# [GameMash Project Manager Feedback](https://docs.google.com/forms/d/e/1FAIpQLSeqrkzqu4p2asBNFLFnEHjbgS8BvV5nXaYeAKvPGJuFWmqx0g/viewform?usp=sharing) Welcome to the Comprehensive Project Guide!

This guide will walk you through everything you need to know about our project. I've made it simple and straightforward for clarity and ease of use. Below, you'll find detailed instructions tailored to each team, starting with the **Front-End Development Team.**

To get a better understanding of our overall project, feel free to review **Lab 1** here:  
[Lab 1 - Project Overview](https://docs.google.com/document/d/1-2jSPW9vKaoZne5U3XwxUv-Gn_9d_RLnykOofRnunZg/edit?usp=sharing)  
  
And our **Lab 2** here:  
[Lab 2- Planning + Outlining](https://docs.google.com/document/d/1kRQLuVK3r_rMWjRf8dv8BCtA0T0vqHycP9C0X3W45Wc/edit?usp=sharing)  
  
And our **Lab 3** here:  
[Lab 3: Sprint 1](https://docs.google.com/document/d/1gfIojQ1D18RP4DQt9FxrJ0LyHoAeX55eyE4w-0-eyqQ/edit?usp=sharing)  
  
And our **High-Level Project Proposal** here**:**[**Group 3 High-Level Project Proposal**](https://docs.google.com/document/d/1NDO-plNh9rYwiRacIsIqpuTzWMmjXfC3cwptJcJgj9A/edit?usp=sharing)And our **Sprint 3 Demo Prep and Work Flow** here:  
[Sprint 3 Demo Prep](https://docs.google.com/document/d/1vpA5n3wCVRRPo8CL-hCR2nVOEiCsUNr_KDAer9RfPsU/edit?usp=sharing)  
  
[SPRINT 3 Work Flow.](https://docs.google.com/document/d/16YRByarxM2S-h8-L38y2xOnAeRym-R5Ofl3xU5_zTxc/edit?usp=sharing)  
  
And our **Sprint 4 Vertical Slices** here:  
[Sprint 4 Vertical Slices](https://docs.google.com/document/d/1x9rHfhEV7DETzaZrSpqSxvzmUW91C6tfKBKcgRle-uw/edit?usp=sharing)  
  
And our **Sprint 5** here:  
[Sprint 5!](https://docs.google.com/document/d/1lqibxSe6XRxs2ejufFXkMjgXzoPLaIQLJP7uMtjN-sw/edit?usp=sharing)  
  
Debriefing:  
[Debriefing](https://docs.google.com/document/d/1q_krD2lsipqgQeLVN_3fl-FYFbXGrgAa1oUktJDRrs0/edit?usp=sharing)  
Final Presentation Guide:  
[Final Presentation Guide](https://docs.google.com/document/d/1A2CApbP5TEfKrZFxo34RcoRJBuK2bh44UBJZ_xgUPqM/edit?usp=sharing)  
Trello:  
[GameMash](https://trello.com/b/THTHXQB3/gamemash)  
Jeni’s art:  
[Jeni](https://docs.google.com/document/d/1H1N1oihUdJ663G6HHBSf-cmMU_1EHpg6fG0i325x0X4/edit?usp=sharing)

