SF assignment 4

Group 5
Julia Potrus
Erik Betzholtz
Rasmuss Per Brodin
Vincent Lagerros

March 5, 2025

1 Project

Name: sir-lancebot

URL: https://github.com/python-discord/sir-lancebot

The project is a discord bot with fun features, such as text-based video games such as tic tac toe. It's code is written in python.

2 Onboarding experience

Did you choose a new project or continue on the previous one?

We chose to work on a new project. The main cause for this is that the previous one was not active, making features and issues outdated and irrelevant. We also thought that the previous project was not very fun so we decided to choose a project that we would find fun and encouraging to work with. The bot project had great issues and seemed to tick our other criteria such as being written in a language that all group members were comfortable with (Python).

If you changed the project, how did your experience differ from before?

Compared to our previous project where we already had all prerequisites installed from Java, this project required a lot of dependencies to run properly. The docs recommended us to use Gitpod to fix this by using a remote PC. This was quite a hassle and took many tries and hours of trying to make it work (since we never had used it before).

We then tried using it locally to see if that would make it less painful, however it required:

1. The exact correct python version installed, otherwise it would throw a random error.

2. A Python packaged manager called poetry, as otherwise the dependencies used by Python would not work in some weird way because of the wrong version. This also required several commands to set up that where not properly documented in the on-boarding experience.

3. Correct environment variables, as it would just throw a weird, and unrelated error if they were not properly configured.

Because this also required a bot, we had to create both a new discord group, and a discord bot account. Additionally they had a very strict pre-commit hook, that required us to document every function and class as well as keeping each line under a specified length. This also caused some issues as we had little experience with this sort of system.

3 Effort spent

We mainly worked using live share in vs code. This made us able to work at the same time and also discuss and evaluate our choices together while writing the code. We also went to lab sessions together where we tried to set up and configure.

For each team member, how much time was spent in:

1. plenary discussions/meetings; 25/02/2025 1h

2. discussions within parts of the group;
03/03/2025 1h
04/03/2025 1h
05/03/2025 2h

3. reading documentation;
03/03/2025 30 min
04/03/2025 2h
05/03/2025 1h

4. configuration and setup;
27/02/2025 3h (Rasmuss and Julia)
04/03/2025 2h (Vincent and Erik)

5. analyzing code/output;
28/02/2025 2h

04/03/2025 1h

6. writing documentation;
03/03/2025 30 min

7. writing code;
27/02/2025 2h
03/03/2025 3h

8. running code;
03/03/2025 1h

For setting up tools and libraries (step 4), enumerate all dependencies you took care of and where you spent your time, if that time exceeds 30 minutes.

The setup was making sure that the pod could run on Discord. This was done in two ways. The first was to configure it using Gitpod, the second was making it work locally. Lastly, we also spent time making the tests run by writing "poetry run task test" in the terminal. Since the project did not have any prior tests, this was not already working. For more details read the **Onboarding experience** section.

4 Overview of issue(s) and work done.

Title: Add a Wordle command

URL: https://github.com/python-discord/sir-lancebot/issues/1034

Summary: We implemented the Wordle command for the bot, the game can be played by running `.wordle (Optional)word_length (Optional)num_of_guesses`

- Collect valid words from large text file `big.txt` and filter by length for faster response time.
- Implemented the game logic, the letter is gray if it is not in the word, yellow if it is in the word but in the incorrect place, and green for correct letters in the correct places. Only valid English words are guessable.
- Created a discord embed that updates according to the game's current state. This allows the user to see past guesses and their outcome and remaining tries.
- Created a testing framework that was missing in the repository, and added unit tests for the Wordle game. Tests can now be run directly using `poetry` with the command `poetry run task test`

Files Affected:

- wordle.py
- test_wordle.py
- pyproject.toml
- big.txt

5 Requirements for the new feature or requirements affected by functionality being refactored

1. Length And Guess: The user can choose the length of the word to guess and the number of guesses. The default is a five letter word and six guesses. This is implemented by the user entering the specifications of the game along with the command to start the game.

2. Words: The user can guess words, however they must be in the dictionary and of the given length.

3. Bot Response: The bot responds to each guess with a color-coded response showing all previous guesses and how close they are to the correct word. This is in accordance with the standard rules for Wordle.

4. Colors: The color green indicates the correct placement of the letter in the word. The color yellow indicates that the letter exists in the word but is in the wrong spot. No color indicates that the letter is not in the word at all.

5. Exit Game: The user should be able to exit the game at any time. If the user does not write a guess in 60 seconds, the game will automatically exit.

For this assignment we only used unit tests, meaning we could only test functions and we could therefore not do end-to-end testing. So we tested 2, 3, and 4 as they are separate functions, while the rest required user interaction.

6 Code changes

6.1 Patch

We have not changed anything in the existing files of code, we have only added new files and our own code in those.

Link to Patch

7 Test results

There was no testing framework or tests in the code-base before we added it. As such we added unit testing to a few functions we used in the bot. We also tried to add end-to-end testing by using a mock bot and user, however after spending a considerable amount of time on it we were unable to get it to work. As such we only tested the core functionality of the bot, by adding 6 test functions for a total of 18 test cases. This helped us find, and fix some bugs on some edge cases we did not find by testing manually.

8 UML class diagram and its description

8.1 Key changes/classes affected

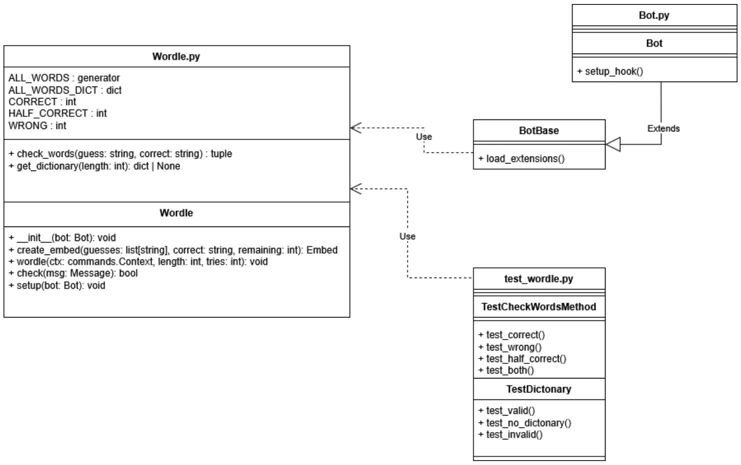

Figure 1: UML.

9 Overall experience

What are your main take-aways from this project? What did you learn?

This project was very fun, we learned more about Gitpod and what it is like to contribute to an open source project. It was much more rewarding to see an actual feature be added that we could use than ”more tests passed”. A lot of the work is simply understanding the previous code, and setting up the project to run. We also learned more about how to work as a group and that communication is key to a great teamwork!

How did you grow as a team, using the Essence standard to evaluate yourself?

We have developed as a team with each task, during this task we agree that the team was in the Performing state. The main difference between the last assignment where the team was evaluated to still be in the Collaborating state was that we focused on eliminating waste continually by working together via discord voice chat and coding live share. By working at the same time as the other team members and being able to discuss concerns and questions directly waste was continually eliminated. The team also produced a high-quality result (a fully functional Wordle game) for the discord bot.

As this was the last assignment the team will be working on together it can however be more accurate to describe the current state as Adjourned. The team is no longer accountable, and all members are available for new projects. There is no one to hand over responsibility to, except maybe, other collaborators working on the open-source project that the team worked on.