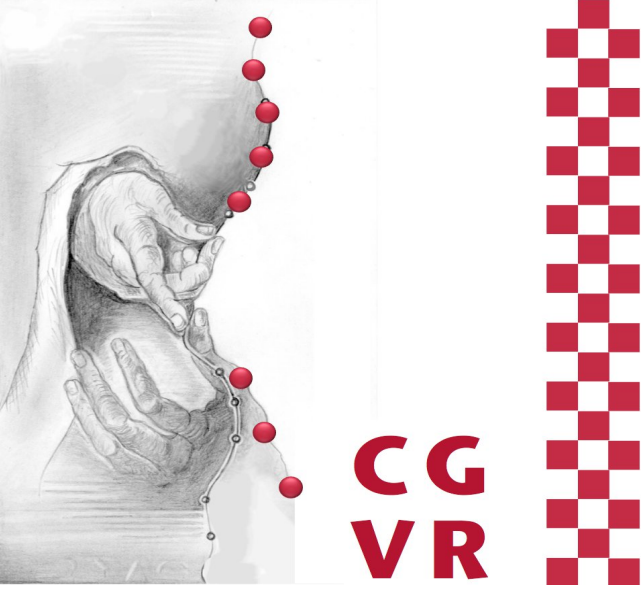




CGVR Lab

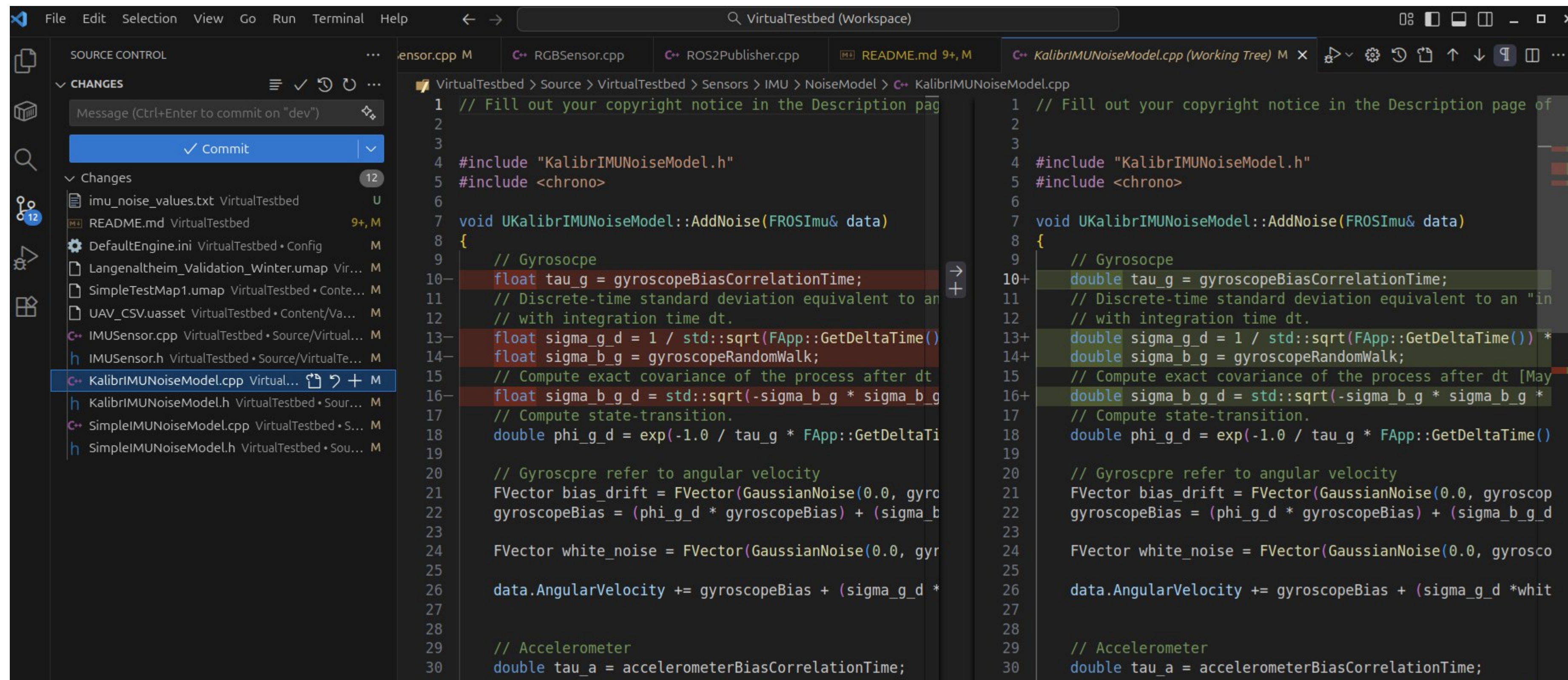
UE5 Projects w/ Teams & git



- Unreal Engine projects are also software projects, so having Version Control (VC) is a necessary tool for maintenance and integration of changes.
