CS5001 Project 5 Report

Shang Xiao

February 22, 2022

1 Problem Description

1.1 The spaceman game is an exciting old game that has been found for years. In the beginning, the game was more of a "solving a puzzle" type of thing, but then it became more complicated as the programming field has been evolving throughout the years.

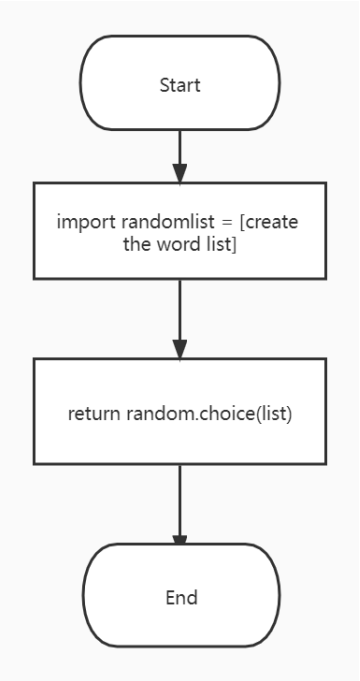
The version we are doing this week is an actual implementation of the spaceman game, each user who gets to play the game is allowed six wrong attempts maximum before they can finally uncover the hidden word. The sample size requirements are a list of 20 words between 5 and 15 characters long.

We want to start by randomly choosing a word from the 20-word list and then hide it with symbols like "*" or "-" while indicating the length of the word. Next, we are going to prompt the user to enter one character at a time, when the user had uncovered a character in the random word, then we remind them that they had found a character and change the hidden word to a combination of the uncovered characters and the remaining hidden characters. For example, for the word "apple", if the user has found "a" and "p", we need to change the hidden word from "*****" to "app**".

Lastly, we need to update the user every time they have guessed a character and prompt them if they had won or lost after six wrong attempts.

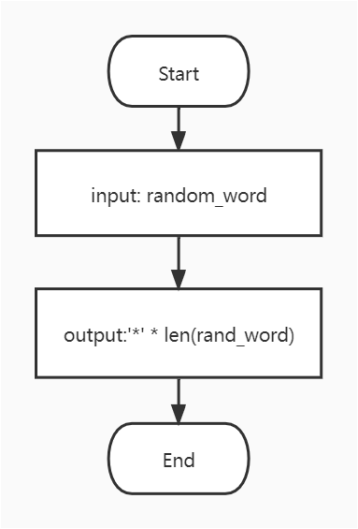
2 Activity Diagrams

2.1 The first function pickRandomWord() picks a random word from a list made of 20 words between 5 and 15 characters long.



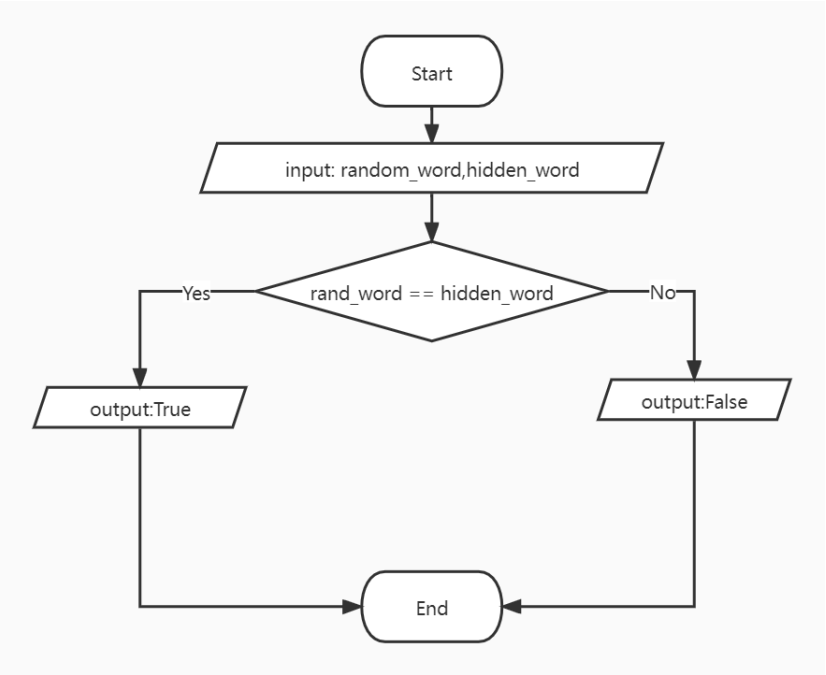
Flowchart for pickRandomWord()

