XR ASSOCIATION DEVELOPERS GUIDE:
AN INDUSTRY-WIDE COLLABORATION FOR BETTER XR

CHAPTER ONE: FUNDAMENTAL DESIGN PRINCIPLES FOR IMMERSIVE EXPERIENCES

XR ASSOCIATION

www.xra.org

REVISED DECEMBER 2024

TABLE OF CONTENTS

1 LETTER FROM THE MEMBERS 03

2 INTRODUCTION 05

3 VIRTUAL REALITY 07

VR Platforms 08

General User Experience 10

User Comfort 12

4 SMARTPHONE AUGMENTED REALITY 25

The User's Environment 26

General User Experience 27

Object Placement 28

5 HEADSET AUGMENTED REALITY 36

6 CONCLUSION 39

7 REFERENCES 38

LETTER FROM THE MEMBERS

THE MISSION OF THE XR ASSOCIATION is to promote responsible development and adoption of XR globally through education, dialogue across stakeholders, and research. This document combines knowledge and experience from Google, HTC VIVE, Meta, Samsung, and Sony Interactive Entertainment, from its first release to today, to create a collective XRA development primer for XR developers across the industry. This document does not express the singular viewpoints of any member company, but rather aims to represent a primer and some application guidelines learned from the nascent days of XR. Example images in this Guide have been provided by the design team at Google.

Representatives from XRA member companies worked together to share what they have learned to date during the development of their respective platforms and to collaborate on a common set of focus areas for development in the XR industry. It is our hope that this guide will improve applications across all our respective platforms, as well as for the global community of XR users.

In the more than five years since the October 2018 release of XRA Developers Guide Chapter 1 immersive technology has advanced significantly. There are new device manufacturers and software developers in the market. Existing developer platforms have improved, and new custom platforms have come into existence. Some have predicted that the Immersive tech market will reach $170 billion USD by 2033. It was the right time to take a look at Chapter 1 to ensure it is standing up to the test of time.

For the most part the recommendations in this guide remain strong.

The 2024 revision includes:

1) General editorial corrections and style updates

2) A new section addressing Headset Augmented Reality

3) Expanding the AR section from smartphones to devices

4) Removing the section on “Inverse Kinematics and Proprioception” and move some of that guidance to “Teaching the Control Scheme” and added a Proprioception call out box.

5) Added a Reference Section

This information is grounded in human physiology, psychology, and ergonomics to achieve better user comfort and safety. While the XR Association member companies are evolving our XR technology in different ways, the attributes and capabilities of our users remain the same.

XR is dynamic, and each platform contains unique features and considerations. This document is not exhaustive. If you are developing an application for a specific AR or VR platform, we encourage you to familiarize yourself with that platform’s specific best practices, requirements, and guidelines, some of which are listed in the Reference section of this Guide. However, we believe that any XR application can benefit from the universal principles set forth in this document.

Thank you to our editorial review committee which included representatives from: Meta, Transfr, MagicLeap, Gamedriver, and others.

INTRODUCTION

XR IS AN UMBRELLA TERM that encompasses augmented reality (AR), mixed reality (MR), virtual reality (VR), and other forms of alternate, expanded, spatial computing, or immersive reality applications, including those not yet invented. By augmenting or digitally enhancing user's reality, XR applications immerse users in alternate places or display information and environments that interact with the user's reality. With this ability also comes responsibilities. The user is trusting that developers will provide them with an experience that is safe, comfortable, and expected. If XR experiences do not integrate basic application design guidelines, they can lead to discomfort in some users, with symptoms clustered around eye strain, disorientation, or nausea.

For example, certain types of images are also believed to be capable of triggering photosensitive seizures in a small portion of the population. The International Standards Organization has published ISO 9241-391:2016 as a standard for image content to reduce the risk of photosensitive seizures. Developers are responsible for reviewing the standards and literature on photosensitive seizures and image safety, and for designing content that conforms to these standards. More broadly, developers are responsible for designing, developing, and testing an experience that is safe and comfortable for users.This document introduces topics that XR developers should focus on for user comfort and safety in AR and VR. They are presented as a starting point and are far from exhaustive. Each device and platform may have different requirements, so it is important to incorporate those specifications into your development.

The XRA member companies have collectively learned a great deal about XR design over the last several years, but it remains early in the overall evolution of the platform, and we are still learning new best practices at a rapid pace. We strongly encourage developers to implement these guidelines while experimenting, trying out new approaches, and running extensive user testing.

As you discover new information, we would love to hear from you to incorporate your knowledge into these guidelines for XR design. You can contact the XRA on the issues at info@xra.org.

INTRO

VIRTUAL REALITY

VR is an immersive medium that has the power to completely replace a user's reality with a new virtual reality. This new reality can be fantastical, like a far away galaxy, or practical, like a training warehouse.

Many VR headsets feature 6DOF (6 degrees of freedom) tracking in which both the headset's orientation and position in 3D space can be tracked. Some headsets might only provide 3DOF (3 degrees of freedom) tracking in which only the orientation of the headset is tracked. The XR primer covers important safety and usage considerations to be aware of when designing and developing the VR experience for both formats.

These considerations center around three aspects of VR: general user experience, physical comfort, and psychological comfort.

Designing for the User Experience presents considerations about how users expect to interact with the VR environment, including recommendations around user acclimation, load screens, and session durations.

Designing for Physical Comfort introduces the difficult topic of what may make users uncomfortable in VR experiences and some mechanisms that can be used to avoid causing VR-specific discomfort, eye strain, and neck fatigue..

Designing for Pyschological Comfort reviews some of the ways that the immersive nature of VR experience may discomfort some users by the nature of the scenario that the user is placed in, including environmental design and architecture features, such as heights, sharp objects, large and small spaces, scale, and user agency.

AUGMENTED REALITY

AR is an augmented environment that layers virtual content, such as digital objects or information, onto a real-world image captured from a device's camera. Because AR experiences interact with the physical environment, environmental considerations, such as lighting and surface detection, become important.

The XR Primer reviews how developers can help users understand the ideal conditions for using the app.

Designing for the User's Environment reviews the concept of designing for many different environments and considerations about how people may use the app, including the private space of a room or the vast openness of a real-world environment.

Designing for the User's Experience details how users are just beginning to become familiar with a 360-degree XR environment and reviews considerations for how users may expect to consume augmented experiences.

MIXED REALITY

Mixed Reality (MR) is a combination of virtual or augmented and actual reality. MR puts the user in a scene where they will view and manipulate 3D objects in a real place. The technology affords the user the ability to interact with digital objects and virtual worlds. For example, an AR app might place a digital dog in a real park. Using MR, the user would also be able to reach out their hand and pet the dog.

MR environment can be designed to digitally copy objects in the real environment affording users the opportunity to virtually modify the object or maintain its persistence within the virtual environment. In addition, similar to AR, mixed reality allows for purely digital objects to be included in the environment. MR experiences interact with the physical environment, therefore environmental considerations, such as lighting and surface detection are important.

Developers should note that the term mixed reality is currently being defined in many ways. Some are more limiting than others but all have similar elements to AR of digital and real objects existing in the same immersive environment. While there is no clear agreement on the definition of the term there are still some considerations that developers should keep in mind when building for a mixed reality environment.

VIRTUAL REALITY

VR IS A POWERFUL IMMERSIVE MEDIUM.

It can create the sensation of being transported into a virtual (or real, but digitally reproduced) three-dimensional world, and can provide a far more visceral experience than traditional screen-based media. It can leverage the user's spatial memory, give the user a sense of scale, and in many cases, serve as an emotional amplifier.

VR is a medium that attempts to replace one's experience in the real world, so users are likely to have an expectation that they will be able to interact with that virtual world in the same ways they do outside of it. This interactive component has benefits and drawbacks. Balancing immersion, usability, and experience is the central challenge in VR design.

VR PLATFORMS

One of the first things an application developer should consider is the type of experience they are going to develop.

When developing, there are currently two categories of VR devices that developers can target, the primary difference being the type of tracking use. 6DOF (six degrees of freedom) platforms support full positional tracking and generally require 360-degree tracking equipment to translate the user's motion into the virtual world in real time. 3DOF (three degrees of freedom) platforms use orientation tracking and kinematic models to offer users an immersive experience.

6DOF (6 DEGREES OF FREEDOM) VR

6DOF devices, like the Meta Quest 3, HTC Vive Pro, and PlayStation VR2 provide tracking in six orientations (forward/back, up/down, left/ right, yaw, pitch, and roll). This means that the device is able to track users' head movements and their position in three-dimensional space.

For 6DOF applications, movement should be restricted to the predefined play area. Each device has a specific play area requirements, which are available on the device websites. Application developers may choose to support any number of 6DOF play styles, from standing room scale to a seated experience, depending on the application they are developing.

Most VR platforms provide some form of warning if the user exits their defined play space, so the platform's default may have existing established boundaries in some cases. If the VR platform does not manage this mechanism at a platform level, we recommend adding some mechanism that warns users if they are about to exit a safe space. This can be achieved in several ways, but some common approaches include displaying grid lines, presenting increasingly dense fog, popup notifications, or arrows indicating the way back to the safe area.

Use subtle cues to denote the safe play area.

Indicators of the safe play area can help set user expectations for how far they can roam before the safety indicators or warnings will activate. This can be achieved through subtle environment changes, like a virtual floor that changes material or texture, or elements that fit with the environment or scene, like a virtual workbench or cockpit.



Avoid game mechanics that encourage users to go beyond the safe play area.

The goal of the platform-level safety systems is to warn users that they should not leave the safe area. Mechanics that encourage the user to move beyond the bounds could encourage users to move so quickly that the platform isn't able to cue them, or they may miss these cues if they are overly focused on the experience itself. For example, a virtual object outside of the play area could be an unnecessary mechanic that creates extra risk to your players.

VR PLATFORMS

3DOF (3 DEGREES OF FREEDOM) VR

3DOF platforms, such as the older, Oculus Go, Samsung Gear VR, and Google Daydream, or the lower cost zapbox, Nreal Air AR, and Pico G3, track users' rotational motions (pitch, yaw, and roll), but not their translational motions (forward/back, up/down, left/right). 3DOF devices that use a controller assume the position of the controller based on arm and kinematic modeling.

