NLR-PT-8 TErdle Development

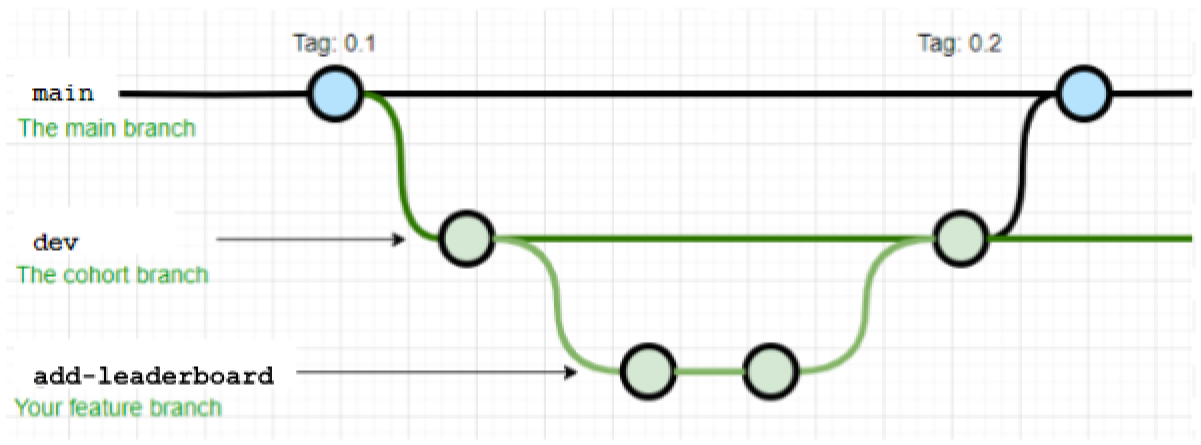
Branches

In Git, branches play an important role in your everyday development process. Think of a Git branch as a pointer to a snapshot of a specific moment in your project's history.

When you're working on a coding project and you want to add a new feature or fix a bug, you should make your changes on a branch. It doesn't matter if your changes are big or small; you'll create a new branch to keep things organized.

Working on a branch in Git isolates your changes from the main project, making it much harder for any shaky code to get into the final product. Creating a branch is like saying, "I'm going to work on something new now, and I don't want to mess up the rest of the project." This isolation gives you the freedom to experiment, make mistakes, and try out new ideas without harming the main project's stability.

Branches also let you clean up your work before merging into the main project. You (and others) can review your changes, fix any mistakes, and ensure your work meets all checkin criteria and is just the way you want it to be.



Main Branch (main)

The `main` branch is special. It's typically considered the stable and production-ready version of your code.   The main branch is expected to be the most stable and reliable version of your project. It should be the one that you can confidently deploy to your production environment without major issues. Working directly in the main branch can introduce instability, as code changes might not be thoroughly tested and could break the existing functionality. **Developers will not work from the `main` branch**! If you try to push to the `main` branch you will be blocked.

Cohort Branch (dev)

The `dev` branch will act as the cohort's branch that contains all the cohort specific changes. This branch will be the basis for cohort feature branches. **Developers will not work not work directly from the dev branch**.

Feature Branches

A feature branch is a dedicated, isolated workspace in your Git repository that exists for the sole purpose of developing a new feature, fixing a bug, or working on a specific task.
Developers can share progress by pushing their branches to the shared repository, allowing for code reviews, feedback, and discussions before the changes are merged into the `dev` branch.

- All feature branches should be created with the `dev` branch as the branch source.
- All feature branches should be associated with and created from a Github issue.
- Developers will always work on, commit, and push changes to a feature branch.

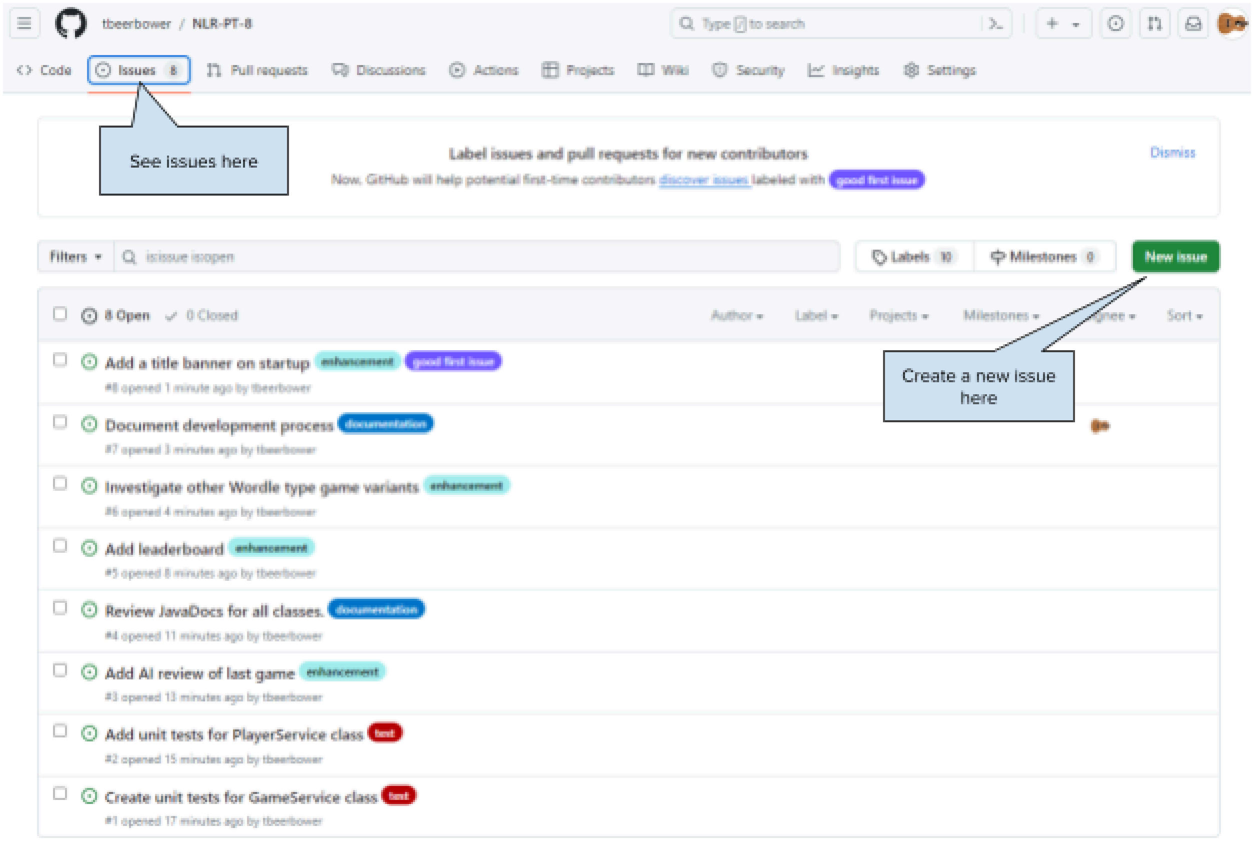
Workflow

The typical workflow in GitHub, from creating an issue to merging it back into the `dev` cohort branch, follows a structured process that allows for systematic tracking, collaboration, code review, and testing throughout the development process. It ensures that changes are well-considered, thoroughly reviewed, and integrated into the project in an organized and controlled manner. Additionally, it provides transparency and accountability by linking code changes to specific issues or feature requests.

Here are the key steps involved:

Issue Creation

1. A team member identifies a need for a new feature, a bug fix, or an improvement in the software.
2. An issue is created in the GitHub repository to document and track this need. This issue describes the problem or feature request, includes relevant details, and may be assigned to a specific team member or labeled accordingly.



Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with good first issue

8 Open ✓ 0 Closed

Add a title banner on startup enhancement good first issue
#8 opened 1 minute ago by tbeerbower

Document development process documentation
#7 opened 3 minutes ago by tbeerbower

Investigate other Wordle type game variants enhancement
#6 opened 4 minutes ago by tbeerbower

Add leaderboard enhancement
#5 opened 8 minutes ago by tbeerbower

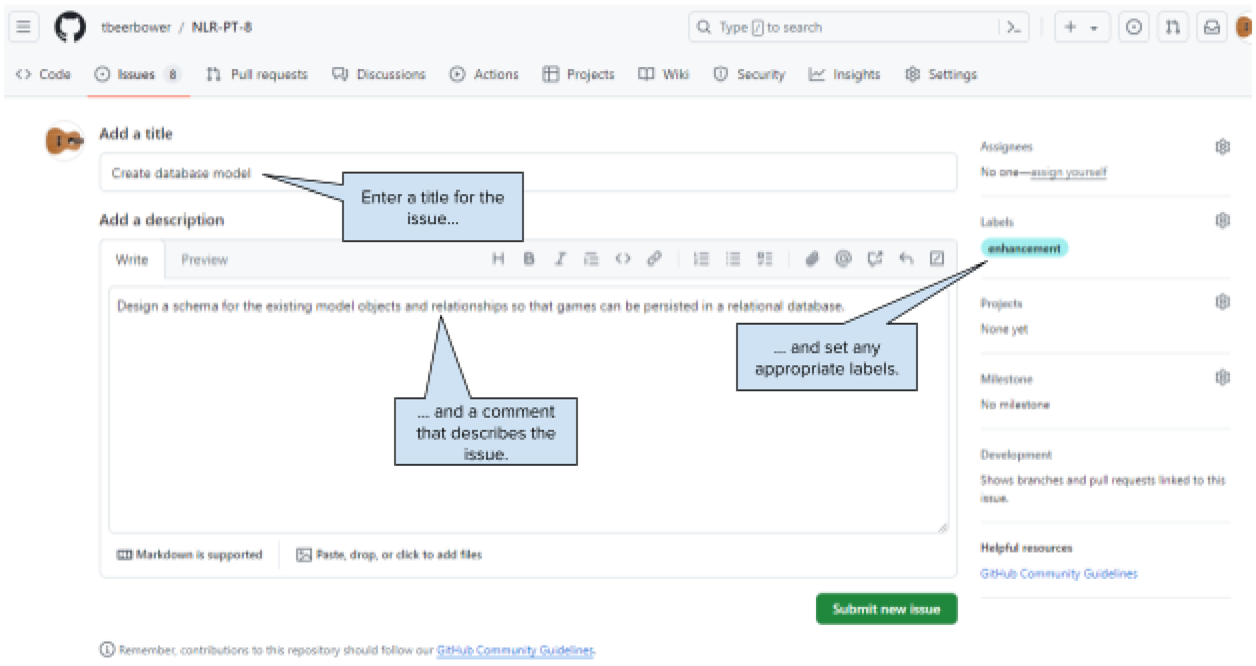
Review JavaDocs for all classes. documentation
#4 opened 11 minutes ago by tbeerbower

Add AI review of last game enhancement
#3 opened 13 minutes ago by tbeerbower

Add unit tests for PlayerService class test
#2 opened 15 minutes ago by tbeerbower

Create unit tests for GameService class test
#1 opened 17 minutes ago by tbeerbower

Enter the required information when you create a new issue including the title, comment and any appropriate labels. You may also assign the issue to yourself if you plan on working on it right away.

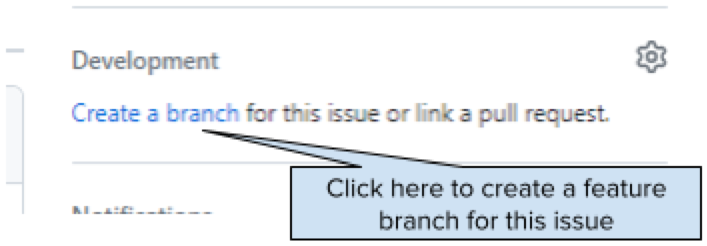

tbeerbower / NLR-PT-8
Code
Issues 8
Pull requests
Discussions
Actions
Projects
Wiki
Security
Insights
Settings
Add a title
Create database model
Enter a title for the issue...
Add a description
Write
Preview
... and a comment that describes the issue.
... and set any appropriate labels.
Markdown is supported
Paste, drop, or click to add files
Submit new issue
Assignees
No one—assign yourself
Labels
enhancement
Projects
None yet
Milestone
No milestone
Development
Shows branches and pull requests linked to this issue.
Helpful resources
GitHub Community Guidelines
Remember, contributions to this repository should follow our GitHub Community Guidelines.

Design a schema for the existing model objects and relationships so that games can be persisted in a relational database.

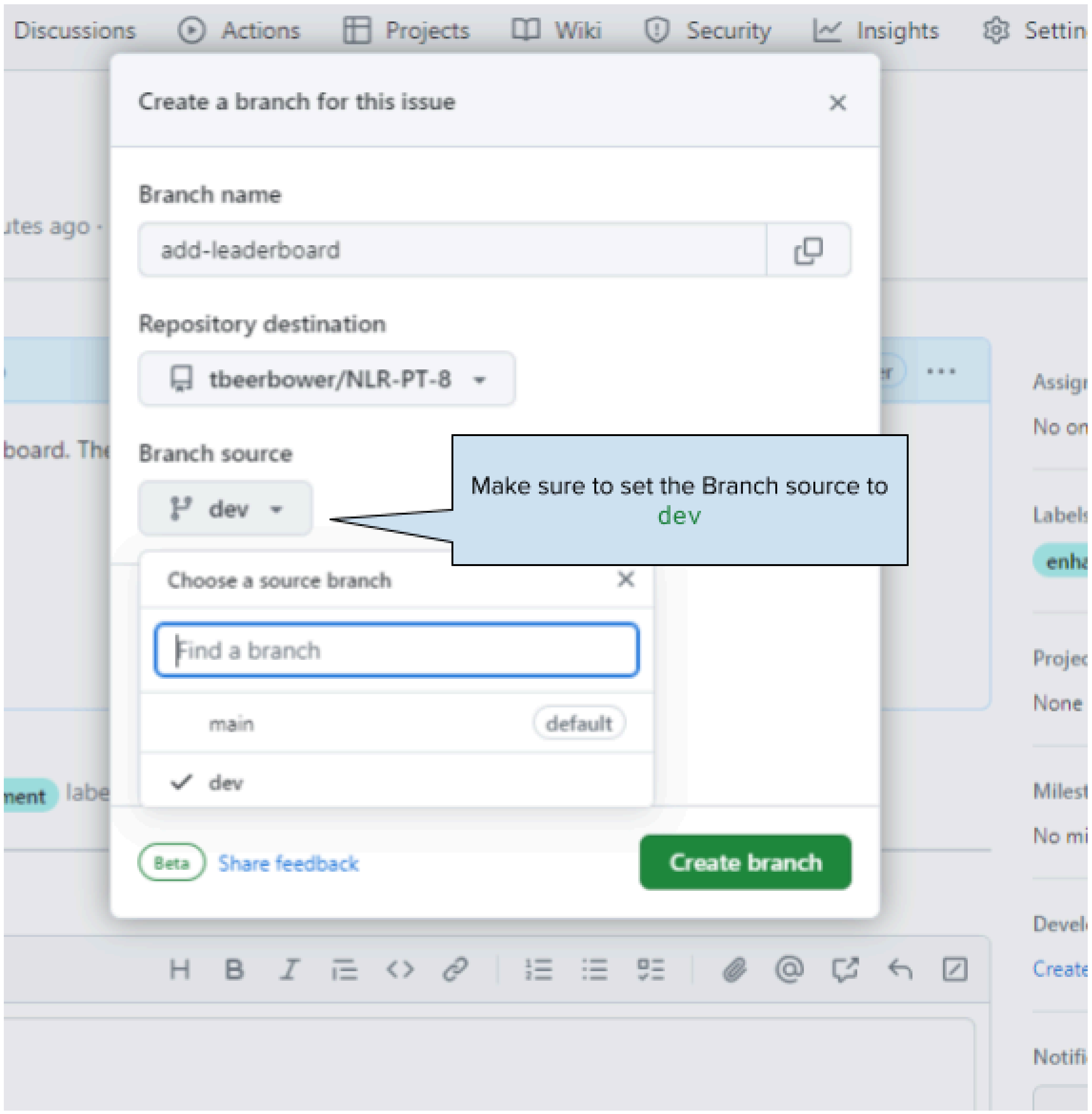
Feature Branch Creation

A developer (usually the one assigned to the issue) creates a new feature branch in the repository based on the `dev` cohort branch. This branch typically has a name related to the issue (e.g. `add-leaderboard`), making it easy to associate the work with the issue.

