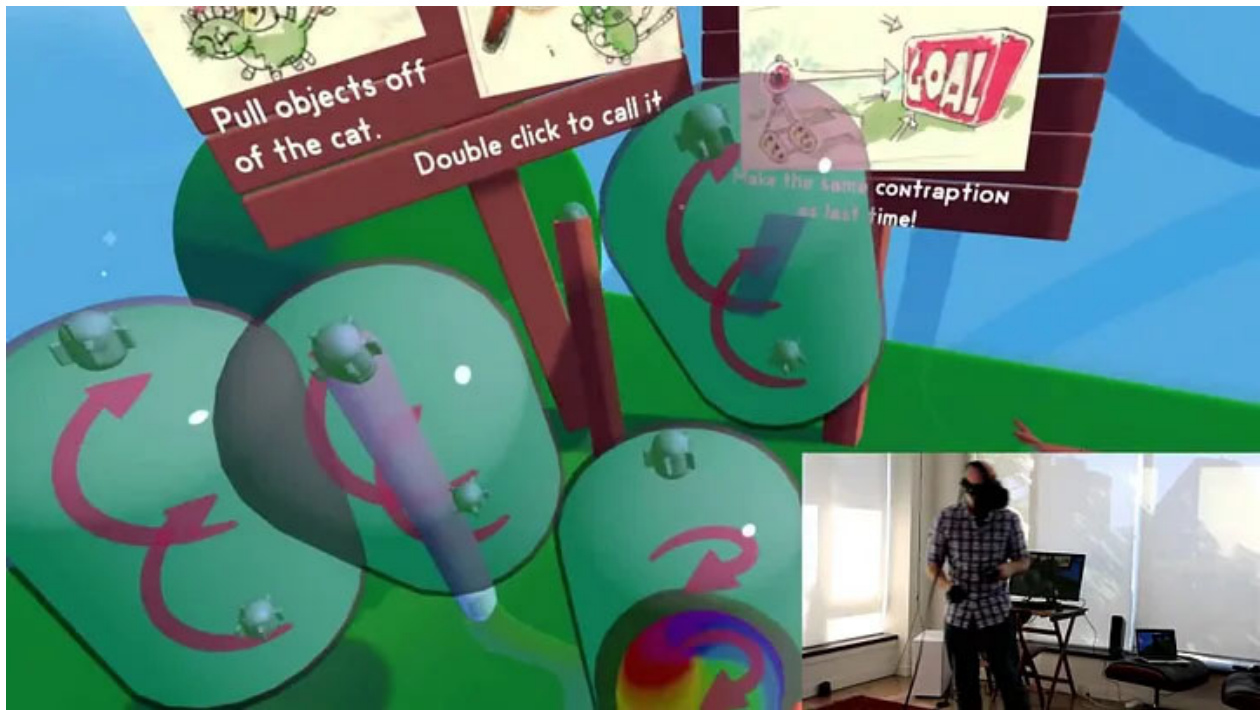
### 1. Give users time to orient themselves.

Users are generally too busy orienting themselves to their new environment to notice or do anything specific for at least 10 seconds, so don't force them to dive right in to the action. [Oculus Story Studio mentioned this phenomena](#) in their talk at Siggraph 2015. For their first short, Lost, they ended up designing something called an 'in' to recreate the settling in that naturally happens when one watches movies or tv: the dimming of the lights, the settling into the chair. Christopher Alexander liked courtyards for the same reason, as a space between spaces, a bit of mental preparation.

The more you get used to VR, the less time you need to orient yourself, but I anticipate it will never go away entirely. **Every new environment will require some amount of time for orientation**, especially if the user has taken on a different body, or has entered a particular awe-inspiring scene.

### 2. A huge percent of users simply do not read static instructions.

No big surprise here, since users never read anything. In fact, kudos to you for continuing to read this post. But it is still a bit surprising to see just how much people ignore instructions, considering how obvious they are in VR.



Fantastic Contraption doesn't require a lot of instruction, but has handy signposts explaining the basics.

Fantastic Contraption has handy signs with clear instructions about how to play the game. Tilt Brush goes one step further, and actually has helpful instructions pointing *directly* at your controllers, telling you exactly what to do. You literally can't change a brush or color in Tilt Brush without looking at your hands (and, presumably, the instructions), and yet, somehow, most people look past them.

