Review Tracker

Reviewer	Due Date	Status	Contact
Initial draft shared with marketing		☑ ~~Done~~	Frederic Plourde
Caster PR initial review		☑ ~~Done~~	Alex Crabb
Khronos marketing initial review		☑ ~~Done~~	Tim Lewis
OpenXR WG Review		☑ ~~Done~~	
Neil review and sign off		☑ ~~Done~~	Neil Trevett
WG / WG Chair final sign off		☐ Done	Alfredo
Author final sign off		☑ ~~Done~~	Fred
Marketing final checks		☑ ~~Done~~	Tim Lewis, Alex Crabb
Post private draft to website		☐ Done	James Riordon
Make public on website and share on social media		☐ Done	James Riordon / Jeff Phillips

Link:
https://www.khronos.org/blog/stepping-up-the-floor-is-yours-with-promotion-of-the-xr-ext-local-floor-extension-to-openxr-1.1-core
Short Link: https://khr.io/138

Stepping Up: The Floor Is Yours with promotion of the XR_EXT_local_floor extension to OpenXR 1.1 Core

By Frédéric Plourde, Christoph Haag and Rylie Pavlik, Collabora

The `XR_EXT_local_floor` extension recently made its way into the core specification with the release of OpenXR 1.1. In this blog post, we will delve into the technical aspects of the `LOCAL_FLOOR` [1] reference space. While `STAGE` space is still available to developers for defining playspace bounds, we will show how `LOCAL_FLOOR` offers a convenient alternative for obtaining a recenterable floor space that does not require user calibration. Additionally, we will explore how this extension includes an estimated floor height, adding further convenience to XR development workflows.

Initial Problem

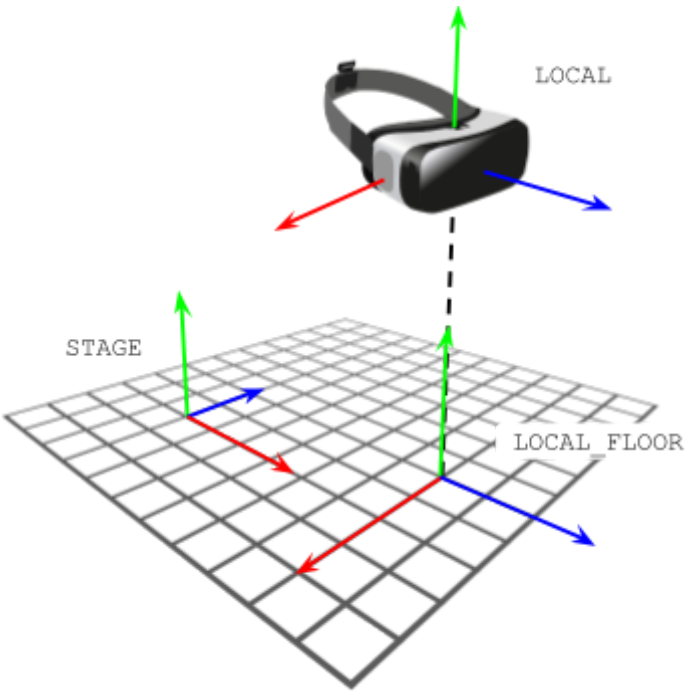
Originally, before the `XR_EXT_local_floor` extension, OpenXR defined two main reference spaces to be used by applications, `STAGE` space and `LOCAL` space. `STAGE` space is typically set up and calibrated by the user once corresponding to their real world room and then typically stays fixed until the user manually starts a new calibration procedure. `LOCAL` space on the other hand is dynamically placed by the OpenXR runtime with no relation to the real world and has no specific relation to the floor or floor height. Typically `LOCAL` space can also be "recentered", meaning that the user's position and forward direction in the XR world gets aligned to the user's current position and forward direction in their physical space. With the rise of standalone headsets, users often want to put on the headset where they stand and start playing without starting a calibration

procedure for stage space, but still have a virtual floor at a height that matches their actual physical floor.

How did `XR_EXT_local_floor` solve the problem ?