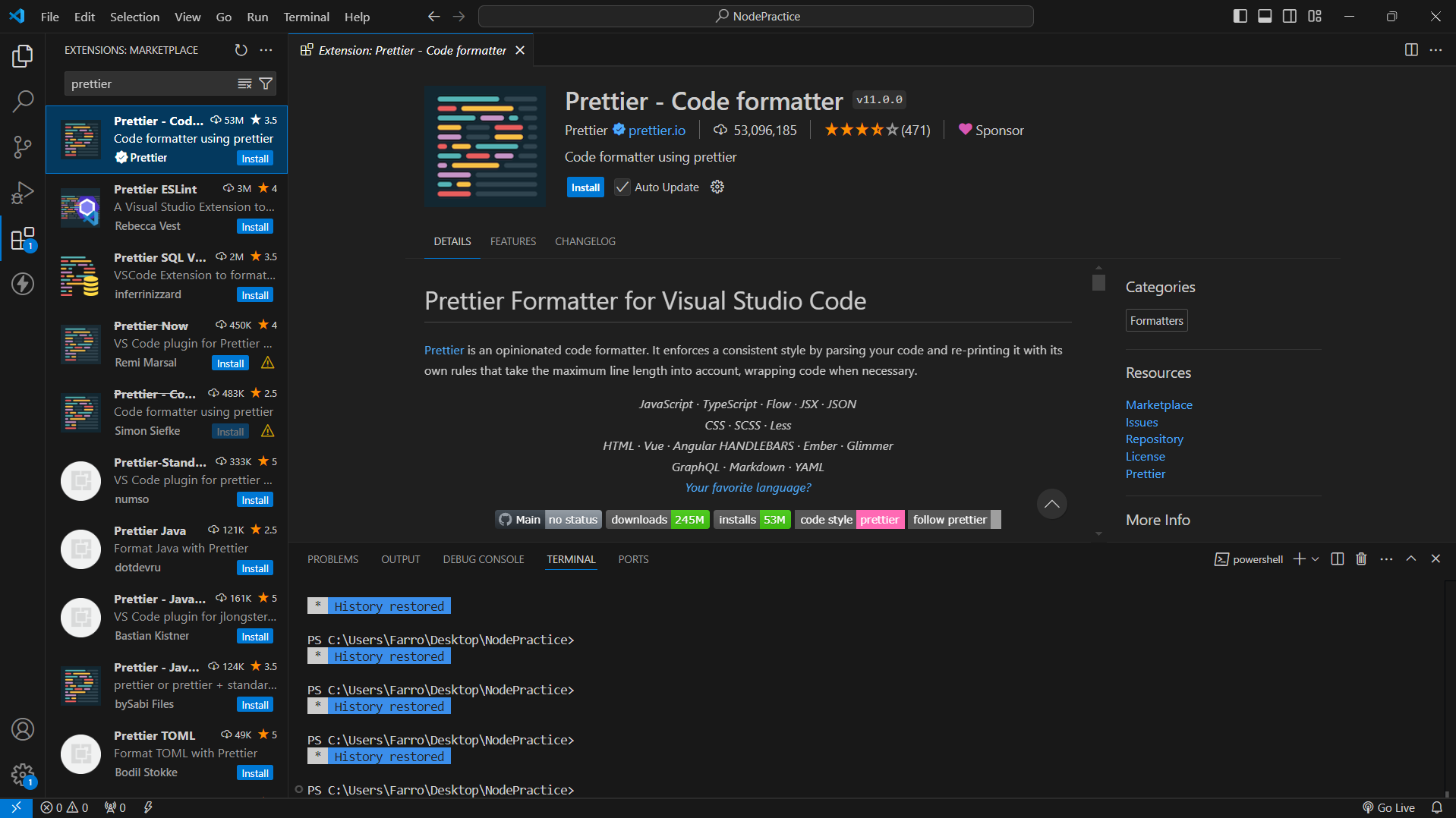
Let’s dive in and get started!

### **Front-End Development Team**

**Team Members:**

* Perla Rodriguez
* Jennifer Martinez
* Javier Castillo

#### **Resources You’ll Need**:

1. **Git**
   * A version control tool for managing code and collaborating efficiently.
   * **Important:** Always keep Git updated.
   * [Beginner's Guide Below]
2. **GitHub Account**
   * Create one here:<https://github.com/>
   * This will be crucial for sharing and managing our code repository.
3. **Visual Studio Code (VS Code)**
   * Our primary code editor.
   * **Tip:** Always update to the latest version for better performance.
   * Download here:<https://code.visualstudio.com/>
4. **VS Code Extensions**
   * **Live Server:** Enables quick live previews of your project.
   * **JavaScript (ES6) Code Snippets:** Provides helpful shortcuts for writing modern JavaScript.
   * **Bootstrap 4:** Makes integrating responsive design frameworks seamless.
   * **Prettier - Code Formatter:** Ensures your code stays clean, consistent, and well-organized.
     + *Why Prettier?* I’ve used it during my internship, and it’s great for maintaining code quality.  
       Visual Example of Prettier Extension Installed:  
       