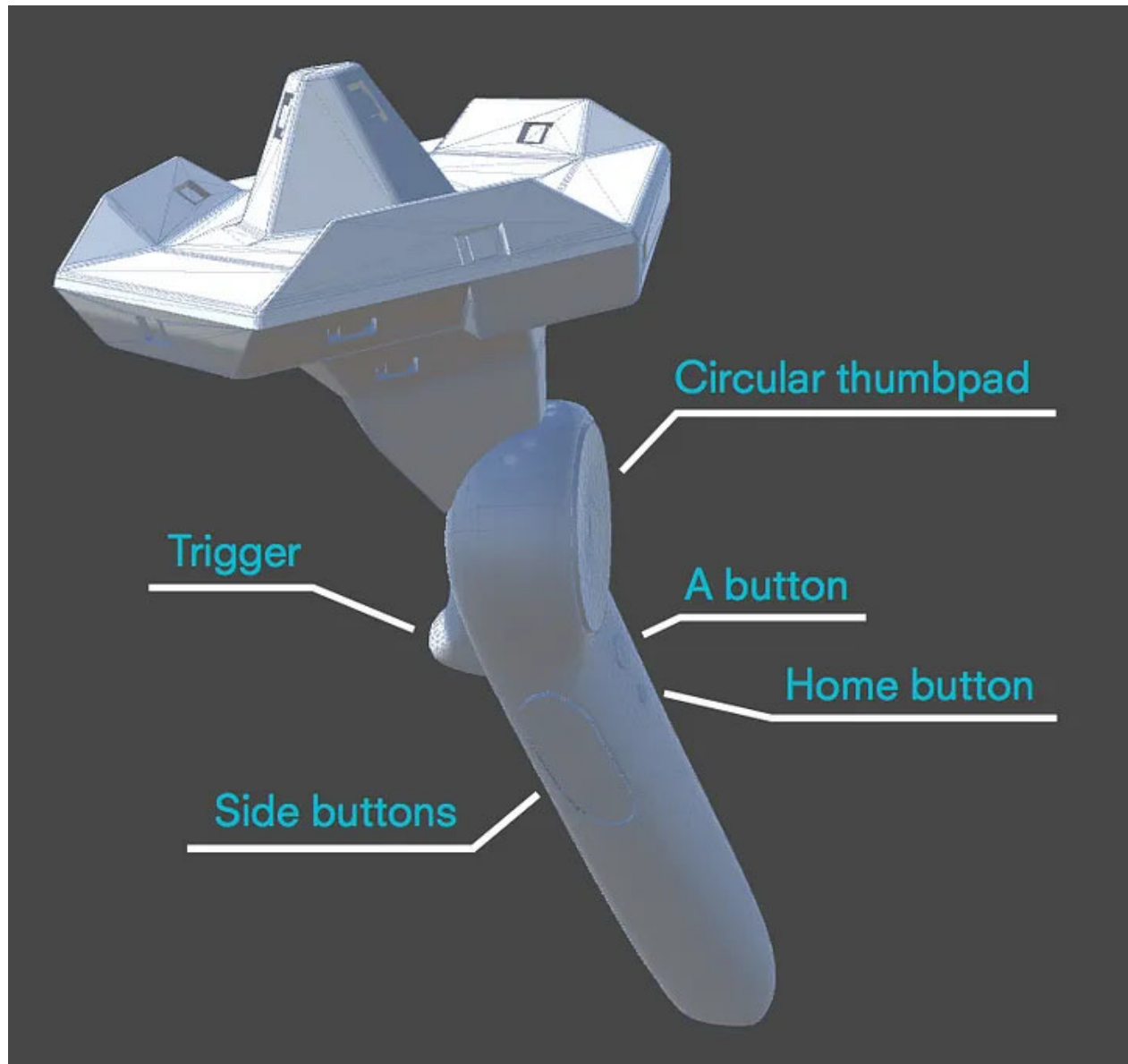
In my experience demoing, there *are* two kinds of people: ones that clearly read all these instructions and start creating, and those who do not. The latter group is much, much larger than the former. **For what it's worth, the ones who read do tend to catch on more quickly, and have more fun.** My advice then is not to remove helper text, but **figure out how to get users to read it.** Animating text, having it swoop in, or including larger arrows: these are all potential options. I am also a fan of **audio cues**, which work remarkably well; I'll talk about this a bit later.

