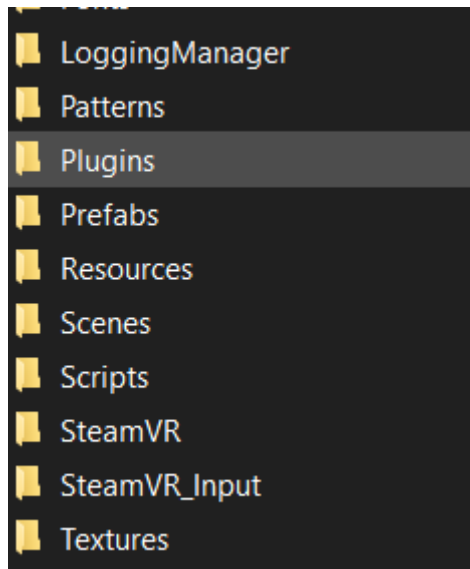
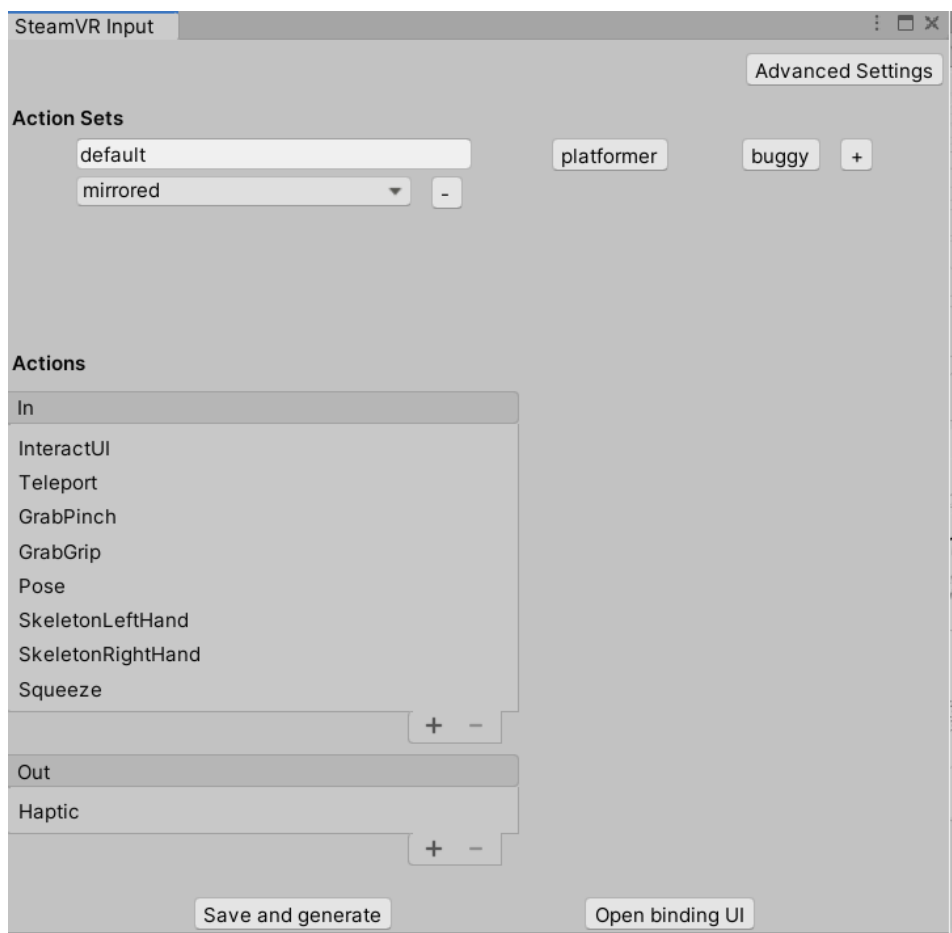


SteamVR Upgrade

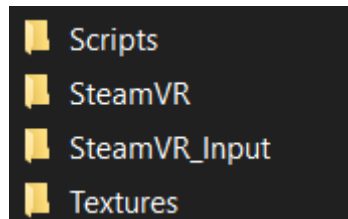


Before to upgrade SteamVR's Version, we need to delete SteamVR and SteamVR_Input folders.