Once the issue is created, you can go back to it and create a feature branch from it.


Development
Create a branch for this issue or link a pull request.
Click here to create a feature branch for this issue

The source branch for the feature branch should be the `dev` branch.



Create a branch for this issue

Branch name: add-leaderboard

Repository destination: tbeerbower/NLR-PT-8

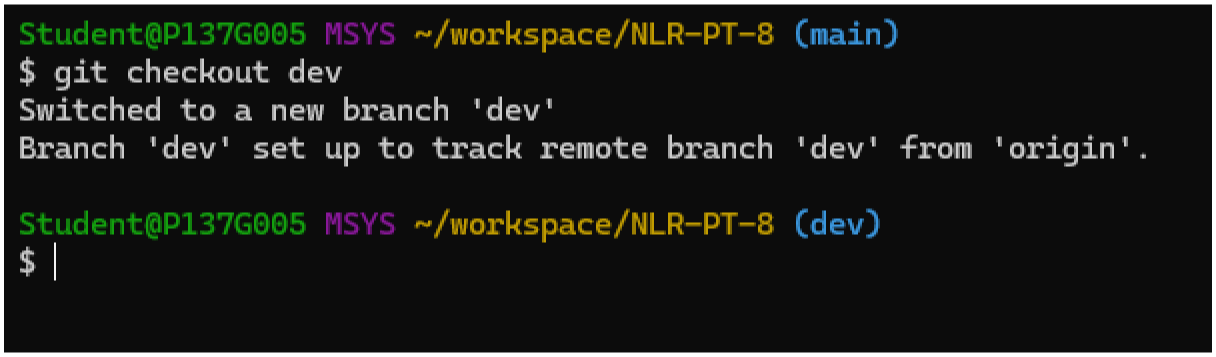
Branch source: dev

Make sure to set the Branch source to `dev`

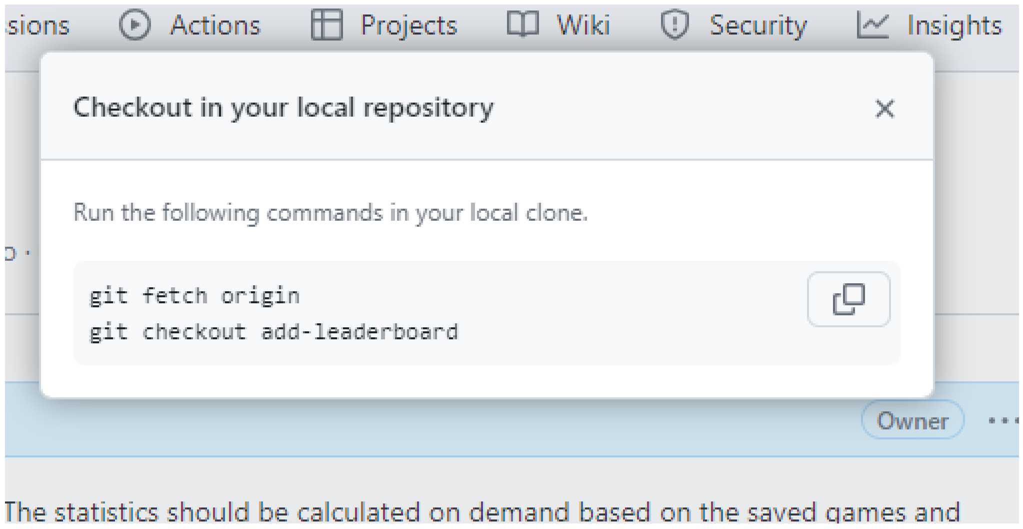
Development

1. The developer works on the issue within the dedicated feature branch, implementing the necessary code changes, bug fixes, or feature additions.
2. During development, the developer makes commits to the branch to track progress. Commit messages should reference the issue number (e.g., "Fixes #123") to link the changes to the issue.

From your local `NLR-PT-8` repo, checkout the `dev` branch. Then follow the instructions from Github to fetch and change to your new feature branch.



```
Student@P137G005 MSYS ~/workspace/NLR-PT-8 (main)
$ git checkout dev
Switched to a new branch 'dev'
Branch 'dev' set up to track remote branch 'dev' from 'origin'.

Student@P137G005 MSYS ~/workspace/NLR-PT-8 (dev)
$ |
```



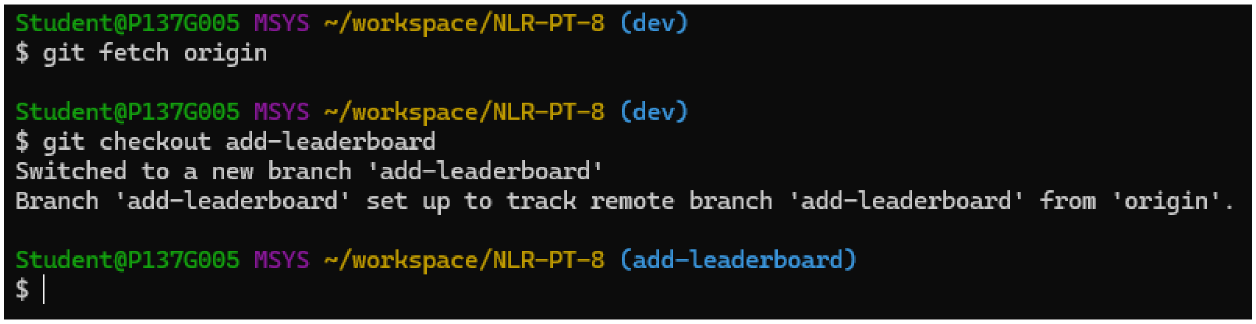
Checkout in your local repository

Run the following commands in your local clone.

```
git fetch origin
git checkout add-leaderboard
```

The statistics should be calculated on demand based on the saved games and

In your local repo, checkout your feature branch so that any changes that you make are done in the feature branch.

