



Omni Application Design Guide

How to Design and Optimize an Application for the Omni

Omni Application Design Goals

1. **Comfortable:** Motion sickness is a big issue in VR. The Omni helps reduce motion sickness from locomotion, but many other elements in your game or application can cause discomfort in players. It is important that you remove any items that cause discomfort.
2. **Accessible for all Players:** Most players will not have experienced VR, and many would not be considered “gamers”. Your game or application should be easy for inexperienced players to understand during their first play.

This guide will provide tips on how to address both issues in your game or application.

Movement

Physical movement is a unique feature for Omni games, so it must work well and be comfortable for players.

Movement Speed

It is important to have the movement speed set correctly in your game to ensure that players have a good experience on the Omni. A high movement speed will cause discomfort.

We suggest starting with a max player movement speed of 7 m/s, and fine-tune the value through testing.

Disable Other Movement Modes (Teleportation / Touchpad)

Since Omni Movement is a key feature of Omni games, all other movement modes (such as teleportation and touchpad movement) should be disabled in your game.

Turning

The ability to turn and walk in any direction is an important feature for the Omni. However, we have found that frequent and drastic turns can cause discomfort for players.

We suggest giving the player the ability to ease into turns. Instead of making a quick 90° turn, split the turn into longer, gradual turns.

We also suggest reducing the number of turns that the player must perform throughout the game. If players are turning frequently, then they are spinning in circles in the Omni. This can make the player dizzy or motion sick.



Collision and Elevation

Any quick change in elevation that moves the camera up or down has the potential to cause discomfort for the player. We've noticed issues in games where collision is enabled on small, decorative objects placed on the ground (small rocks, debris, foliage). The sudden camera movement caused by walking over these small objects can quickly lead to player discomfort. We recommend you remove player collision from these objects in your game to keep the player's walking experience as smooth and comfortable as possible.

Similarly, if there are stairs in your game, use a ramp instead of stepped stair collision.

Falling can also be an uncomfortable experience for some in VR, especially if it is abrupt. We find it helpful if gravity is reduced to be lower than realistic gravity (i.e. 50% of real world gravity) so that when the player is falling, they "glide" to the ground and don't become disoriented while falling. Even with lower gravity, this can still cause discomfort if it is done frequently.

Collision on walls should also be smooth. Objects that protrude from or are placed next to a wall need to have simple collision that prevents the player from getting stuck. In most cases, it's best to treat collision for walls as if they were flat walls.

Vive Controller Interactions

Interactions with the Vive controllers can be effective in VR games, however there are some limitations that need to be considered when creating Vive controller interactions for an Omni game.

The Omni Ring

The Omni Ring is very important for safety in the Omni as well as for movement tracking. It does, however, come with some physical limitations.

The Omni Ring limits the player's ability to reach the ground with their hands. This means that there should be no instances in the game where the player must pick up an object from the ground. They also will not be able to reach out far from the center of the Omni.

If you have any objects that the player needs to interact with, place them at chest height and let the player get close so that they don't have to reach far. We also suggest increasing the interaction radius of the objects, or creating a system where the player can interact with objects through pointing. This could allow the player to point the controller at an object on the ground and press a button to pull that object to their hand.

Sometimes the Omni Ring can get in the way of certain objects. Do not create interactions where the player will be swinging their arms by their hips or any other interactions that may cause the player to forcefully hit the ring.



Player Direction / Objective

Many Omni players will be inexperienced with video games and with VR. Ensure that the player can always figure out what their objective is and where they need to go. The player should never be required to remove the VR headset and ask for help from an operator.

Clear Player Objective

Present the player's objective in a clear and concise statement so that the player knows what they are trying to do, and why they are trying to do it.

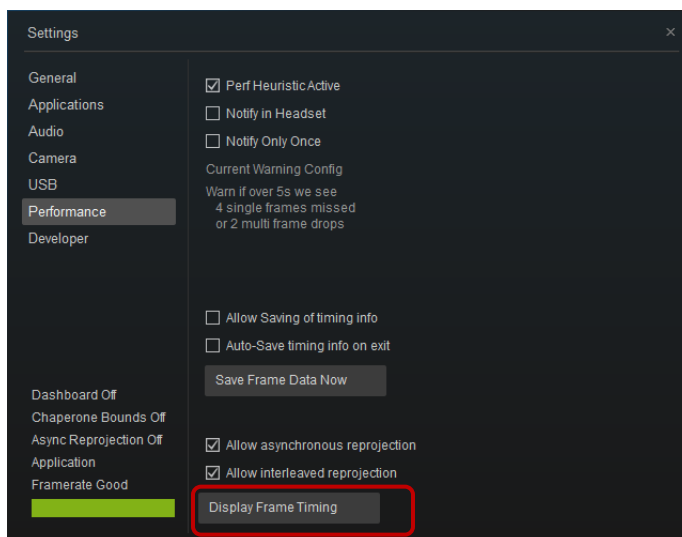
Player Direction / Navigation

Many players who are inexperienced playing video games have a very difficult time understanding where to go. Make it extremely obvious where the player needs to go, for example by using large arrows or lighting effects.