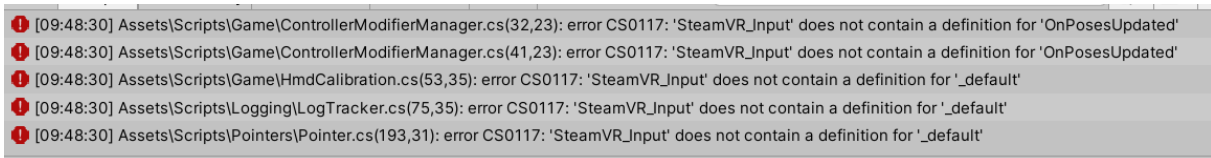
When it's done, we can import the UnityPackage of SteamVR and reload the project.



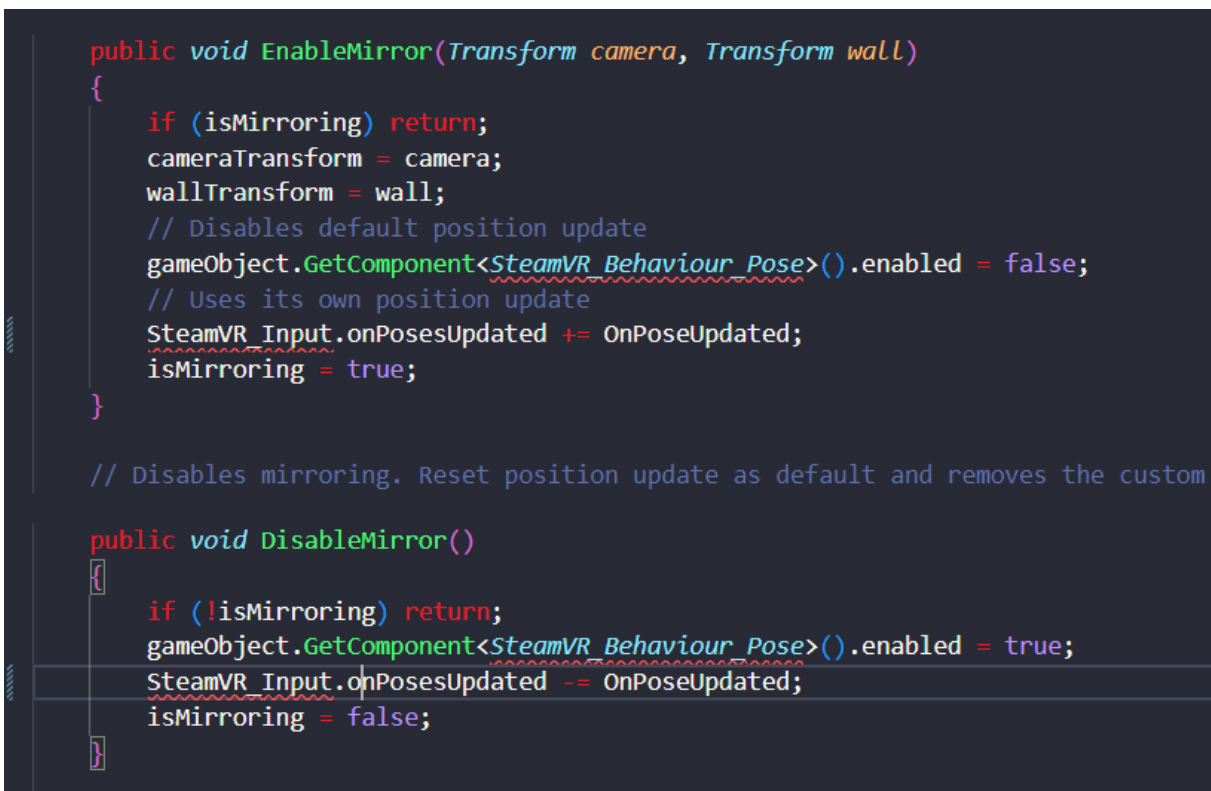
After that, we need to re-generate the input folder that contains the actions of the controllers.