Having clear instructions is especially important in cases where you are using unusual button mappings. Users are fairly forgiving about trying to click around, but if they haven't noticed a button, they won't try to use it in VR. Instead, they'll think the game is broken, they did something wrong, or—gasp!—worst of all, that your game is lame.

**One general caveat:** everybody notices huge text that overlays the environment. For example, the credits at the end of The Gallery: Six Elements and TheBlu. If you must tell the user something important, try doing the visual equivalent of shouting in their face.

### 3. Controllers are a tricky beast.

The Vive's controllers are new and unfamiliar: they do not feel like an xBox or Playstation controller, for example, and they don't have a joystick.



Buttons on the HTC Vive's hand controllers. The small squares on the tops and side are sensors.

Tilt Brush makes good use of the thumbpads for secondary actions, like increasing the brush stroke size (right hand) and rotating the cube menu (left hand), but people generally only find out

that the thumbpads are useful by accident — even though Tilt Brush has clear signs sticking out of the controllers, pointing directly to the touchpad and saying what they do.



In Aperture Robot Repair, set in the Portal world, your controller is skinned and simplified.



In Fantastic Contraption, your controllers are wooden version of the real-life controllers.

Moreover, each game has the ability to completely customize what the game controller looks like. This is not simply a matter of changing button mapping, which most games can do today. This means you can actually change what the controllers look like in VR: they can look like guns, magic wands, horses, 20-sided die, entire life-sized houses — you name it, it can look like that. For the most part, though, apps tend to keep at least the general shape of the controllers, or a simplified version of them, with visible buttons in equivalent places to their real-world counterparts. The other common option, which is Oculus' favorite, is using dummy hands.



In The Gallery: Six Elements, your controllers are gloved hands, with which you can pick up and use objects & weapons.

In The Gallery: Six Elements, you have two disembodied glove hands, with which the user grabs things. In the Gallery, the grab action is mapped to the side buttons — the only demo on my list in which the side buttons are used for a primary action. **No one has instinctively figured this out, in part because you can't easily feel the side buttons.** Most don't notice they're there. Instead, they try to grab with the trigger or thumb, then think that either the controllers are broken, they're doing it wrong, or that they simply can't interact with things in the game.

Chunks uses the secondary small button to switch between controller modes. This is actually quick useful, but unusual — I didn't notice it until someone else did a walkthrough.

In Aperture Robot Repair, you must click the touchpad to grab things at a distance. Most people ignore the huge blue blinking button the touchpad has turned into, and instead try to use the trigger, fail, and again assume the demo is broken, or that maybe they aren't supposed to open *those* particular drawers, etc.

**It's also worth noting that long ray casting gets tricky fast, selecting or manipulating objects at a distance.** Try easing motion-to-speed at long distances. Aperture Robot Repair solved this by having the controller 'jump' out of your hand quickly, rather like the hookshot from Ocarina of Time. Gaze-based selection can also work, but gets increasingly less useful at longer distances, too.

My coworker Dioselina Gonzalez pointed out that, **given the fact that no controller tracks perfectly yet anyway, a small piece of visual, audio or haptic feedback is almost necessary to confirm to users that they are still being tracked.** Make liberal, and consistent, use of them. Haptic and audio feedback in particular feel good and contribute directly to heightening the sense of presence.