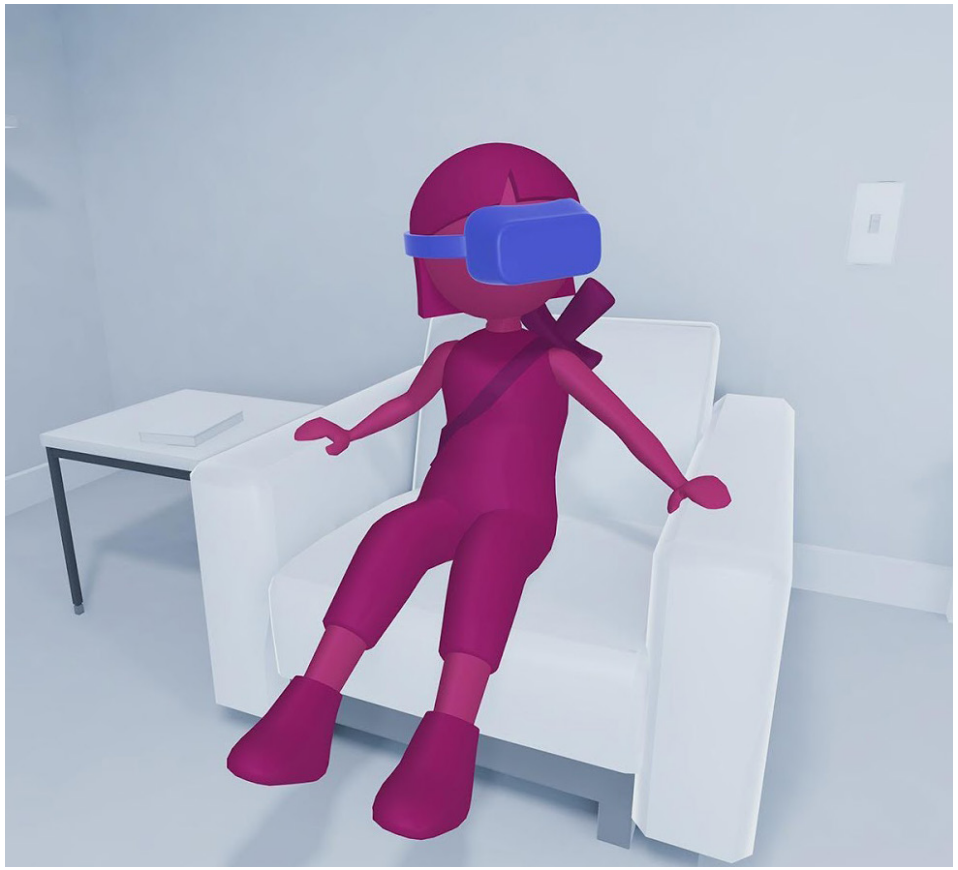
Designing for 3DOF VR can be challenging. Many users will use their 3DOF VR platform while seated in a comfortable chair or couch. Because not all chairs rotate in 360 degrees the players' accessible Field of View (FOV) may be limited, making immersive game design more difficult.

Provide users with controls for them to virtually rotate their view horizontally, in case they can't swivel in real life.

There are a number of implementations to achieve this effect, from discrete swipes that trigger a sudden jump of a fixed number of degrees, to momentarily restricting the user's FOV (called tunneling) while they do a smooth rotation of the camera.

There are a number of implementations to achieve this effect, from discrete swipes that trigger a sudden jump of a fixed number of degrees, to momentarily restricting the user's FOV (called tunneling) while they do a smooth rotation of the camera.

If the expected use case is for your users to be sitting, one approach developers have used is to place the virtual camera height at a distance off the ground based on a seated user in order to avoid the feeling of sitting in a virtual levitating chair. This decision should be content-specific. For example, if the user is in an environment with many standing characters, it may feel more comfortable for the user to be placed in a levitating chair and at the same eye-line as the avatars they are interacting with.



Use caution when representing the 3DOF controller as a hand, or hand-like object, implying that the controller may be used to grab or manipulate objects.

Multiplayer User Experience

Consider supporting local multiplayer where some of the players are in VR and others are engaging with 2D devices. A device agnostic approach may improve user engagement.

GENERAL USER EXPERIENCE

The guidance for general user experience focuses on the basic interaction between users and the VR environment. These include allowing users to define their own session duration, easing users into the new virtual world, avoiding blank or frozen load screens.

USER BREAKS

There is at least one way in which VR is similar to the traditional application ecosystem: ideally users should be able to pause most VR experiences any time or at intervals and not lose their progress. There are some VR experiences, specifically synchronous multiplayer apps, and learning and development, where this may not be feasible. In these instances, offer plenty of opportunity between sessions for users to take a break if needed.

Traditionally, some game design has used the pacing of save points in a game to intentionally create tension. For example, designing so players must navigate through a challenging sequence or solve a complex puzzle before they can reach the next save point.

The save point mechanic maybe problematic for VR for a few reasons:

The user may have a real-world interruption.

The user may want to take a break from an intense experience.

The user may want to look at or respond to a notification from another application.

For these reasons, it is best to offer users the ability to both save progress in real time and allow the user to resume an experience at any point. Additionally, well-timed suggestions to take a break could be a good reminder for users who might otherwise lose track of time.



GENERAL USER EXPERIENCE

ACCLIMATING TO A VIRTUAL ENVIRONMENT

It takes users time to acclimate to a new virtual world or to a new scene within a familiar virtual world. Immediate instructions or choices are harder for users to process while they are still under the cognitive load of acclimating to the new environment. When introducing users to a new virtual environment:

Start slow.

Begin with more sedate, slower-paced interactions that ease users into the experience and allow them to acclimate to the new environment.

Avoid presenting information or asking the user questions immediately following a scene change.

Users are unlikely to be paying attention to informational cues during the first moments in a new scene. They may not be able to effectively process additional information or make immediate choices.

Avoid unnecessary scene changes.

If the users are interacting with the User Interface (UI) in front of them, or accessing a menu, suddenly changing the environment around them will introduce cognitive load and detract from their ability to interact with the interface.

Avoid rapid scene changes.

Putting users into a sequence of rapidly changing environments will quickly become overwhelming. While this effect may not be nauseating, it can be confusing.



LOADING SCREENS

Loading screens may be unavoidable, but users should experience them as infrequently and as briefly as possible. Avoid presenting users with a blank loading screen, as after a few seconds of a blank scene, users may assume that the application has crashed and remove their headset.

Additionally, when you have to present a loading screen, always maintain orientation tracking and motion of the head.

USER COMFORT

Users report that they may feel uncomfortable wearing a VR headset for reasons that are beyond an app developers control. These include headset weight, headset warmth, or the amount of airflow inside the headset.

However, there are at least three areas of physical comfort VR application developers can affect:

VR-specific discomfort: a feeling similar to being motion sick or car sick.

Neck and arm ergonomics: tired or sore neck or arm muscles.

Eye strain: increased strain on the muscles used for eye movement.

All app developers should consider these three comfort areas when designing any VR experience.

VR-SPECIFIC DISCOMFORT CAUSES

One of the challenges of building VR apps is Visually Induced Motion Sickness (VIMS), a feeling of nausea or dizziness that may be triggered for some users by performance issues or design choices in a VR application. Regardless of a user's predisposition to nausea or dizziness, VIMS can be generated by a number of poor design choices.

The only way to gauge how users will respond to potential discomfort triggers in an application is to test the application with a large, diverse, and novice set of users. An application's user population is unlikely to be well represented by simple observations of its development team. Developers' familiarity with the application sets expectations about the experience ahead of time, reducing the likelihood of discomfort. Applications should therefore be tested by representative and unfamiliar user sets and include a range of user ages expected to engage with your experience.

Additionally, settings that allow users to modify the VR experience to match their comfort level may reduce the likelihood that users will experience discomfort. For example, providing different locomotion methods for users to select from allows users to engage with the application in their preferred locomotion method. This may be the most comforting method for users who are prone to motion discomfort or the most intense experience for users who are less likely to experience motion discomfort.

A NOTE ABOUT PERFORMANCE

Performance issues in VR applications, such as low frame rates, freezing, and stuttering, can cause some users to experience discomfort. Delivering VR applications which consistently render at 60 frames per second or higher (optimally at 90 or 120 frames per second, depending on the device's native refresh rate) can significantly reduce the likelihood that users will experience discomfort.

Many platforms introduce techniques to mitigate the impact to users in the event that an application fails to maintain frame rate or suddenly freezes, by continuing to track the user's head movement. However, developers should avoid relying on these mitigation devices. An application that freezes or drops frames in VR is a poor experience for all users. The goal is to create an application that doesn't rely on mitigation techniques.

The following sections outline some general design information to be aware of when designing a VR application.

Moving the User's Perspective

The player's perspective is commonly referred to as "the camera" in traditional 2D game design. In the past, this has meant what the player sees on their TV or computer monitor. In VR the camera is the user's head, or more specifically the user's eyes, as VR devices render two stereoscopic images. Developers should avoid taking control of the camera away from users. Moving the camera without the users moving their heads breaks immersion and quickly leads to discomfort.

This discomfort arises when the user's vision tells them they are moving through space, but their body perceives the opposite (vestibular sense and proprioception). This illusory perception of self-motion from vision alone has been termed vection, and is a major underlying cause of VIMS. Generally, avoid creating any unexpected disparity between what the user is seeing and what their vestibular system is sensing. The vestibular system is the inner-ear system that collects sensory information critical for motion, equilibrium, and spatial orientation.

Consider how poor graphics performance may induce discomfort in users. In a scenario where an app unexpectedly freezes while the user is moving their head, many users will experience immediate discomfort as their vestibular system is telling them that they are moving, but their vision is telling them that they are stationary.

To avoid creating an unexpected disparity in what the user is seeing and sensing, developers should avoid exaggerating or scaling rotation of the user's perspective, as this creates a disparity in what the user is sensing and seeing. Movement in VR should exactly match the real-world movement of a user's head.

Acceleration

Acceleration is any change in velocity in any direction. Acceleration can refer to decreasing the speed of movement, rotating or turning, and moving (or ceasing to move) sideways or vertically. Acceleration conveyed visually — but not to the vestibular organs — constitutes a sensory conflict that can cause discomfort.

The human vestibular system does not perceive constant velocity, only acceleration and deceleration. For example, flying in an airplane at 500 mph often feels like sitting still, aside from turbulence that causes changes to the plane's speed. The rider feels takeoff and landing because there is acceleration and deceleration.

Moving in VR at a constant velocity creates no disparity between what users see and what they feel, because both sitting still and moving at a constant velocity exerts zero force. This is aside from the constant 1g of force user's feel due to the Earth's gravity, which is also consistent between the user's real and virtual experiences. However, when observing an acceleration or deceleration in VR, for instance when riding a roller coaster or maneuvering a fighter plane, users expect the g-forces that come with these changes in velocity, which can cause discomfort.

USER COMFORT

Discomfort may increase as the frequency, size, and duration of the acceleration increases. Any period of visually-presented acceleration represents a conflict between senses, and it is best to avoid them as much as possible.

Unnatural Movement

Another trigger for discomfort is unnatural movement. Strafing (moving side to side) is common for first person video games played with a gamepad or mouse and keyboard, but humans are not physically constructed to quickly move side to side. Allowing users to move laterally in VR or allowing them to move in the direction of a controller they are pointing regardless of where they are looking, may make some users uncomfortable. Forward user movement is generally most comfortable.

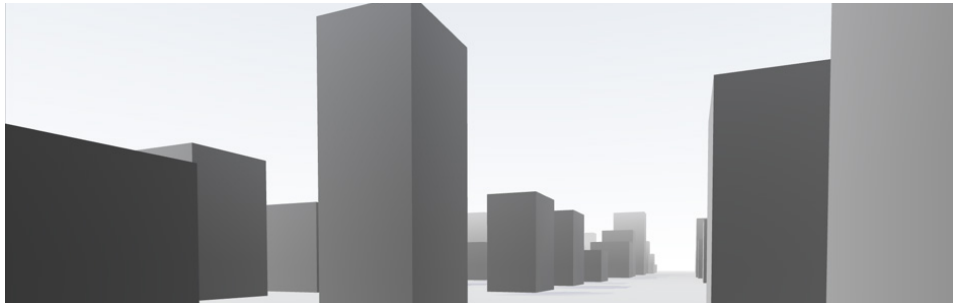
Vection Illusion

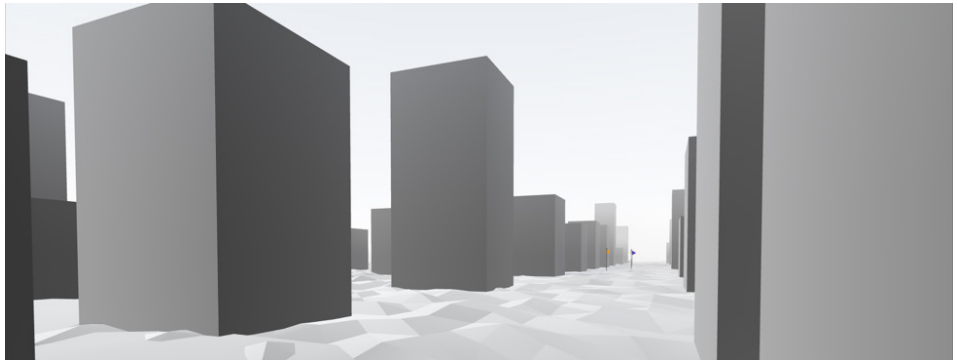
Unlike real life, VR users can't rely on their vestibular system to determine if they are moving; instead, they must make a judgment using only visual information. In some cases, users may misinterpret moving objects for objects that are stationary, thinking they are the one moving. This is also vection, which is an underlying cause of VIMS.

