Software Development Process (SDP)

Principles
Process
Roles
Tooling
Definition of Done (DoD)
Release Cycle
Environments

Principles

- We will always do our best to respond to communications on our discord server and on our Microsoft Teams chat within 24 hours of a message being sent.
- We use a github project board for our team to keep track of what tasks still need to be completed, and who is currently working on what task, so that duplicate work can be avoided.
- The backlog will always have work items ready for the next two weeks at the minimum.
- All changes need to be developed in a separate git branch, preferably with a separate git branch for each major task.
- Once the feature, bug, or refactoring is done, a Pull Request is created from the separate branch to the main branch.
- Each Pull Request has to be reviewed by at least one team member before being merged.
- The Pull Request needs to comply with the Definition of Done (see later) and should be linked to the corresponding work item.

Process

- Backlog and Planning, once per week.
- GitHub Project Board, contains Ready, Backlog, In Progress, In review, and Done sections that task cards can be dropped into.
- Weekly check-in meeting with all team members and our Project Partner (Chris Patton), once per week.
- Weekly check-in meeting with all team members and our TA (Ananya Sundararajan), once per week.

Roles

- Subgroups
    - Terrain: Benny, Ezra
    - Multiplayer: Greggory
    - UI: Stanley
    - Audio: Stanley

We will all rotate the responsibility of leading the weekly meetings with our TA, with a different team member leading each week.

This list will expand/change over time as needed.

Tooling

Version Control	GitHub
Project Management	GitHub Issues and Projects
Documentation	README
Linting and Formatting	Rust with Visual Studio Code (https://code.visualstudio.com/docs/languages/rust)
CI/CD	GitHub Actions
IDE	Visual Studio Code
Graphic Design	LucidChart
Others	Bevy Engine (https://bevyengine.org), Rust (https://www.rust-lang.org)

Definition of Done (DoD)

- Acceptance criteria are validated
- Changes are merged to the main branch
- Documentation is updated, including deployment instructions if any

- Breaking changes are evaluated/avoided
- Demo is prepared for next TA/project partner meeting

Release Cycle

- Automatically deploy to staging every merge to main branch
- Deploy to production every release
- Release every three months
- Use semantic versioning `MAJOR.minor.patch`
  - Increment the `minor` version for new features
  - Increment the `patch` version for bug fixes
  - Increment the `major` version for breaking API changes

Environments

Environment	Infrastructure	Deployment	What is it for?	Monitoring
Production	GitHub Releases and the Main Branch	Release	Fully tested and verified builds that function as intended with no major bugs (hopefully)	Can be viewed on our public GitHub page under “Code”, and in our releases tab, once we have GitHub Actions set up.
Staging (Test)	GitHub Issues, Separate GitHub Branches	Minor Changes & Patches	New unreleased features and integration tests	Can be viewed on GitHub issues, and listed in the GitHub Project Board under “In progress” or “In review”.
Dev	Local (macOS and Windows)	Commit	Development and unit tests	N/A