- Unreal uses both text files (e.g. C++-files) and binary files (`.uasset` for all Unreal assets, e.g. blueprints) which all need to be tracked.
 - Tracking changes for binary files is very difficult.
 - Resolving merge conflicts is also difficult.
 - —> Need special workflows.

VCS for Text Based Files

- Works just like on any other project, so no issues here.



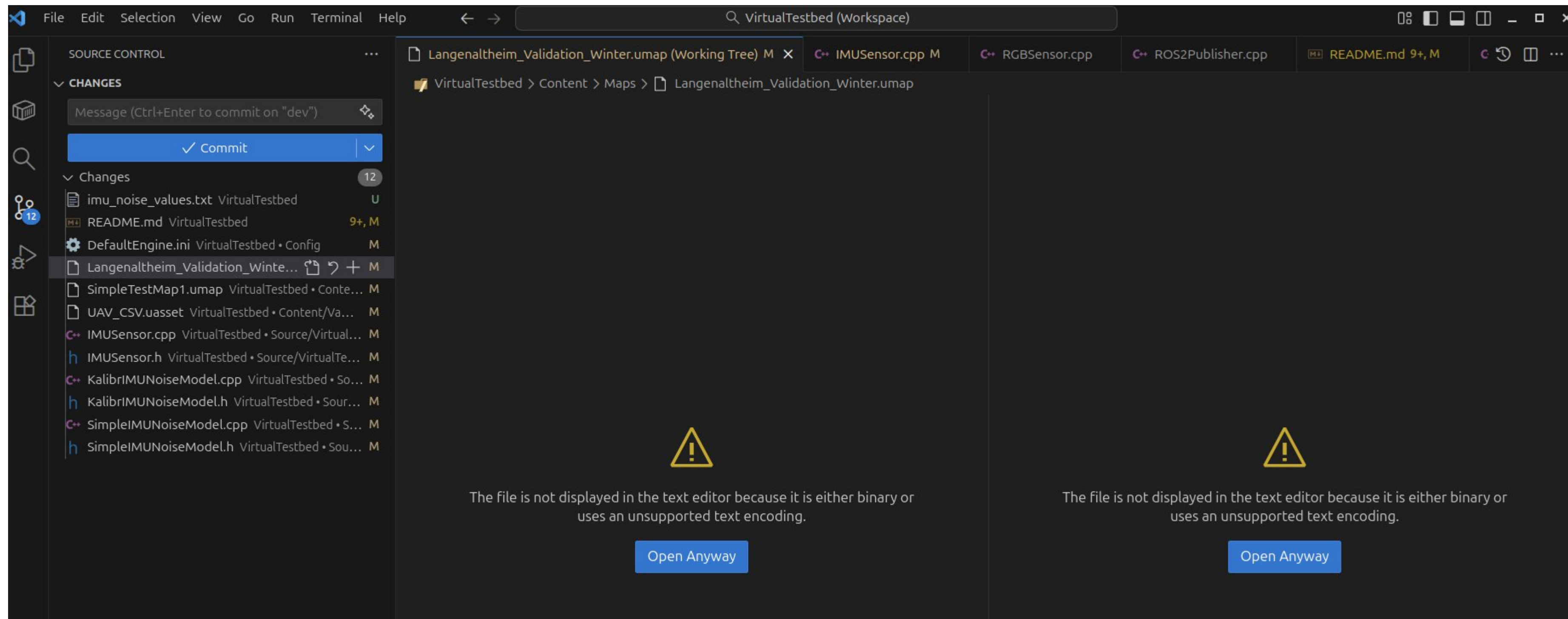
```

1 // Fill out your copyright notice in the Description page of
2
3
4 #include "KalibrIMUNoiseModel.h"
5 #include <chrono>
6
7 void UKalibrIMUNoiseModel::AddNoise(FROSimu& data)
8 {
9     // Gyroscope
10-    float tau_g = gyroscopeBiasCorrelationTime;
11    // Discrete-time standard deviation equivalent to an
12    // with integration time dt.
