

Purpose

Detail which defect database, testing and logging frameworks (and code coverage/mocking/style checking if using) will be used for the project and why

Defect Database

Github has built in issue tracking which our team will be utilizing

Testing Frameworks

We will be using NUnit to extend Unity's default framework as it is relatively simple to do and the extra functionality of measuring tests and more in depth assertions will be helpful

User Testing

Will be documented using users with a variety of physical/mental attributes that will be outline in the testing document

Manual tests will also be done by the testers frequently and may remain relatively undocumented with the exception of catching a bug

Scenario Testing

Currently no frameworks available for this that I have found

As such we will implement this ourselves by writing scripts to:

- Press random keys (ASCII only)
- Move and click the mouse randomly
- Rotate the oculus randomly
- Press the oculus controller buttons randomly
- Consider making the above scripts not as random and give them each a list of things that they can do next

We will then use that input to test the output against tests that have yet to be imagined.

Logging Framework

UberLogger is our logging framework of choice as it is open source and extends the capability of Unity's built in logging by adding separate channels which can be useful if implemented properly.

Code coverage

We will be using OpenCover as it is open source and supports branch and statement coverage, these will be enough for the testing purposes

SharpCover will be used for Namespace, class, method, line and instruction inclusions/exclusions

Conditional coverage tools don't exist that are good for C#, and We don't want to make one when we are already covering statements and branches with OpenCover

Mocking

NSubstitute as it is open source, it uses Arrange Act Assert, and is particularly readable

Style checking

Visual Studio has a built in style checker that we will be using