Criss-Cross Multi-Step Word Guessing Game

Assignment 1
CSSE1001/7030
Semester 2, 2020
10 marks

Due Date: 4th September, 17:00

1 Getting Started

To start, download *a1.zip* from Blackboard and extract the contents. The *a1.zip* folder contains all the necessary files to start this assignment. Some support code has been provided to assist with implementing the tasks. You will be required to implement your assignment in *a1.zip*.

The other provided file is *a1_support.py*, which contains some code to help you implement your assignment. You are not compelled to use this file to implement your assignment but it is strongly recommended. Do not make changes to the *a1_support.py* file. The only file that you should submit is *a1.zip*. It could cause unexpected errors submit more than one file or if you made changes to the *a1_support.py* file as well.

Note: The functionality of your assignment will be marked by automated tests. This means that the output of each function, as well as the output of your overall program, must be exact to the specifications outlined below in the **Implementation section**. This includes whitespace, grammar, etc. If you add your own messages throughout the game, you will likely fail most of the automated tests. Some sample tests will be provided to give you an idea of whether your output differs from the expected output. These tests are not necessarily the complete set of tests that will be used in marking, so if you pass all sample tests you are not necessarily guaranteed to achieve full functionality marks. However, if you do not pass all sample tests, you are guaranteed not to achieve full functionality marks.

2 Concepts

At the start of the game the user selects a difficulty, and then a word is chosen at random based on that difficulty. The word length will depend on the difficulty selected by the user; either "**FIXED**" (meaning the word will be exactly eight letters long), or "**ARBITRARY**" (meaning the word will be anywhere between six to eight letters long). The goal of the game is for the player to guess that word through a series of guesses of "subwords". The player will have a different number of guesses (with different subwords to guess; see *GUESS_INDEX_TUPLE*), depending on the difficulty selection. If the player chooses the "**FIXED**" difficulty, the word must be selected at random from the **WORDS_FIXED.txt** file. If the player chooses the "**ARBITRARY**" difficulty, the word must be selected at random from the **WORDS_ARBITRARY.txt** file.

At program startup, the user is asked to specify one of three actions:

Input	Action
’s’	Start game
’h’	The game rules will be printed out and then the game will commence
’q’	Quit game
Invalid command	Print invalid command message (see below) and restart. “Please enter a valid command: “

Table 1: List of valid actions.

When the game is started the player should be prompted with:

```
>>> "Do you want a "FIXED" or "ARBITRARY" length word?"
```

2.1 If the user specifies “FIXED”

The game randomly selects an eight-letter word from the **WORDS.txt** file and the correct guess sequence from the *GUESS_INDEX_TUPLE* tuple.

Each vowel guessed in the correct position gets 14 points. Each consonant guessed in the correct position gets 12 points. Each letter guessed correctly but in the wrong position within the substring gets 5 points. You can assume that the words do not contain repeated letters and all guesses are lowercase letters.

2.2 If the user specifies “ARBITRARY”

The game randomly selects a word from the **WORDS_ARBITRARY.txt** file and the correct guess sequence from the *GUESS_INDEX_TUPLE* tuple.

The scoring system is the same as for the ”**FIXED**” word length.

2.3 Guessing procedure

2.3.1 For an eight-letter word

The user will be prompted to guess the word, step by step. The guessing procedure involves 8 steps, where the guess slices will depend on the *GUESS_INDEX_TUPLE*. At each of the 7 first steps the user guesses a subsection of the word and receives feedback (their score) for that guess. The final 8th step involves guessing the whole word. After the 8th guess, the user is informed of whether they ‘won’ (i.e. guessed the word correctly) or ‘lost’ (in which case they are told what the word was). If, at any stage, the player enters a guess that is of the incorrect length then the game should repeatedly prompt for the correct word length until the player enters a guess that matches the length of the substring to be guessed in that step.

The guessing and scoring procedure is illustrated in Table 2, for the 8 letter word, ”crushing”.