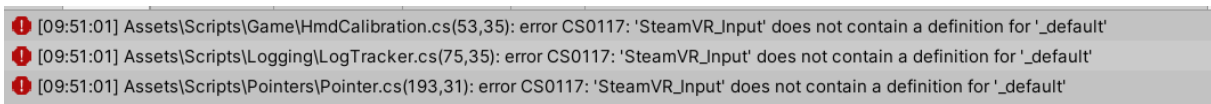
If the process is successful, the two folders that we've deleted few steps ago should be appeared at the same path.



However, we can see that we have several compiler errors. This is because variable and class names have changed.



On the first case, we have just to change in ControllerModifierManager script the variable OnPosesUpdated by onPosesUpdated.



When it's done, we have the same thing to do for three scripts.

```
113 references
public partial class SteamVR_Actions
{
```

In the last SteamVR Version, the class name was SteamVR_Input but now, it's SteamVR_Actions.

```
// Update is called once per frame
1 reference
public void PositionUpdated() {
    // HACK: for some reason we need invoke the calibrationUpdate a couple of times..
    // need to investigate why this is..
    if (time < timeout) {
        if (SteamVR.active)
        {
            if (SteamVR_Actions._default.inActions.GrabPinch.GetStateDown(controller))
            {
                CloseInstructionPanel();
            }
        }
    }
    if (calibrated) {
        time++;
    }
}
}
```

So, we have just need to change the class name in these three files.

```
[09:51:59] Assets\Scripts\Game\HmdCalibration.cs(53,46): error CS1061: 'SteamVR_Input_ActionSet_default' does not contain a definition for 'inActions' and no accessible extension method 'inActions' accepting a first argument of type 'SteamVR_Input_ActionSet_default' could be found (are you missing a using directive or an assembly reference?)
[09:51:59] Assets\Scripts\Logging\LogTracker.cs(75,35): error CS0117: 'SteamVR_Input' does not contain a definition for '_default'
[09:51:59] Assets\Scripts\Pointers\Pointer.cs(193,31): error CS0117: 'SteamVR_Input' does not contain a definition for '_default'
```

However, when we only change the class name, we have another error message that appeared. It's because in the last SteamVR Version, the SteamVr_Input class had a definition for inActions, but in the new version, it's not the case.

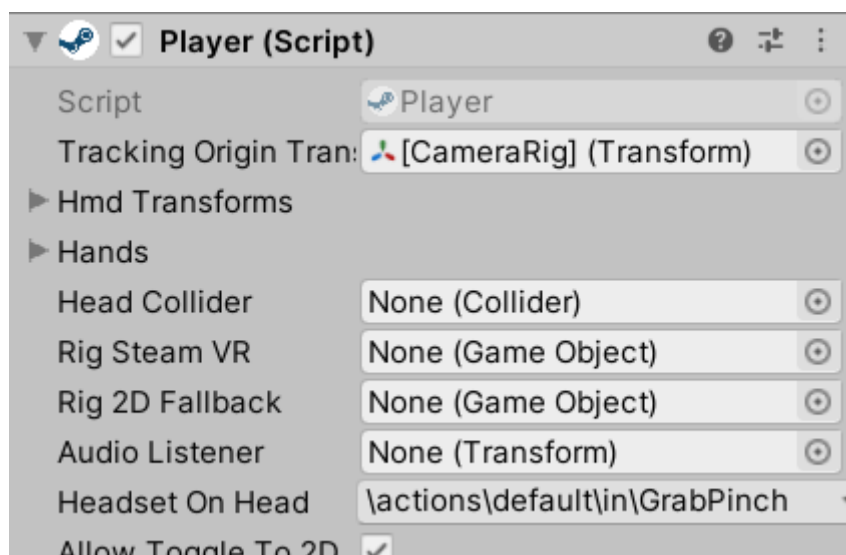
```
// Update is called once per frame
1 reference
public void PositionUpdated() {
    // HACK: for some reason we need invoke the calibrationUpdate a couple of times..
    // need to investigate why this is..
    if (time < timeout) {
        if (SteamVR.active)
        {
            if ((SteamVR.Actions._default.GrabPinch.GetStateDown(controller)))
            {
                CloseInstructionPanel();
            }
        }
        if (calibrated) {
            time++;
        }
    }
}
```

So, we just have need to delete this element in each file.