#### **Visual Studio Code Setup**

Visual Studio Code is easy to set up, so I’m confident you’ll be able to follow the simple steps without any issues. To help, I’ve included a short YouTube video below for a smooth setup process. This video covers how to install Visual Studio Code and the required extensions (excluding Prettier, which you’ll need to install yourself).  
[Watch the Setup Video](https://www.youtube.com/watch?v=mL1IcxIUd5Y)

#### **Git Setup**

1. **Create a GitHub Account:**
   * If you haven’t already, create an account at [GitHub](https://github.com/).
2. **Download Git:**
   * Go to [Git Downloads](https://git-scm.com/downloads) to download Git for your operating system.
3. **Quick Git Setup Tutorial:**
   * For an easy-to-follow setup guide, watch this tutorial:  
     [Git Installation Tutorial](https://www.youtube.com/watch?v=iYkLrXobBbA)
4. **Git Cheat Sheet:**
   * Save this cheat sheet for easy reference throughout the project. It’s a helpful guide for common Git commands:  
     [Git Cheat Sheet](https://about.gitlab.com/images/press/git-cheat-sheet.pdf)
   * Feel free to explore other resources as well!
5. **Updating Git:**
   * If you already have Git installed but it’s an older version, simply uninstall it and install the latest version using the link above.

#### **Operating System Note**

I primarily use Windows for development, and I recommend you use it as well for the best compatibility. However, Visual Studio Code works seamlessly on macOS too. If you run into any issues on a Mac, don’t hesitate to reach out—I’m happy to help!

#### **Part 1: Cloning the GitHub Repository into Your Project**

In this section, we’ll go over how to clone the GitHub repository I created so you can start using Git with your project.

To get started, I'll need to add you as a collaborator to my GitHub project. Once you've created a GitHub account and installed Git, the process should be straightforward. Simply provide your email address to me via direct messages or in the Discord server. Once I've added you, I'll let you know by replying with "done."

Here’s the repository link:  
[GitHub Repository: GameMash-Front-End](https://github.com/FarronJones/GameMash-Front-End-)

**Setting Up Your Environment**

1. I recommend creating a folder called **"Front-End Development"** (or a name of your choice) on your computer.
2. Open this folder in **Visual Studio Code** or through your command line/terminal.

**Cloning the Repository**

I found a helpful video that explains the process of cloning a GitHub repository using Visual Studio Code:  
[Watch the video here](https://www.youtube.com/watch?v=eLmpKKaQL54).

* **Start at 0:50**, where the cloning process begins, and watch until **6:04**.
* You can follow the steps shown in the video using either the command prompt/terminal or Visual Studio Code’s built-in terminal—it’s the same process.

**Pushing and Pulling Changes**

The video also covers how to push and pull changes, which are essential skills for collaborative software development. These processes closely resemble how real-world development teams work.

As a collaborator, you'll be able to make pull requests. I'm in charge of reviewing and merging them, so don't worry about that part.

**Why Git is Important**

Git simplifies collaboration by allowing us to share code efficiently. Without Git, we’d be stuck copying and pasting code, which would quickly become a nightmare.

Additionally, using Git is a valuable skill for your resume and future software development work. I highly encourage you to learn and follow this process. However, if you encounter any difficulties or cannot use Git for some reason, let me know, and I'll do my best to accommodate you.

#### **Part 2: Day-to-Day Work**

The goal for the front-end development team is straightforward: we need to build a website inspired by [Cool Math Games](https://www.coolmathgames.com/0-fireboy-and-water-girl-in-the-forest-temple), specifically creating a platform that’s engaging, user-friendly, and supports playing games online.