13-    float sigma_g_d = 1 / std::sqrt(FApp::GetDeltaTime())
14-    float sigma_b_g = gyroscopeRandomWalk;
15    // Compute exact covariance of the process after dt
16-    float sigma_b_g_d = std::sqrt(-sigma_b_g * sigma_b_g *
17    // Compute state-transition.
18    double phi_g_d = exp(-1.0 / tau_g * FApp::GetDeltaTime())
19
20    // Gyroscope refer to angular velocity
21    FVector bias_drift = FVector(GaussianNoise(0.0, gyro
22    gyroscopeBias = (phi_g_d * gyroscopeBias) + (sigma_b
23
24    FVector white_noise = FVector(GaussianNoise(0.0, gyro
25
26    data.AngularVelocity += gyroscopeBias + (sigma_g_d *
27
28
29    // Accelerometer
30    double tau_a = accelerometerBiasCorrelationTime;
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

VCS for Unreal Assets (uassets)

- Tracking detailed changes for `.uasset` files (stored as binaries) is more or less not possible with the standard version of git.
- Can only detect that a file has changed and upload the newer version.



VCS for Unreal Assets (uassets)

- Tracking detailed changes for `.uasset` files (stored as binaries) is more or less not possible with the standard version of git.
 - Can only detect that a file has changed and upload the newer version.
- Usually game developers use other VCS for tracking changes for Unreal assets, e.g. Perforce.
 - However, these options are proprietary and the free tier is usually quite limited.

Git Large File Storage (LFS)