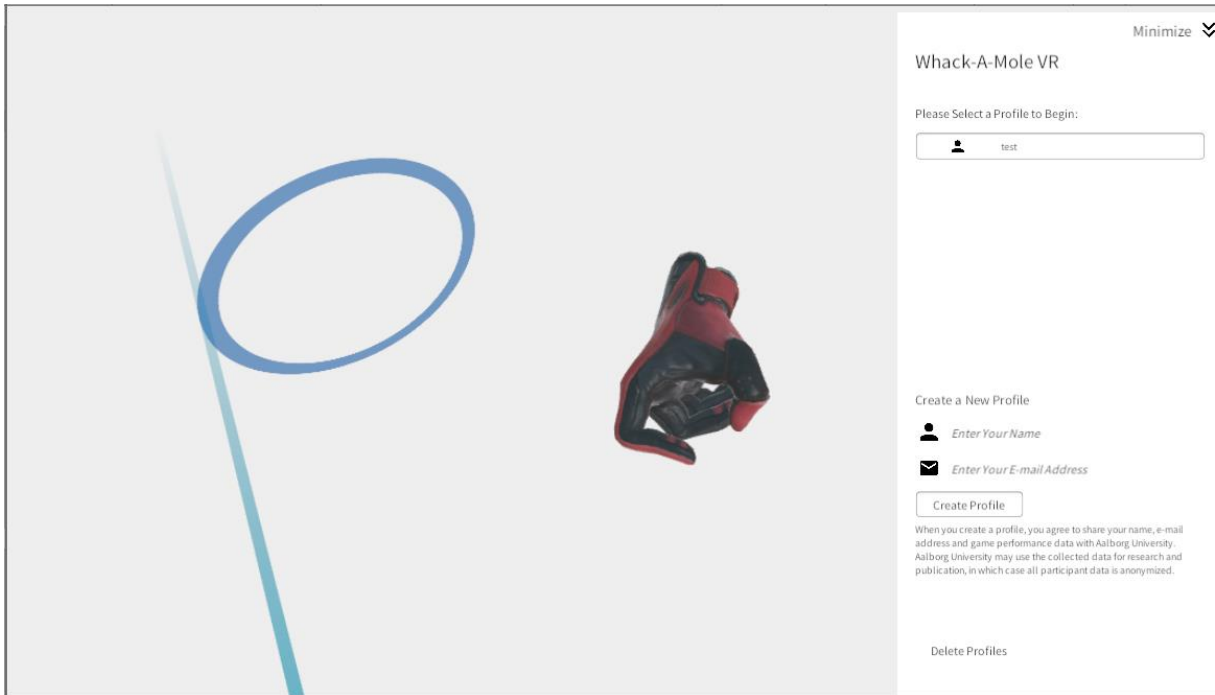
After that, all the compiler errors have disappeared.

