Pendream: An Interactive Spelling Practice System with an OCR Model

Project Members:

Haiwei Du, Linwei Wu, Yiran Wang

Project Description

Pendream is an English vocabulary study application tailored for beginners who want to assess and improve their spelling skills. The project utilizes a curated set of 1000 common vocabulary words to create interactive “fill-in-the-blank” style questions. Words are partially masked, and users guess the missing parts based on definitions or contextual hints. This approach encourages both recall and application of vocabulary in an engaging manner.

For instance, the word "apple," with the definition "fruit with red or yellow or green skin and sweet to tart crisp whitish flesh," might appear as "a___e." The user fills in the blanks with "ppl" on a virtual writing board. The tool evaluates their input, providing instant feedback and a comprehensive report after completing all questions. To enhance learning, users can select their difficulty level and practice specific parts of speech, such as nouns, verbs, or adjectives/adverbs, or opt for a mixed mode for random challenges.

Class Diagram.

See at the end of the document.

Test coverage reports

Coverage report: 99%

Files | Functions | Classes

coverage.py v7.6.8, created at 2024-12-03 22:09 -0500

File ▲	statements	missing	excluded	coverage
FillInTheBlank.py	56	0	0	100%
interface.py	130	5	0	96%
OCR_model.py	13	0	0	100%
Question_set.py	45	0	0	100%
Question.py	12	1	0	92%
test_FillInTheBlank.py	50	0	0	100%
test_interface.py	99	0	0	100%
test_OCR_model.py	7	0	0	100%
test_Question_set.py	52	0	0	100%
test_Vocab.py	59	0	0	100%
test_writing.py	54	0	0	100%
Vocab.py	42	0	0	100%
writing.py	34	0	0	100%
Total	**653**	**6**	**0**	**99%**

coverage.py v7.6.8, created at 2024-12-03 22:09 -0500

main.py	67	12	0	82%

The reason why we cannot reach 100% coverage for interface.py, Question.py, and Main.py is complicated. First, the reason why the __init__ function of main.py and the load_next_question function in interface.py do not have automated tests: This is because my Qt interface uses a dropdown menu, which can only be generated after the __init__ function has been executed. Testing this menu leads to an access violation error, as the pytest file encounters insufficient memory access permissions. The reason the latter function cannot be tested is that the question generation function generates questions purely at random, making it impossible to establish precise and specific test cases.

