Week 10 Code Review

Overview

This code review was a peer review and was performed on 12 May 2023 .

The file being reviewed is NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py. The review was not of the whole file, but just certain methods of the StateManager class; the following methods were reviewed:

- handle_word_received
- check_stop_request_received
- respond_to_demonstration
- respond_to_demonstration_with_full_word
- publish_word
- wait_for_shape_to_finish
- respond_to_test_card
- stop_interaction
- start_interaction
- wait_for_word
- wait_for_feedback
- wait_for_robot_to_connect

Below is the completed peer code review template, which outlines the comments pointed out by reviewers. After the completed template, some reflections on the peer review and a comparison to ChatGPT code review is included.

Code Review Completed Template

ORGANISING YOUR CODE REVIEW (A FEW DAYS /HOURS BEFORE MEETING)	
Name and local of artifact (on Github) to be reviewed:	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py - link here: https://github.com/COMP90082-2023-SM1/NA-Boxjelly/blob/main/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py
What to be reviewed?	Methods (class StateManager): handle_word_received, check_stop_request_received, respond_to_demonstration, respond_to_demonstration_with_full_word, publish_word, wait_for_shape_to_finish, respond_to_test_card, stop_interaction, start_interaction, wait_for_word, wait_for_feedback, wait_for_robot_to_connect
When's the code review meeting happening:	11am-12pm Friday 12/05/2023
Reviewers:	Jun Li Chen, Chang Shen, Tianyang Wang

CODE REVIEW MEETING (STARTING THE MEETING)								
Team:	NA-Boxjelly							
Date:	5/12/2023							
Time:	11am-12pm							
Facilitator:	Joel Towell							
Reviewers:	Jun Li Chen, Chang Shen, Tianyang Wang							

CODE REVIEW (DURING MEETING TIME)

Item	Artifact (on GitHub)	Location (where the issue was found in the reviewed artifact?)	Severity	Type	Defects Category	Description	Fixed by the author?	Verified by the Moderator?
1	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method handle_word_received	Trivial	Improvement	Documentation Defects (Comment)	Docstring for method runs over 72 characters, and is thus not PEP 8 compliant	Yes	Yes
2	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method respond_to_demonstration	Trivial	Improvement	Documentation Defects (Comment)	Docstring for method runs over 72 characters, and is thus not PEP 8 compliant	Yes	Yes
3	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method respond_to_demonstration_with_full_word	Trivial	Improvement	Documentation Defects (Comment)	Docstring for method runs over 72 characters, and is thus not PEP 8 compliant	Yes	Yes
4	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method respond_to_demonstration_with_full_word	Trivial	Improvement	Documentation Defects (Comment)	Redundant comment about displaying word	Yes	Yes
5	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method publish_word	Trivial	Improvement	Documentation Defects (Comment)	Docstring for method runs over 72 characters, and is thus not PEP 8 compliant	Yes	Yes
6	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method publish_word	Trivial	Improvement	Documentation Defects (Comment)	Redundant comment about instance variables	Yes	Yes
7	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method publish_word	Trivial	Improvement	Visual Representation Defects (Long Line)	Method signature runs over 79 characters, not PEP 8 compliant	Yes	Yes
8	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method wait_for_shape_to_finish	Trivial	Improvement	Visual Representation Defects (Long Line)	Method signature runs over 79 characters, not PEP 8 compliant	Yes	Yes

9	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method wait_for_shape_to_finish	Trivial	Improvement	Documentation Defects (Comment)	Docstring for method runs over 72 characters, and is thus not PEP 8 compliant	Yes	Yes
10	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method respond_to_test_card	Trivial	Improvement	Documentation Defects (Comment)	Docstring for method runs over 72 characters, and is thus not PEP 8 compliant	Yes	Yes
11	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method respond_to_test_card	Trivial	Improvement	Visual Representation Defects (Long Line)	Method signature runs over 79 characters, not PEP 8 compliant	Yes	Yes
12	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method start_interaction	Trivial	Improvement	Documentation Defects (Comment)	Docstring for method runs over 72 characters, and is thus not PEP 8 compliant	Yes	Yes
13	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method start_interaction	Trivial	Improvement	Visual Representation Defects (Long Line)	Method signature runs over 79 characters, not PEP 8 compliant	Yes	Yes
14	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method wait_for_word	Trivial	Improvement	Visual Representation Defects (Long Line)	Method signature runs over 79 characters, not PEP 8 compliant	Yes	Yes
15	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method wait_for_word	Trivial	Improvement	Documentation Defects (Comment)	Docstring for method runs over 72 characters, and is thus not PEP 8 compliant	Yes	Yes
16	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method wait_for_feedback	Trivial	Improvement	Documentation Defects (Comment)	Docstring for method runs over 72 characters, and is thus not PEP 8 compliant	Yes	Yes
17	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method wait_for_feedback	Trivial	Improvement	Visual Representation Defects (Long Line)	Method signature runs over 79 characters, not PEP 8 compliant	Yes	Yes

