CSCI 2081 Introduction to Software Development - Fall 2024
Homework 03 - Following a Development Process

Figure 1 - Follow a detailed process to find and fix bugs.

Due Date: Sunday, November 10, 2024 @ 11:59pm

Instructions: This homework assignment involves following a process for creating a backlog and fixing bugs in your particle simulation. This homework also involves learning how to document a project and code.

Objectives:

- **Git** - Learn how to use source control with git..
- **Github** - Use the bug tracking and pull request features in Github.
- **Branching and Merging** - Learn how to work on several feature or bug branches to fix issues while maintaining high quality code.
- **Documentation** - Start documenting your code and project so you can release it to the world.

Getting Started: Before you begin this assignment, be sure to become familiar with the following course content:

- **Video Tutorial: Lecture 23 - Branching and Merging**
- Markdown Tutorial: Basic Syntax | Markdown Guide
- In-Class Assignment: Create a GitHub Repository

Submission: Your submission is entirely on github (code, documentation, branches, pull requests). You should make sure all tasks in this assignment are completed.

Task 1: Create a GitHub Repository (30 points)

Task: Create a github repository where you can check in your code.

1. Navigate to the organization: https://github.umn.edu/orgs/csci2081-f24

2. Create a Repository named <x500>_hw
    - Create default with a README.md
    - Install Git Locally (Download)
    - Clone your repository locally (using terminal - e.g. Git Bash)
        - `git clone <path to repo>`
        - `cd repo`
    - Copy your files from Homework 2 into the repository
    - Create a .gitignore file in the local repository. Add *.class to the text file to ignore class files.
    - Add your files (stage files)
        - `git add .`
    - Commit your changes
        - First time using git:
            - `git config --global user.email "<x500>@umn.edu"`
            - `git config --global user.name "<Name>"`
        - `git commit -m "Added Homework 2"`
    - Push your changes to GitHub
        - `git push origin main`

3. Edit your README.md using Markdown. You may format it however you like, but it should look professional and include the following:
    - **Title of your project** - You may name your project whatever you want
    - **Project Owner** - state your name
    - **Short Description** - Provide a description of your project
    - **How to use your program** - Provide how to open your project in Processing and interact with your particles

4. Pull your changes locally.
    - git pull origin main

Rubric:

- 10 points - GitHub repository exists.
- 10 points - Particle Simulator (Homework 2) is checked into the repo.
- 10 points - The README.md has the project information and is professional.

Task 2: Create a backlog (20 points)

Task: Create a backlog of issues to be fixed or worked on.

1. Go to your GitHub <x500>_hw repository stored in https://github.umn.edu/orgs/csci2081-f24

2. Create a Backlog by adding 7-10 issues with the following labels:
    - **Bug** (at least 1) - If you don't have a bug, create one (e.g. remove semicolon in your code). Log as many bugs as you know exist in your program.
    - **Documentation** (at least 1) - You will need to add at least one issue for documentation to create javadoc documentation.
        i. One of your documentation issues should be to document your code for javadoc creation.
    - **Enhancement** (at least 5) - Suggest several enhancements to your code.
        i. One of these enhancements should be a recommendation for using a design pattern

3. Be sure all of your issues are professionally documented with enough detail so that any developer could come in and implement items on the backlog.

Rubric:

- 10 points - GitHub Repository has at least 7-10 valid issues (things that could be completed)
- 5 points - Issues include at least 1 bug, 1 documentation, and 5 enhancements.
- 5 points - At least one enhancement should detail the use of a design pattern.

Task 3: Resolve issues through Branching and Merging (50 points)

Task: Close at least 3 issues by branching and merging. The following process is described below:

Do the following at least 3 times (1 for bug, 1 for documentation, and 1 for one other issue):

1. Choose an issue to work on.

2. Pull all changes locally:
    - `cd <x500>_hw`
    - `git pull origin main`

3. Create a branch **(-b below will create a new branch)** to fix the problem. Name your branch after the fix. (e.g. : bug_missing_semicolon, feature_colliding_particles, doc_javadocs):
    - `git checkout -b <branch_name>`

4. Implement the fix or change on your local branch.

5. Check in your changes:
    - `git add .`
    - `git commit -m "description of your changes."`

6. Push your branch to GitHub:
    - `git push origin <branch_name>`

7. Checkout your main branch:
    - `git checkout main`

8. On GitHub, create a **pull request** from <branch_name> to main.
    - Specify in the pull request that it fixes #<issue number>.
    - This will map the pull request to the issue.

9. Merge the pull request!

10. Pull the latest from GitHub.
    - `git pull origin main`

Rubric:

- 5 points - GitHub has at least **4 branches** named appropriately (including main).
- 5 points - GitHub has at least **3 closed pull requests** that map to **3 separate closed issues**.
- 10 points - At least one bug is fixed.
- 10 points - At least one documentation issue is fixed.
- 10 points - At least one additional issue is fixed.
- 10 points - Code is documented so that Javadocs can be created.