Step no	Step by step letter guesses								Score
1	c	r							12+12=24
2		u	i	t					1*5=5
3					c	a	l	s	0
4				b	s	h			2*5=10
5				h	s	i	n		12+14+2*5=36
6						i	j	k	1*14=14
7			u	s	h	e	r	k	2*12+14=38
8	c	r	u	s	h	i	n	g	You win

Table 2: Step by step guessing procedure for the word "crushing".

2.4 Game over

If the user guesses the correct word at the end of the game, the following message should be printed out:

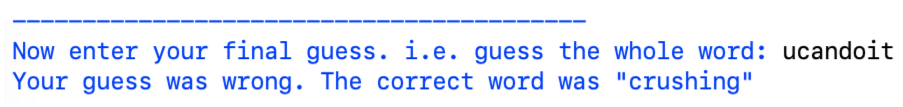
`"You have guessed the word correctly. Congratulations"`.

If the user guesses the wrong word at the end of the game, the following message should be printed out:

`"Your guess was wrong. The correct word was "{word}""`

(Where word represents the word the player was trying to guess.)

2.4.1 Examples



```
-----------------------------------------
Now enter your final guess. i.e. guess the whole word: ucandoit
Your guess was wrong. The correct word was "crushing"
```

Figure 1: Example of final guess.

3 Implementation

Within this section, the following variables hold meaning as defined below:

- `word_select`: A string representing a FIXED or ARBITRARY word selection.
- `guess_no`: An integer representing how many guesses the player has made.

- `word`: A string representing the word being guessed by the player.
- `word_length`: An integer representing the length of the word being guessed by the player.

You must write the following functions as part of your implementation. You are encouraged to add your own additional functions if they are beneficial to your solution.

`select_word_at_random(word_select)-> str:`
Given the `word_select` is either "FIXED" or "ARBITRARY" this function will return a string randomly selected from WORDS_FIXED.txt or WORDS_ARBITRARY.txt respectively. If `word_select` is anything other then the expected input then this function should return None.
Hint: see a1_support.load_words() and a1_support.random_index()

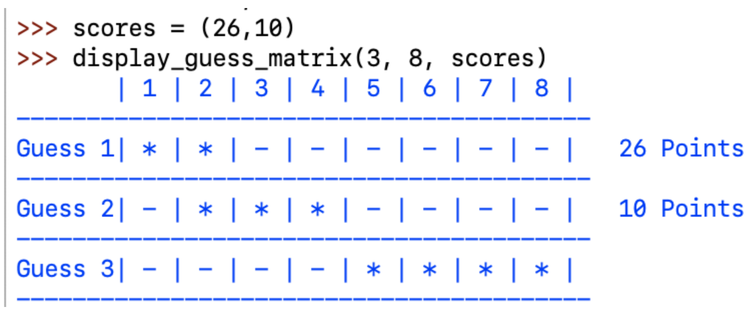
`create_guess_line(guess_no, word_length)-> str:`
This function *returns* the string representing the display corresponding to the guess number integer, `guess_no`.

Example:

```
>>> create_guess_line(2, 8)
'Guess 2 | - | * | * | * | - | - | - | - |'
```

`display_guess_matrix(guess_no, word_length, scores)-> None:`
This function *prints* the progress of the game. This includes all line strings for guesses up to `guess_no` with their corresponding `scores` (a tuple containing all previous scores), and the line string for `guess_no` (without a score).

Example:



```
>>> scores = (26,10)
>>> display_guess_matrix(3, 8, scores)
       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
-----------------------------------------
Guess 1| * | * | - | - | - | - | - | - |   26 Points
-----------------------------------------
Guess 2| - | * | * | * | - | - | - | - |   10 Points
-----------------------------------------
Guess 3| - | - | - | - | * | * | * | * |
-----------------------------------------
```

`compute_value_for_guess(word, start_index, end_index, guess)-> int:`
Return the score, an integer, the player is awarded for a specific guess. The `word` is a string representing the word the player has to guess. The substring to be guessed is determined by the `start_index` and `end_index`. The substring is created by slicing the word from the `start_index` up to and including the `end_index`. The `guess` is a string representing the guess attempt the player has made.