Using all your front-end skills, work day by day to create an attractive and functional UI for the website. Be sure to reference the [document](https://docs.google.com/document/d/1NDO-plNh9rYwiRacIsIqpuTzWMmjXfC3cwptJcJgj9A/edit?usp=sharing) I’ve shared for additional guidance.

Your task is to build a unique and creative website with cool features for playing games. Feel free to utilize any resources—such as YouTube tutorials, Google searches, or other documentation—but remember, originality is key. Use the internet for inspiration, but our focus is on developing something distinct and reflective of your own ideas.

I don’t have a YouTube guide for you like the backend team, but website development is relatively simple. You can always reach out to me if you have questions or need help with specific tasks. Remember, the internet is your best friend.

#### **Daily Reporting**

In the Discord channel for your team, I’d like you to post your daily accomplishments. Use the following format:

* **Accomplishments Today:** *E.g., Reviewed the Group 3 project guide.*
* **Struggles/Questions Today:** *E.g., How do I pull from Git? How do I push changes to Git?*
* **Goals for Tomorrow:** *E.g., Create the first version of the website and set up columns.*

This format is similar to what I used during an internship for a startup, and it’s a great way for me to track progress and see who’s contributing. Plus, it helps keep the team organized and ensures everyone stays aligned.

As the project manager, I’ll still be monitoring progress closely, but this process will help keep communication clear and consistent. Let’s get started and build something amazing!

### **Back-End Development Team**

**Team Members:**

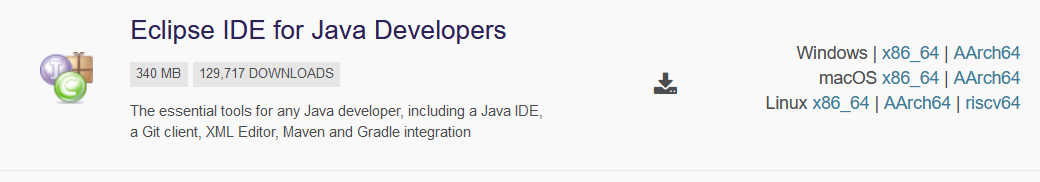
* Wayne Sterling
* Edmon Demirovski

#### **Resources You’ll Need**:

1. **Java JDK 23**
   * Java is the programming language we’re using for the back end, and the JDK (Java Development Kit) provides the necessary tools for coding.
   * Tip: Always use the latest version of the JDK.
   * Download JDK 23 here: [Oracle JDK 23 Downloads](https://www.oracle.com/java/technologies/downloads/#jdk23-windows)
2. **Setup Tutorial**
   * Watch this short tutorial to guide you through the installation process:  
     [Java JDK 23 Installation Tutorial](https://www.youtube.com/watch?v=WGQN0HKGnu0)
   * Note: You can ignore the part about environmental variables. Instead, check the installation by running java -version in the Command Prompt. If it shows your installed version, the setup is correct.
3. **Updating Your JDK**
   * If you have an older version of Java installed, simply uninstall it and then install the latest version of JDK 23.
4. **Verifying Your JDK Installation**
   * You can verify your JDK version anytime by running java -version in the Command Prompt. This will display the installed version.

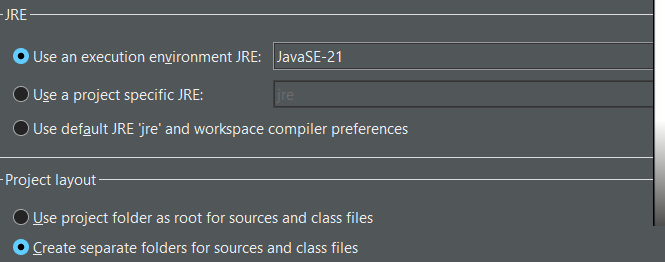
#### **Eclipse IDE Setup**

Eclipse IDE is an integrated development environment (IDE) that we'll use for our project development. You can download it from the official site here:  
[Eclipse IDE Official Website](https://eclipseide.org/)