18	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method wait_for_robot_to_connect	Trivial	Improvement	Visual Representation Defects (Long Line)	Method signature runs over 79 characters, not PEP 8 compliant	Yes	Yes
19	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method wait_for_robot_to_connect	Trivial	Improvement	Documentation Defects (Comment)	Docstring for method runs over 72 characters, and is thus not PEP 8 compliant	Yes	Yes
20	NA-Boxjelly/src/cowriter_letter_learning/letter_learning_interaction/nodes/learning_words_nao.py	method wait_for_robot_to_connect	Medium	Investigate	Structure Defects (Dead Code)	If True means else block will never execute, so code inside the else block is dead code	Yes	Yes
END OF CODE REVIEW MEETING								
Number of severe/critical errors:		0						
Number of medium errors:		1						
Number of trivial errors:		19						
Total inspection time (hs):		1 hour						

Reflections on Peer Code Review

There are two points to make about the faults pointed out in the code review:

1. 19/20 identified flaws were stylistic in nature; they mostly point out violations of PEP 8 guidelines, and in a couple of cases note redundant comments.
2. One of the identified flaws was a logic error, but a fairly minor one; there was dead code, but not strictly incorrect behaviour.

Going through the code review process with peers was useful in several ways. First and foremost, it helped improve the code by drawing attention to a number of stylistic faults, as well as a logic error which caused a dead code block. Without the scrutiny of code review, some or all of these faults may have made it into the final product, so this review was definitely a fruitful exercise from a quality assurance point of view. Additionally, the code we reviewed does quite a lot of work in the application we are working on; reviewing these methods provided team members who were not involved in writing the code additional insight into how some key logic of the program works. Team insight into parts of the application that they did not write is valuable, and can help them more easily write code which is compatible with the code under review.

In summary, the benefit of review was two-fold: it improved the quality of the code under review by drawing attention to faults, and it made the rest of the team become familiar with some important code in the application.

Comparison to ChatGPT Code Review

In the week 8 code review, it was noticed that ChatGPT failed to pick up on a number of PEP 8 violations. Interestingly, the majority of the faults pointed out by peer reviewers related to violations of PEP 8. This suggests that human reviewers may be more sensitive to stylistic violations than ChatGPT, particularly in relation to line length. (LLM's like ChatGPT may be unable to reliably count characters per line; it is not obvious that a model designed to predict the next token in a sentence can count characters per line.) There were no logic errors in the week 8 code review, so it is difficult to compare ChatGPT vs peers in this respect.

Of the benefits of peer review noted in the reflection above, only one of them relates to improving the code under review. The other benefit is that it makes other team members have a better understanding of parts of the code which they did not directly work on. This is a benefit which code review with ChatGPT clearly cannot deliver, at least not to the extent of peer code review. (Summary generated by ChatGPT could be read by others, but this is obviously less informative than reviewing the code itself.)

There is one respect in which ChatGPT is clearly superior to peer review, time. ChatGPT reviews code in seconds, the peer review involved a time commitment from all reviewers to read the code in advance, then took an hour of the reviewers and facilitators time. This is a considerable drain on resources compared to automated code review. For simpler and less important functionality, using ChatGPT for code review can increase productivity by saving lots of time which would be spent on peer review. However, when the code is important, complex, and perhaps when has a particular concern about stylistic issues, peer review offers clear benefits over ChatGPT review, and if the time can be spared, it is probably worth undertaking a detailed peer review of the code in question.