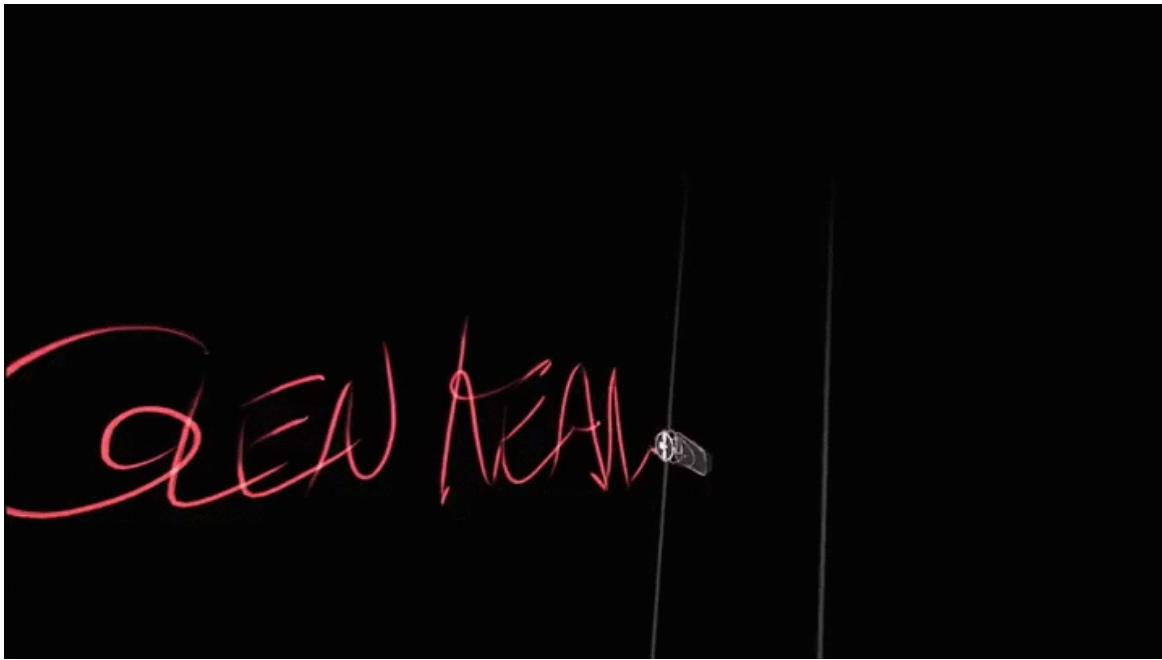
In all cases like these, **it's worth doing a quick overview of the customized controllers,** unless you want figuring out these actions to be part of the game puzzle.

#### **4. Most users' sense of movement is slightly off. VR makes this more obvious.**

The two-dollar word we're looking for here is proprioception, and it has to do with how we normally gauge distance and our physical bodies in space. For example, you know about how long your arm is: you can pick up a glass of water easily without having to double-check from all sides.

Most of the time, our proprioception just fine: maybe not exact, but good enough for everyday life. **For smaller, precise movements like writing or drawing, humans are used creating on 2D planes, physically supported by a table, desk, or something similar.** We can't draw gravity-defying sculptures in real life, and when we can see our motions over time — like, say, the trail a sparkler leaves — we generally can't move around it quickly enough to see just how off our actions are in z-space.

This disconnect between what you are doing and what you *think* you are doing is very obvious very quickly in Tilt Brush, where attempts to write your name or draw a house turn into Lichtensteinian sculptures once you move three feet to the side and see what your drawing looks like from another angle.



From Glen Keane: Step Into the Page

It is less obvious in Fantastic Contraption, where about 1/3 of users reach out to grab an object with the trigger button, and miss it by a foot or two. For what it's worth, this almost never happens in Job Simulator, where objects are no more than a few feet away. **It may be there is some small perception point glitch in our brains around the five-foot mark.** I am interested in whether anyone else has noticed this.

In environments where most anything is useable, like Job Simulator, it's fine to simply let users figure it out for themselves. But in experiences like The Gallery: Six Elements, where only certain objects can be acted on, the designers sensibly highlight items when you get close enough to grab them.

##### **5. Users seem to learn best from audio cues.**

On some level, this makes sense. Often tutorials in games involve someone telling you what to do: in quest form (the old village wise man showing you how to user your sword), as a narrative arc (the boss on the intercom, telling you there's a poisoning), or simply a HUD when you're about to enter battle for the first time, telling you to hit the X or Y button.



If immersion is your goal in VR, it can be hard to justifying stopping the action, and tempting to try to simplify the UX to the point there is no user confusion. **I would recommend trying audio cues instead, in the form of a narrator, character, or God-like all-seeing AI.**

Aperture Robot Repair uses the uppity British Robot to good effect, guiding you around the room, telling you what to do next. He doesn't tell you *how* to do it — therein lies the game component — but at least you know you're on the right track when you get audio feedback.

