

# Overlays for OpenXR

Brad Grantham, LunarG, Inc.  
[brad@lunarg.com](mailto:brad@lunarg.com)

Thanks to Pluto VR for their sponsorship of this project!

# Background

LunarG is working on Pluto VR's behalf to drive forward the development of an OpenXR overlay extension and API layer that implements the overlay extension.

This allows for the ability to render two simultaneous applications using OpenXR.

This presentation provides a brief introduction to overlays, describes the work LunarG has done to create the overlay extension, and shares a link to the GitHub repository. Links to report bugs and issues, or to request additional information are included as well.

# Introduction - Overlays

Existing XR APIs support the notion of 2D composition layers “on top” of an existing application’s content

- For example, Valve’s IVROverlay

Overlay applications can add a rich variety of content into other XR Applications

- Desktop OS windows in-world
- In-game HUD
- Virtual keyboard
- Chat (e.g. Pluto VR’s Pluto)

# EXTX\_overlay - Overlays for OpenXR

“Extensions” add new functions, enumerants, and structures to the OpenXR API

XR\_EXTX\_overlay is an experimental extension intended to prove the concept and uncover issues

We encourage feedback

- Security and access control needs refinement
- Input focus needs to be addressed
- We hope OpenXR runtimes will incorporate a future version of the extension

Incorporated into OpenXR 1.0.8 specification

- After discussion by and with input from other members of the Khronos OpenXR Working Group

# EXTX\_overlay - Early Access Implementation

Pluto VR and LunarG have released an open source implementation of an OpenXR API Layer providing the extension!

- And a test overlay app that can be run with unmodified OpenXR “hello\_xr”
- A user must explicitly load the layer to get the extension (XR\_ENABLE\_API\_LAYERS and XR\_API\_LAYER\_PATH)
- Windows Direct3D 11 is supported at first, other platforms and APIs are in the plan.

# EXTX\_overlay - Early Access Implementation

Not a full implementation of the API in this release

- Notably no input is passed to the Overlay app
- Action, ActionSet, and Haptic functions will return `RUNTIME_FAILURE`

Different Graphics APIs in Main and Overlay are not supported in this release

- Main and Overlay applications must use Direct3D 11

Only one Overlay session at this time

No stability guarantees

But this release allows experimentation with the API and public feedback!

## EXTX\_overlay - Creating an Overlay Session

Create an `XrSessionCreateInfoOverlayEXT` and chain it off the `XrSessionCreateInfo` passed into `XrCreateSession`:

```
typedef struct XrSessionCreateInfoOverlayEXT {  
    XrStructureType type;  
    const void* next;  
    XrOverlaySessionCreateFlagsEXTX createFlags;  
    uint32_t sessionLayersPlacement;  
} XrSessionCreateInfoOverlayEXT;
```

## Try the `EXTX_overlay` extension

Where to get it:

- <https://github.com/LunarG/OpenXR-OverlayLayer>
- [README.md](#) describes how to build and use the API layer and test

Help us improve the overlay extension. Report bugs or issues:

- <https://github.com/LunarG/OpenXR-OverlayLayer/issues>

We welcome feedback. Submit questions or requests:

- [overlay-extension@lunarg.com](mailto:overlay-extension@lunarg.com)