Example:

```
>>> compute_value_for_guess("crushing", 0, 1, "rc")
10
```

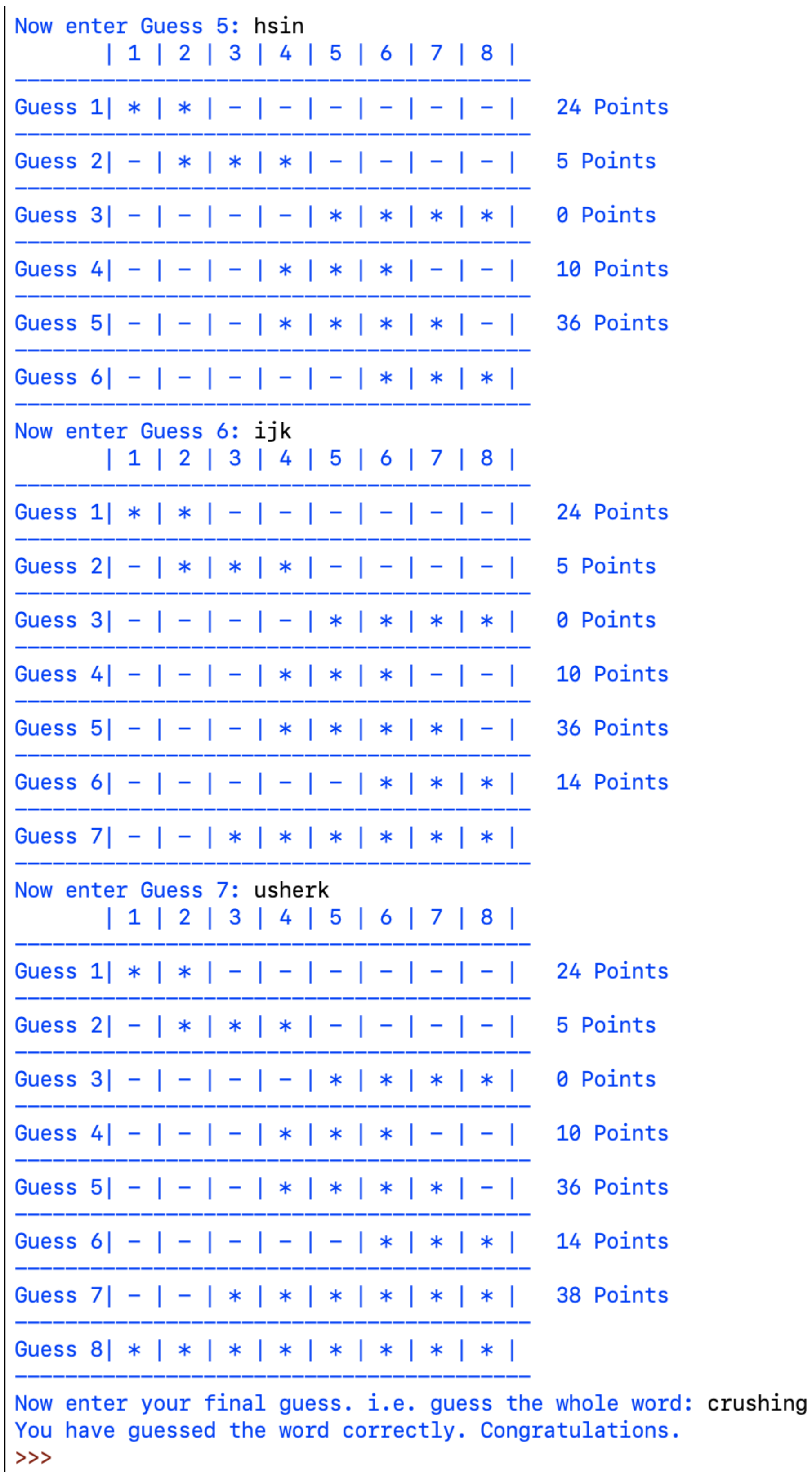
`main()-> None:`
This function handles player interaction. At the start of the game the player should be greeted with the **Welcome message**. Once the guessing sequence commences the game should loop for the correct number of rounds until either the player wins by guessing the correct word or loses by guessing the incorrect word.