VR UI can cause vection when a large screen of thumbnails is presented in a large portion of the user's FOV. When the user begins to scroll, the user may feel like their head is moving, not the thumbnails in VR.

A real-world example of vection is when a stationary driver, watching the car ahead of them drive forward at a green light, has the sensation that their own car is traveling backwards.

In VR, any moving objects can create vection. For example, the lack of information on the ground in the scene above may cause confusion to whether the user or the cubes are moving in the scene.





Providing a more stable visual reference with texture on the ground makes it clear which of the two are moving.

Collisions

In 6DOF VR, users are able to move their heads into nearby virtual objects. If possible, applications should avoid rendering solid virtual objects close enough to the player that they may collide with them. For instance, if an app sends an enemy too close to the user, the user may move their head and clip into the enemy models and view their hollowed side, breaking immersion. One solution would be to design the game with the enemy always maintaining a distance from the player so they will not intersect with the virtual object in normal movement.

If it is not possible to keep distance from virtual objects, there are mitigation techniques that can be considered. However, there are a few mitigation techniques that should be avoided, such as freezing the camera so that it stays outside of the object, or using a physics collider. Both will create a disparity between what the user is seeing and sensing.

The ability for users to walk through virtual walls (either in room scale or with a controller) creates challenges for motion design. A common technique is to expose a black void, blurred effect, or grid environment beyond the wall. This helps to communicate that the player is exiting the expected play space without manipulating their head in a way that will cause discomfort.

USER COMFORT

If the user moves into an object that contains a physics collider and this object is outside of the user's current field of view, the sudden stop will be as discomforting as an app that randomly freezes due to performance issues.

Similarly, objects moving through a users' personal space and causing them to have to dodge out of the way can be a challenge, just as having objects or other players move through their virtual body can cause a feeling of "violation," which should be avoided.

Physics collisions are important for the interactions between different virtual objects in an environment, but should never affect the movement of the camera.

Horizon Movement

Moving the virtual horizon line is one of the quickest ways to induce discomfort in a user. Similar to moving the camera, there is a stream of visual information that does not match the forces that the user is feeling.

Take the example of a user on a boat at sea. While sitting still, the user is feeling 1g of force directly perpendicular to the real- world horizon. However, an oscillating virtual horizon line as the boat bobs on virtual waves implies that 1g of force should be pushing on the user from many angles. Even though the camera remains still, moving the horizon creates a disparity between the perceived and actual force being felt by the user.

Another way to induce a similar effect is to maintain the virtual horizon at a fixed angle that does not match the real-world horizon. For instance, angling the camera to give the user a better view of a sports stadium or a playing board will cause the user to notice the difference between the 1g of force being applied to them in real life versus what they are seeing and the constant disparity will bother them.

If there is no visible horizon in a virtual scene, the user will set expectations for the direction





USER COMFORT

of gravity by observing the objects around them. For instance, placing the user floating in front of an angled virtual pillar, or floating on an angled virtual platform, may convince the user that the virtual horizon has tilted, even without a visible virtual horizon line to confirm it.

Any sudden change to the horizon line or rotation of the camera may cause the user to experience discomfort. While rare, this is most commonly due to bad data coming from hardware sensors, software bugs, or someone accidentally knocking over an external tracker. In cases where the virtual horizon suddenly moves, users may react to steady themselves. Depending on how dramatically the virtual environment changed, a quick shift of balance to remain standing in the virtual environment could result in the user tripping in reality.

Backwards Movement

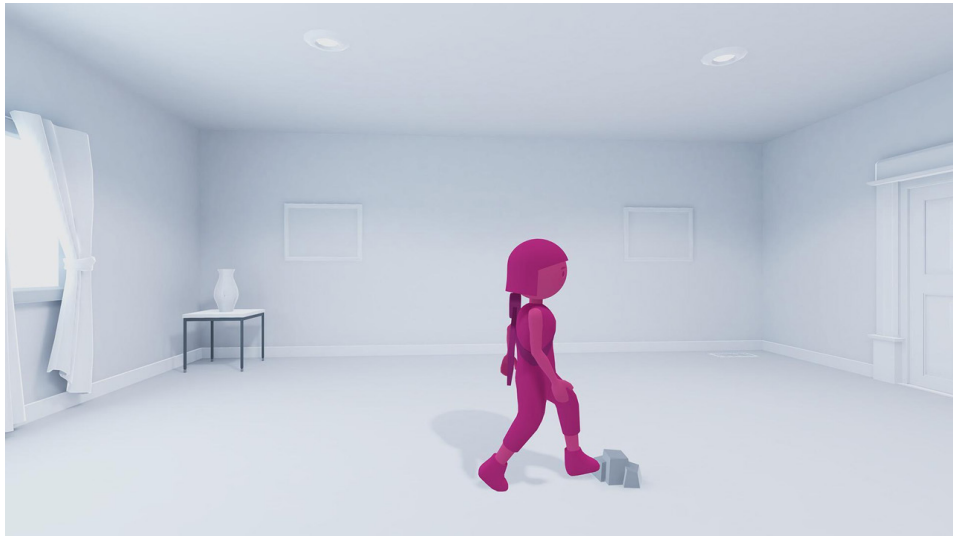
Avoid having users take steps backward as their ability to see objects like furniture or clutter in their environments behind them is limited when they are in a headset. Backwards stepping may occur when a virtual object is looming too close in their environment or when a virtual object is approaching the user rapidly and the user feels the need to make space. Developers should avoid those types of object placements or movements when designing an experience.

A NOTE ABOUT MOTION INTENSITY

Different users have varying tolerances for virtual motion, so it is important to separate the concept of motion intensity of an application from the concept of user comfort.

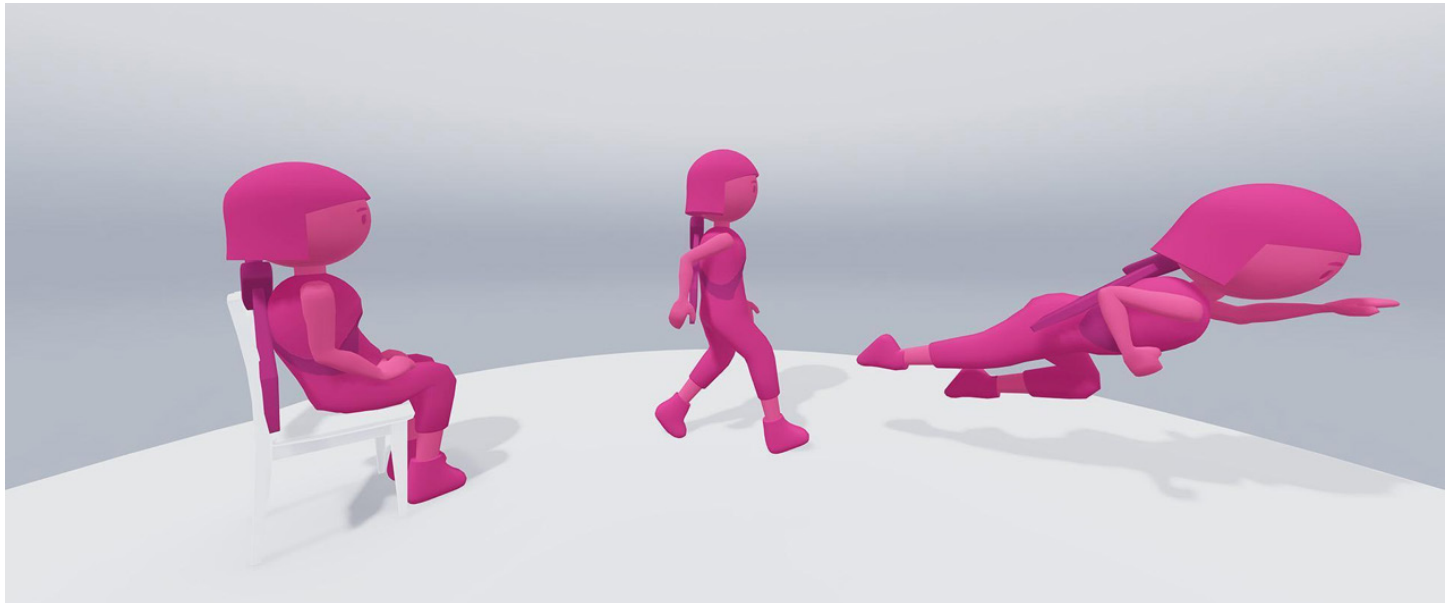
Comfort is a property of each specific user. Individual users will experience discomfort differently. Individual users may or may not be in the segment of the population that experiences discomfort from any particular mechanic.

Comfort consists of many factors beyond motion intensity. These include (but are not





USER COMFORT



limited to) neck and arm ergonomics, eye strain, headset weight, headset warmth, air flow, optics, and the general psychological comfort of an application. There are cases where VR applications that have no virtual motion are still considered to be "uncomfortable" by users.

Comfort holds an innate value judgment. Comfort is generally considered to be good, and discomfort to be bad. However, this may not be true for motion intensity. Some users who are less susceptible to motion discomfort may find that their favorite VR applications are the ones that contain the most intense sequences of virtual motion.

Developers are encouraged to review and understand how their apps will be rated before submitting an app for distribution.

MITIGATING VR-SPECIFIC DISCOMFORT

The previous section detailed the many ways that motion can cause discomfort for users in VR. This section will introduce some ways to move users around in VR without triggering discomfort (or are less likely to trigger discomfort). User movement in a virtual environment (beyond simply walking in real life) is commonly referred to as "locomotion." Below are a few ways to achieve virtual locomotion, in order of the amount of motion mitigated by the method. The common theme among these methods is that they are all intended to convince the brain that the user is not in fact moving, that essentially their senses are correct. This means that sufficient visual cues need to be present to reinforce the perception of the actual motion.

As mentioned earlier, to reach the broadest possible audience for an app, consider including a comfort mode with less intense motion.

No Motion by Design

The most effective means of avoiding discomfort from locomotion is to simply have no locomotion. Moving the player around a VR environment without input or explanation can feel artificial and break the immersive nature, so good narrative design is required to provide a mental model that maintains immersion.

One approach is to have players progress through discrete levels or areas, providing some mechanism for the transportation that doesn't expose them to vection. For example, the user might ride an elevator or other encapsulated mode of transportation, or have their view of the world fade to black for story reasons (e.g., losing consciousness, being blindfolded).

