GitHub Collaboration Guide

1 Repository Overview

Repository name: `maze-visualizer`
Structure:

- `frontend/` – HTML, CSS, JavaScript
- `backend/` – Flask Python code
- `docs/` – Project specs, LaTeX, PDFs
- `test_cases/` – Sample input/output data

Main Branches:

- `main` – Production-ready code (Be very careful)
- `dev` – Integration branch

2 Branching Rules

Each member creates a feature branch from `dev`:

- FE: `fe/grid-ui`, `fe/animation`
- BE: `be/dfs-api`, `be/bfs-api`
- QA: `qa/test-case-1`

Do not commit directly to `main` or `dev`. Always make a pull request.

3 Commit Message Format

Format:

`<type>(<scope>): <short description>`

Example:

- `feat(fe):  build initial map editor`
- `fix(be):  correct DFS edge case bug`
- `docs:  update API format in README`

Common Types:

- `feat` – New feature
- `fix` – Bug fix

- `refactor` – Code refactoring

- `docs` – Documentation only

- `test` – Adding tests

4 Pull Request (PR) Policy

- Title format: `[FE] Add grid toggle`, `[BE] DFS engine v1`

- PR should describe what was changed, why, and how to test.

- Assign reviewer (PM or related teammate)

- Do not merge your own PR without review

5 Tips and Reminders

- This is just a guideline not meant to be followed rigorously, but I believe this really enhances our workflow efficiency

- Write meaningful commit messages

- Use `.gitignore` to avoid committing generated or temporary files

- Communicate in Discord

- As always, very wecolmed to give advices and suggestions and correct me at anytime