Style Guide for VR Simulation Design

This is an excerpt of an 80 page document created for Instructional Designers to create learning experiences in VR in collaboration with SMEs and 3D developers.

Related documents

(updated 8-24-22 VL)

This Style Guide provides detailed information about VFX, SFX, and haptic feedback, as well as the authoritative guide to writing VO. It also includes general information about designing in VR.

Other types of information can be found in the following documents -

For IDs:

- See Template Appendix for a library of interaction templates and additional VO parameters
- Duplicate the following document templates for each new sim:
    - Outline template
    - Design Doc template (non-CE) - provides structure and general standards to follow when writing design documents
    - CE Design Doc template - provides structure and general standards for Career Exploration sims
    - Domain model planning template (optional support/not required)
- For additional documents, see Confluence or ask in the Slack channel for your module.
- Many common interactions/mechanics can be found in the Demo Scene.

For Developers: See the Frontend Development Guidance page for vendor facing, developer-friendly guidelines across our simulations.

For anyone (but probably for no one):

- If you are supporting an older sim, see Legacy Storyboard Template for structure and general standards used for those storyboards. This format was deprecated in Spring 2022 - do not use it to create new sims.

Sim scope and structure

(updated 8-24-22 VL)

How do sims fit into a trainee's learning journey?

Our VR sims are just one part of a trainee's learning experience. They focus on hands-on processes, which they are uniquely suited to teach, and are often augmented by textbooks and classroom lessons, which are better suited to teaching factual content knowledge, as well as by mentorships and other resources).

You cannot be certain how much knowledge the trainee has before using a sim, so in general, structure your design assuming the trainee has little or no prior experience with the topic.

How are sims packaged?

Sims are grouped into modules, which are further organized into verticals.
For example:

- *Vertical: Diesel*
- *Module: Diesel, Part 1*
- *Sims:*
    - *1 Lifting Devices*
    - *2 Tire Inspection*
    - *3 Tire Replacement*
    - *4 Oil and Filter Service*
    - *5 Fuel System Service*
        - *6 Fundamental Analysis - Air Compressor Replacement*
    - *7 Fundamental Analysis - Air Distribution System*
    - *8 Fundamental Analysis - Air Valve Arrangement*
    - *9 Drum Brake Inspection and Repair*
    - *10 Parking Brakes Analysis and Repair*
    - *11 CAM Brake Replacement*
    - *12 Wheel Ends*
    - *13 Steering Gearbox Replacement*
    - *14 Tie Rod End Replacement*
    - *15 Remove & Replace Steering Linkages*
    - *16 Kingpin Assembly Inspection*
    - *17 Kingpin Replacement*
    - *18 Shock Absorber Replacement*

- *19 Leaf Spring Replacement*
- *20 Drivetrain Inspection*
- *21 Battery Replacement*
- *22 Starter Replacement*
- *23 Engine Analysis*

Trainees working within a vertical may or may not complete all modules and sims within it, and may or may not complete them in order.

Thus, sims are generally designed to be both **non-overlapping** and **modular.** ●
Non-overlapping - a trainee who completes all sims will find them mutually exclusive and comprehensively exhaustive.
● Modular - sims do not reference one another, and a trainee's ability to complete each sim is not dependent on completion of prior sims

How is a single sim structured?

Section summary

In most modules, each sim has two sections: Animated Tutorial (AT) and Hands-on Practice (HOP). *(Notable exception: Electrical Construction includes a third section by customer request: On Your Own.)*

In the AT, the trainee follows instructions given by the virtual coach, walking them through the process the sim is intended to teach, usually in sequential order. AT usually starts with a brief introduction and donning PPE, but this may vary by module.

In the HOP, the trainee repeats the process, this time with very little guidance from the coach. Directives are generally removed or replaced with a generalized prompt, such as "What's next?." Hints and redirection are provided when idle prompts and conditionals are triggered. The HOP is the graded portion of the sim.

Step sequence

In most cases, there is a single correct sequence in which the steps of a process must be completed, and the sim follows that sequence in both the Tutorial and the HOP.

In some cases, a person would be able to do some steps out of order in the real world, and it would be just as valid.

In these cases, you will have to pick a sequence to use for the Tutorial. If it seems important, you can have the virtual coach verbally note that the steps could be done in a different order.

In the HOP, you can either:

- Provide more supportive VO and visual hints in the HOP to enforce the same order as the tutorial.
- Use blue text to make a note for developers about which steps can be done out of order. See this example in Paint Gun Inspect and Clean. This is more work for the developer, so use this option only when it is crucial to the learning objective(s).