USER COMFORT

Teleportation

One common approach is teleportation. Teleportation allows the user to move around a virtual environment by magically appearing at a location that the user has specified. This is commonly done with a fade to black. Applications that leverage teleportation for movement can still maintain a motion intensity level of "no virtual motion."

While teleportation can be an effective tool to mitigate motion discomfort, it can also be disorienting if users are not given adequate information to acclimate them to their new position in the scene. The additional mental energy required to adapt to their new position may reduce users' sense of immersion or their ability to effectively interact with other app mechanics.

A technique that slightly modifies the teleport approach to locomotion is to very quickly warp the user to their new position. If this warping to the new location is done fast enough (<500ms), the virtual movement does not seem to cause discomfort for most users. Additionally, the warp effect allows users to maintain a constant sense of where they are in the virtual environment without having to look for unique landmarks and reorient after the teleportation has occurred. Warping between teleportation points is especially effective in environments that do not contain a lot of unique visual landmarks where users may be easily disoriented.

Tunneling

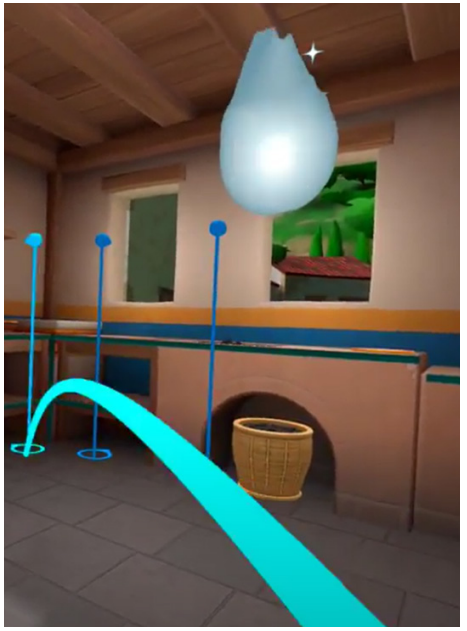
For first-person movement in VR, one approach to mitigating discomfort is to fade the user's peripheral vision out during movement, then fade back in when the user is no longer moving.

One reason tunneling is effective is because it leverages certain properties of human vision. The central vision, or fovea, can perceive more detail than the peripheral vision. "Foveated rendering" refers to intentionally rendering at a lower resolution in the user's peripheral vision, where they are less likely to notice visual degradation.

Peripheral vision has one major advantage over central vision: it is much better at detecting motion. For instance, you may only notice the flickering of a fluorescent light or an LCD monitor out of the corner of the eye. This is a result of the extreme motion sensitivity of peripheral vision.

By fading away virtual motion in the user's peripheral vision and replacing it with a static image, the user not only perceives less motion, they perceive less motion where they are most sensitive to motion. Since users have low resolution peripheral vision, users may not notice when tunneling effects are used.

While apps commonly fade to black in the user's periphery, an effective addition for mitigating discomfort is to introduce stable grid lines in the user's peripheral vision. Vertical gridlines are especially effective when





Images courtesy of Schell Games

USER COMFORT

the user is rotating, and horizontal gridlines are effective when the user is moving forward and back.

There are a few variations on tunneling that are more advanced. One approach is to only turn on tunneling when there is a surface close to the user, causing increased peripheral motion. For example, tunneling may be used on both sides when the user flies into a canyon, and only one side when the user flies next to a cliff.

Instead of replacing the user's peripheral vision with a changed virtual environment, an immersive approach is to obscure their peripheral vision with an object that fits the application's narrative. For example, this could be a helmet or the cockpit of a submarine. Allowing the user to sit in a virtual living room and watch a movie beyond one wall, or on a large 3D television, is also a type of tunneling.

Another variation of tunneling is to remove the texture information in the user's peripheral vision but maintain geometry and color. Similar to foveated rendering, this presents the user with less movement in their peripheral vision, and they are unlikely to notice the effect.

Blinks and Snap Turns

As mentioned previously, acceleration is one of the common causes of VR discomfort. This applies to rotational acceleration, not just linear acceleration. One technique to rotate the user comfortably is to use blinks and snap turns.

Blinks and snap turns omit the visual information that is likely to cause vection by simply cutting to black or instantaneously changing the viewpoint. The human brain is well suited to filling in interruptions in perceptual streams. For instance, visual interruptions that occur when blinking or looking around a room go unnoticed.

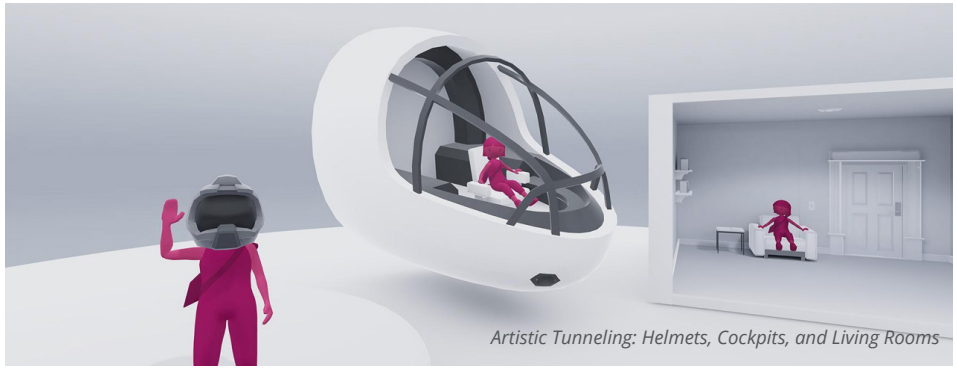
Leveraging this ability can help transition the user through motions that are likely to cause vection.

EYE STRAIN

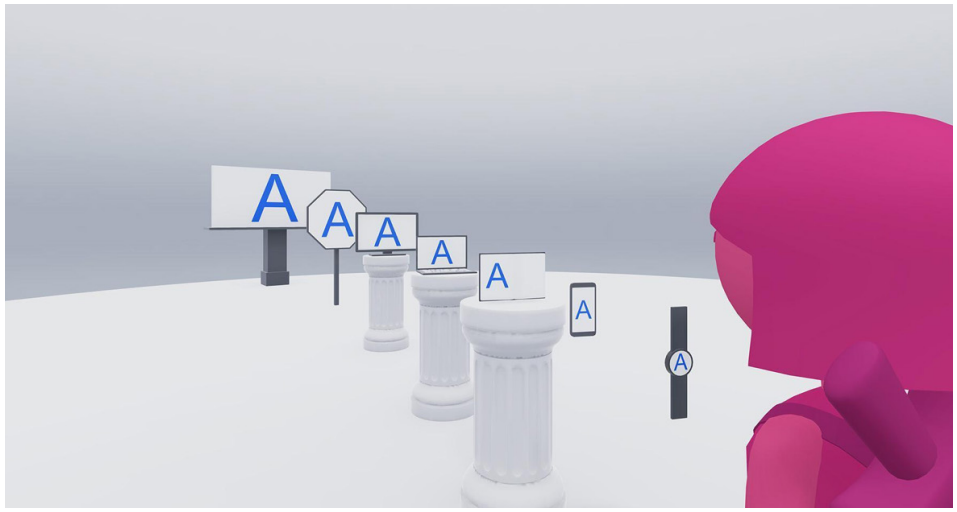
After motion-induced discomfort, the second source of physiological discomfort is eye strain. This section reviews some issues that can result in eye strain and an inability to focus on virtual objects due to conflicting visual cues.

Legibility of Text

Since objects can be placed at any virtual distance, the notion of a "readable font" needs to consider how far away that font will appear for the user. An easier way of thinking about text legibility is to make sure that all text is larger than a particular angular size. At farther distances, the font size may need to be increased so the angular size remains the same.



Artistic Tunneling: Helmets, Cockpits, and Living Rooms



USER COMFORT

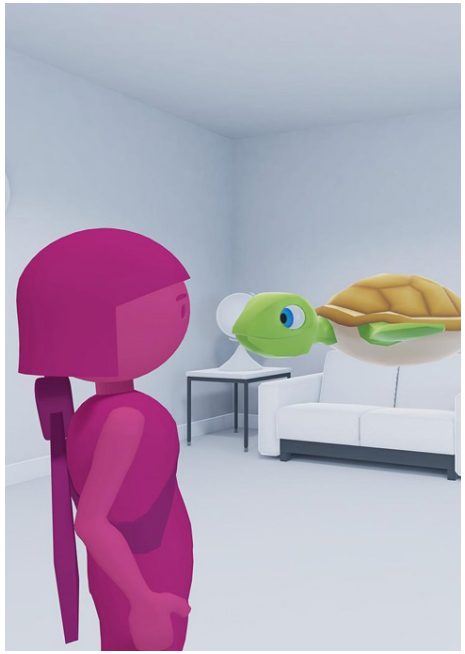
The minimum angular size depends on the pixel density of the display in the VR headset. Headsets vary widely in the pixels per degree (PPD). At that resolution, text that is comfortable to read will take up 1.5 degrees of the user's visual field of view.

Consider the languages the app supports. The complexity of combined letter combinations may affect the legibility of text. For instance, consider using a font that supports East Asian languages. Localization may also affect text layout as some languages employ more letters than others for the same copy.

When displaying virtual text, it is generally best to place it on an opaque surface in the scene. Text floating in space, without any background, or with a sufficiently transparent background, can create eye strain for users over periods of time. This is because the user's eyes may involuntarily refocus between the surface of the text and the background scene in the spaces between words or in open letters. These repetitive refocus operations exercise the user's eye muscles, which will eventually fatigue. It is especially important to background protect long passages of text. One option to avoid any potential eye strain for long passages is to employ a narrator to read the text to the user.

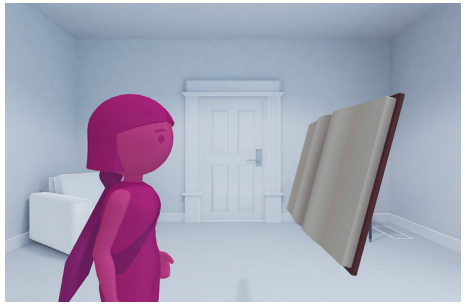
Focus

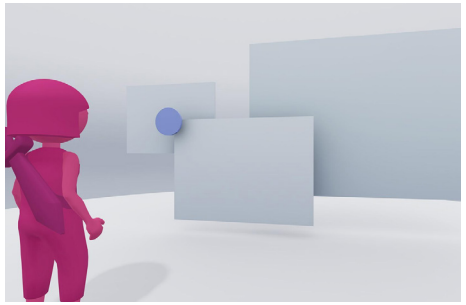
Two issues are of primary importance to understanding eye comfort when the eyes are focusing on an object in VR: accommodative demand and vergence demand. Accommodative demand refers to how your eyes have to adjust the shape of their lenses to bring a depth plane into focus (a process known as accommodation). Vergence demand refers to the degree to which the eyes have to rotate inwards so their lines of sight intersect at a particular depth plane. In the real world, these two are strongly correlated with one another; so much so that we have what is known as the accommodation-convergence reflex: the degree of convergence of your eyes influences the accommodation of your lenses and vice-versa.



VR creates an unusual situation that decouples accommodative and vergence demands, where accommodative demand is fixed but vergence demand can change. This is because the actual images for creating stereoscopic 3D are always presented on a screen that remains at the same distance optically, but the different images presented to each eye still require the eyes to rotate so their lines of sight converge on objects at a variety of different depth planes. If an object is placed too close to the user, they may not be able to converge their eyes on it, resulting in double vision.