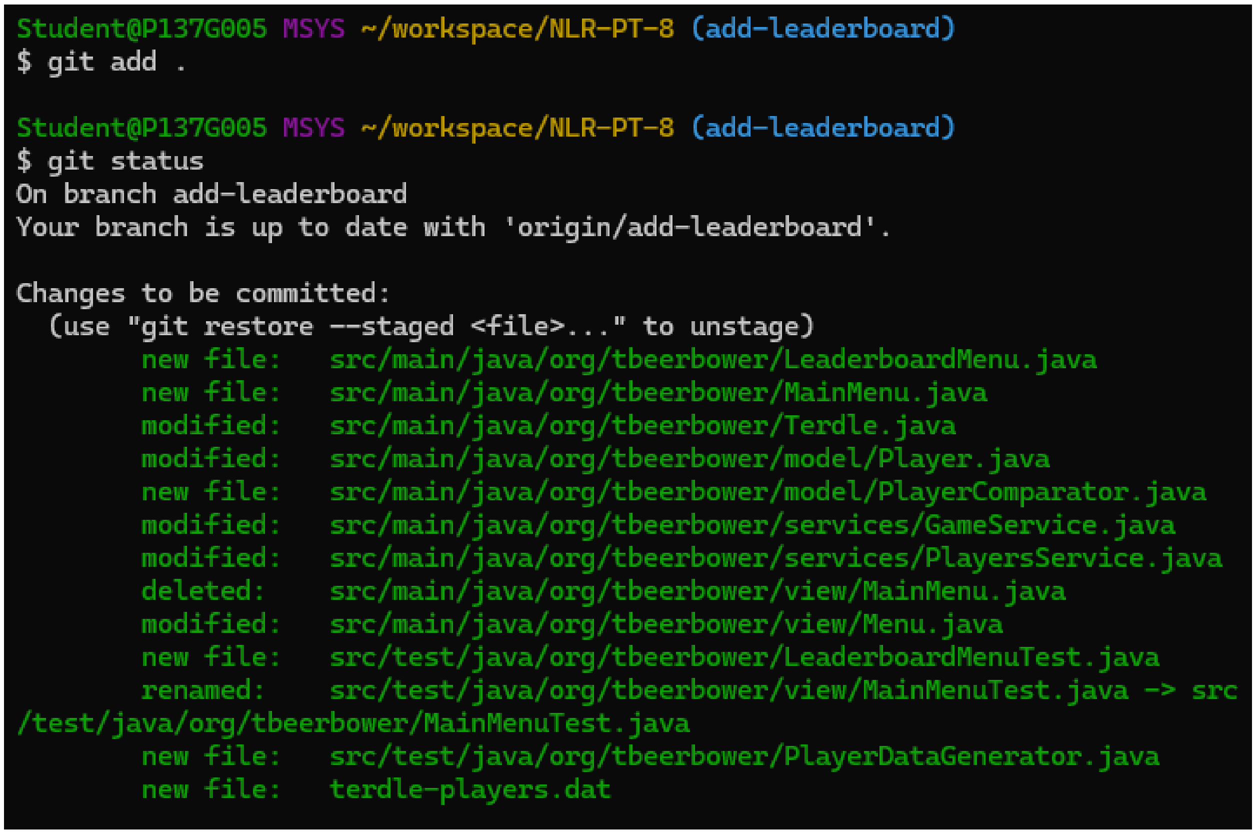


```
Student@P137G005 MSYS ~/workspace/NLR-PT-8 (dev)
$ git fetch origin

Student@P137G005 MSYS ~/workspace/NLR-PT-8 (dev)
$ git checkout add-leaderboard
Switched to a new branch 'add-leaderboard'
Branch 'add-leaderboard' set up to track remote branch 'add-leaderboard' from 'origin'.

Student@P137G005 MSYS ~/workspace/NLR-PT-8 (add-leaderboard)
$ |
```

When you are done writing code and are satisfied that the code meets all coding standards and passes all required tests, then commit and push your changes into the feature branch.

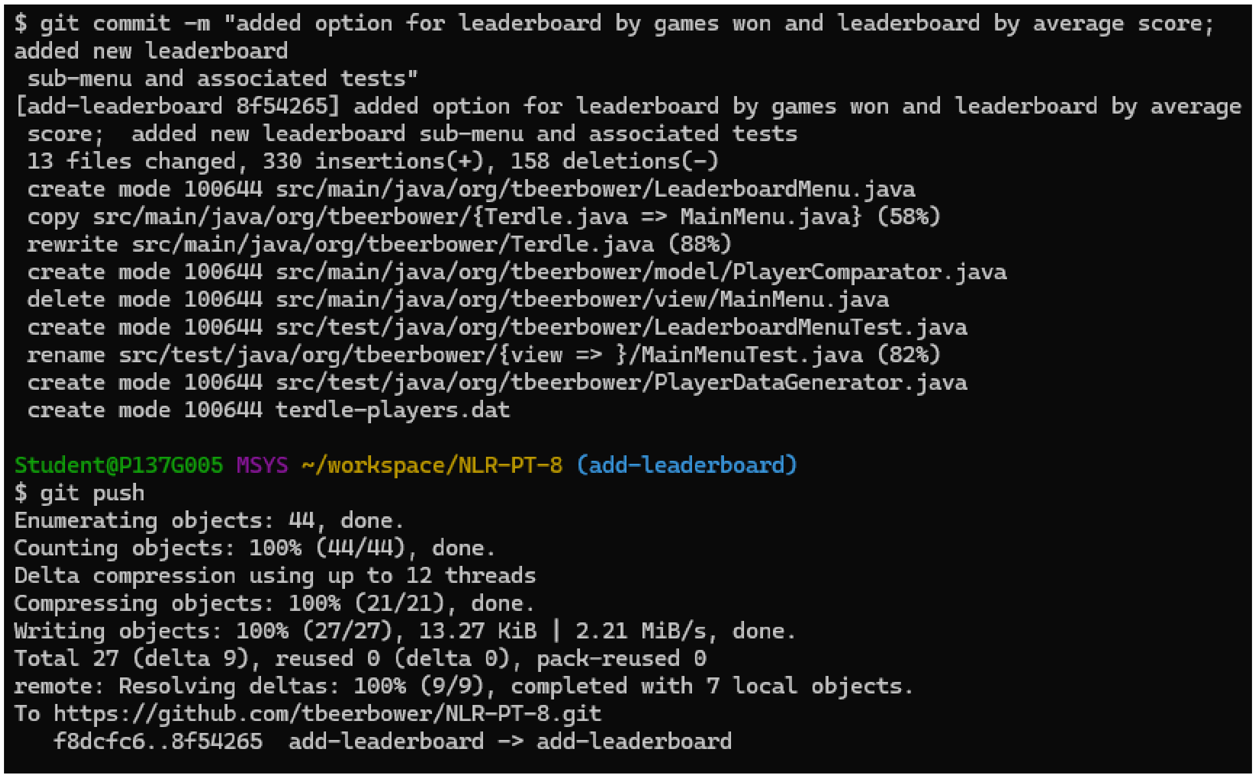
```
git add.
git commit -m "<your commit message>"
git push
```

Add the changes.



```
Student@P137G005 MSYS ~/workspace/NLR-PT-8 (add-leaderboard)
$ git add .

Student@P137G005 MSYS ~/workspace/NLR-PT-8 (add-leaderboard)
$ git status
On branch add-leaderboard
Your branch is up to date with 'origin/add-leaderboard'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   src/main/java/org/tbeerbower/LeaderboardMenu.java
        new file:   src/main/java/org/tbeerbower/MainMenu.java
        modified:   src/main/java/org/tbeerbower/Terdle.java
        modified:   src/main/java/org/tbeerbower/model/Player.java
        new file:   src/main/java/org/tbeerbower/model/PlayerComparator.java
        modified:   src/main/java/org/tbeerbower/services/GameService.java
        modified:   src/main/java/org/tbeerbower/services/PlayersService.java
        deleted:    src/main/java/org/tbeerbower/view/MainMenu.java
        modified:   src/main/java/org/tbeerbower/view/Menu.java
        new file:   src/test/java/org/tbeerbower/LeaderboardMenuTest.java
        renamed:    src/test/java/org/tbeerbower/view/MainMenuTest.java -> src
/test/java/org/tbeerbower/MainMenuTest.java
        new file:   src/test/java/org/tbeerbower/PlayerDataGenerator.java
        new file:   terdle-players.dat
```

