OKTO DESIGN TEST OVERVIEW

To simulate the video feed that would be provided by users sharing content, I opted to recreate each user in their own scene with their own background, objects, effects, and camera.

The camera for each scene was then hooked to a render texture that was used in UI canvas images which would be similar to how I would display video content from users in the actual app.

Each render texture image object existing in a masked viewport and is automatically laid out using Unity's build in UI layout scripts. These scripts not only help standardize the component objects but make the solution extensible --- adding additional images just increases the layout below the screen ad infinitum. To test this simply duplicate the objects under the "feed" object in the scene hierarchy.

One simple script was needed to run the scrolling/swiping. While Unity does have a built-in scroll window, I opted to not use that as it gives limited control to the user, especially for snapping to predetermined windows.

Instead, I set up an update function that runs in 4 stages:

- Drag started
- Dragging
- DragEnded
- Snapping

This loop defines how and when a user can interact with the scrollable element. Left exposed are variables for tweaking scroll speed, snap distances, and drag sensitivity. Like the UI setup, the script runs off of number of child objects (referred to as "pages"). Everything else --- the snap locations, beginning/end of scroll --- are calculated dynamically.

The challenges I faced were mostly in tweaking the scroll sensitivity and making sure that additional user touches during the swiping did not create too many variables as the system adjusts to UI positions. Another challenge was recreating backgrounds and UI elements from the limited direction provided. I may have taken some artistic liberty here for my own enjoyment.

Lastly, optimizations would probably be in moving the functions to an OnDrag event handler rather than in the update function, but past experience has found this to be negligible when properly gating your logic.

Let me know if you have any questions.

Cheers,

Cameron