Hint: the main function should be your starting point but also the last function you finish implementing.

3.1 Example game

```
Welcome to the Criss-Cross Multi-Step Word Guessing Game!


Enter an input action. Choices are:
s - start game
h - get help on game rules
q - quit game:
a

Please enter a valid command.


Enter an input action. Choices are:
s - start game
h - get help on game rules
q - quit game:
h

Game rules - You have to guess letters in place of the asterixis.
Each vowel guessed in the correct position gets 14 points.
Each consonant guessed in the correct position gets 12 points.
Each letter guessed correctly but in the wrong position gets 5 points.
If the true letters were "dog", say, and you guessed "hod",
you would score 14 points for guessing the vowel, "o", in the correct
position and 5 points for guessing "d" correctly, but in the
incorrect position. Your score would therefore be 19 points.
```

```
Do you want a 'FIXED' or 'ARBITRARY' length word?: Fixed
Do you want a 'FIXED' or 'ARBITRARY' length word?: FIXED
Now try and guess the word, step by step!!
       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
-----------------------------------------
Guess 1| * | * | - | - | - | - | - | - |
-----------------------------------------
Now enter Guess 1: cr
       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
-----------------------------------------
Guess 1| * | * | - | - | - | - | - | - |   24 Points
-----------------------------------------
Guess 2| - | * | * | * | - | - | - | - |
-----------------------------------------
Now enter Guess 2: uit
       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
-----------------------------------------
Guess 1| * | * | - | - | - | - | - | - |   24 Points
-----------------------------------------
Guess 2| - | * | * | * | - | - | - | - |   5 Points
-----------------------------------------
Guess 3| - | - | - | - | * | * | * | * |
-----------------------------------------
Now enter Guess 3: cal
Now enter Guess 3: cals
       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
-----------------------------------------
Guess 1| * | * | - | - | - | - | - | - |   24 Points
-----------------------------------------
Guess 2| - | * | * | * | - | - | - | - |   5 Points
-----------------------------------------
Guess 3| - | - | - | - | * | * | * | * |   0 Points
-----------------------------------------
Guess 4| - | - | - | * | * | * | - | - |
-----------------------------------------
Now enter Guess 4: bsh
       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
-----------------------------------------
Guess 1| * | * | - | - | - | - | - | - |   24 Points
-----------------------------------------
Guess 2| - | * | * | * | - | - | - | - |   5 Points
-----------------------------------------
Guess 3| - | - | - | - | * | * | * | * |   0 Points
-----------------------------------------
Guess 4| - | - | - | * | * | * | - | - |   10 Points
-----------------------------------------
Guess 5| - | - | - | * | * | * | * | - |
-----------------------------------------
```



```
Now enter Guess 5: hsin
       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
-----------------------------------------
Guess 1| * | * | - | - | - | - | - | - |   24 Points
-----------------------------------------
Guess 2| - | * | * | * | - | - | - | - |   5 Points
-----------------------------------------
Guess 3| - | - | - | - | * | * | * | * |   0 Points
-----------------------------------------
Guess 4| - | - | - | * | * | * | - | - |   10 Points
-----------------------------------------
Guess 5| - | - | - | * | * | * | * | - |   36 Points
-----------------------------------------
Guess 6| - | - | - | - | - | * | * | * |
-----------------------------------------
Now enter Guess 6: ijk
       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
-----------------------------------------
Guess 1| * | * | - | - | - | - | - | - |   24 Points
-----------------------------------------
Guess 2| - | * | * | * | - | - | - | - |   5 Points
-----------------------------------------
Guess 3| - | - | - | - | * | * | * | * |   0 Points
-----------------------------------------
Guess 4| - | - | - | * | * | * | - | - |   10 Points
-----------------------------------------
Guess 5| - | - | - | * | * | * | * | - |   36 Points
-----------------------------------------
Guess 6| - | - | - | - | - | * | * | * |   14 Points
-----------------------------------------
Guess 7| - | - | * | * | * | * | * | * |
-----------------------------------------
Now enter Guess 7: usherk
       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
-----------------------------------------
Guess 1| * | * | - | - | - | - | - | - |   24 Points
-----------------------------------------
Guess 2| - | * | * | * | - | - | - | - |   5 Points
-----------------------------------------
Guess 3| - | - | - | - | * | * | * | * |   0 Points
-----------------------------------------
Guess 4| - | - | - | * | * | * | - | - |   10 Points
-----------------------------------------
Guess 5| - | - | - | * | * | * | * | - |   36 Points
-----------------------------------------
Guess 6| - | - | - | - | - | * | * | * |   14 Points
-----------------------------------------
Guess 7| - | - | * | * | * | * | * | * |   38 Points
-----------------------------------------
Guess 8| * | * | * | * | * | * | * | * |
-----------------------------------------
Now enter your final guess. i.e. guess the whole word: crushing
You have guessed the word correctly. Congratulations.
>>>
```