- Since the asset files can become quite large, it is necessary to use Git LFS.
- Most public git providers (GitHub, GitLab) have a file size limit of 25-100 MB, and limits for the repository size for the free tiers (e.g. 2 GB for GitHub).
- GitLab Uni-Bremen has much higher limits for students, so no worries on that end.

Git Large File Storage (LFS)

- Installation: <https://git-lfs.com/>
 - Install regular git with its command line extension beforehand.
- Setup:
 - After installation run: `git lfs install`
 - Navigate to the folder of your project.
 - Then specify all file types to be tracked with LFS:
`git lfs track "*.<file type>"`
 - For UE5 this can for example be `.umap`, `.uasset`, `.dll`, `.lib`, `.fbx`

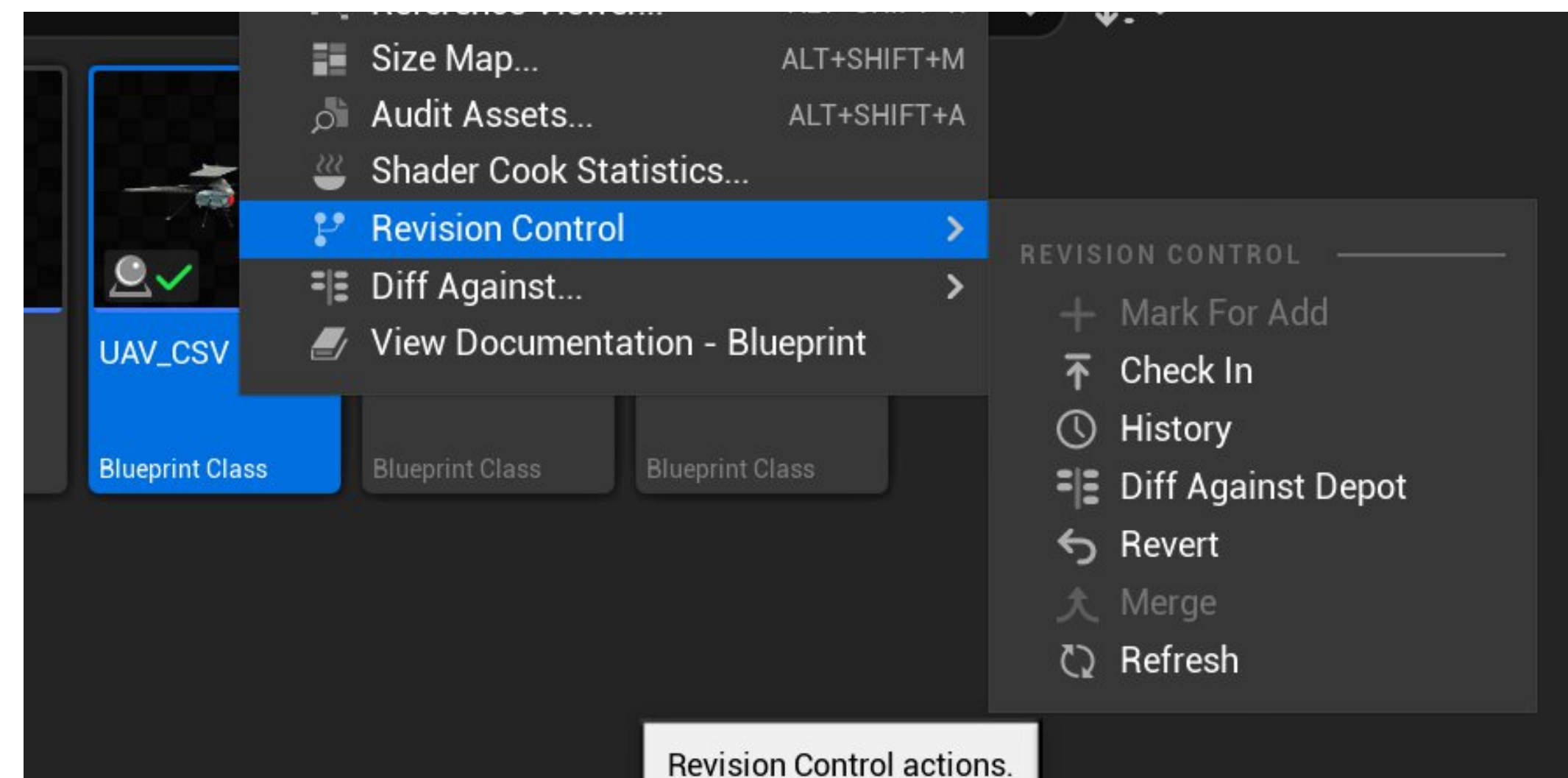
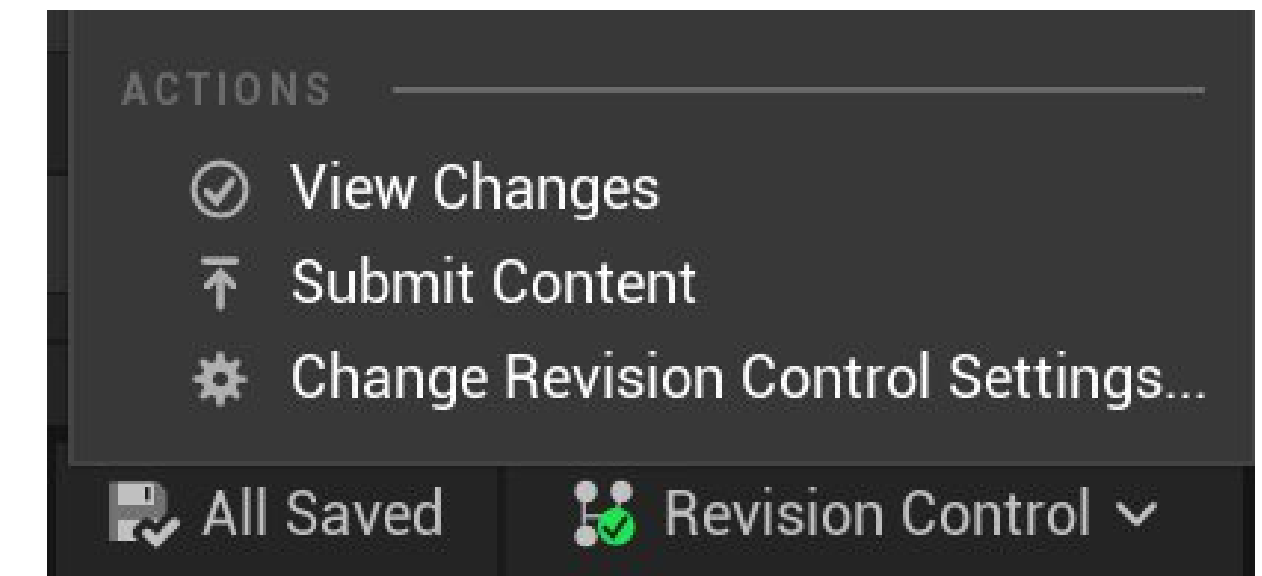
Possible Workflows... (unfortunately)

- Work on branches.
 - In case of a conflict, sometimes a simple solution can just be to copy a file to another location and then back, once the conflict has been resolved.
- Coordinate who works on what asset and perhaps at what time.
- Regularly reincorporate remote changes.
- Clone the repository twice (yes, really...)
 - In case of a conflict, integrate remote changes (can discard local changes in that repository, given they are present in the second repository) and manually re-add your local changes from the second repository.
- Or use one of the plugins on the next page.

