How to use VR Controls C# knowledge needed.

There are main scripts you will be using

1, The control object itself [ButtonControl.cs, LeaverControl.cs, KnobControl.cs]

These act as the controllers on your game objects, when added to a Game Object they perform the logic and pass to the ControlObject.cs. They all inherit from one BaseControl.cs

2, ControlObject.cs

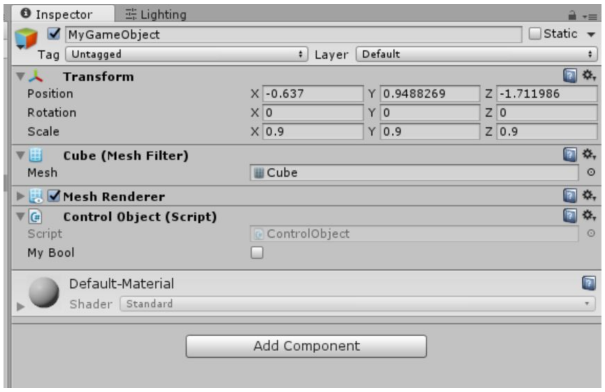
This is the component that is placed onto any gameobject in the scene you want to apply specific functions to such as ‘Toggle light, Scale Color, Pass value, Toggle Bool’

Let us make a simple function to toggle an object's active state.
In order to add a function to an object you need to update the ControlObject.cs.

```
public class ControlObject : MonoBehaviour {
    public void ToggleObject(float value)
    {
        this.gameObject.SetActive(!this.gameObject.activeSelf);
    }
}
```

When you set the Enum on the *control.cs it will call the delegate corrasponding to the ToggleObject method.

Now attach this ControlObject.cs to a game object.



☑ MyGameObject □ Static

Tag Untagged Layer Default

Position X -0.637 Y 0.9488269 Z -1.711986

Rotation X 0 Y 0 Z 0

Scale X 0.9 Y 0.9 Z 0.9

Mesh Cube

Script ControlObject

My Bool □

Default-Material

Shader Standard

Now all you have to do is add the ToggleObject method we made to the BaseControl.cs in three places.

1, Add the function to the class in order to set the Delegate.

```
        void ToggleGameObjectState(float value = 0)
        {
            foreach (ControlObject controlObject in gameObjectToInteractWith)
                controlObject.ToggleObject(value);
        }
```

2, Add the name to the Controls Enum in order to be able to set the function in the inspector.

```
    public enum Controls
    {
        NotSet, //LEAVE, this just tosses an error if it’s not set in the inspector
        ToggleGameObjectState,
    }
```

3, Finally add the case statement to set the delegate to the proper method decided by the users choice of Enum.

```
        void SetDelegate(Controls control)
        {
            switch(control)
            {
                case Controls.ToggleGameObjectState:
                    delegateMethod = ToggleGameObjectState;
                    break;
            }
        }
```