Started by with a brief discussion to get the expert's background experience. It is primarily VRTK for VR development.

Gave a quick explanation of how it relates to VRTK ("interact" is the VRTK "use" action, etc.)

Talk occurred of making custom inspector windows

- "They're a pain to make"
- "it's rarely worth it unless it's something you really, really need to have a nice layout for with a nice explanation"
- "do it at the very end when [the feature] it is locked down"

Object attachment and connections

- The VRTK method "abstracts away" the type of attachment that is made which is more flexible for technical developers and might be better in the long for non-technical developers
- "what happens when I want a different type of attachment altogether"

Event bridge built into the object

- I have abstracted out the event bridge from within objects whereas VRTK does not
- Mine is simpler whereas the VRTK system has more capabilities (setup one for a selection of objects rather than set it up on each individual object as occurs in my toolkit)
- Expanding the event bridge
    - "Can I stop some event midway through execution"
    - "Can I pass some argument to the event bridge"
    - etc.
- "What if I wanted to define a series of steps to occur all of the objects on a grab event"
    - VRTK has a setup once for this whereas mine requires setup on each of the objects (this would be painful for a lot of objects)
    - Perhaps an "advanced event bridge is needed" or "extension to event bridge"

He pointed out a small bug in some of the example scripts setup:

- Thanks

Controller Configuration

- Very straightforward system
- VRTK action-to-button logic probably due to the developer saying we have this small number of possible actions total hence scalability is not really a concern
- Suppose the two controllers are setup to have different functions. Is there a way of saying, for example, that the first controller turned on gets this functionality and the second gets this functionality. I ask because SteamVR for a while worked where the first controller turned on was the right controller. Then it changed so that one of the controllers was always the right controller and one always the left controller.

SDK Management

- Maybe if your at the point where you need to use an SDK then perhaps a more technically driven toolkit (such as VRTK) would be better for you

Limitation of the evaluation

- No direct comparison was made between the SVRA toolkit and the VRTK toolkit
- Therefore you cannot conclude it is better than VRTK
- Instead you can say that there is the desire for such a toolkit
- And list the reasons why you didn't compare it against VRTK
- You need to address this in the evaluation
- There are two ways of looking at the system:
    - Proof of concept for extension to VRTK to simplify it
        - Custom editor windows
            - Modes - simple, medium, everything
            - How we would build it (summer project)
        - Pros and Cons again of an extension to existing toolkit opposed to development of your own toolkit
        - Suspected reason for VRTK developer stopping being that VRTK 4 would be a massive rewrite in how it works and break most of the compatibility with existing applications
    - First iteration of a new system

Suggestions

- If you important a script that requires another script to function correctly then automatically import the required script if it has not already been imported
- Switching toolkits - so you prototype in SVRA and then switch to development with VRTK and the functionality is represented across both