**Installation Steps:**

1. **Download the Latest Version**
   * Always download the latest version from the Eclipse website for the best performance and features.
   * If you're new to Eclipse, follow this [tutorial for downloading Eclipse](https://www.youtube.com/watch?v=wMTdB7ElrIQ). The installation process in the video is quite similar to what we’ll be doing.
2. **Note on Java**
   * Eclipse comes bundled with a version of Java, but since you’ve already installed JDK 23, you don't need to worry about this.
3. **If You Have a ZIP Folder:**
   * If you download Eclipse as a ZIP file, simply extract all contents, then open the .exe file inside the extracted folder to launch Eclipse.

**JRE Version for Our Project:**

* For our project, we will be using JRE 21 (Java SE 21).
* Other versions like SE 22 and SE 23 are in beta, so they won’t work with our setup.  
  

#### **Git Setup**

* **Git** helps with version control and easy collaboration on our project. Always ensure that you have the latest version installed.
* **GitHub Account:**
  + If you don't already have one, create a GitHub account here:<https://github.com/>

**Installing Git**

1. **Download Git:**
   * You can download Git from the official website: [Git Downloads](https://git-scm.com/downloads).
2. **Quick Setup Tutorial:**
   * For a simple, step-by-step guide on how to install Git, watch this tutorial:  
     [Git Installation Tutorial](https://www.youtube.com/watch?v=iYkLrXobBbA)
3. **Git Cheat Sheet:**
   * Save this Git cheat sheet for easy reference during the project. You can also explore other guides if needed:  
     [Git Cheat Sheet](https://about.gitlab.com/images/press/git-cheat-sheet.pdf)

#### **Operating System Notes**

* I primarily use **Windows**, and I recommend using Windows for consistency, whether you’re working on campus or from home.
* **Mac Users:**
  + If you're using macOS, the process should work, but there may occasionally be issues that I can't help with. I can provide additional tutorial videos if you run into trouble.

#### **Part 1: Cloning the GitHub Repository**

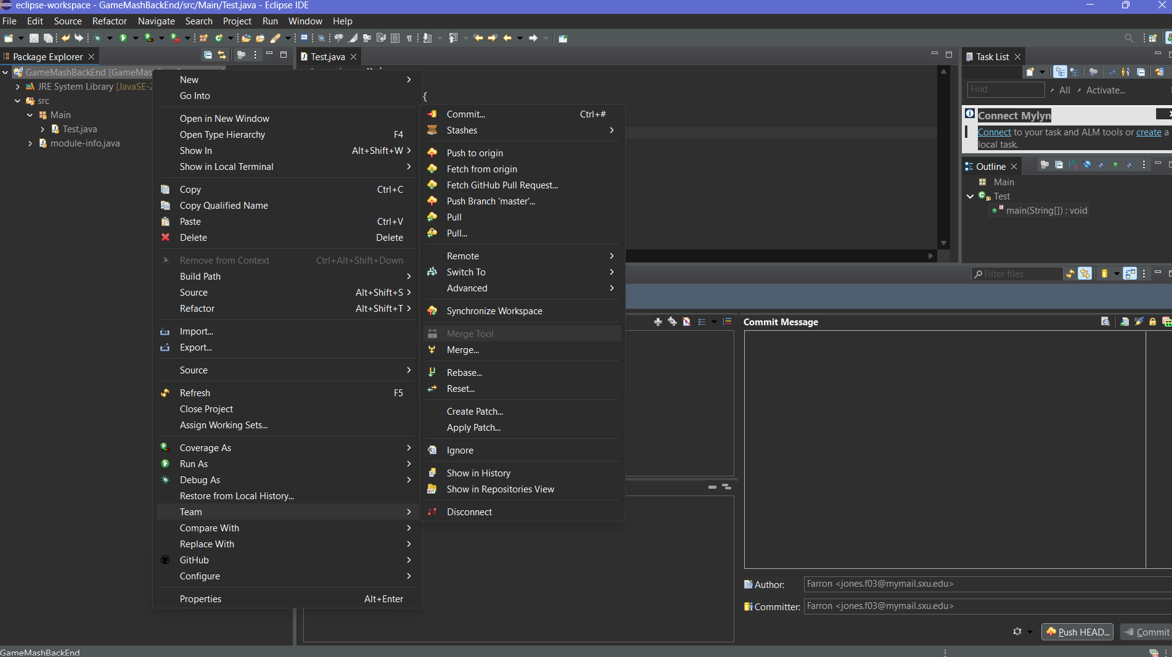
In this section, we’ll go over how to clone the GitHub repository I created into your project to start using Git.