While each VR headset is different, a general rule of thumb is to avoid rendering objects closer than 0.5 meters from the user.





Resizing Objects for Close Inspection

In some situations, the user may wish to inspect an object up close. Perhaps they are reading a note in the margin of a book or viewing details on objects in their inventory.

One way to facilitate this is to allow users to bring objects close to them, but then lock the objects at a distance users can still focus on.

If the user continues to "zoom in" the object remains locked at a fixed distance from the user, but becomes larger in the scene. From the user's perspective it appears that the object is getting closer and closer as it scales up. With this approach, the user maintains focus on the object without experiencing double vision.

USER COMFORT

Using Visual Cues for Depth

Another consideration for visual focus in VR is to give users enough details to understand where to focus. A completely featureless surface placed in front of the user may cause the user's eyes to refocus out to infinity.

Adding visual noise, gradients, or shadows cast onto an object will give users a sense that the object actually exists at a specific distance in space in front of them and is not a window to an infinite void.

Depth Perception Tricks

Stereopsis, the perception of depth based on disparity between the viewpoints of each eye, is the most salient depth cue, but it is only one of many ways the human brain processes depth information. Another depth cue is from motion parallax, the degree to which objects and different distances appear to move at different rates during small head movements. 6DOF VR headsets can translate motion parallax, but 3DOF VR headsets cannot.

ERGONOMICS

Beyond VR-specific discomfort and eye strain, a third aspect of the user's physiological comfort is ergonomics. Muscle fatigue in the wrist and neck is a common concern in VR.

Wrist Ergonomics

When the user is holding a flat controller (like a flashlight) in a natural position for the hand and wrist, the controller will be aimed upwards towards the ceiling, shown in red in the diagram to the right.

For wrist comfort there are at least two common techniques to resolve this (see insert).

Controllers that are physically constructed to have an angled grip.

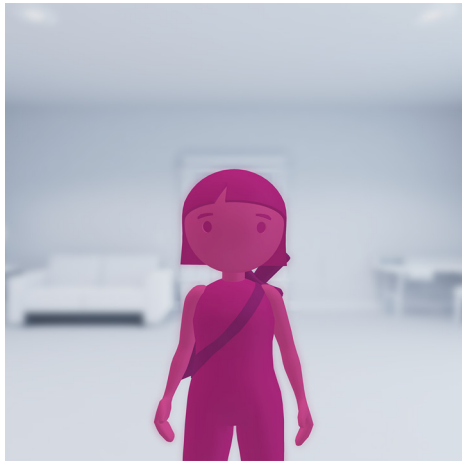
Many VR controllers are designed to have a section that naturally rests in the user's hand at an angle and a section that naturally points forward. These platforms do not require further design considerations.

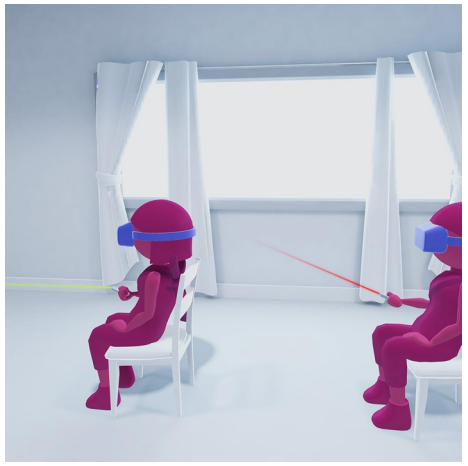
Software that angles the raycast of the controller downward.

For VR platforms that use a flat controller that does not have an angled grip, angling the raycast from the controller downward (shown in green) at an angle of 19 degrees or more may improve user wrist ergonomics.

Neck Ergonomics

For neck ergonomics, developers should avoid placing content that users will focus on for long periods far away from what you have defined as the center of the FOV for the app. Otherwise, users may need to engage their neck muscles to hold a turned position. This problem is often mitigated by many VR platforms having system-level recentering, which means that the controller and the headset's current pose will be redefined as "forward" in the scene. However, neck ergonomics should still be a consideration when designing VR applications.





PSYCHLOGICAL COMFORT

When designing VR environments, be mindful not only of the user's physical comfort, but also their feeling of psychological comfort.

There are several ways a virtual environment can have an emotional impact on users. Heightened reality may make virtual elements feel more real and the emotional impact may not always be positive (i.e. triggering fear or horror), but it is important that developers are intentional. The following are some common ways in which environment design and architecture can unintentionally trigger fear and psychological discomfort in users:



Heights

Some users are uncomfortable with places far off the ground. Use caution when placing the user in a scene where they will be more than a couple feet off the ground.



Sharp Objects

People are generally uncomfortable around sharp objects. This fear can be triggered by seemingly harmless objects in VR. For example, a hummingbird flying too closely to a user with their sharp beak may make users uncomfortable.



Large Spaces

Some users may be uncomfortable in large spaces; placing them in an expansive galaxy may make them uncomfortable.



Small Spaces

Some users may be uncomfortable in small or confined spaces; placing them in an elevator for an extended time may make them feel uncomfortable.



User Agency

Small details in an environment, such as a cup set on a wood table without a coaster or a crooked painting on a wall, can create unneeded distractions and be discomforting if they are not able to fix them.



Scale

Use caution when placing a user in a virtual scene where they are much smaller than the other objects in the scene, especially adversaries. This may make the user feel powerless or vulnerable.



Graphic Warning

Some users may be uncomfortable experiencing intense graphic elements such as blood, nudity, realistic weapons or violence. Including a graphic warning(s) for content before the user enters the experience can give the users an option to forgo the experience.



User Harassment

Developers should be aware of the tools that operating systems and device manufacturers have to assist users in managing their safety and level of experience while in immersive applications where there are multiple people. Functions, like muting and space bubbles can be made available to users.

INPUT AND CONTROL

Most current VR devices, both 3DOF and 6DOF, include some form of controllers. Each manufacturer gives recommendations for the use of their specific controller, but what follows is applicable to every controller and platform.

TEACHING THE CONTROL SCHEME

Controllers appear in both physical and virtual space, so one way to teach users the control scheme is to use labels on the virtual controller that directly map to the actual controller buttons. This is more intuitive for users than text placed somewhere in the environment that refers to the controls with names or symbols.

While it is common for 3DOF VR applications to render the virtual controller, 6DOF applications often render hands in place of the virtual controllers or hands with virtual controllers. It is important to get the hands right if they are part of a 6DOF app. Even small differences in position or orientation are noticeable and can break immersion. This also affects virtual hand models that contain fingers if the system does not do accurate finger tracking.

Even if the user's hand position is rendered exactly, differences between the virtual avatar and the real user can break proprioception. Users may readily accept rendered hands that mimic their real-world hands, but still break presence if the hands are not in keeping with the scene. For example, a set of realistic hands would be out of place in space, where users expect a space suit.

Another consideration when implementing realistic hands is that not all users look the same. Allow users to customize their hands to match their skin tone.

MAINTAINING PROPRIOCEPTION

Perfectly reproducing the actual controller in virtual space can feel boring or even reduce the user's feeling of presence since the controller becomes a constant visual reminder of the outside world. This can be mitigated by adapting the virtual controller to a game's or experience's art style.

However, there is one important caveat: it is important to maintain the exact size and position of any controls (buttons, touch pads, joysticks, triggers, etc.) for the virtual controller as for the physical controller. Any detail that the user can feel on the actual controller will bother them if the same detail is presented slightly differently on the virtual controller. For instance, shifting a button or making the entire controller thinner, may feel wrong to users.



Image courtesy of Sony Interactive Entertainment

PROPRIOCEPTION

Proprioception is the body's ability to sense its position, movement, and actions. This sense allows us to move smoothly and coordinate our actions effectively, and is crucial for activities including walking, touching our nose with our eyes closed, or typing without looking at the keyboard. For VR developers, developing applications with accurate proprioception feedback can enable users to interact with virtual objects naturally, enhancing realism and the overall user experience. In applications such as VR training simulations, precise proprioception is essential in developing real-world skills.



In the image on the top right of this page, the overall size of the virtual controller and precise position of buttons match the industrial design of a physical controller, but the aesthetic of the controller has been adapted to the application's art style.

Issues with controller design affecting the user's ability to maintain presence seem to be

INPUT AND CONTROL

most significant when the virtual controller design is only slightly wrong. If the controller is completely replaced in VR with another object, like a flashlight, banana, or fishing rod, the effect doesn't seem to be as noticeable.

The effect also only impacts areas where the user has engaged their muscles and can feel the controller. For instance, if the hilt of a virtual sword that the user is holding matches the width of the controller, the user will probably still be satisfied with the design of the sword.

Equally as important as the virtual representation of the controller is how the user moves it in the scene. Avoid scaling or exaggerating the virtual movement of the controller, and instead move the virtual controller exactly as the user moves the controller in the real world.

PERCEIVED OBJECT WEIGHT

While replacing the controller with an entirely different object (like a sword or fishing rod) isn't likely to trigger proprioception issues, a user may notice when the weight and center of mass are wrong. The more the user is presented with an obvious difference in object weight, the more it is likely to reduce their sense of presence.

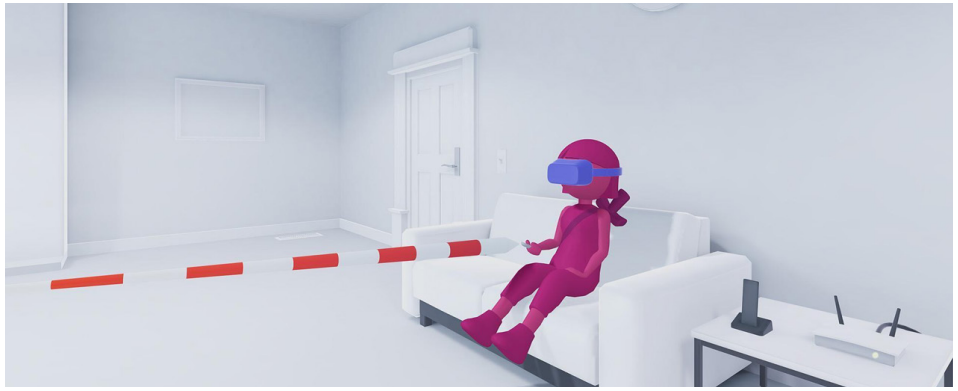
While this effect is usually triggered with more realistic virtual objects (battle axes, swords, etc.), it can be equally problematic for interface components. For instance, a large and solid raycast coming off from the controller to help with targeting can seem like it should have some weight associated with it.

This can also be true of large solid menu panels attached to a controller, versus more holographic or open visual designs.

CONVEYING VIRTUAL WEIGHT

While virtual objects cannot exert any pressure or resistance on the controller to convey a sense of weight (at least not without





robotic or haptic technology included a part of the users VR rig), there are some software approaches to conveying virtual weight.

For instance, the virtual object can resist the raycast used to select it, causing the raycast to bend. This will create the perception that the object has mass. Light objects will only bend the raycast a small amount, while very heavy objects will bend the raycast significantly. Heavier objects may also be harder for the user to move around than lighter objects.

HEAD GAZE INTERACTIONS

While most active interactions in VR are mediated through the controller, the user's head gaze can also be useful for information or a passive interaction, especially in 3DOF modalities where controller input options are limited. For instance, head gaze can display annotations or additional information in the virtual environment, acting as a form of focus selection.

