Build Policy

Feature Branches

To begin development on a feature, developers must create a new branch off the "development" branch in the GitHub repository. Each commit to these feature branches are expected to be free of compilation errors, and should not include changes to elements not relevant to the feature that the branch is intended for. Before merging a feature branch back to development, development must be merged into the feature branch, and all merge conflicts must be resolved on the feature branch.

Before a feature branch can be merged back into the development branch, it must adhere to the following guidelines.

- The feature is considered complete in functionality.
- There are no compilation errors/warnings present.
- Adequate unit testing has been created and passed for the corresponding elements.
- All scripts adhere to the style guide (fixed through the Visual Studio's menus).

After a merge occurs to development, Unity Cloud Build will be triggered and begin the build process.

Unity Cloud Build

Once Unity Cloud Build is triggered, the repository's contents will be downloaded to their site and the build request will be put into the queue. The build request may experience a "cooldown" before being put into the queue, if a previous build request has been done within the past hour. The build will still show up as being in the queue, if this cooldown is being experienced.

Build Process

Once the build request is sent to the builder, the unit tests are first run to ensure there is no failure in the build. If the tests fail, the build is considered a "failed build" and stops before attempting to continue the build. After completing the tests, the builder will proceed to make a build for all platforms we have chosen. If the builder experiences an error during the build process (invalid scene registered, asset cannot be found, etc.) the build is then considered a "failed build", and will stop the build process.

Failed builds

In the event that the merge of a feature into "development" causes the build to fail, the developer should "check out" the development branch and see if compilation errors are present as a result of the merge. If none are present, unit tests should be run to observe the point of failure. If there are no observable issues from the results, the Cloud Build log should be consulted.

Issue tracking

If a bug is found, it should be reported to the GitHub issue tracking. Things to be noted in the issue submission should include:

- Area the bug is present
- Observed behaviour
- Whether it is repeatable or not (Heisenbug)

Documentation

Documentation that is to be included in the repository should be added to the “master” branch, as opposed to “development”. This is to avoid unnecessary builds on the development branch, and to ensure documentation is present for deliverables.