Simulation Sickness

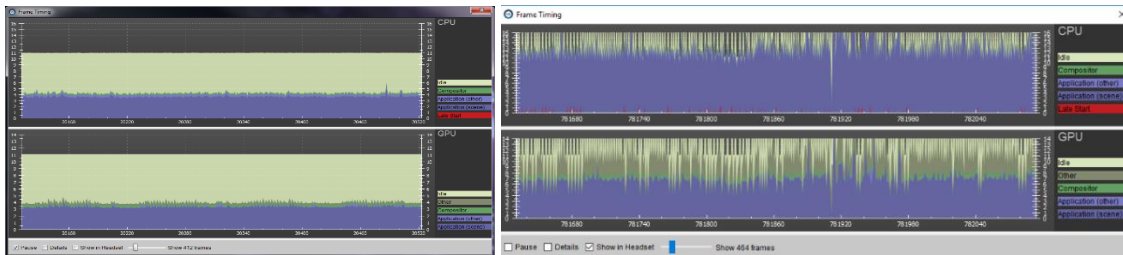
Simulation Sickness is a form of motion sickness that occurs when using HMD devices in a VR world. Simulation Sickness can greatly affect a user's VR experience and in some cases, ruin the VR experience altogether. To help reduce the likelihood of your user having a bad VR experience, follow the best practices listed below closely. If you do not do this, your users could end up having a very unpleasant VR experience.

- You must maintain a framerate of 90 fps, and ideally with extra buffer to make sure you're always exceeding the HMD's native framerate. Low framerate is the biggest trigger for Simulation Sickness, so make sure to optimize your game as much as possible. To test framerate in your application, use the SteamVR "Frame Timing Window". To view the "Frame Timing Window", go to SteamVR > Settings. Under the performance tab, click "Display Frame Timing".





The “Frame Timing Window” shows you the time needed by SteamVR to render your application, both on the CPU and the GPU. If either the CPU or GPU spike above 11 ms, then your application is not maintaining 90 frames per second. The image below on the left shows the application running consistently under 11 ms and maintaining 90 fps. The image on the right shows the application spiking above 11 ms for many frames and not running at a consistent 90 fps.



- Avoid cinematic cameras, or anything that takes control of camera movements away from the player, as this tends to be a big cause of users getting uncomfortable.
- Do Not use a "Walking Bob" camera effect (as is typical in most first-person games). Causing the camera to move up and down to mimic the movement of the human body will give player Simulation Sickness, and ruin their VR experience.
- Do Not "Shake" the camera when trying to relay an event to the player. If a grenade goes off next to a player, a camera shake may make sense in non-VR games, but in VR games, it can trigger Simulation Sickness easily.
- When you are designing worlds and levels for your VR game, make sure to use dimmer lights and colors than you normally would. Strong and vibrant lighting in VR games can cause Simulation Sickness to occur more quickly. Avoid this by using cooler shades and dimmer lights than you normally would.
- Avoid stairs and use lifts instead. When the player is moving quickly, especially up and down stairs, it can be very disorienting.
- Do Not use Depth of Field or Motion Blur post processes, because they can greatly affect what the user is seeing and, more importantly, they can result in Simulation Sickness.
- Do Not override the Field of View (FOV) manually, and Do Not expose this setting to the end user for editing purposes. The value needs to match the physical geometry of the headset and lenses, which should be automatically set through the device's SDK and internal configuration. If there is a mismatch, the VR world will appear to warp when you turn your head, leading to discomfort or nausea.
- Developers make the worst test subjects, because they are typically already used to VR and immune to Simulation Sickness. Test your game as much as you can, with as many different people as you can, to make sure that you are not causing Simulation Sickness.

Please note that this list is not a complete list of root causes for Simulation Sickness; however, it should give you a good idea of what might be causing your players to fall ill while playing your VR game.