Head gaze can also be used to trigger events in the scene, like the progression of the story, or something happening where the user is not looking (most commonly used

INPUT AND CONTROL

for jump scares). For certain interactions, head gaze can be used in combination with the controller raycast. For instance, if the controller raycast is intersecting multiple objects that can be selected, head gaze can be used to disambiguate which object the user is attempting to select.

EYE TRACKING

In-headset eye tracking technology is evolving, and only a few suppliers are using it for user interface. One example of eye tracking, is the Apple Vision Pro, where eye tracking is used as a primary input modality. In this instance, the eye gaze is used as a highlighting mechanism in their main menu and other interface elements. It combines with hand tracking of pinch-gestures to allow the user to select an item from a menu simply by looking at it, and making a small gesture with their hand. Other examples of the use of eye tracking are the Quest Pro and MagicLeap ML1. This new modality has not been extensively studied for its effect on eye strain, mental fatigue, and error-rates for input. As more headsets start to support eye tracking, we will learn more about how effective this modality is for different applications.





SMART DEVICE AUGMENTED REALITY

AN AUGMENTED ENVIRONMENT LAYERS VIRTUAL CONTENT.

Digital objects and information are layered onto a real-world image captured from a device's camera. The user can reorient their device to explore the objects from different angles and, if appropriate for the experience, interact with objects using gestures and movement.

AR software platforms, on devices like smartphones, use a process called concurrent odometry and mapping, or COM, to understand where a user's device is relative to the world around it. These platforms identify visually distinct features in the captured camera image, called feature points, and use the points to compute the device's change in location. The visual information is combined with inertial measurements to estimate the pose (position and orientation) of the camera relative to the world over time.

Virtual content is rendered from the correct perspective by aligning the pose of the virtual camera that renders 3D content with the pose of the device's actual camera provided by the device AR software platform. The rendered virtual image can be overlaid on top of the image obtained from the device's camera, making it appear as if the virtual content is part of the real world.

AR is well known by users and can be found in retail, entertainment, and enterprise solutions. However, the rapidly developing nature of the technology and industry means that there are many aspects of it that require additional research. The guidelines for AR app development outlined below should be considered to be evolving.

THE USER'S ENVIRONMENT

Designing for device AR means designing for many different environments, from the private space of a room, to the vast openness of the outdoors.

Anticipate and communicate any potential challenges that may hinder users' physical capabilities and movements, such as difficulty with tracking and occlusion in public spaces. Help users understand ideal conditions for using the app. Give them a clear understanding of the amount of space needed for the task at hand, from a tabletop, to an entire room, to large-scale space. And, consider how device movement and use of phone-based AR might be unsafe or hinder real-world interactions.

Considering that many of those who have access to such technology sometimes put themselves in harm's way because they use augmented reality experiences in unsuitable situations for AR programs, developers should consider including precautions for their programs to keep users aware of their surroundings. For example, if a program detects someone is driving, they could stop users from using their experience.

SURFACES

Smartphone AR software platforms are constantly improving their understanding of the real-world environment. They can cluster and recognize feature points that appear to lie on common horizontal and angled surfaces and make these surfaces available to the app as planes.

For now, limitations that may hinder accurate detection of surfaces include:

Flat surfaces without texture, such as a white desk.

Low-light environments.

Extremely bright environments.

Transparent and reflective surfaces, such as glass.

Dynamic or moving surfaces, such as blades of grass or rippling water.

When users encounter environmental limitations, indicate what went wrong and point them in the right direction.

LIGHTING

AR requires an exceptional level of detail to overlay believable-looking objects onto real-world environments. Accurately detecting the light type, source, and intensity are important components of selling realism.

To improve fidelity, smartphone AR software platforms use light estimation features. When no light estimation APIs are present, false ambient lighting can add depth and realism to assets in the scene.





GENERAL USER'S ENVIRONMENT

General user experience considerations in AR include user movement and comfort as well as environmental limitations and safety hazards.

USER MOVEMENT

Users are becoming familiar with 360-degree XR environments. When initially engaging in AR device experiences (i.e. smartphones, tablet, glasses, etc.), they tend to remain stationary due to habits built by 2D mobile interactions.

Nudge users if the experience demands exploration or if they're so immersed in the experience that they freeze. For example, placing a virtual character slightly out of view will compel the user to move to explore it.

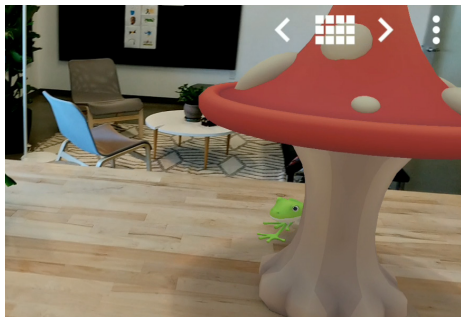
Arrows along the bottom of the screen can likewise guide users toward offscreen virtual objects.

Based on each user's environment and comfort level, their movement while using a smartphone AR application can go through four phases:

1. Seated, with device fixed
2. Seated, with device moving
3. Stationary with device fixed
4. Full range of motion

AR ADAPTIVE WARNINGS

When designing AR applications, it is crucial to implement adaptive warnings that respond to users' movement and context. If an application detects that the user is engaged in activities that may compromise their safety or the safety of others, it should display a clear notification advising them to use the application responsibly. Ensuring that warnings are dynamically updated based on the user's context can protect them from potentially hazardous activities.



For each phase:

Design for comfort, making sure not to put the user in an uncomfortable situation or pose. Avoid large or sudden movements.

Provide clear direction when a user needs to transition from one movement to another.

Educate the user about specific movements required to trigger an experience.

Give clear guidance on movement range and necessary adjustments to user's position, pose, or gesture.

While movement can be engaging, avoid enforcing user movement until it is necessary.

GENERAL USER'S ENVIRONMENT

ACCESSIBILITY AND EASE OF ACCESS

If the user isn't able to move around, consider providing an alternative experience. For example, instead of encouraging users to move closer, emphasize other activities, such as object interaction or transformation.

Keep in mind that users may move less as they fatigue. This makes it particularly important to ensure that any text or annotations are clear and legible from any angle.

USER COMFORT

Holding up a mobile device for a prolonged period of time can be tiring. Allow the user to define what is comfortable for them.

Consider users' physical comfort during all application phases.

Be mindful of the length of play experiences. Consider when users might need to take a break.

Be especially mindful of hand movements and the need for the user to take a break.

Let users pause or save their progress, making it easy for them to continue an experience where they left off even if their physical location has changed.

REDUCING USER FRUSTRATION

Try to anticipate and reduce user frustration. User frustration can be subjective. Consider a tutorial or user guide which the user can access during the experience.

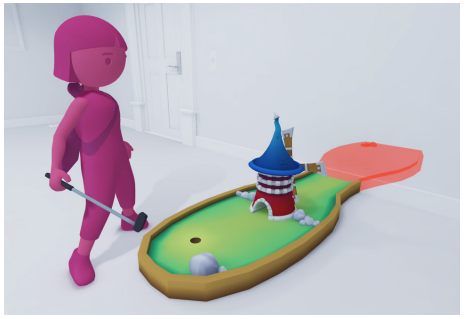
Anticipate the limitations of users' spaces — both indoor and outdoor, the space's physical size and any obstacles, including furniture, things, clutter, pets, and people. While it is impossible to know where the user will be when using an app, try to provide suggestions or feedback to reduce user frustration.

Give users a clear idea of the amount of space needed for the experience.

Remind users to be aware of their surroundings.

Avoid asking the user to walk backwards.

Avoid placing large objects directly in front of users, as this will cause them to physically back up.



OBJECT PLACEMENT

Object placement refers to putting virtual content in a real-world environment. For users to do so successfully, they must first scan their environment for surfaces on which to place the asset, then select an appropriate viewing distance, and finally, make the placement through manual or automatic action.

SURFACE DETECTION FEEDBACK

Educate users how to move their devices to scan their environments by providing clear instructions and visual cues. For example, a clockwise or circular motion animation can help users understand what movement is required. As users move their devices, provide simultaneous feedback to indicate surface detection and instructional cues about what to do next (such as placing an object or modify device movement).

Design seamless transitions.

As the user moves quickly, the system might lose tracking. Design for a smooth and subtle transition between surface finding and detection.

Be clear about which surfaces have been detected.

Distinguish between surfaces that have yet to be detected and ones that have been successfully detected. Consider visually highlighting valid surfaces with a semi-transparent treatment.

Aim for visual consistency.

The visuals used for each state should also share aesthetic attributes for the sake of consistency and visual harmony.

Use progressive disclosure.

Communicate changes in system status in a timely and relevant way. Visual highlighting or displaying text can help communicate successful surface detection.



SURFACE VISUALIZATION

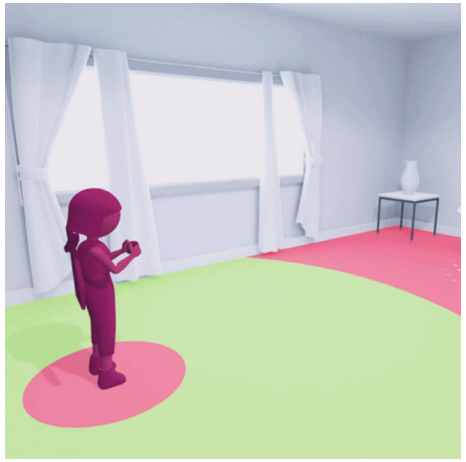
A lack of visual cues during plane discovery can cause confusion about whether or not virtual objects have been placed on valid planes. For example, when placing a digital piece of furniture in a room, make sure it doesn't pass through the floor or wall.

To avoid this:

Highlight surfaces that have been detected and are ready for object placement.

Create visual distinctions between different planes to avoid any confusion for future placement of 3D assets.

Visually highlight only the surfaces that the user is viewing or pointing at, taking care not to highlight multiple surfaces at once.



OBJECT PLACEMENT

OPTIMAL PLACEMENT RANGE

An optimal placement range indicator helps ensure that the user will place an object at a comfortable viewing distance. Guide users to optimal object placement by clearly labeling regions of the screen, setting maximum placement distance defaults and/or using visual indicators for object destinations.

Scene Staging

The limited field of view on a mobile screen or glasses can pose challenges for users to perceive depth, scale, and distance. This may impact users' experience of the app and their ability to interact with objects. In particular, object placement alters depth perception. For example, placing an object too close to the user can be startling or alarming, while placing large objects close to the user may cause them to step back.

Maximum Placement Distance

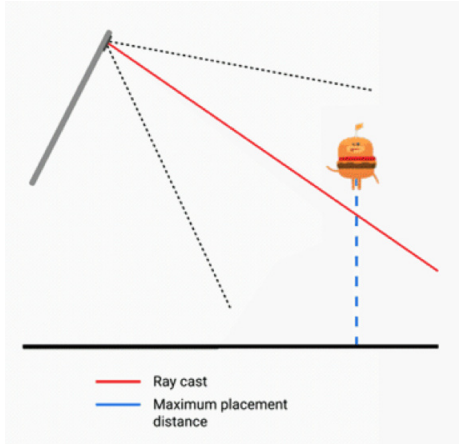
Setting maximum placement distance defaults helps ensure users place objects at comfortable viewing distances. The object will then maintain realistic scale as the user continues to position it.