2.2 The second function createHiddenWord() replace the random word using "*" indicating its length.



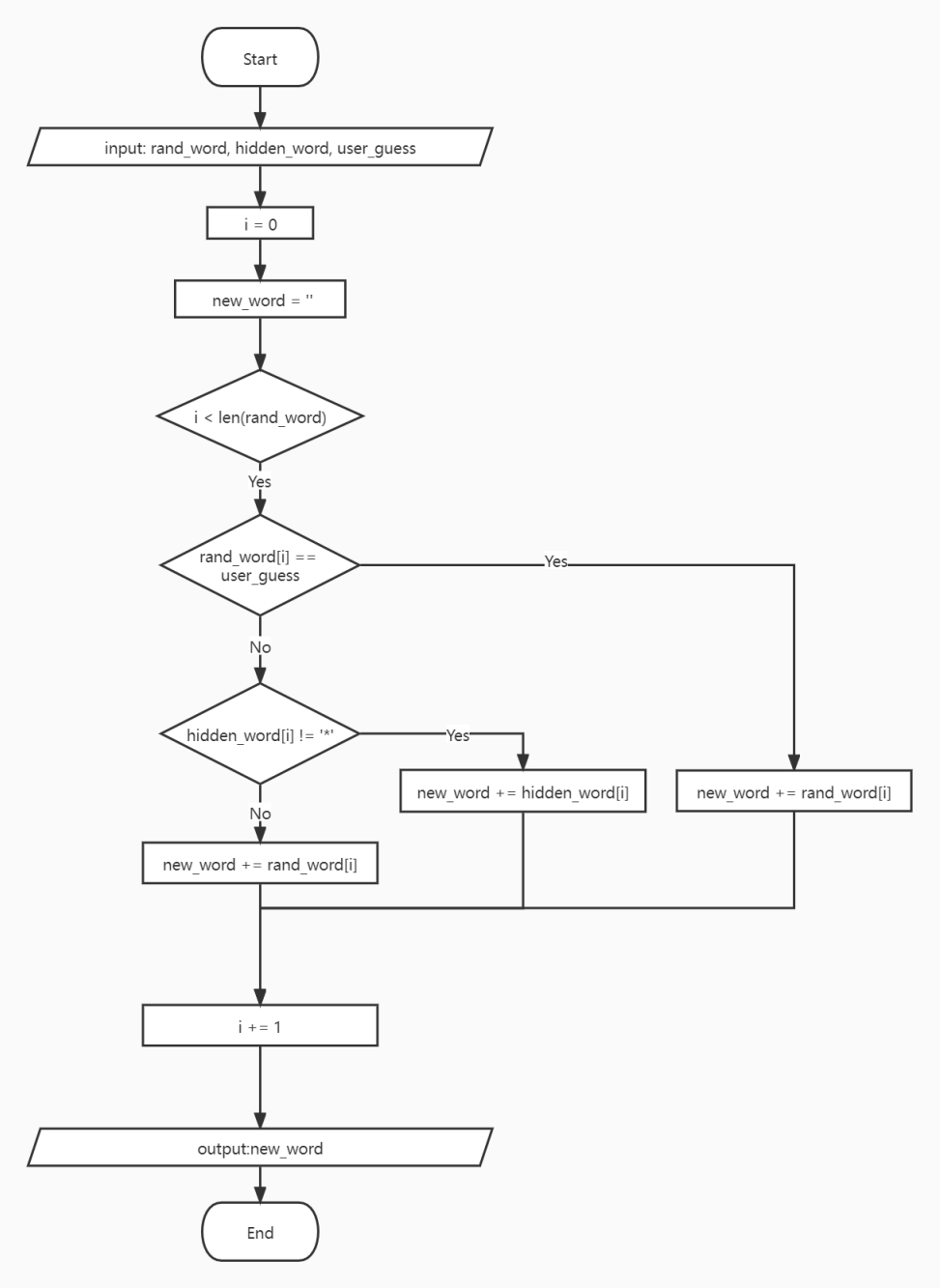
Flowchart for createHiddenWord()

2.3 The third function wordFound(randWord, hiddenWord) takes two parameters randWord and hiddenWord to deteect whether the random word has been uncovered.



