We are using GitHub for our storage platform. GitHub is a widely used as versional control system, giving it many qualities that differ from simple file storage.

Management Features

Issues

Within GitHub, much of a projects conceptual organization is carried out through issues. Issues are opened in reference to the entire repository and may contain content stating a new idea that may improve the project, a requirement that must be implemented, or a bug in the code.

Tasks

Similar to Azure DevOps, GitHub allows collaborators to create checklists of items within an issue that need to be completed on a project. These tasks are both reorderable and can be checked without editing the issue. Though most of our task management will be done through Azure DevOps, this feature may still be beneficial for highly specific parts of the project. For example, a list of changes that need to be made in a particular piece code for it to function correctly could be created.

Assignees

Unlike Azure DevOps, multiple collaborators (10 max) can be assigned to an issue. This allows collaboration to be tracked and easily determined when referring to particular progress in the project.

Labels

An issue can additionally be given a label, which identifies the category it falls under. This has the benefit of making it easy to find through searching. Some of the default labels are “bug”, “documentation”, “enhancement”, and “question”. These, along with the fact that new labels can be created, is more than enough to manage and neatly organize the many issues that will arise over the course of the project.

Project Board

This feature is typically used in the same way as Azure DevOps. It allows issues to be placed into different columns, which could be “To do”, “In progress”, and “Done” depending on the way it is set up. It is unlikely that this feature will be used due to its overlap with Azure DevOps.

Usage Features

Commits

A commit is a specific iteration of the project created after changing parts of the project. A name and description can be given to a commit to allow the easy identification of its purpose and an overview of its changes. All the commits of a project may be viewed separately and may be used as backup to revert to if necessary.

Branches

The default branch within a project is named “main” and represents the current state of the project. Alternative branches of the project can be created at a specific iteration in its development. These branches can then be used to isolate commit changes and evaluate how they interact with the rest of the code. This feature is beneficial as it prevents collaborators from potentially breaking each other’s code while still allowing simultaneous progress to be made.

Pull Requests & Merging

A pull request allows the changes of alternative branches to be merged into the main branch. By being a request, other collaborators are given the opportunity to review the changes and provide feedback on them before the changes are put in place for everyone. Additionally, the request shows the differences between the branches and distinguishes between additions and subtractions.

Other Notable Features

Wiki

Documentation for the project can be provided on this separate page which accessible through the repository. Unlike inline documentation specific to pieces of the code, documentation on the wiki describes the design and structure of the project. For example, the usage of a particular set of classes could be given or a process that must be undergone to achieve a desired result (a tutorial) could be defined.

Releases

GitHub also provides a means to distribute releases of a project. Releases can be drafted with a version, title, and description. They can also target other branches besides the main branch if necessary. All the releases will be shown on the releases page of the repo. By doing this, the repository would not need to be cluttered by binaries which would also use up available space.

Comparisons

Limitations

One potential issue with GitHub is the storage limitations. For the free plan, repositories must be kept under 500 MB. GitLab, a competitor, allows up to 10GB of storage on free repositories. When considering potentially large scenes involving many assets (textures, models, sounds), it is a very real possibility that GitHub’s storage limit may be encountered. If this were to occur, the only solution would be a team or enterprise subscription providing 2GB and 50GB of storage respectively.

Distribution

Despite the convenience of having releases directly embed in the repository page, we could also use alternative sites for the sole purpose of storing releases. Any form of cloud storage platform, such as Dropbox with 2GB, OneDrive with 5GB, or Google Drive with 15GB of storage on their free plan, would all fit this use case.