How long is a sim?

● Ideal: Does not exceed **15 minutes** of simulation play time for a competent user. ● Maximum acceptable: Does not exceed **20 minutes*** of simulation play time for a competent user.

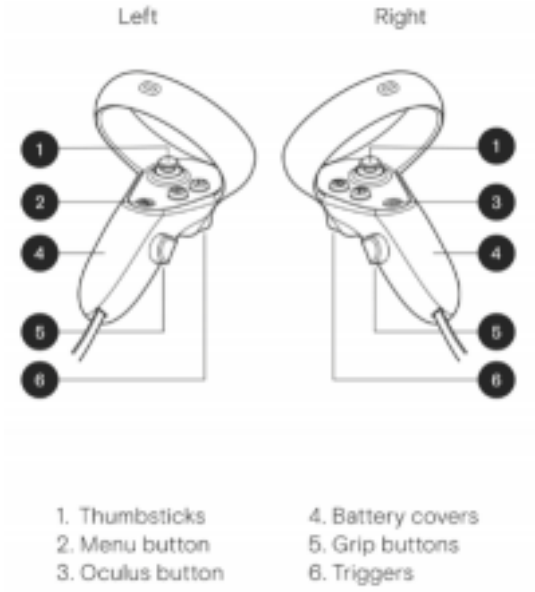
*Source: *Su, Y., & Wang, J. (J. (2022.). UX Summit: Research on visual fatigue when playing games. GDC Vault. Retrieved April 28, 2022, from https://www.gdcvault.com/play/1027671/UX-Summit-Research-on-Visual*

Note of exception: Electrical Construction sims often exceed 30 minutes per Client requirements.

Designing for VR

Controllers

(updated 8-24-22 VL)



Buttons

Note: Full details for applying controller instruction in VO are covered under Controller Instruction in VO—please read.

Grip: Located under middle finger
Used to grab an object, such as picking up a tool, grabbing a dial to twist, or grabbing a plug to remove, etc.

Trigger: Located under index finger
Used to activate something the trainee grabbed, or to confirm a selection the trainee is pointing to with a ray cast. For more information on ray casts, see Pointing at Objects. For CE (and possibly other verticals) appear a 3D guide on first callout/use when the user is holding something with the grip and needs to activate the tool with the trigger (such as a drill).

Example: [3D VR controller appears in front of student. A semi-transparent hand holds the controller with trigger button highlighted]

Thumbstick: Located under the thumb
Used to adjust or control something the trainee grabbed.

Because thumbsticks (formerly referenced as “joysticks”) are seldom used, show a floating 3D guide with the thumbstick highlighted in any idle prompt for a thumbstick interaction.

Example: [3D VR controller appears in front of student. A semi-transparent hand holds the controller with thumbstick highlighted]

Touch

“Touching” an object is achieved by bringing the full controller in close proximity to it - i.e., reaching out your hand to “touch” it while holding the controller.

Generally used to identify, open, or close something that doesn't require grabbing. It can also be used to simplify a more complex interaction if necessary (such as opening a door).

Haptic

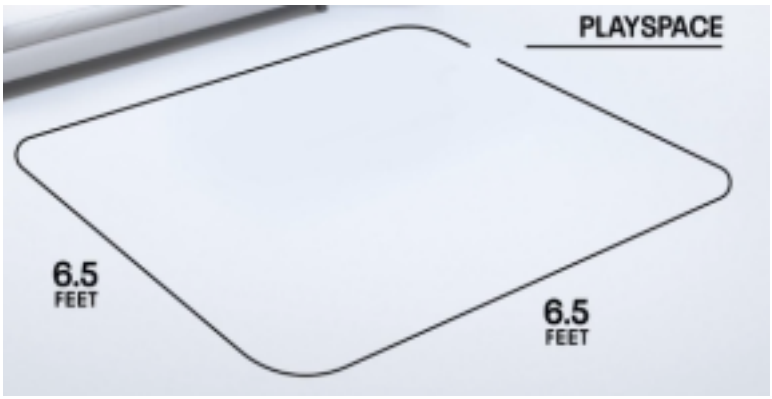
The controllers for Oculus Quest 2 come with vibrating motors. Using these is called [haptic feedback]. Haptic feedback can vary in duration and intensity.

See section on Haptic Feedback for when and how to use haptic in our sims.