4 ASSESSMENT AND MARKING CRITERIA

4.1 Functionality Assessment

The functionality will be marked out of 7. Your assignment will be put through a series of tests and your functionality mark will be proportional to the number of tests you pass. If, say, there are 25 functionality tests and you pass 20 of them, then your functionality mark will be $20/25 * 7$. You will be given the functionality tests before the due date for the assignment so that you can gain a good idea of the correctness of your assignment yourself before submitting. You should, however, make sure that your program meets all the specifications given in the assignment. That will ensure that your code passes all the tests. Note: Functionality tests are automated and so string outputs need to exactly match what is expected.

4.2 Code Style Assessment

The style of your assignment will be assessed by one of the tutors, and you will be marked according to the style rubric provided with the assignment. The style mark will be out of 3.

4.3 ASSIGNMENT SUBMISSION

You must submit your completed assignment electronically through Blackboard. The only file you submit should be a single Python file called **a1.py** (use this name – all lower case). This should be uploaded to

```
Blackboard>Assessment>Assignment 1.
```

You may submit your assignment multiple times before the deadline – only the last submission will be marked.

Late submission of the assignment will **not** be accepted. In the event of exceptional personal or medical circumstances that prevent you from handing in the assignment on time, you may submit a request for an extension. See the course profile for details of how to apply for an extension.

Requests for extensions **must** be made no later than 48 hours prior to the submission deadline. The application and supporting documentation (e.g. medical certificate) must be submitted to the ITEE Coursework Studies office (78-425) or by email to enquiries@itee.uq.edu.au. If submitted electronically, you **must** retain the original documentation for a minimum period of six months to provide as verification should you be requested to do so.

5 Appendix

5.1 Welcome message

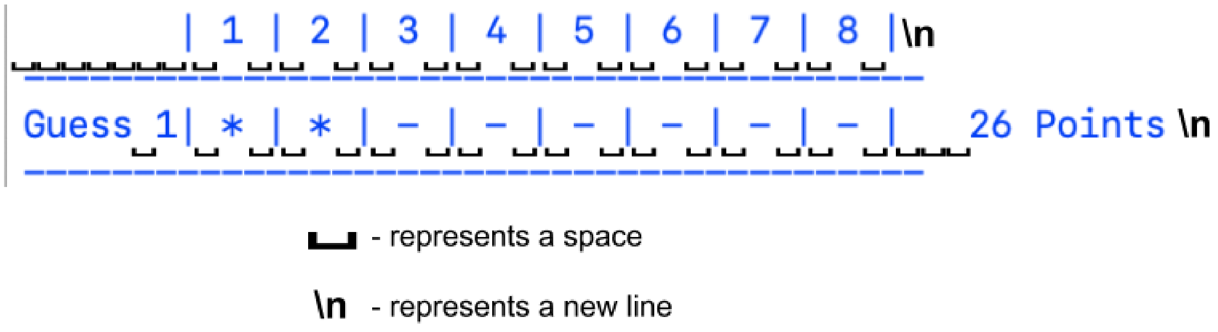
```
"Welcome to the Criss-Cross Multi-Step Word Guessing Game!

Enter an input action. Choices are:
s - start game
h - get help on game rules
q - quit game:
"
```