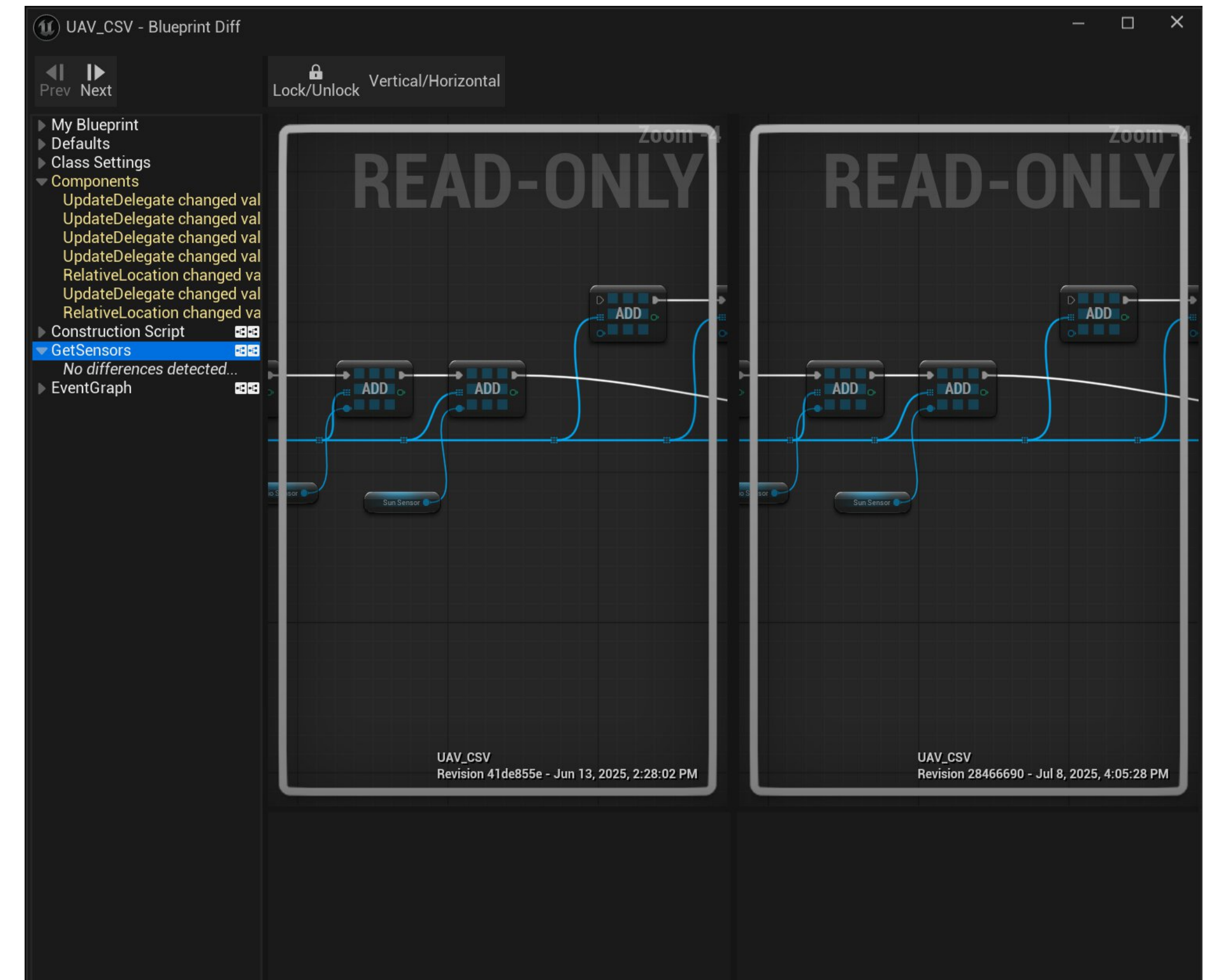
- UE5's integrated git plugin
 - Poorly documented, but should be able to do many things such as viewing changes, committing, and pulling changes.
- Community UE5 git plugin (<https://github.com/ProjectBorealis/UEGitPlugin>)
 - Supports file locking -> no two people can work on the same file simultaneously.
 - We have not tested it, yet.
- UE5's Multi-User Editing (<https://dev.epicgames.com/documentation/en-us/unreal-engine/getting-started-with-multi-user-editing-in-unreal-engine>)
 - Fairly new; Also not tested by us, yet.

UE5 Integrated Git Plugin

- Activate it on the bottom right.
- Changes can be loaded and viewed per asset:



- Other functions such as merge seem to be available as well.



Additional Resources

- Set up UE5 with git ([Youtube](#))
- How to Handle Merge Conflicts in UE5 – Diversion ([Link](#))
 - ~10 min read