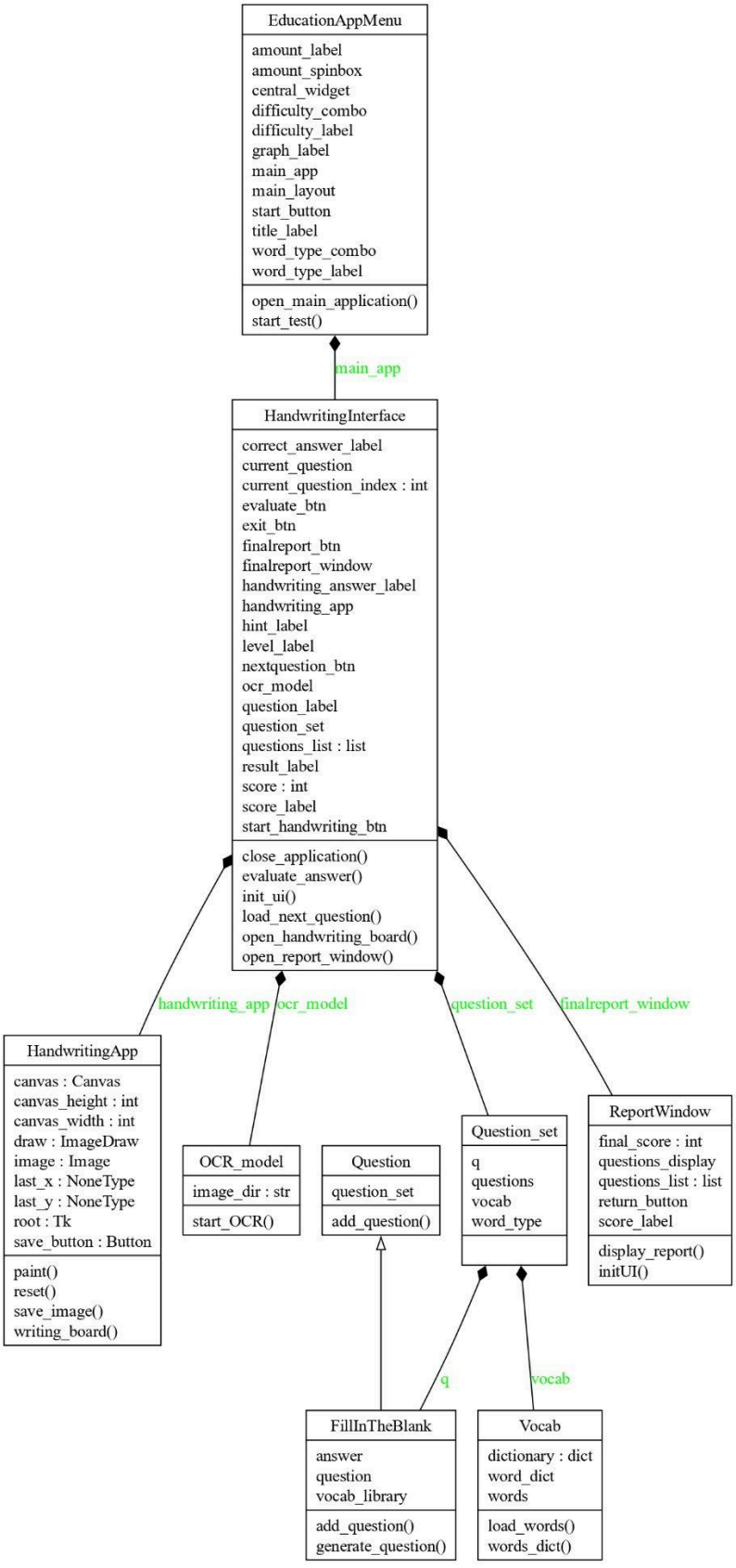
Secondly, for the "No more questions available." branch in interface.py that cannot be tested, the reason is that this message only appears when the question list reaches its end. However, the number of questions in the list (i.e., the number of times the "next question" button can be pressed) is not determined by this class, making it untestable.

Third, the reason why question.py does not achieve full coverage is that it contains an abstract method which cannot be covered by tests.

Nevertheless, I have conducted manual testing on these functions, including as many comprehensive corner cases as possible and cover all branches and statements. I believe that these two functions and the class they belong to have been thoroughly refined.

References

[1]Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., & Wei, F. (2021). TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. arXiv preprint arXiv:2109.10282. https://arxiv.org/abs/2109.10282
English Dictionary https://github.com/nightblade9/simple-english-dictionary/tree/main
Vocabulary Reference
https://www.scribd.com/document/287974164/1-1000-Word-Little-Language-Vocab-List


EducationAppMenu
amount_label
amount_spinbox
central_widget
difficulty_combo
difficulty_label
graph_label
main_app
main_layout
start_button
title_label
word_type_combo
word_type_label
open_main_application()
start_test()
main_app
HandwritingInterface
correct_answer_label
current_question
current_question_index : int
evaluate_btn
exit_btn
finalreport_btn
finalreport_window
handwriting_answer_label
handwriting_app
hint_label
level_label
nextquestion_btn
ocr_model
question_label
question_set
questions_list : list
result_label
score : int
score_label
start_handwriting_btn
close_application()
evaluate_answer()
init_ui()
load_next_question()
open_handwriting_board()
open_report_window()
handwriting_app
ocr_model
question_set
finalreport_window
HandwritingApp
canvas : Canvas
canvas_height : int
canvas_width : int
draw : ImageDraw
image : Image
last_x : NoneType
last_y : NoneType
root : Tk
save_button : Button
paint()
reset()
save_image()
writing_board()
OCR_model
image_dir : str
start_OCR()
Question
question_set
add_question()
Question_set
q
questions
vocab
word_type
ReportWindow
final_score : int
questions_display
questions_list : list
return_button
score_label
display_report()
initUI()
q
vocab
FillInTheBlank
answer
question
vocab_library
add_question()
generate_question()
Vocab
dictionary : dict
word_dict
words
load_words()
words_dict()