Final Approach has a similar narrator approach, this time with a jovial British narrator breezily explaining the basics of how to select an airplane and land it on the tropical island. His instructions are partnered with large visuals, including flashing green squares and arrows, but — surprise surprise — people don't notice them at first.

As for the rest of the demos, I generally provide audio cues myself. For example, in Tilt Brush, there is a large sign pointing directly to the thumbpad, saying "SWIPE HERE" with an animated arrow swiping left to right. In my experience, maybe one person has noticed this sign. The rest discover it by accident, or not at all. **But the second I tell them what to do, they learn, and they don't forget.**

I'm looking forward to seeing what new clever ways people come up with for teaching game interactions in VR. But so far, **having a clear bodiless narrator introducing the space and initial tasks seems to be the most useful way of orienting users without breaking immersion.**

## **6. Encourage users to move around as much as they can.**

First-time user don't move around very much: they're either afraid of tripping on wires, or they just feel a bit weird in a space that doesn't match reality. To counter this, Skyworld forces users to move off a solid platform into 'thin air', which feels odd, but is a reasonable learning mechanic.

Fantastic Contraption is impossible to play without moving around to build your creation, and the UX is so seamless users don't notice they are forced to move.

Aperture Robot Repair gets a gold star here, in particular. By manipulating the user into a specific position in the demo, they make it seem as if they have some procedural stuff going on based on the user's position. It's not — the demo is a simple cinematic loop — but it feels spooky, like the robots know exactly where you are. That's some great UX.

The big question around space, of course, is what you do when the user is seated at a desk, with a high chance of knocking over something hot onto something expensive and electronic. In this case, having the ability to grab or move objects or the scene is very key. Oculus Medium allows you to move and scale your project very quickly simply by 'grabbing' with the double trigger buttons. It feels great and lets you manipulate objects at any size — think taking something the size of city block, reducing it to a foot wide, then rotating it around. It saves you a city block's worth of walking. This is the magic of VR.

## Final thoughts

You may have noticed that I didn't talk much about heads-up displays, peripheral interfaces, floating inputs, and other glamorous cinematic VR UI tropes. This is because none of the demos people like most have them.

But I hesitate to take away any solid conclusions from this. It could be that VR is so new, people must start with the familiar, for the same reason a computer desktop is called a 'desktop' and originally had skeuomorphic iconography.



The best toolbox-slash-pet a VR user ever had, in [Fantastic Contraption](#).

If that's the case, something like [Fantastic Contraption](#)'s toolbox seems like a natural way to test the waters. The toolbox is a floating, purring cat who farts pink clouds to fly. It is adorable, familiar but weird, introduced at the right time, and easily one of the most lovable details in any VR game out there.

Like everyone else, I can't wait to see where VR is headed. And now is the time to try *everything*, see what works and doesn't, or what things might work in the future once users are primed. Eventually we'll codify behaviors, but VR can, and should, actively avoid falling into overworn UI guidelines like the classic 'users don't scroll below the fold' stereotype of the early 2000s.

[Open in app](#)[Sign up](#)[Sign In](#)

Search Medium



This lady knows what's up

VR creators have a great advantage: the cost of prototyping new UIs, inputs, and mechanisms is quite low compared to the cost of designing a new physical mouse, keyboard, or touchscreen. In VR, the same physical controllers can be rejiggered to be guns, lightsabers, laser pointers, slingshots, grapplehooks, spray cans, or anything else you might dream up. And unlike the keyboard, mouse and phone, there's no set conventions around selection & movement.

This is crazy awesome and exciting, and why I advocate using audio commands or text instructions rather than avoiding 'weird' controllers and button mapping.

Try it out! See what works! We got nothing but all the time and space in the virtual world.

*Special thanks to Sebastian Gutierrez, Corey Johnson, Laura Gluhanich, Alex Bowles, Beau Cronin, Amir Ebrahimi, and Pete Moss for their feedback. Also, I love talking VR with anybody and everybody, especially about creation tools. Please get in touch if you'd like to talk more about VR, Unity, or future tech in general.*

[Virtual Reality](#)[Oculus Rift](#)[Htc Vive](#)<https://medium.com/@timoni/ux-pointers-for-vr-design-dd52b718e19>

14/15



692



7

[About](#) [Help](#) [Terms](#) [Privacy](#)

Get the Medium app