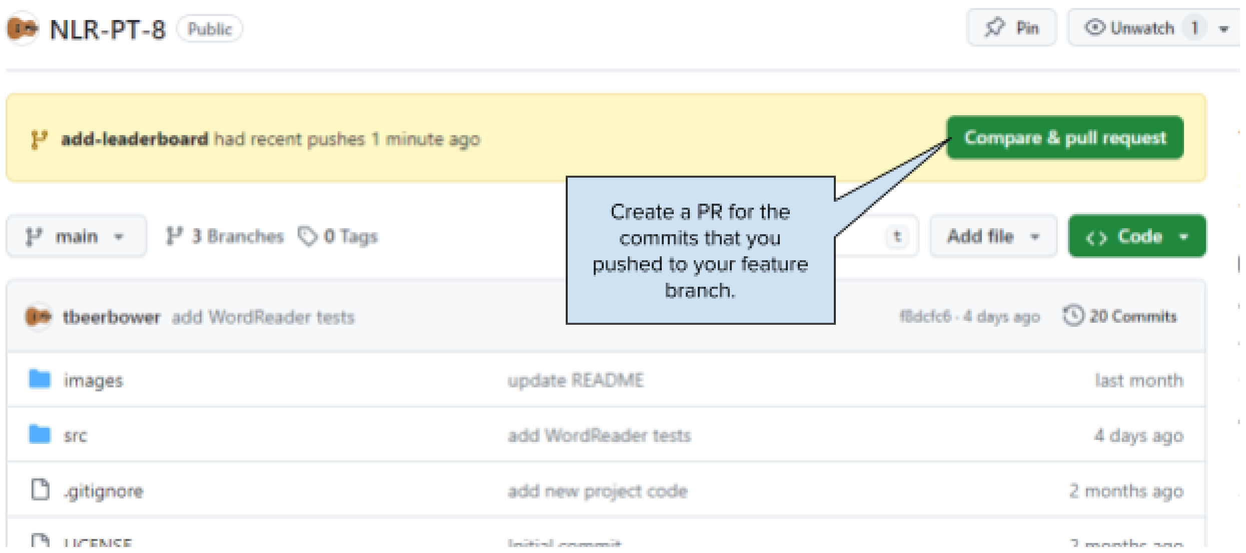
Commit and push the changes.



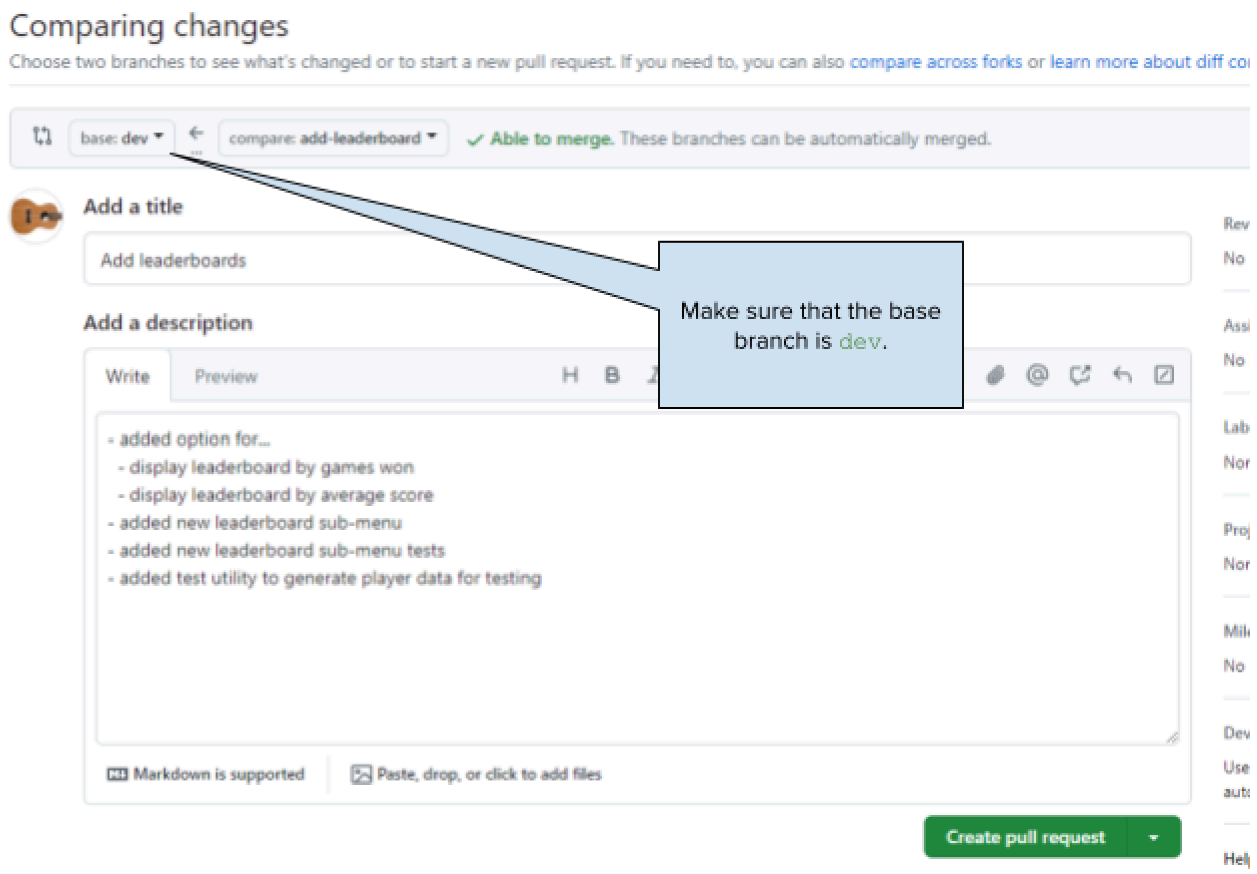
```
$ git commit -m "added option for leaderboard by games won and leaderboard by average score;
added new leaderboard
 sub-menu and associated tests"
[add-leaderboard 8f54265] added option for leaderboard by games won and leaderboard by average
 score;  added new leaderboard sub-menu and associated tests
 13 files changed, 330 insertions(+), 158 deletions(-)
 create mode 100644 src/main/java/org/tbeerbower/LeaderboardMenu.java
 copy src/main/java/org/tbeerbower/{Terdle.java => MainMenu.java} (58%)
 rewrite src/main/java/org/tbeerbower/Terdle.java (88%)
 create mode 100644 src/main/java/org/tbeerbower/model/PlayerComparator.java
 delete mode 100644 src/main/java/org/tbeerbower/view/MainMenu.java
 create mode 100644 src/test/java/org/tbeerbower/LeaderboardMenuTest.java
 rename src/test/java/org/tbeerbower/{view => }/MainMenuTest.java (82%)
 create mode 100644 src/test/java/org/tbeerbower/PlayerDataGenerator.java
 create mode 100644 terdle-players.dat

Student@P137G005 MSYS ~/workspace/NLR-PT-8 (add-leaderboard)
$ git push
Enumerating objects: 44, done.
Counting objects: 100% (44/44), done.
Delta compression using up to 12 threads
Compressing objects: 100% (21/21), done.
Writing objects: 100% (27/27), 13.27 KiB | 2.21 MiB/s, done.
Total 27 (delta 9), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (9/9), completed with 7 local objects.
To https://github.com/tbeerbower/NLR-PT-8.git
   f8dcfc6..8f54265  add-leaderboard -> add-leaderboard
```

Pull Request Creation

Once the developer completes the work and is ready for feedback and review, they create a Pull Request (PR). A PR is a request to merge the commits from the feature specific branch into the development branch (dev in this case).



When creating the Pull Request, make sure to select the base base of dev. Also, add a title and a description that lists the changes included in the feature branch being merged.


Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff co

base: dev ← compare: add-leaderboard ✓ Able to merge. These branches can be automatically merged.