MainScene Changes

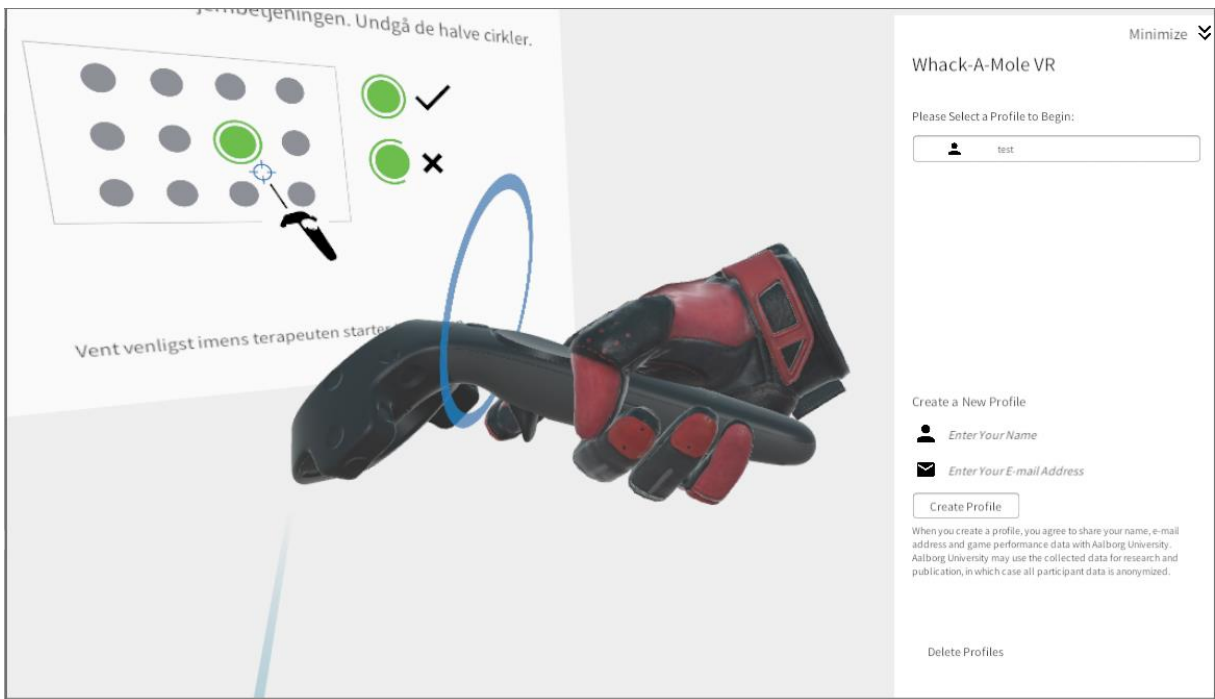


To see the controller and the hand, we need to add in the CameraRig element this script and to reference one element (CameraRig Transform) and one action (GrabPinch because we use this action in all our scripts).

