**Instructions for version control using GitHub and SourceTree**

1. Cloning Repository
   1. Create a github account
   2. Join ‘OU-2D-GameDev’
   3. Download bitbucket and login with github account
   4. Go to remote and click clone on the repository
   5. Select the root folder (where the repository folder will be located) then add ‘/<repo name>’ to the end of the path
   6. Checkout master branch
   7. Initialize git flow
2. Create a feature
   1. Using git flow in sourcetree, use the start new feature command
   2. Name the feature and select latest development branch
   3. Push the newly created branch so that it can be tracked and others can check it out
3. Committing to a feature
   1. Make sure you are on your feature branch before making changes, there is a way to fix it if you are on the wrong branch and make changes
   2. Make your changes in unity, to the assets, or to the scripts
   3. In source tree, under workspace and file status, stage ALL files, not just the ones that you specifically created. Unity creates meta files for all assets
   4. Type a commit message then hit the commit button
   5. The push your changes so that they get sent to the github servers
4. Updating your feature from develop
   1. If a feature gets implemented into the develop branch before you are done with your feature, your feature branch will fall behind the develop branch
   2. To fix this, merge the develop branch into your feature with sourcetree by pressing merge then selecting the most recent commit to the develop branch
   3. It is likely that there will be some conflicts between the files you have edited, and the files that were edited on the develop branch. These will have to be fixed manually using a text editor then committing.
   4. Push so that the changes reflect on github
5. Finishing a feature
   1. Merge from develop to make sure that your feature is fully up to date with the current version of the game
   2. From github.com, create a pull request
   3. Others are supposed to comment and make suggestions about how to improve that feature before it gets implemented into the develop branch.
   4. Once it has been adequately reviewed, an admin will merge the branch into the develop branch
   5. After this happens, go to sourcetree and delete the feature branch from your working copy since it has already been merged and deleted from github
   6. Switch to the develop branch then pull to update your local copy with the develop branch that now has your feature implemented

Notes:

* Only admins have the power to edit the develop and master branches, any changes that need to be made to the develop branch MUST go through a pull-request which means they also REQUIRE a new feature branch
* Merge the develop branch into your feature often to make sure you don’t fall too far behind and end up with tons of conflicts when you are finished with your branch