5.2 Help message

```
"Game rules - You have to guess letters in place of the asterixis.
Each vowel guessed in the correct position gets 14 points.
Each consonant guessed in the correct position gets 12 points.
Each letter guessed correctly but in the wrong position gets 5 points.
If the true letters were "dog", say, and you guessed "hod",
you would score 14 points for guessing the vowel, "o", in the
```

```
correct position and 5 points for guessing "d" correctly, but in the
incorrect position. Your score would therefore be 19 points."
```

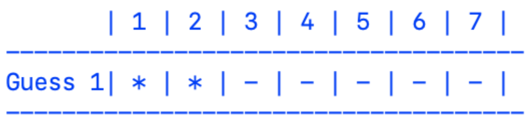
5.3 Printing Example



```
        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |\n
-----------------------------------------
Guess 1| * | * | - | - | - | - | - | - |   26 Points \n
-----------------------------------------
```

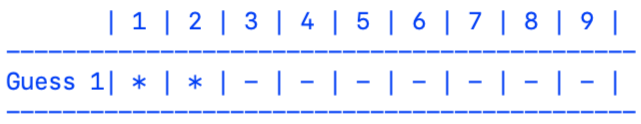
␣ - represents a space

\n - represents a new line



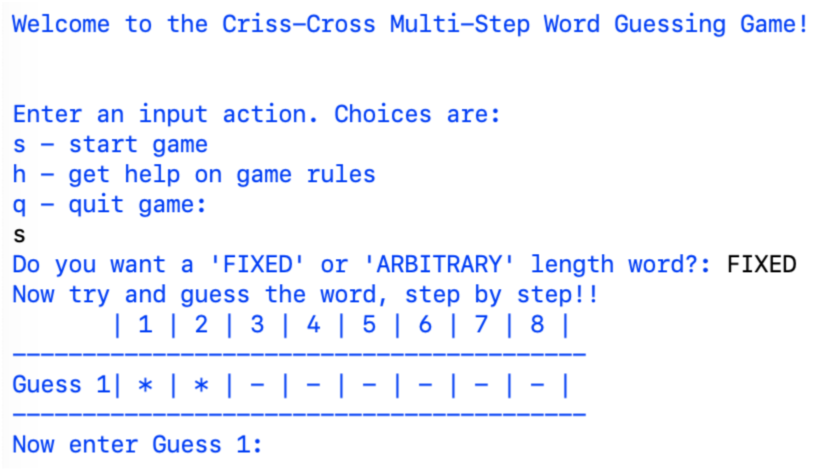
```
       | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
-------------------------------------
Guess 1| * | * | - | - | - | - | - |
-------------------------------------
```

Figure 2: Display for seven letter words.



```
       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
---------------------------------------------
Guess 1| * | * | - | - | - | - | - | - | - |
---------------------------------------------
```

Figure 3: Display for nine letter words.



```
Welcome to the Criss-Cross Multi-Step Word Guessing Game!


Enter an input action. Choices are:
s - start game
h - get help on game rules
q - quit game:
s
Do you want a 'FIXED' or 'ARBITRARY' length word?: FIXED
Now try and guess the word, step by step!!
       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
-----------------------------------------
Guess 1| * | * | - | - | - | - | - | - |
-----------------------------------------
Now enter Guess 1:
```

Figure 4: Example where the player chooses ”s” to start the game.