Environment

Playspace and position

(reviewed 8-24-22 VL)



A trainee's playable space is 6.5' x 6.5' (or 2 meters by 2 meters). This means they can generally take a step to any side, as long as they return to their point of origin.

- When possible, no interactive items should be outside the play space.
- If this is not possible, any interactive item that needs to be reached outside the area either:
    - Requires a programmed-in teleport, or
    - Should be used / selected with a laser-type selection mechanic.
- All lift mechanics must happen inside the play area.
- Trainee should be instructed to recenter themselves in the play area when needed by voiceover or step mechanic to start a new simulation or reset for a new task.

Assets

(updated 9-2-22 VL)

Assets are all objects in the sim. When, where, and how they appear and disappear has a big

impact on the trainee's experience.

Examples of assets include:

- large and small objects the trainee interacts with (e.g., hammer, screwdriver, table, engine, etc)
- non-interactable objects that create ambiance (e.g., plane, truck, scaffolding, etc) ● sim UIs (e.g. menu, virtual coach, medal, hint UIs, floating text) and any other visual content.

Appearance and Disappearance of Interactable Assets

Timing

The appearance or disappearance of tools, supplies and equipment can serve as a hint for the trainee, so it should be timed thoughtfully. Also, the appearance of too many objects on the work surface at once can lead to a very cluttered environment. Below are the standard timings used in our sims; you should generally adhere to these unless your sim has a special situation that requires a variation:

	Appear	**Disappear**
Tutorial	Each appears when the trainee needs it. Example: [bottle brush appears on table] [Simon VO:] We're gonna use this brush to clean the inside of the paint gun. [pulse highlight bottle brush] [Simon VO:] Pick it up.	When the trainee is all finished with it, or if they will need it again, but not for several steps. CAUTION: Never disappear an object while the trainee is holding it - this is jarring to the trainee. Always include a "Put Down" interaction first, then disappear it from the table.
HOP All are available either at the very beginning of HOP or—more often—after the trainee finishes putting on PPE. *(The HOP serves as an assessment, so the trainee is expected to know when they need to use the objects they just learned to use. Spawning or highlighting them every step of the way would give those answers away.)* Example:	[all tools needed for sim appear on the cart: socket wrench, cup-type end cap on ratcheting socket wrench, torque wrench, new filter, bottle of oil with no cap on it, a rag] [Simon VO:] Here's everything you'll need to drain the oil and change the filter.	All should remain available until the end of HOP as much as possible.

In sims where that would make the table too crowded, you might: - specify the devs use an extra-large cart or table - have the items spawn and disappear in batches. - If there is teleporting, spawn each object only in the playspace where it will be used	- Disappear some tools when the trainee is finished with them, if it is unlikely to give anything away 7

	NOTE: Some sims have SO MANY tools and supplies. Here are some options for avoiding a crowded table: - specify the devs use an extra-large cart or table - have the items spawn and disappear in batches. - If there is teleporting, spawn each object only in the playspace where it will be used

Location

Tools always appear on the cart, table, or other logical surface. Do not spawn a tool in a trainee's hand.

Ideally, tools should also disappear from the table, cart, or other logical surface. (In real life, the trainee would put the tool back where they got it from.) Avoid having a tool disappear from their hand unless the sim is running extremely long and there are no other options.

Arrangement of non-interactable assets and sim UIs

See Environment Best Practices in Frontend Design Guidance for notes on how best to place these assets within the environment

Height Adjustments

Some tasks require a trainee to reach high or low on a vehicle, machine, wall, or other large object. When possible, avoid having the trainee crouch down or stand on tip toe because it is easy to lose balance.

Instead, either the object or the floor should be raised or lowered so the trainee can easily reach where they need to work and see any hints or highlights they'll need to see. Keep in mind this also might be necessary to accommodate both tall and short trainees. Many vehicles can simply be placed on lifts similar to what technicians use in the real world.

Teleporting

(updated 8-4-22 VL)

Teleporting is used when the trainee needs to move to a different environment or work on something outside their playspace. There are two primary types of teleports: visible teleports and blind teleports. A visible teleport is when a trainee can see their destination. A blind teleport is when they cannot see it.

Learn more about how teleports work and how to design them in the Template Appendix document under Teleporting.

Transparency

(reviewed 9-2-22 VL)
If you need the trainee to have an interior view of something, such as a part hidden within a truck, you can indicate that objects are transparent. Do this by stating [object turns transparent]. You can also say [object switches to XX% opacity] if you need it to still be slightly visible, but see-through.