This is where `LOCAL_FLOOR` space was born: A space that can typically be recentered to the current position at the press of a button without a whole calibration procedure, and has an estimated floor height built in. `LOCAL_FLOOR` is also required to be available on all runtimes, including those that do not provide a `STAGE` space.

`LOCAL_FLOOR` space is not a complete novelty: before the XR_`EXT_local_floor` extension, developers could already locate `STAGE` space relative to `LOCAL` space and then use `LOCAL` space, but at stage space height. `LOCAL_FLOOR` space eliminates that one step and so, as an explicitly supported space now, it is more likely that developers make use of this functionality.



What are the immediate consequences for developers ?

In OpenXR 1.1, applications do not need to enable the `XR_EXT_local_floor` extension anymore to use the `XR_REFERENCE_SPACE_TYPE_LOCAL_FLOOR` reference space. Runtimes may still make the extension available, but because the `XR_REFERENCE_SPACE_TYPE_LOCAL_FLOOR` value in OpenXR 1.1 is aliased to the `XR_REFERENCE_SPACE_TYPE_LOCAL_FLOOR_EXT` value in the `XR_EXT_local_floor` extension, enabling the extension in OpenXR 1.1 is redundant.

As a developer, be aware that `LOCAL_FLOOR's` horizontal coordinates (X and Z) and orientation always match `LOCAL` space ones. If the runtime you are targeting provides a

STAGE space, and `LOCAL_FLOOR` location is within that `STAGE`'s horizontal bounds, then `LOCAL_FLOOR`'s Y coordinate will be set to 0. However, if there's no `STAGE` (or if that space is not currently tracked), your `LOCAL_FLOOR`'s Y axis origin will get set by the OpenXR runtime using its best estimate of the floor level under the origin of the `LOCAL` space.

In addition, even if the user has not calibrated the `STAGE` space, the `LOCAL_FLOOR` space will still have the best estimate of the floor height that the OpenXR runtime can determine. `LOCAL_FLOOR` space defines clear conditions when its height (Y coordinate) must match `STAGE`'s height.

That's nice, but what happens when `LOCAL` space changes? The `LOCAL_FLOOR` space will then change accordingly, and the runtime will emit an `XrEventDataReferenceSpaceChangePending` event. `xrLocateSpace`(s) or `xrLocateViews` will only take the updated `LOCAL_FLOOR` space location into account when using an `XrTime` greater than the `changedTime` parameter returned in the event.

Finally, while OpenXR runtimes will make a best effort to estimate a proper floor level that's coherent with the user's physical floor, the `LOCAL_FLOOR` space does not have any requirements about the floor level being walkable, or being an established play area. In some environments, it might just be a static offset from the `LOCAL` space: the OpenXR 1.1 specification requires this space to be provided on all systems, and it is not requiring environment-understanding capabilities to do so. Applications that want to support runtimes both with and without an established floor are still encouraged to use `STAGE` and `LOCAL` space respectively for two different modes of interaction.

Conclusion

All that being said, applications that need more information about floor levels can look at the `STAGE` space or use an environment-understanding extension, if available. However, developers that do not want to implement a separate interaction mode for devices without a `STAGE` space will find the `LOCAL_FLOOR` space handy with its ability to easily locate objects relative to a floor-based, recenterable space across all systems.

With the promotion of the XR_EXT_local_floor extension in OpenXR 1.1 core specification, we sincerely look forward to seeing increased adoption of the LOCAL_FLOOR space by OpenXR developers.

Read more about everything new in OpenXR 1.1 and please share your feedback via the OpenXR Discord channel, OpenXR Forums, or the GitHub Issue Tracker.

[1] *To make the text lighter, we have used the terms `STAGE`, `LOCAL` and `LOCAL_FLOOR` in place of their actual `XR_REFERENCE_SPACE_TYPE_STAGE`, `XR_REFERENCE_SPACE_TYPE_LOCAL` and `XR_REFERENCE_SPACE_TYPE_LOCAL_FLOOR` counterparts from the specification.*