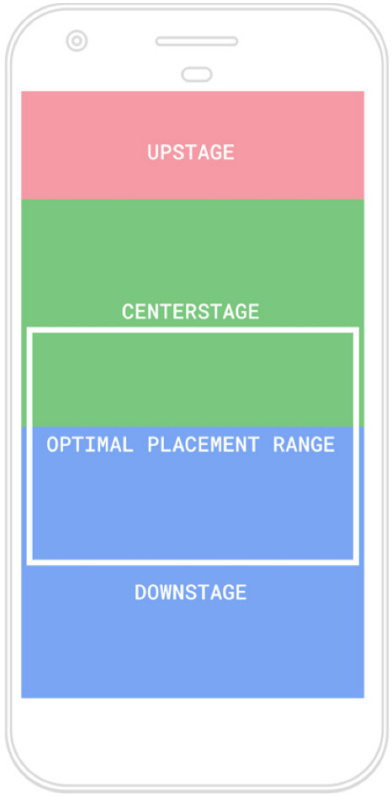
Destination Point

The destination point refers to where an object will eventually be placed.

Visual indicators for the destination point, such as shadows, can make it easier for users to understand where an object will be placed on a detected surface.







OBJECT PLACEMENT

TYPES OF OBJECT PLACEMENT

Placing objects in AR within a real-world environment can occur in two primary ways: automatically or manually. In automatic placement, the AR system's built-in sensor detects suitable surfaces or areas for objects to be placed. In manual placements, users have full control over where and how virtual objects are placed.

Auto Placement

Auto placement is app-controlled and refers to when virtual objects populate the scene once a surface is detected.

Auto placement works well when:

A virtual environment is augmented over the real-world space, such as in an RPG game.

Minimal or no interactions are required. Precise object placement is not required.

AR mode is crucial and should kickstart on launch.



Image courtesy of Google ARCore

OBJECT PLACEMENT

Manual Placement

Manual placement is user-controlled and refers to users' ability to physically position and move an object in the scene. This could include anchoring a playspace or setting up a location to start a smartphone AR experience. There are at least three types of manual object placement.

Tap-to-Place

Tap placement allows a user to drop a pre-selected virtual object by tapping a location within the scene.

Tap behavior generally comes naturally to the user, and works best when:

The virtual object doesn't need significant adjustment or transformation (scale/rotation) before placement.

Quick placement is required.

Tap behavior doesn't work well when multiple objects automatically appear or spawn in the scene. When tapped, multiple objects can pile on top of each other.

Drag-to-Place

Drag placement is ideal for high-precision object placement, allowing the user to drag a virtual object from a gallery into the scene.

Drag behavior works best when:

The virtual object needs significant adjustment or transformation.

The placement requires a high degree of precision.

Scale is a critical part of the experience.

Drag behavior doesn't work well when the user is not informed about the placement gesture in advance. Since drag behavior can be less familiar to users, provide clear guidance and onboarding when enabling it.

Free Placement

Free placement allows for object placement when no surface is detected. If levitation or elevation is intentional, provide clear visual cues and guidance to the user. If a surface isn't detected, let the user know that the object isn't anchored to a surface.

Guide the user to successful object placement by:

Blocking input until surface detection is complete. This prevents users from placing objects in the scene without a valid plane.

Providing visual and/or textual feedback that the object hasn't been placed yet. For example, use hover animation and text to communicate what the user should do next.

If virtual objects appear in the environment without surface detection, it can create confusion, since the object may appear to be levitating. This also bars the user from interacting with the object or space in a meaningful way.





Image courtesy of Google ARCore

OBJECT PLACEMENT

MANIPULATING VIRTUAL OBJECTS

Any interactive objects placed by the user in a scene should be easily manipulated, from selecting the object, to moving it between surfaces, to scaling it larger or smaller. There are multiple ways to do this including Selection, Translation, Multi-surface Translation, etc.

Selection

Selection allows the user to identify, manipulate, or interact with virtual objects.



Design for intuitive object selection by:

Creating visual indicators that highlight the objects users can interact with. This is especially critical in a scenario where multiple objects can be selected.

Maintaining the original visual integrity of the virtual object, while taking care not to visually overpower the virtual asset.

Translation

Translation allows the user to move a virtual object along a surface or from one surface to another. To move an object, the user can select it and either drag it along the screen with one finger or move the actual device.

Multi-Surface Translation

Avoid sudden transitions or changes in scale during translation:



Visually distinguish between two surfaces.

Show surfaces that are ready for placement before the user releases their finger.



Avoid sudden transitions that could create the illusion of changes in scale.

Translation Limits

Adding a max translation limit can prevent users from translating the object so far into the scene that it is rendered impossible to view or manipulate.

ROTATION

Rotation is the ability to orient the object in a desired direction. Rotation can be either auto or manual.

Manual Rotation

Support a two-finger gesture on the device's screen for manual rotation. To avoid conflicts with scaling, require that the fingers move in the same direction on a horizontal axis.

Auto-Rotation

Avoid auto-rotation unless it is an intentional part of the experience. Persistent auto-rotation can be disconcerting to users. For example, if the object's orientation is locked to face users, restrict auto-rotation if they try to manually change the object's orientation.

SCALING

Scaling is the ability to increase or decrease the size of an object, most commonly with a pinch gesture. It is generally appropriate when an object has no intrinsic size, such as a game character or toy. Scaling is generally not appropriate when an object has a definite size in relation to the real world, such as a piece of furniture.

Constraints

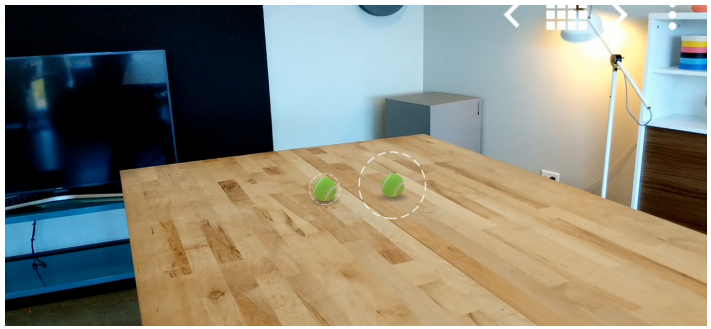
Setting minimum and maximum scale limits can improve the user experience. Smaller scale values allow users to have more precise compositional scenarios. Consider adding a bouncing effect to indicate maximum and minimum size. Or, snap the object if the object has achieved true scale.

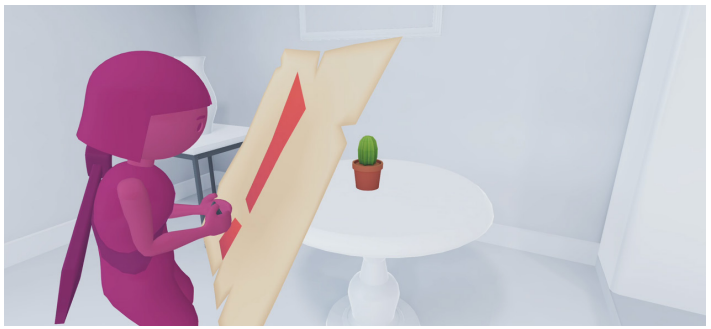
Scale and Gameplay

Scale can be a playful aspect of the user experience. For example, a large dinosaur in the scene can add a surprise element. Consider using scale to elicit delight.

The addition of sound can also be used to communicate scale and make the experience even more delightful. By utilizing an altering scale pitch, for example from low notes to high notes as the object grows and shrinks, can encourage immersion.

OBJECT PLACEMENT





VOLUMETRIC INTERFACE DESIGN

Volumetric interfaces allow users to interact with objects that appear to have three-dimensional depth. Volumetric interfaces necessitate a range of design considerations.

Gestures

Direct touch interaction with a virtual object is both more immersive and more intuitive than interaction through a separate control scheme. However, direct manipulation of virtual objects on a smart device screen poses its own challenges. Touch gestures are inherently two-dimensional, but the AR environment involves the three dimensions of the real world.

Proximity

Precise handling may be challenging to users for objects that are small or distant. Assume that a user is attempting to interact with an object if the app detects a gesture near an object. Provide reasonably sized touch targets despite the object's small size. Consider the size of touch targets to allow for optimal interaction.

Accidental Gestures

Create a consistent system for gestures and interactions. Assign gestures to specific interactions or tasks, while avoiding the use of similar gestures to complete different types of tasks. For example, a two-finger pinch gesture used for object scaling should not also be used for object rotation. Accommodate various two-finger gestures. Two-finger gestures are commonly used to rotate or scale an object.

These can include:

Use of index finger + thumb rotation.

Use of thumb + index finger with thumb as anchor and index rotating around the thumb.

Use of two thumbs independently.

Include these touch interactions as a part of two-finger gestures.

Users with a hands up or other position for too long may experience muscle fatigue. Build in breaks or force a break so users can put down their hands.

MAINTAINING IMMERSION

Design for an immersive experience, but empower users to feel in control. Allowing users to customize their experience to fit their personal preferences and needs, can ensure a more engaging and satisfying interaction with the AR application. This can include allowing customizable avatars, control inputs, notification preferences, and audio settings. By incorporating these customizable aspects, AR applications can better meet the diverse needs of preferences of users, enhancing their overall engagement and experience.

Avoid full screen takeovers unless the user explicitly selects it.

Avoid persistent 2D overlays that can disrupt immersion.

Allow users to perform primary and secondary tasks within the app.

For example, allow users to select, customize, change, or share objects without leaving AR, and avoid flows that frequently take the user in and out of a scene.

OBJECT PLACEMENT

ONBOARDING AND INSTRUCTIONS

Allow users to launch the smart device AR experience quickly and teach users how to perform key tasks contextually in the first-run experience. This will tie helpful instructions to relevant tasks and build retention.

While adding contextual onboarding tips:

Dismiss cues once the task is completed.

Offer tips or resurface key onboarding visuals if the user repeats the same mistakes.

Rather than relying solely on text, use a combination of visual cues, motion, and animation to guide users. For example, users understand a swipe gesture better if it is demonstrated to them on-screen, rather than communicated via text-only instructions.

FAMILIARITY

Leverage familiar digital UI patterns and conventions. Align to standard UX interaction models and patterns without breaking the immersiveness of the experience. The use of familiar icons and actions will help reduce the need for providing instructions or detailed onboarding (see Resources). For example, the use of "x" as a close icon is well recognized by users.

LANDSCAPE AND PORTRAIT MODES

Supporting both landscape and portrait modes creates a more immersive experience and increases user comfort. If this isn't possible, select the one that is best for the AR experience. Consider the following:

Camera and button placement

For each mode, be aware of how camera positioning affects depth sensing, spatial awareness, and accurate surface measurements.

Layout

When appropriate, change layouts for secondary targets.

One-mode support

Make it clear to the user if support for only one mode is available.

Key target placement

Avoid displacing key targets or allowing for rotate animation.



Image courtesy of ReframeXR

OBJECT PLACEMENT

AUDIO EXPLORATION

Audio can encourage engagement and enhance the user's experience. Be mindful of the following though while infusing audio in 3D objects or the 360-degree environment:

Add attenuation for sound effects.

Allow audio to fade or stop if user is not handling the object.

Allow users to manually turn off the audio of selected objects.

Avoid playing sounds simultaneously.

Always ensure sounds add to the experience rather than distract from it. Sounds are also excellent tools for navigation and environmental understanding for users with low vision.

VISUAL EXPLORATION

Visual or audio cues can encourage off-screen exploration. For example, a bird flying off-screen until the user finds it in the scene, can help direct the user to complete the intended goal.