Please note that transparency can be costly in development, so only use it if absolutely necessary for instruction.

Trainee Actions

Discrete interactions with concise directives

(updated 8-24-22 VL)

Using concise directives and breaking interactions down into discrete steps brings clarity both to the trainee, who may be learning a new type of media, and the developers, who may misinterpret your design. Concise prompts also prevent your sim from feeling too wordy/verbose.

To that end:

- Break down a sequence of actions into discrete interactions; do not combine multiple actions or control commands into a single command.
    - Exception: If the sim has a similar sequence of actions throughout, it is okay to combine in some cases. For instance, if the trainee needs to pour ten cups of

coffee, you can give them the breakdown of control commands for the first two or three cups, then a combined command for the remainder.

Success conditions

(added 8-24-22 VL)

For each interaction, you'll describe the correct behavior as precisely as possible (e.g., "trainee picks up coffee pot")

Also describe any cues (VFX, SFX, haptic, VO) that occur during the interaction (e.g., "cup fills with liquid", "sfx liquid pouring") or upon completion (e.g., "cup appears full", "sfx confirm", "Simon VO: That looks delicious!").

Note: As in all parts of your Design Doc, pay attention to the Global Rules stated at the top of the Design Doc template and to the Frontend Design Guide. Avoid restating anything that is already covered by a global rule in either of those places)

Conditionals

(updated 8-24-22 VL)

For each interaction, you'll also describe the errors the trainee might make and specify any cues that should occur as a result. These errors are called "if conditionals".

- Each discrete interaction should have its own conditionals. At a minimum, every interaction must have an idle prompt (see next subsection).
- Take all *important* possible trainee actions into account when writing conditionals.
    - Include conditionals for actions that likely indicate misconceptions or common errors (e.g., turning something the wrong direction, choosing a specific incorrect object, doing something out of order)
    - When appropriate, also include conditionals for actions that relate to VR experience (stopping too soon, pressing the wrong button, etc)
    - Do NOT include conditionals for every silly thing someone might do in the environment (e.g. throwing the wrench across the room)
- Consider how the conditionals may create complexity in development, and try to keep it simple.
    - Avoid a very generalized "grabs wrong object/does wrong thing" conditional, which would require developers to code every single object in the environment as potentially grabbable
        - Instead, if a specific wrong object or action is called out in the TA or by the SME as dangerous/incorrect, call out that specific object/action in the conditional

Idle Prompts

(updated 7-22-22 VL)

Idle prompts are given if the trainee has not started/completed an expected action within a set number of seconds after the directive is given.

Idle prompts always include a VO reminder (often just repeating the directive), and usually also include a visual hint such as a ghost or a highlight.

Idle time allowances are specified globally in the Frontend Design Guide and do not need to be specified in individual design docs. Note that since idles are considered incorrect and often count as mistakes in HOP, the time allowed is generally longer in HOP than the Tutorial.

Examples

The following examples provide examples of how (not) to apply the guidance given above. *Note that these are written in the old sim and VO style - please disregard the format and language.*

Pick up the pot and pour the coffee into the cup. [trainee pours the coffee] [SFX dribble] [SFX confirmation] ➔ [if trainee doesn't complete both actions, repeat every 7 seconds] Pick up the pot and pour the coffee into the cup.

Next you're gonna pour the coffee into the cup. Pick up the pot.
[trainee grabs pot] [SFX confirmation] [ghost of coffee pot appears above cup in a pouring position]
➔ [if trainee doesn't pick up pot, repeat every 7 seconds] [pulse highlight handle of pot] Pick up the pot using the grip button on your controller.

Lift it over the cup to pour the coffee.
[trainee holds pot in or near the ghost over cup] [sfx confirmation] [completion circle UI appears and gradually fills up] [coffee pours] [SFX dribble]
[trainee holds/pours coffee for 3 seconds until completion circle UI is full] [SFX

confirmation]

➔ [if trainee doesn't hold pot in correct place, repeat every 7 seconds] [up arrow appears] Lift the coffee pot over the cup to pour the coffee.

➔ [if trainee lets go of handle] [coffee pot respawns on table] Lift the coffee pot over the cup to pour the coffee.

➔ [if trainee moves pot away from ghost pot before 3 seconds] [pot moves with trainee's hand] [ghost pot reappears] Lift the coffee pot over the cup to pour the coffee.

