## System Configuration – Adventures of an Adventurer

### Purpose

This document should give an overview of the used versioning techniques and why they were chosen.

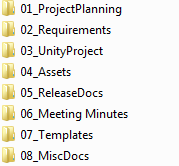
Which tools do we use?

For our project we use the service called GitHub and manage our data via the Windows/Mac application SourceTree, which provides a graphical interface for Git to get a better view on libraries etc. We tried to figure out the pros and cons of using this service and compared it to other ones, but due to its easy user interface and the crucial features it provides we decided to continue using Git in combination with Source Tree.

Which documents are under version control?

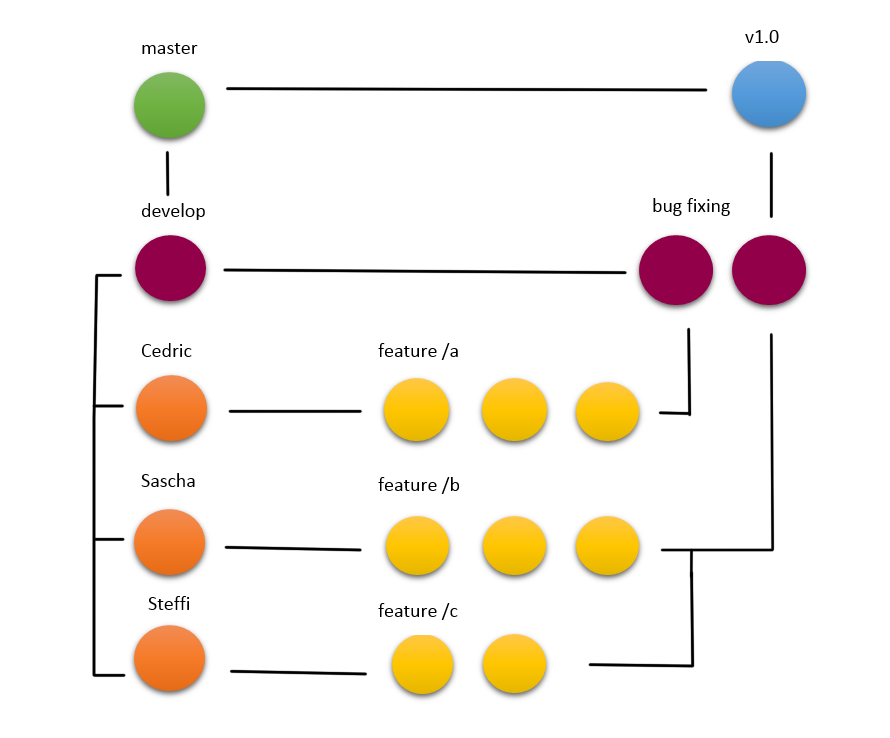
Nearly every file in our project repository is under version control. Obviously all the documents and the project files plus the assets of our project. The files which are not under version control would be all the temp files which are always generated new upon starting Unity, but never get deleted when Unity closes. There’s also the unity lock file, which just prevents important files to be removed, while the project is open.

Organization of Artifacts



The main purpose of this folder structure is to make the files of our project more accessible for new and old team members. In order to achieve this we have chosen the folder structure shown above. Some folders are obviously more important than others and some may also be never filled, but never the less we think, that this organization is a well approved structure.

|  |  |
| --- | --- |
| Name | Description |
| 01\_ProjectPlaning | This folder contains several documents like an organization Excel-Sheet which documents the work of every team member or a Bug-Log with Bugs, that need to be fixed |
| 02\_ Requirements | This folder contains documents which represent which requirements a level has to fulfill to be completed |
| 03\_UnityProject | This is the main folder of our project, it contains the project, the game objects and all the source code associated with it |
| 04\_Assets | This is the Assets fsolder which contains all Sprites and GUI which were created |
| 05\_ReleaseDocs | The containment of this folder are the changelogs for all the different versions and an License Agreement for Alpha-/Beta-Users |
| 06\_MeetingMinutes | A folder for all the Organizing stuff outside of the virtual world like for example, the team meetings. |
| 07\_Templates | The content of this folder are all the Temp-Game objects, which are used for testing purposes. |
| 08\_MiscDocs | This folder serves the purpose of collecting all the documents which don’t fit to a topic of the folders above. Like for example, all the random ideas we have for the game which need to be further elaborated or some story elements which at the time don’t belong to a level. |

Variant Management

* **Production branch** (master)
* **Development branch** (develop)
* **Feature branches** (Steffi/, Cedric/, Sascha/)
* **Release branches** (bug fix/)

Because our project is 2 years old, we began only using the “master” branch which was very chaotic. After some time, we decided to use different branches for each person, because our team only consists of three persons. We know, that this method may not be the best, but in our group this branch-management works really well.

So let’s explain how our branches work with a simple example:

* Each time a new feature is implemented, the person which is responsible for it pulls the current version from the master branch.
* When another person develops another feature this person commits his version to the branch with his/her name.
* After the person is finished he/she commits the finished feature to the development branch. This branch connects to the bug fixing branch, which is used to fix bugs before an update or a version is released.
* After all, features and bug fixes are committed, the branch connects to the master branch, which is also used for the version which is meant for public release.