AVOIDING VISIBLE DEPTH COLLISIONS

To avoid depth collisions — when a virtual object appears to intersect a real-world object— be mindful of reasonable room sizes and the various environments in which users will use the app. Set expectations up front by clearly communicating how much space is needed for the experience, from a tabletop, to an entire room, to a world-scale space.

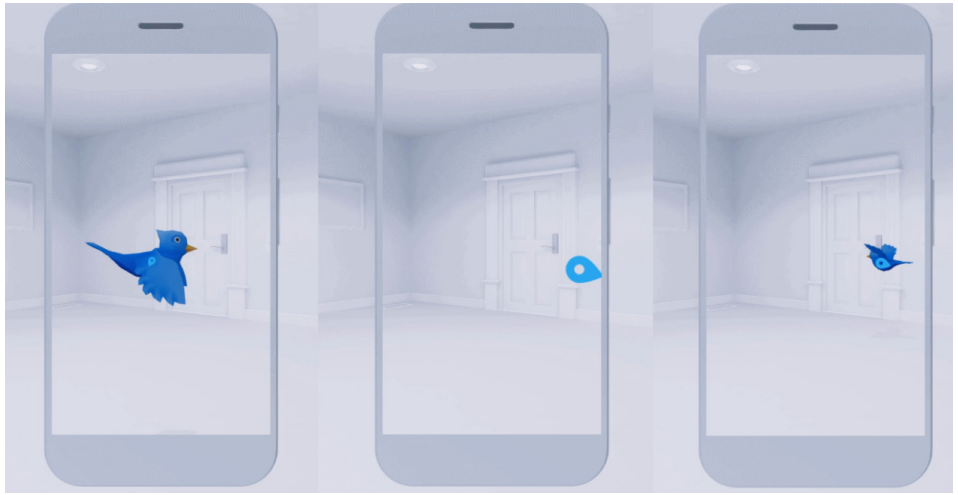
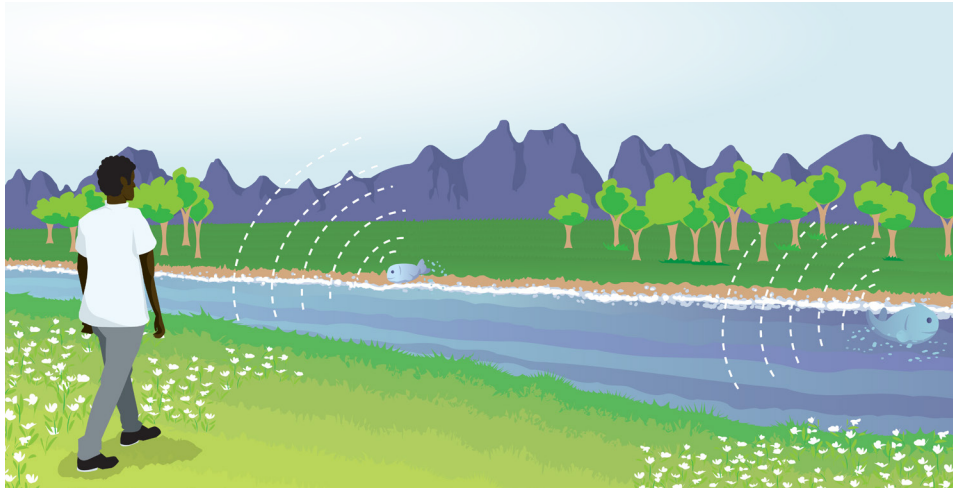


Image courtesy of Google ARCore



HEADSET AUGMENTED REALITY

HEADSET AUGMENTED REALITY DESIGN IS EXCITING AND RAPIDLY EVOLVING.

Using Augmented Reality (AR) and Mixed Reality (MR) in a head mounted display developers can use spaces and physical surfaces like streets, statues, walls, tables, and ceilings as a creative canvas. in MR developers can digitally recreate physical spaces and surfaces like the ones mentioned above creating a real time realistic yet digital canvas. Developers can place virtual content and objects on these surfaces and make them persist like real objects. Virtual objects can also be made to interact with the environment. For example, developers can have users climb a sofa, jump off a stair, or hide behind a chair.

The advances in AR and MR are driving solutions for new and exciting entertainment and gaming innovations. But it also drives solutions that are solving everyday problems in precision agriculture, retail sales, surgical innovations, manufacturing factory management and more. The ability to model new concepts with digital assets in real spaces can be a game changer for many industries.

While this is simultaneously an exciting new opportunity for designers, it presents a design challenge because developers have no control over the user's environment. When designing an experience that uses a particular type of surface, make sure there are alternative options. For example, the environment might not have a sofa or table, or may not have enough floor space. Design alternative placement options to accommodate this kind of dynamically changing environmental situation.

As this technology requires that digital objects be added to real spaces much of the guidance provided in the section on AR Devices applies here as well. The guidelines and concepts for MR application development outlined below should be considered to be additive to prior guidance as well as preliminary and evolving.

HEADSET AUGMENTED REALITY

SCENE PERCEPTION

The scene perception is the system's ability to perceive environmental objects like floor, ceiling, wall, desk, couch, door, window etc. The MR content which includes scene elements can recognize those scene objects and interact with them to immerse in the real world. Scene elements are sometimes defined as Plane 2D, Anchor, Scene mesh, and Object 3D.

- **Plane 2D:** support vertical/horizon plane and plane mesh.
- **Spatial Anchor:** A coordinate marker in virtual space. Anchoring and tracking virtual objects to real world positions and orientations.
- **Scene mesh:** support visual mesh and collider mesh for room.
- **Object 3D:** object in cube;

Refer to "Surfaces (pg.36)" in the AR section of this guide for more information about guiding users to scan scenes as environments for applications.

ALIGNMENT OF VIRTUAL OBJECT AND PASSTHROUGH

MR may use the cameras and sensors of the devices to reconstruct a single high-resolution color camera image into a stereo passthrough frame for MR rendering. By doing so, the perspective of passthrough should match the user's field of view. With the correct perspective, the virtual objects will be placed in the right position and make the user feel more naturally in mixed reality.

Misalignment

Passthrough (reconstruction with depth data) is mainly based on visual clues, supplemented by depth sensors. In some conditions, the virtual objects might not align with the environment properly, such as places at a distance, places that are too close to the cameras, or a messy area that is being scanned. These misalignments might cause the user to feel uncomfortable and impact the MR experience.

For better MR experience, developers should place or interact virtual objects in the following conditions:

Floor.

Furniture, such as the user's table or chair, is an object with lots of depth clues.

Drawing or books, easy to place and track.

User-defined plane or objects.

To avoid misalignment, developers should not place or interact virtual objects in the following condition:

Place in less than 0.5 meter or farther than 3 meters from viewing position.

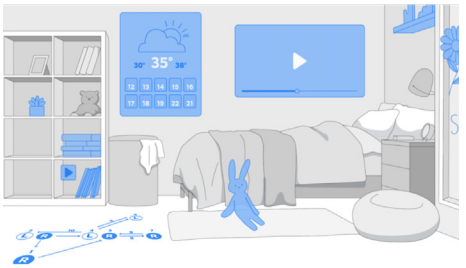
Messy areas without any empty space, the depth data will also be messed up.

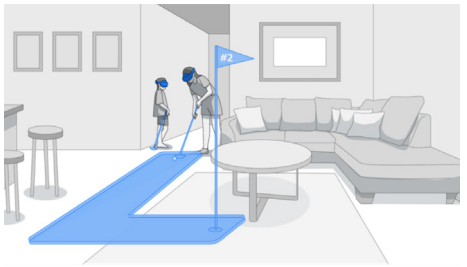
A plane lacking any visual clues, such as white wall without anything hanging on it.

Spatial Anchors

A Spatial Anchor serves as a bridge between virtual and augmented reality applications, allowing for consistent object placement and tracking across different sessions and devices.

- Spatial anchor represents an indicated location of a tracking map within the space which the user locates.





Images courtesy of Meta

- Spatial anchors are created and destroyed by the application according to content design.
- The spatial anchor is destroyed when the application session is terminated or the tracking map is relocated.
- The spatial anchor can be stored as cached anchor or persisted anchor with a specified anchor name for reuse across application sessions.

HEADSET AUGMENTED REALITY

Object placement

When placing virtual information, or content in a space, place it at a comfortable distance and location. In MR experiences poor object placement may cause a user to have a quick reaction to step back or jerk to avoid an object. For example, if an object appears too close to what a user perceives as their physical body. In general, it is recommended to place objects at a roughly 1 meter distance slightly below the user's line of sight. In addition, the following is recommended:

- When launching an app, avoid objects that suddenly appear too close or large to the user. Also, provide visual or audio cues to the user when placing objects on surfaces in real space (e.g., tables, walls, ceilings, etc.)
- If an object's size can be visually overwhelming to the user, consider gradually revealing the object with fade-in animation.
- When placing an object on a table, wall, or other surface, make sure the object is aligned with the surface so that it feels realistic.

Design with user safety in mind

Even though the user can see their real space, they can be distracted by virtual objects, which in turn can cause injury. For example, a virtual object may collide with a real physical lamp or vase, obscuring it. Or, virtual content may be misplaced behind a wall, causing the users hand or body to collide with the wall when trying to interact with it. Accordingly, it is crucial to design MR experiences to accurately and safely blend virtual content with the user's real world.

PROTOTYPING

Even if something on a 2D screen is carefully designed is often different than what was anticipated. Accordingly, it's important to experience creations on a head mounted device, such as Quest 3 or HTC Vive Pro, as soon as possible. Test and iterate frequently to avoid surprises in behavior.

By placing 2D images or 3D dummy placeholder objects (e.g., like a cube in the physical space), the developers can experience and gather great insights on many design elements, including optimal size, distance, placement, and overall spatial layout.

CONCULSION

It is our hope that this guide serves to deepen developers to better understand some of the fundamental design principles to be considered when starting to build immersive experiences at all levels.

The XRA member companies have collectively learned a great deal about XR design over the last several years, but it remains early in the overall evolution of the platform, and we are still learning new best practices at a rapid pace. All of our members publish best practice guides for developers on their platforms. Developers should rely on their guidance.

An essential part of making these guidelines better and more comprehensive over time is to hear from you, the community of XR developers and designers. If you have discovered something new, from an effective design pattern to a common pitfall, we would love to hear from you and to incorporate what you have learned into these guidelines for the broader industry.

You can reach the XRA on these issues at **info@xra.org**.

Together we can help establish this new industry, and pull the future into the present.

REFERENCES

XRA Developers Guides: An Industry Wide Collaboration for Better XR, https://xra.org/developers-guide/

XR Accessibility Github, https://github.com/XRAccessibility/xraccessibility.github.io

HTC Vive Software Developers Kit, https://developer.vive.com/resources/downloads/

Meta Horizon Best Practices, https://developers.meta.com/horizon/resources/

Magic Leap Developer Docs Best Practices, https://developer-docs.magicleap.cloud/docs/category/best-practices/

Google ARCore Design Guidelines, https://developers.google.com/ar/design

Apple VisionIOS Best Practices, https://developer.apple.com/design/human-interface-guidelines/designing-for-visionos#Best-practices

ISO/IEC JTC 1/SC 35 -User Interfaces, https://www.iso.org/committee/45382.html

Note: *This Resource list is not intended as an endorsement nor exhaustive list of all the development resources available to Immersive experience developers.*



XR ASSOCIATION

www.xra.org

DECEMBER 2024