Add a title

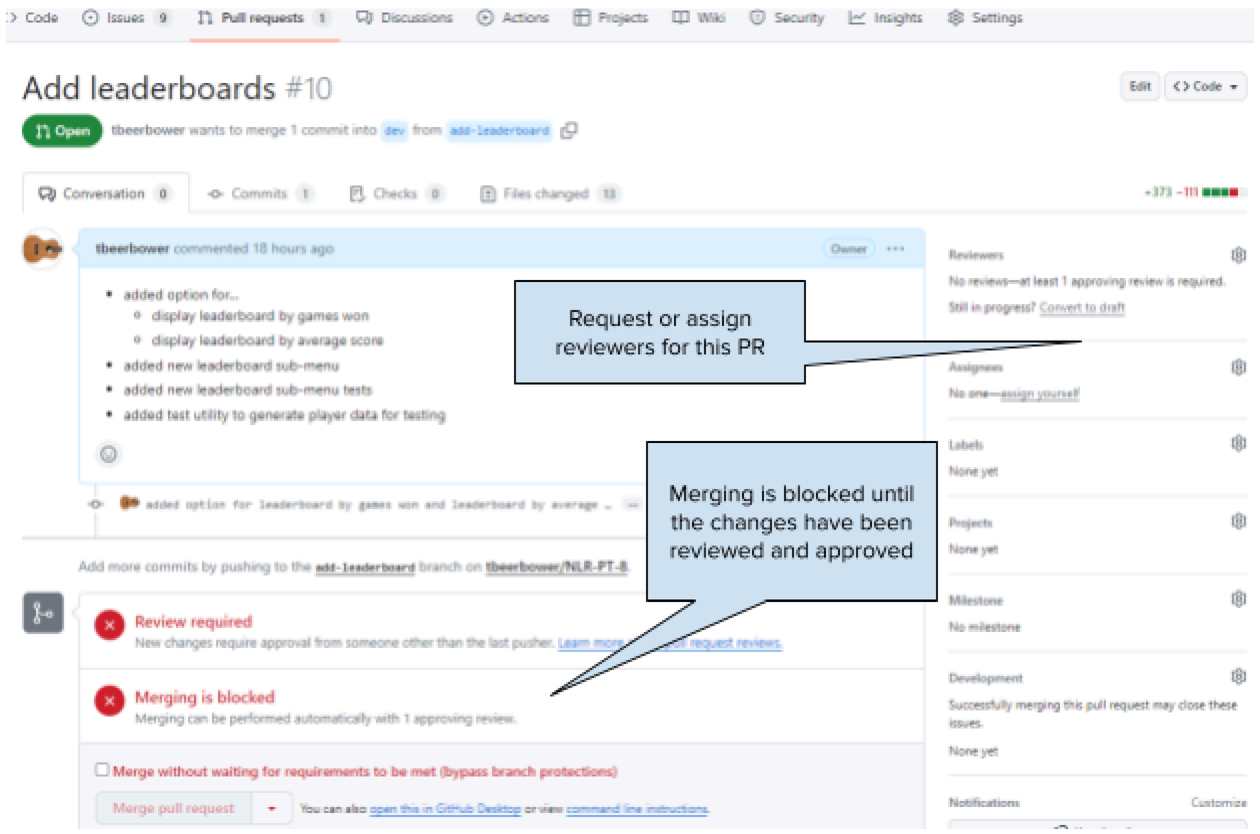
Add leaderboards

Add a description

```
- added option for...
  - display leaderboard by games won
  - display leaderboard by average score
- added new leaderboard sub-menu
- added new leaderboard sub-menu tests
- added test utility to generate player data for testing
```

Write
Preview
Markdown is supported
Paste, drop, or click to add files
Create pull request
Make sure that the base branch is dev.

Once the PR is created, you can find it under the `Pull Requests` tab. Note that the PR can not be merged until it has been reviewed and approved.



Add leaderboards #10

tbeerbower wants to merge 1 commit into dev from add-leaderboard

tbeerbower commented 18 hours ago

- added option for...
  - display leaderboard by games won
  - display leaderboard by average score
- added new leaderboard sub-menu
- added new leaderboard sub-menu tests
- added test utility to generate player data for testing

added option for leaderboard by games won and leaderboard by average ...

Add more commits by pushing to the add-leaderboard branch on tbeerbower/NLR-PT-8.

Review required
New changes require approval from someone other than the last pusher. Learn more about request reviews.

Merging is blocked
Merging can be performed automatically with 1 approving review.

☐ Merge without waiting for requirements to be met (bypass branch protections)

You can also open this in GitHub Desktop or view command line instructions.

Reviewers
No reviews—at least 1 approving review is required.
Still in progress? Convert to draft

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Notifications

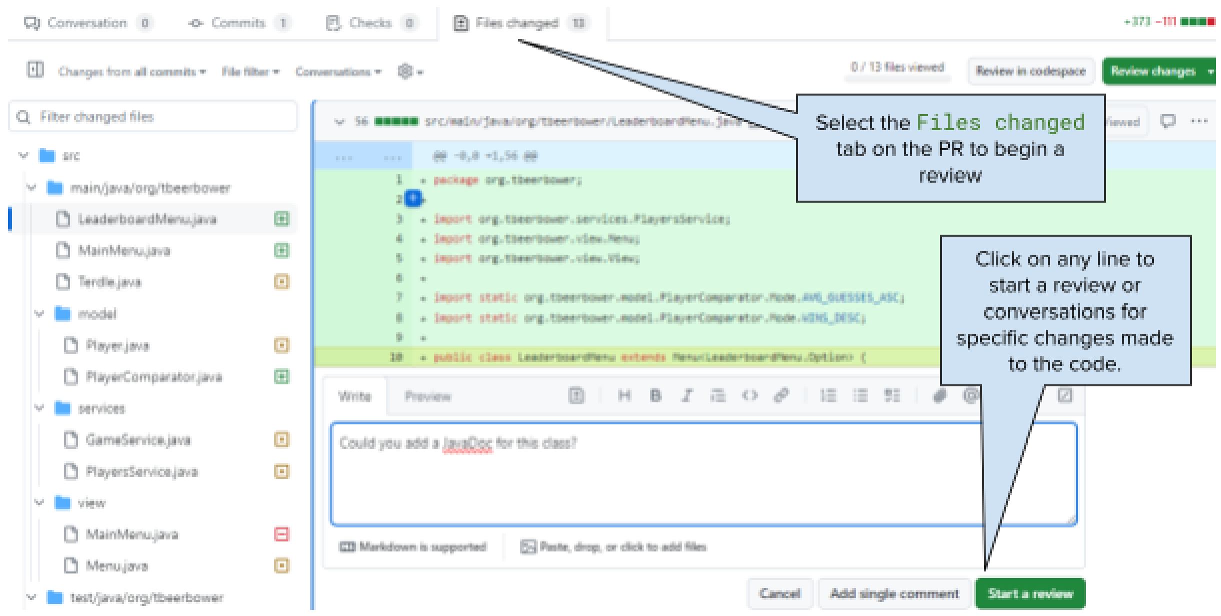
Code Review

At least one other team member needs to review the code changes within the PR.
During the review, the reviewers can leave comments, suggest improvements, and discuss the changes to ensure code quality and adherence to project standards.
All team members are responsible for reviewing PRs. The choice of reviewers may depend on the complexity of the changes, the area of the codebase affected, or their domain expertise.
Reviewers should thoroughly examine the code changes within the PR. This involves:

- Evaluating the quality of the code, ensuring it adheres to coding standards, and looking for potential improvements.
- Checking that the proposed changes address the issue or feature request and do not introduce unintended side effects.
- Ensuring that code is well-documented, comments are clear, and variable/method names are descriptive.
- Verifying that the proposed changes include or update tests to cover new functionality and that existing tests pass.
- Reviewing for security vulnerabilities and following secure coding practices.
- Confirming that the changes follow best practices for the technology stack being used.

Reviewers may leave comments on specific lines of code or within the PR discussion to provide feedback, ask questions, or suggest improvements. This feedback can result in a constructive

dialogue between the developer and the reviewers, aiming to improve the code quality and resolve any issues.
Based on the feedback received, the developer may need to make additional changes to the code in response to the review comments. This process can involve multiple iterations of review and revision until the code meets the project's standards.