To get started, I'll need to add you as a collaborator to my GitHub project. Once you've created a GitHub account and installed Git, the process should be straightforward. Simply provide your email address to me via direct messages or in the Discord server. Once I've added you, I'll let you know by replying with "done."

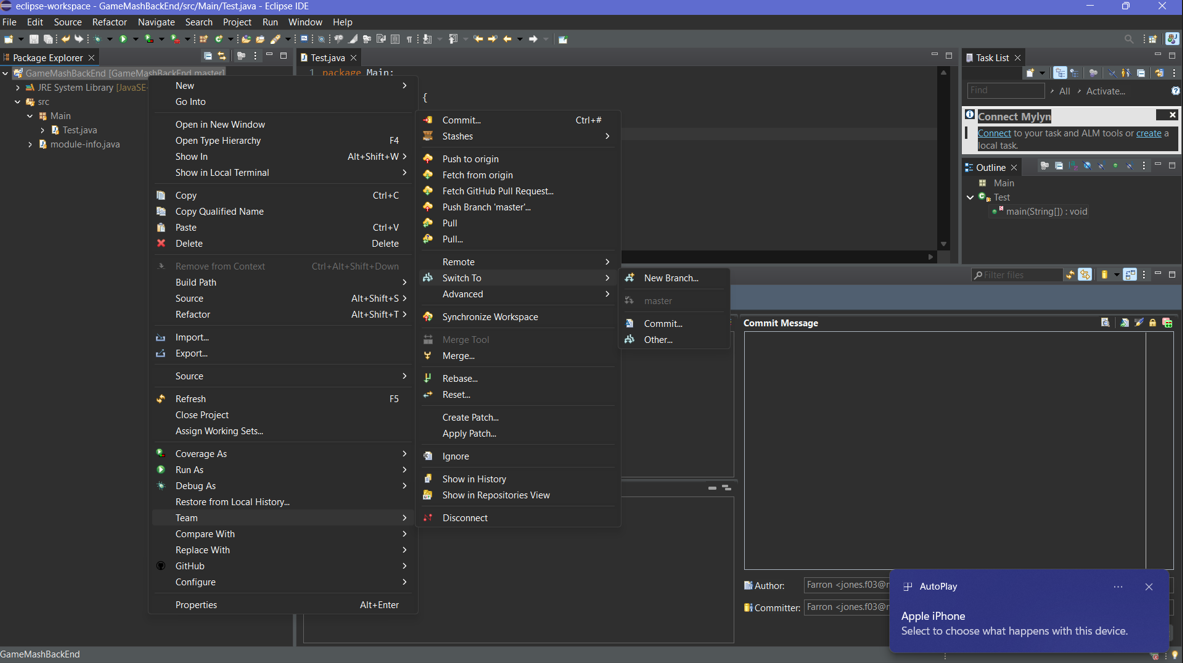
Here’s the repository link:  
<https://github.com/FarronJones/GameMashBackEnd>  
  
  
To get started, you'll need to clone the GitHub repository using Eclipse IDE. This process is straightforward, but make sure you have your Git username and password handy, as you'll use them frequently in Eclipse. I’ve found the perfect guide for this process—it's recent and very clear. Watch the video below from **0:00 to 5:10** to learn how to clone the project:  
[Cloning a GitHub Repository with Eclipse](https://www.youtube.com/watch?v=5iNuVKC2z1Q)

Eclipse IDE Git WorkFlow Full Steps(Use this to help you, this is my day-to-day guide for our project)

https://docs.google.com/document/d/1Eje9iW4\_34N96LgJehLlXk-PDKCLaqDKwwVVvGAmLAs/edit?usp=sharing

For pushing your code, use the personal access token below instead of a password:  
**Personal Access Token:** ghp\_WeUkhlaUXxbBsEfNH2s5oKCV0rGMku1FipuR  
  
Here’s my personal input: Although there aren’t many comprehensive guides for using Git with Eclipse, the process is still fairly simple. After cloning the project, right-click on it in Eclipse and select **Team**.  