Handedness

When right versus left hand matters to the sim design, there are a few ways you can specify this in your design doc: ask the trainee, infer from trainee actions, or tell the trainee which hand to use. **The infer method should be used whenever possible, because it is the most seamless experience for the trainee.**

Ask Trainee

When handedness impacts a large part of the sim and the nature of the setup itself, specifically ask the trainee what their dominant hand is. An example of this can be found in the Handsaw sim. The dominant hand in this sim was required both for tool use as well as the set up of the work environment.

Example of asking trainee:

This is a one-handed tool, so before we get started, I need to know if you're right-handed or left-handed. Touch one of these buttons.

[RIGHT-HANDED button appears] [LEFT-HANDED button appears]

[trainee touches right/left button] [SUBMIT button appears underneath]

➔ [if doesn't touch a right/left button and is idle, repeat every 7 seconds] This is a one-handed tool, so before we get started, I need to know if you're right-handed or left-handed. Touch one of these buttons.

[trainee touches SUBMIT]

➔ [if doesn't touch SUBMIT and is idle, repeat every 7 seconds] Make sure you press SUBMIT.

➔ [if selects LEFT-HANDED] [Sim layout flips horizontally from what is outlined in this design doc to accommodate learner's interaction]

Infer

When handedness impacts how tools are used or how ghost hints are displayed, but not the setup of the environment, you should infer handedness from the trainee's actions. An example of this can be found in the Two-Handed Pour Interaction Storyboard Example in the Pouring Fluids section of this Style Guide. Also see the Knife Skills sim (read the global conditionals in particular). The dominant hand in this sim adjusted how virtual hands appeared and how food items were displayed, yet the trainee could change the hand they used to hold the knife at any time.
This feature has been referred to as "ambidextrous ghosts" in conversations with devs in Jira and on Slack.

Tell Trainee

When the use of a dominant hand is not crucial, but the use of a specific hand is, you can tell the trainee which hand to use. For example, if a tool must be picked up in the right hand to read a measurement/value, you can tell the trainee to use their right hand specifically, and add a conditional VO if they use their left hand.

Example:

[This hacksaw interaction was modified to restrict set up for right-handed use. Trainee will be redirected to use right hand in any instances when they try to use their left.] [highlight hacksaw]
Use the grip on your right controller to pick up the hacksaw from the table.
[trainee grabs saw with right hand] [SFX confirmation]

- ➔ [if idle, repeat every 7 seconds] Use the grip on your right controller to pick up the hacksaw from the table.
- ➔ [if picks up using left hand] [do not allow trainee to grab saw with left hand] For now, pick up the hacksaw using the grip on your RIGHT controller.

[ghost hand on unclamped, loose side of hose]
Hold the hose with your left hand.
[trainee grabs hose with left hand] [SFX confirmation][ghost hand disappears][hose is held taut]

- ➔ [if idle, repeat every 7 seconds] Hold the hose with your left hand.
- ➔ [if drops hose] Hold the hose with your left hand. [ghost hand on unclamped, loose side of hose]

➔ [if drops saw and grabs hose] [do not allow trainee to grab saw with left hand] Hold the hacksaw with your right hand and the hose with your left hand.

Templates and supporting documents

(updated 8-24-22 VL)

See the Related documents box at the top of this docs for links to the other documents you'll need to get started.

Many common interactions/mechanics can be found in the Demo Scene. Many templates

for specific interactions can be found in the Template Appendix document.

Multiple Choice Knowledge Checks

(updated 8-4-22 VL)

Multiple choice questions are used sparingly—only when there is critical information that must be taught or assessed and there is no other way to do it that would make better use of the rich interactions VR affords.

For templates, VO notes, and other information related to designing knowledge checks, see the Template Appendix.

Visual, Sound, and Haptic Effects

Haptic Feedback

(reviewed 9-2-22 VL; table is incomplete, not sure who should complete it)

We generally use haptic feedback to simulate real world sensations, but not as an in-game indicator of any other kind. Examples of when to use haptic feedback include: ● The trainee is holding something that vibrates, such as a motorized tool being used. ● Alert the trainee that they're doing something wrong *or* something that can endanger them, such as: holding their hand too close to a saw blade or touching something that could cause electric shock. This warning should be apparent in relation to the danger (e.g., a saw blade would injure the hand, so the hand receives haptic feedback).

When adding haptic feedback, there are three key areas to document in your design: ● Controller (left, right, both)
● Intensity (low, medium, high)