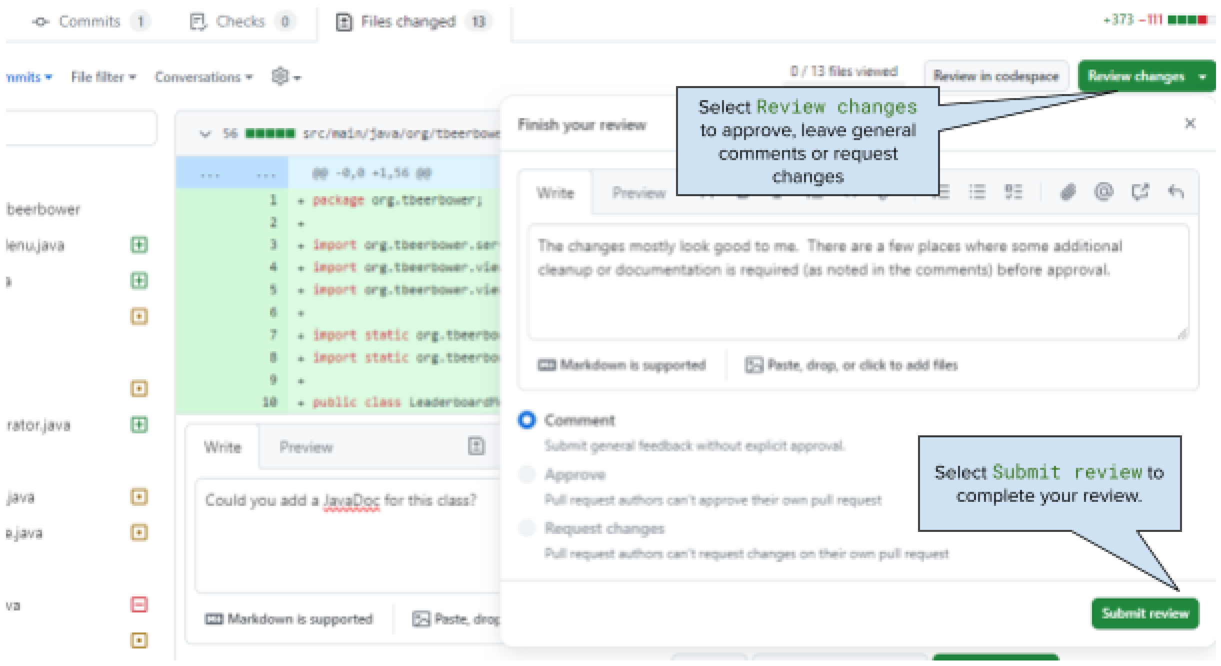


Select the `Files changed` tab on the PR to begin a review

Click on any line to start a review or conversations for specific changes made to the code.

```
56 src/main/java/org/tbeerbower/LeaderboardMenu.java
@@ -0,0 +1,56 @@
+ package org.tbeerbower;
+
+ import org.tbeerbower.services.PlayersService;
+ import org.tbeerbower.view.Menu;
+ import org.tbeerbower.view.View;
+
+ import static org.tbeerbower.model.PlayerComparator.Mode.AVG_GUESSES_ASC;
+ import static org.tbeerbower.model.PlayerComparator.Mode.WINS_DESC;
+
+ public class LeaderboardMenu extends Menu<LeaderboardMenu.Option> {
```

Could you add a JavaDoc for this class?



Select `Review changes` to approve, leave general comments or request changes

The changes mostly look good to me. There are a few places where some additional cleanup or documentation is required (as noted in the comments) before approval.

[✓ Comment]
Submit general feedback without explicit approval.
Approve
Pull request authors can't approve their own pull request
Request changes
Pull request authors can't request changes on their own pull request

Select `Submit review` to complete your review.

Could you add a JavaDoc for this class?

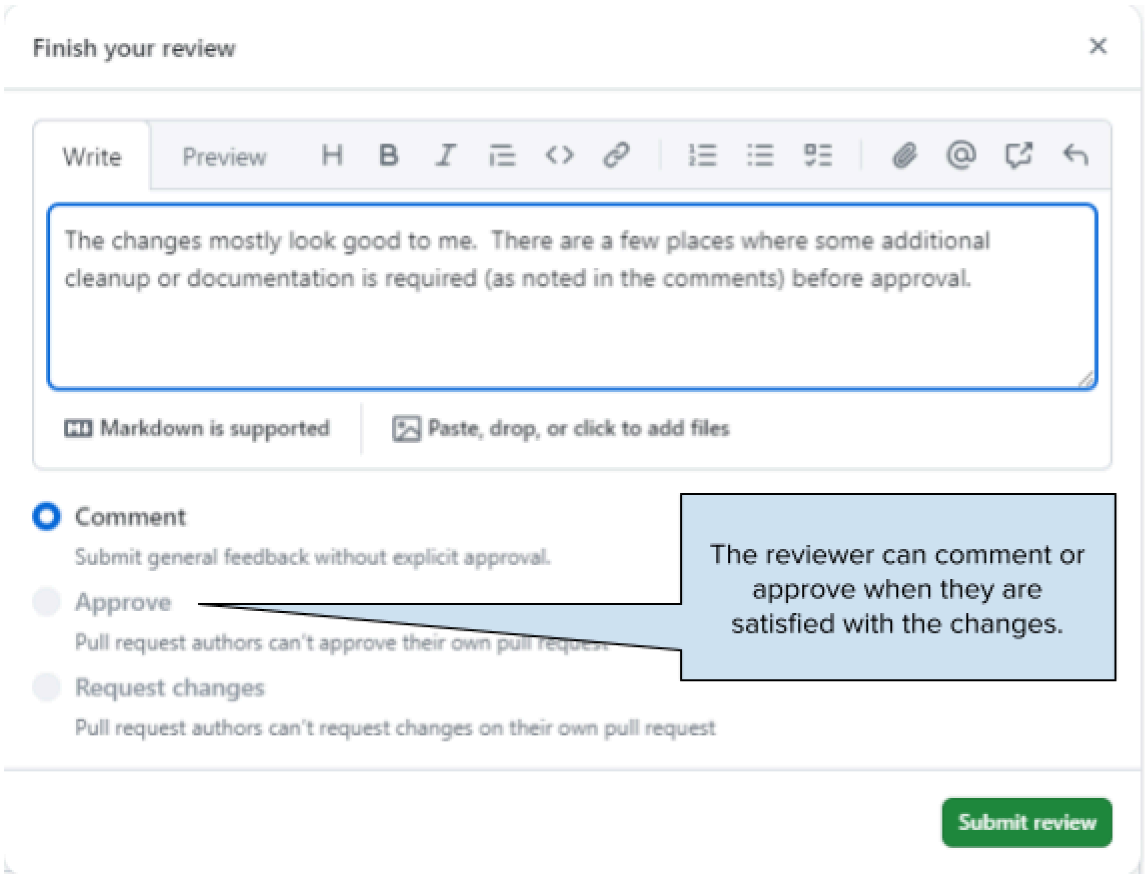
Testing

Before merging the PR, it's common practice to conduct thorough testing. This includes unit tests, integration tests, and any other relevant tests to verify that the changes work correctly and don't introduce regressions.

Approval

The changes in the PR can not be merged until at least one other developer has done a code review and approved the PR. This is enforced by the branch protection rules for the `dev` branch.

Once the reviewers are satisfied with the code changes and consider them ready for integration into the development branch, they approve the PR. The approval often takes the form of a comment or a specific action in GitHub.


Finish your review

The changes mostly look good to me. There are a few places where some additional cleanup or documentation is required (as noted in the comments) before approval.

[✓ Comment] Submit general feedback without explicit approval.