Without Player Script



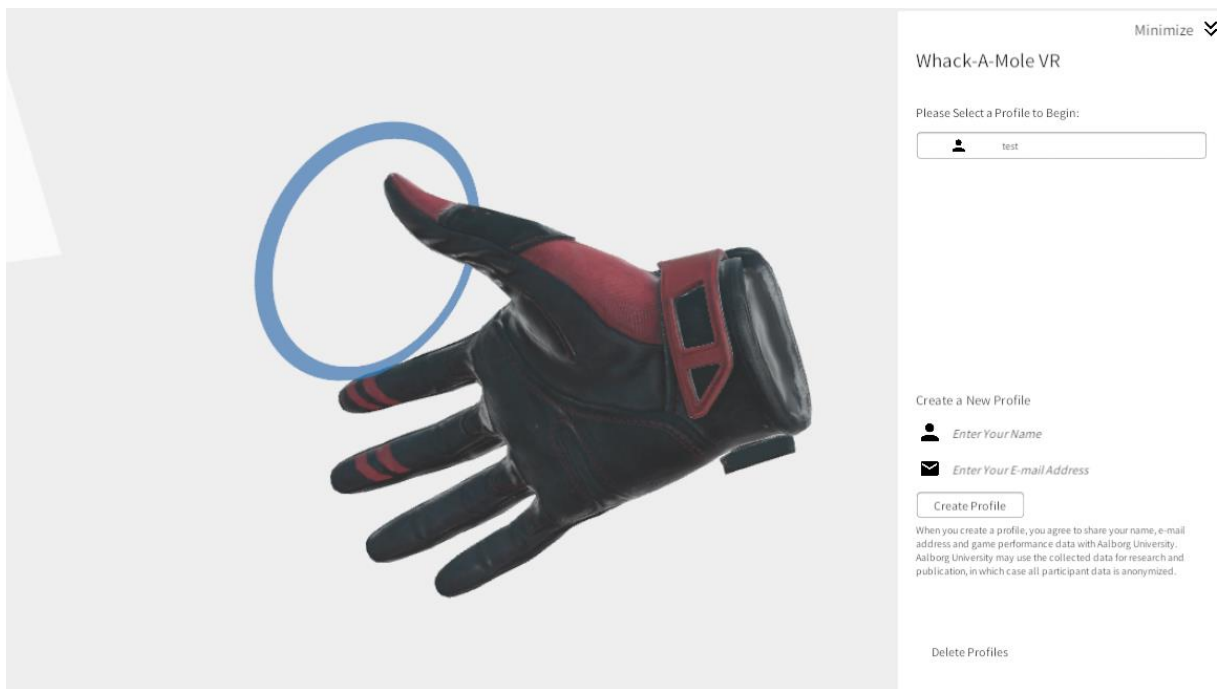
Within Player Script



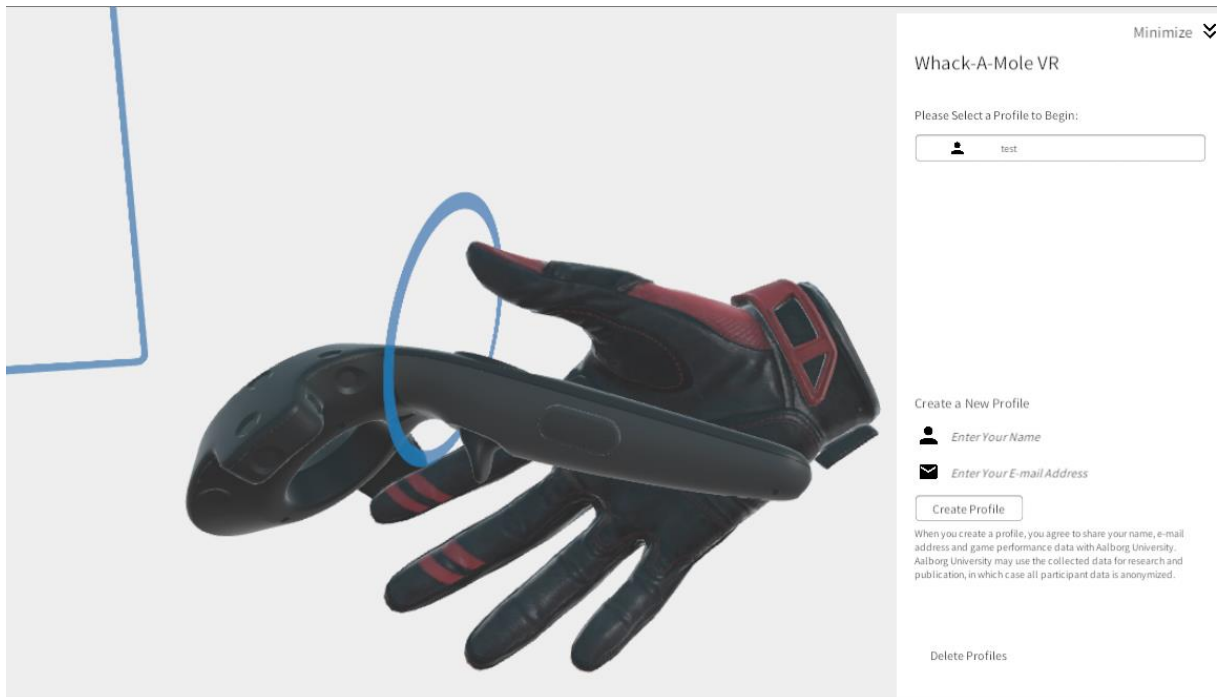
Informations

We now use SteamVR version 2.2.0 because it's the only one after our last version that don't cause problems with the hand and controller rendering and positioning. Indeed, in the other versions, the hand position and renderer are right. However, when we try to add the controller render, we can see the hand and the controller but when we click on the button behind the controller, we can see the hand movement but it go through the controller and the fingers position are not right.

When we run the game, we can see that the controller didn't appear.



So, to add it, we need to add SteamVr_Render_Model script. However, as we can see, the hand position on the controller isn't right. Indeed, the index finger isn't on the button behind the controller and the other fingers don't touch the controller.



Finally, when we press the button behind the controller, we can see that we really go through the controller.