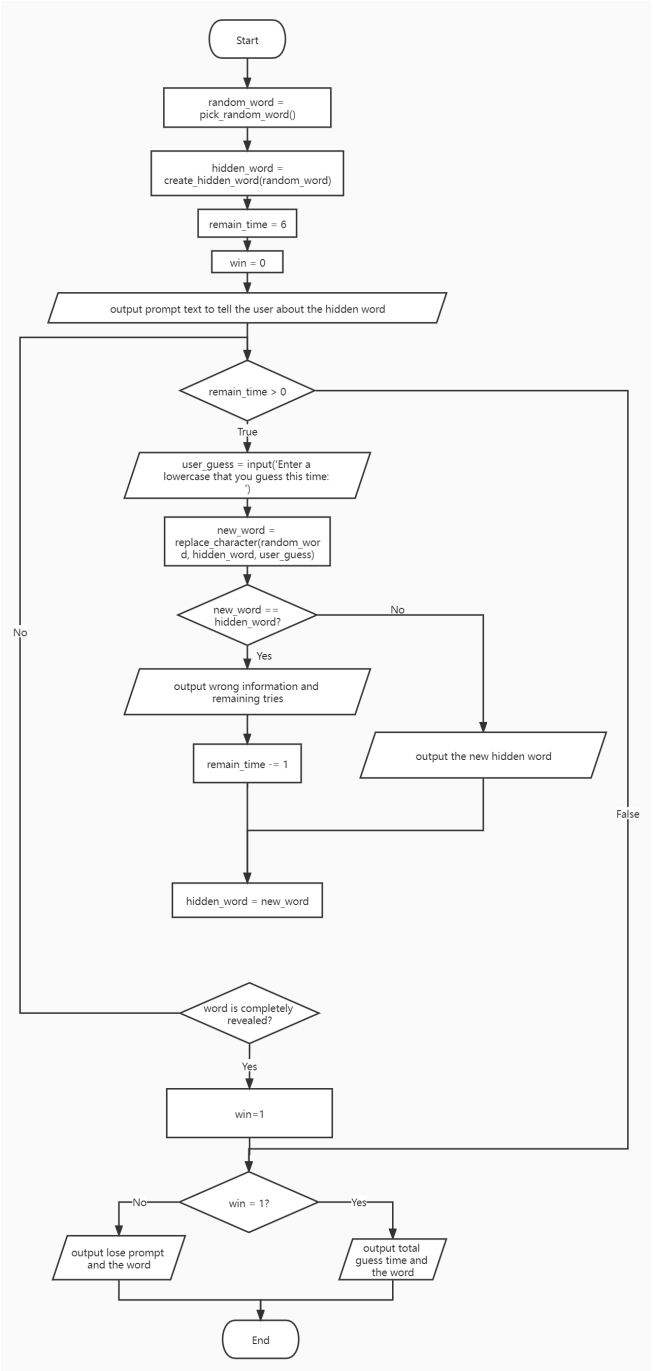
Flowchart for wordFound(randWord, hiddenWord)

2.4 The fourth function replaceCharacter(randWord, hiddenWord, userGuess) replaces all the occurences of userGuess in hiddenWord with userGuess.



Flowchart for replaceCharacter(randWord, hiddenWord, userGuess)

2.5 The main() function integrates all four functions. Starting with a brief introduction of the game to make it more user-friendly, next we prompt the user with the hiddenWord replaced by "*" indicating its length and start asking the user for their first attempt. Repeat this process until the user has found/failed to find the random word, and lastly, prompt the user of their end game result.



Flowchart for main()

3 Reflection

3.1 The spaceman game is an excellent opportunity to enhance our experiences with the string object and while loops. At the same time, the project smoothly makes the transition into lists.

One of the essential takeaways from this project is to structure my logic. For example, when figuring out how to update the newWord after the user uncovered a new character from the random word, I spent hours searching YouTube and Geeksforgeeks.com.

In the end, I was able to combine my research with my logical thinking to build this part of the structure into my code, which when the newWord is not equal to the most recent hiddenWord, that means the user must have found something, or it will be equal, using this logic I wrote my "if-else" conditions in my while loop condition and finialized the project.

4 Acknowledgements

4.1 Website consulted (lecture notes and documentations can be found in these links):

`https://www.geeksforgeeks.org/python-list/?ref=gcse` - Documentations for lists implementations, as well as how to write a list.

`https://docs.python.org/3/library/random.html` - Python documentation for the random module, under "Function for sequences" it teaches how to use random.choice().

`https://northeastern.instructure.com/courses/102943` - Module videos from professor Maria Jump.

`https://tekhnologic.wordpress.com/2017/03/01/spaceman-an-alternative-to-hangman/#:~:text='Spaceman'%20is%20an%20alternative%20to,easily%20identifiable%20by%20its%20name.&text=If%20students%20guess%20a%20letter,is%20drawn%20on%20the%20board.` - in section "Using the template", number 17th bullet point, "With every incorrect guess, a new part of the spaceship is built. Click the build button nine times to complete the spaceship.", where it solved my question of how do I update the hiddenWord with newWord everytime after user uncovered a new character? - It turns out that, using this logic here, if newWord != hiddenWord, then the user must have uncovered something, which means they were correct on that guess.

4.2 Website used for flowchart: `https://www.processon.com/`

Website used for debugging codes and loops for each line execution:
`https://pythontutor.com/visualize.html#mode=edit`