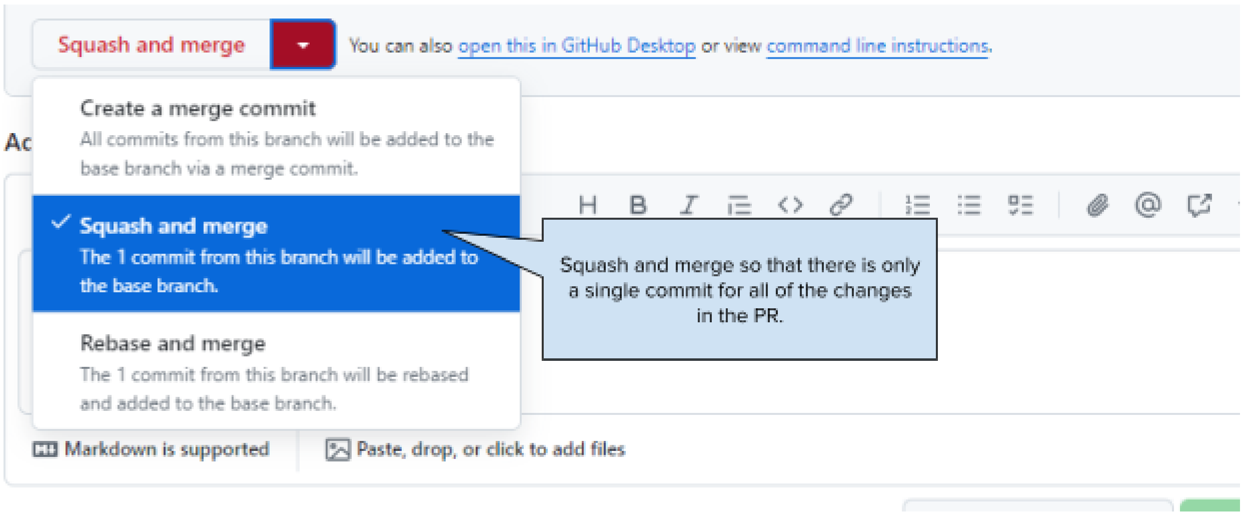
Approve — Pull request authors can't approve their own pull request

Request changes — Pull request authors can't request changes on their own pull request

The reviewer can comment or approve when they are satisfied with the changes.

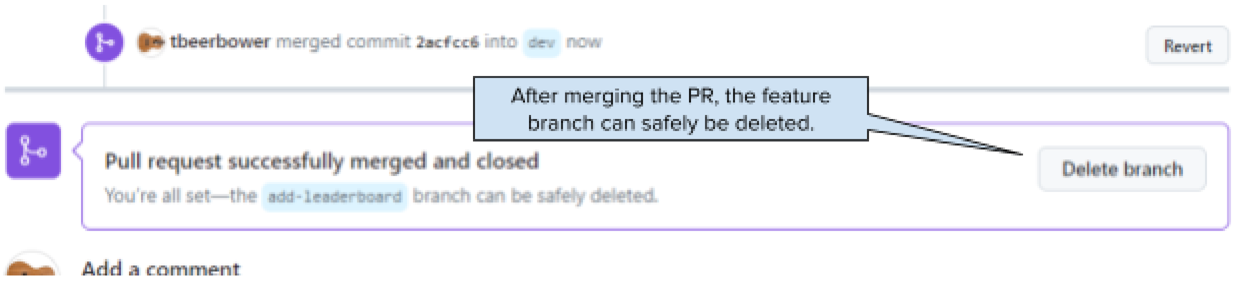
Merging

With the PR approved, the changes are merged into the development branch `dev`. This action incorporates the new feature or bug fix into the project.


Squash and merge
You can also open this in GitHub Desktop or view command line instructions.
Create a merge commit
All commits from this branch will be added to the base branch via a merge commit.
✓ Squash and merge
The 1 commit from this branch will be added to the base branch.
Rebase and merge
The 1 commit from this branch will be rebased and added to the base branch.
Markdown is supported
Paste, drop, or click to add files

Squash and merge so that there is only a single commit for all of the changes in the PR.

After merging you should delete the feature branch.



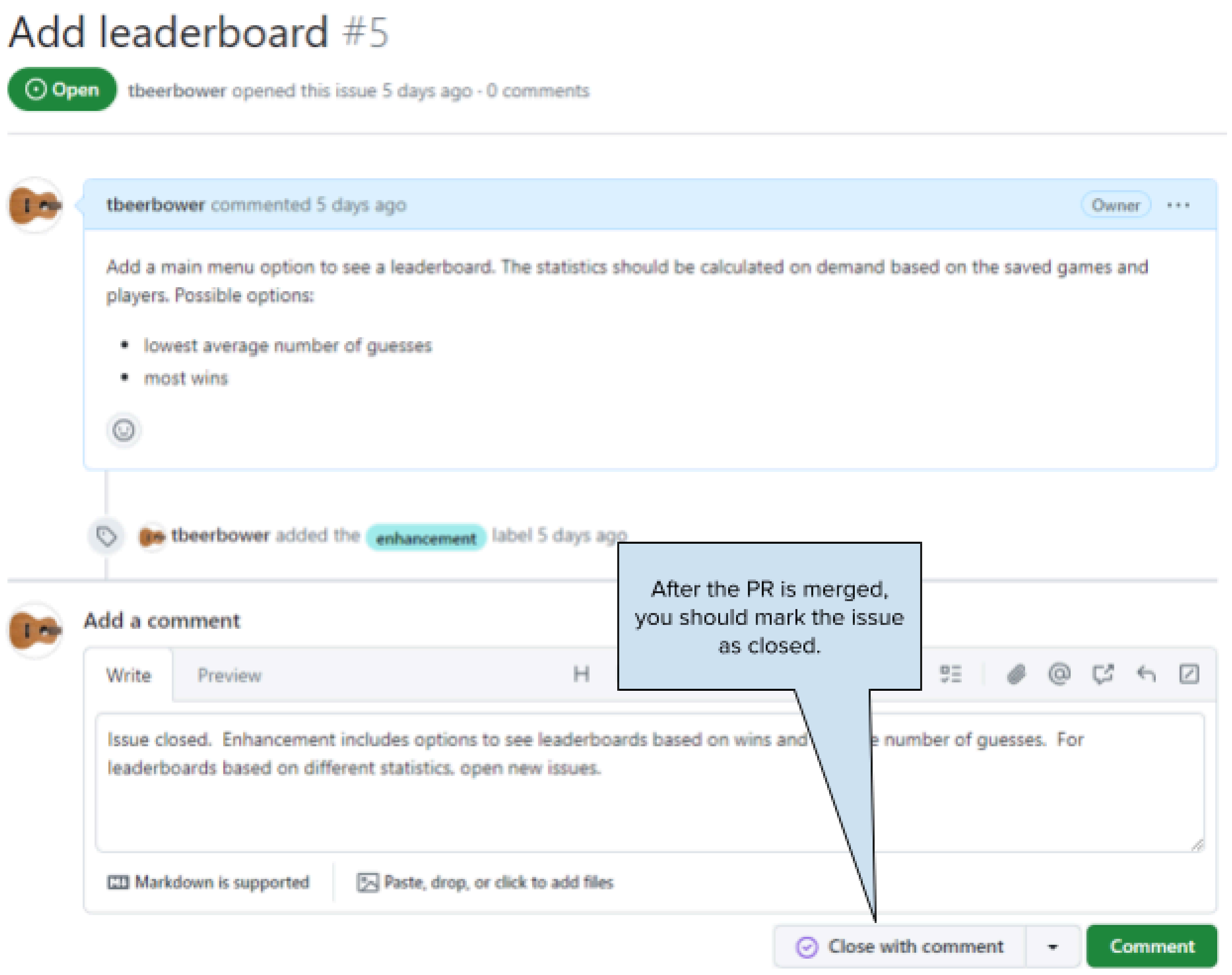
After merging the PR, the feature branch can safely be deleted.

Pull request successfully merged and closed

You're all set—the `add-leaderboard` branch can be safely deleted.

Issue Closure

Upon successful merging, the issue that initiated the process is typically marked as "closed" or "resolved" in the GitHub repository, indicating that the work has been completed.



Add leaderboard #5

Open tbeerbower opened this issue 5 days ago · 0 comments

tbeerbower commented 5 days ago

Add a main menu option to see a leaderboard. The statistics should be calculated on demand based on the saved games and players. Possible options:

- lowest average number of guesses
- most wins

tbeerbower added the enhancement label 5 days ago

Add a comment

Issue closed. Enhancement includes options to see leaderboards based on wins and [illegible] number of guesses. For leaderboards based on different statistics, open new issues.

Owner
Write
Preview
Markdown is supported
Paste, drop, or click to add files
Close with comment
Comment
After the PR is merged, you should mark the issue as closed.