While my Eclipse UI might look slightly different from yours, you can customize it if needed. The essential Git tools, such as viewing commits, are all easily accessible under the **Team** menu.

You can also switch branches or create new ones, which is an important step for managing your work effectively.  


To better understand branch management, I recommend following the front-end development team tutorial:

* [Watch the video here.](https://www.youtube.com/watch?v=eLmpKKaQL54) (Watch from **0:50 to 6:40**)  
  Although the video demonstrates these steps in a different IDE which is Visual Studio Code, the process is the same in Eclipse.

For additional reference, here’s another similar tutorial that reinforces the same concepts but in Eclipse IDE:

[Eclipse IDE Branching](https://www.youtube.com/watch?v=xVzAKN2YCCs)  
  
**Why Git is Important**

Git simplifies collaboration by allowing us to share code efficiently. Without Git, we’d be stuck copying and pasting code, which would quickly become a nightmare.

Additionally, using Git is a valuable skill for your resume and future software development work. I highly encourage you to learn and follow this process. However, if you encounter any difficulties or cannot use Git for some reason, let me know, and I'll do my best to accommodate you.

#### **Part 2: Day-to-Day Work**

The goal of the backend development team is to create a game inspired by *Fireboy and Watergirl*. You can check out the gameplay in this [video](https://www.youtube.com/watch?v=og0inOdAh8w). The game should support two separate players(Hopefully) in a 2D platformer environment and provide smooth gameplay. Refer to the [document](https://docs.google.com/document/d/1NDO-plNh9rYwiRacIsIqpuTzWMmjXfC3cwptJcJgj9A/edit?tab=t.0#heading=h.f8ezna55j4db) I’ve provided for specific details about the game’s goals and user experience.

To assist us in building the game, I’ve found a great reference online. While we can use a decent amount of the code from this resource, we won’t use all of it since our game will differ in significant ways. Our focus is on making something unique while leveraging useful ideas from existing projects.

The main reason we’re using **Eclipse IDE** and **Java** for this project is because they’re well-suited for game development, and there are plenty of tutorials available to help us along the way. I’ve found a helpful [YouTube playlist](https://www.youtube.com/playlist?list=PL4rzdwizLaxYmltJQRjq18a9gsSyEQQ-0) that covers Java 2D platformer game development in Eclipse.

Additionally, I’ve worked on a similar game project myself, so I have some firsthand experience in this area. You can check out my work here: [Personal Java Game Project](https://github.com/FarronJones/PersonalProject).

Like the front-end development team, we’ll rely heavily on online resources since game development can be challenging. However, with the provided guides, references, and my support, I’m confident we’ll successfully bring this project to life.

Let’s focus on building a smooth, engaging game while making it unique and creative!

#### **Daily Reporting**

In the Discord channel for your team, I’d like you to post your daily accomplishments. Use the following format:

* **Accomplishments Today:** *E.g., Reviewed the Group 3 project guide.*
* **Struggles/Questions Today:** *E.g., How do I pull from Git? How do I push changes to Git?*
* **Goals for Tomorrow:** *E.g., Complete the first game development tutorial and push the changes to the GitHub repository.*

This format is similar to what I used during an internship for a startup, and it’s a great way for me to track progress and see who’s contributing. Plus, it helps keep the team organized and ensures everyone stays aligned.

As the project manager, I’ll still be monitoring progress closely, but this process will help keep communication clear and consistent. Let’s get started and build something amazing!