## OptitrackMonitorPointer Integration Example

*This package demonstrates how to use Optitrack to create mixed reality 2D displays that you can operate with Optitrack events.*

### Overview

The system works by marking the corners of standard computer monitors with Optitrack markers and then using those points to create flat planes that render screenspace Unity UI panels. Each set of points corresponds to both a worldspace driver plane that tracks worldspace raycasts, as well as a screenspace canvas view that can be rendered as a separate window on a monitor. The code in this package handles creating those planes as well as reflecting events from worldspace into screenspace.

### Getting Started

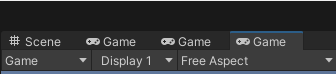
* Import the package into a new project and open **Assets/Scenes/optitrackMontitorExample**.
* Find the SceneRunner object and the **OptitrackMontiorCreator** object. This is the object that creates the planes and bootstraps the worldspace and screenspace correspondence.
  + Find the **DebugWithoutOptitrack** checkbox - this allows you to test the UI without Optitrack running.
  + Run the scene. You can run the scene right away and see that three test monitors are created. Use your mouse to drag the **optitrackedPointer** object in the scene to interact with them.

### Setting up Optitrack

* For each monitor you want to track, mark the upper left, upper right, and lower right corners with one marker each.
* Create a rigid body for each monitor in Motive. Note - as you create the rigid body, select the upper left, upper right, and lower right markers in order – the order of those points determines the orientation of the plane. If you do it in a different order, the plane will be reversed.
  + If your monitors don’t track well with just three marker points, you may add more markers. The first three markers in the rigid body are used for plane orientation, and extras will be ignored by the monitor system.
* Go back to Unity, find the **OptitrackMonitorCreator** object.
  + Open the **OptitrackMonitorNames** – These must correspond to the names of the TrackedRigidBodies in motive.
* In motive, create a rigid body for the pointer.
* In Unity, find the **optitrackedPointer** object and assign the correct rigidBodyId
* In Unity on the SceneRunner object, make sure the OptitrackStreamingClient is set to the IP for data streaming in Motive.
* Uncheck DebugWithoutOptitrack on SceneRunner to use the Optitrack data flow.

### Setting up the Game Views

Each screenspace UI window will be a separate Game view rendering UI.

* For each monitor you want to support, add a new Game view tab.
  + 
* For each Game view, set the display dropdown. The default game view is Display1, so if you have 2 monitors they should be Display2 and Display3.
* Note - the default display1 associated with the mainCamera is used for raycast event handling. It is important that your mainCamera be positioned so it can see all Optitrack monitors.
  + The symptom of NOT doing this is that the pointer will not appear at the correct location on the screen canvas.
  + Check that when you run the scene the Display1 view can see all monitors. You can adjust the FOV if the angles in your space make this difficult.
* Detach your Game views for your monitor displays and maximize them on the monitors in your lab.
* Run the scene!
* If everything is working correctly, you should see your optitrack pointer object moving, and it should cast a cursor onto the lab monitors.

### Adding new Activities

There are 4 example tasks provided. Each one is controlled by a class that inherits from **Assets/Scripts/MonitorActivity**. New tasks are registered in ActivityManager.