

Endigit WordBattle

Object

The object of Endigit WordBattle is to find all the words in a 4 by 4 grid of letters that are in a given dictionary. Instead of finding all of these words yourself, you will be writing LabVIEW code that will find the words for you. The goal of the competition is to write code that will solve the puzzles the fastest.

How to Play

[Download the Endigit WordBattle project](http://Endigit.com/wordbattle) from Endigit.com/wordbattle. This download includes a zip file with a LabVIEW project. In that project there are three VIs: Framework.vi, Player.vi, and Generate Puzzle.vi.

Framework.vi

Framework.vi is used to test how fast your code runs. It has an array of 10 sample puzzles, an array of the solutions to those sample puzzles, and a sample dictionary.

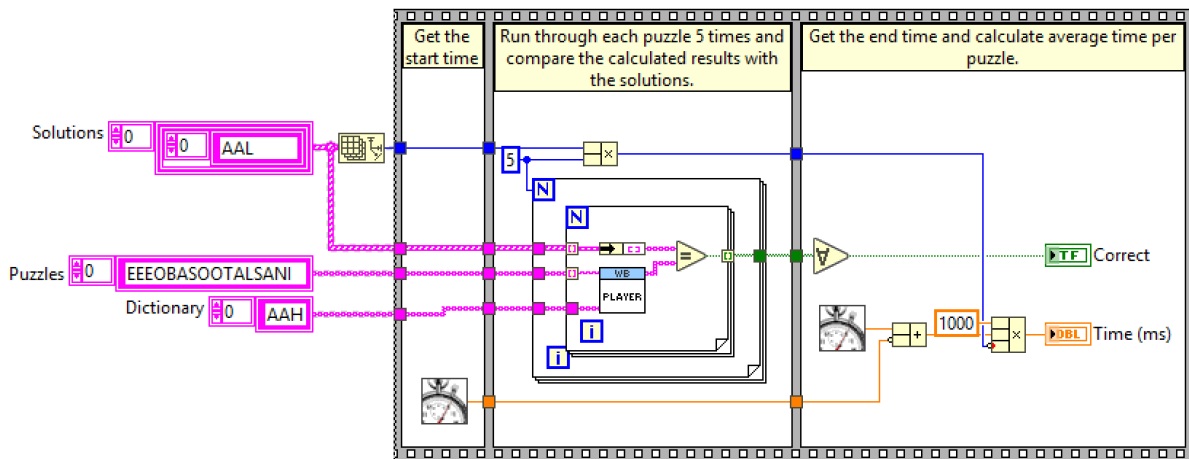


Figure 1: Framework.vi

The framework gets the start time, runs your solution 5 times on each of the 10 puzzles and then calculates the average time per puzzle.

Player.vi

Player.vi is where you implement your solution to the WordBattle challenge. There are two inputs, Puzzle and Dictionary, and one output, Solution.



Puzzle will be a string with 16 characters representing a 4x4 grid
Dictionary will be a list of upper-case, valid words; 3 letters or longer

Figure 2: Player.vi

Puzzle is a 16 character string with the 0th index being the top left corner of the playing board and the 15th index being the bottom right corner of the board.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Figure 3: Letter grid layout

Dictionary is an array of potentially acceptable words. In these example puzzles the dictionary is the same, but it could change on each puzzle. Your solution needs to be able to handle any sorted dictionary.

Solution is an array of sorted strings that represent your solution to the puzzle. Each word must be at least three (3) letters long and be in the puzzle and in the dictionary.

Generate Puzzle.vi

Generate Puzzle can be used to make puzzles. It is not currently used in the framework, but you can use it if you want to try different puzzles.

Rules of the Game:

A valid word:

- Must be 3 or more letters in length
- Every letter in the word must be adjacent or diagonal to the previous letter in the word. As an example using Figure 3, a word starting in index 5 could have index 0,1,2,4,6,8,9, or 10 as the next letter. See Figure 4 for a valid example and Figure 5 for an invalid example.

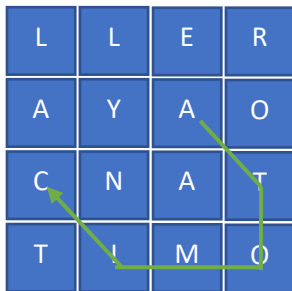


Figure 4: Valid word ATOMIC

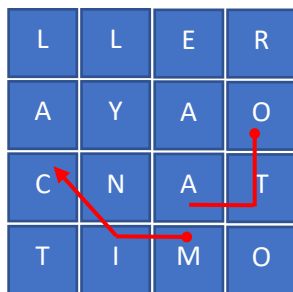


Figure 5: Invalid word ATOMIC

- Can only contain the letter at an index once. For example, a word cannot use index 2 more than one time in the same word. Figure 6 shows examples of building the word Endigit. The green path is acceptable and the red path is unacceptable.

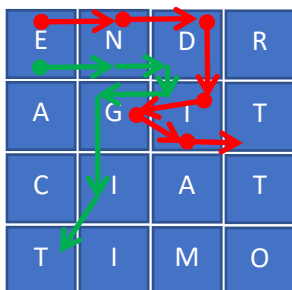


Figure 6: Examples of using letter once per word

Example

To help you get started we will show you images of the block diagram for an example that Jon or awesome intern did. His solution runs in three (3) seconds.

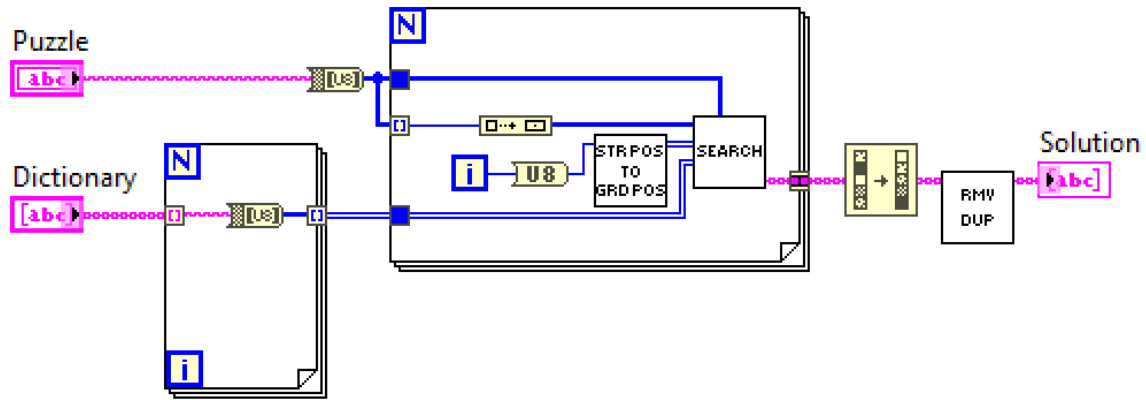


Figure 7: Jon's Player.vi

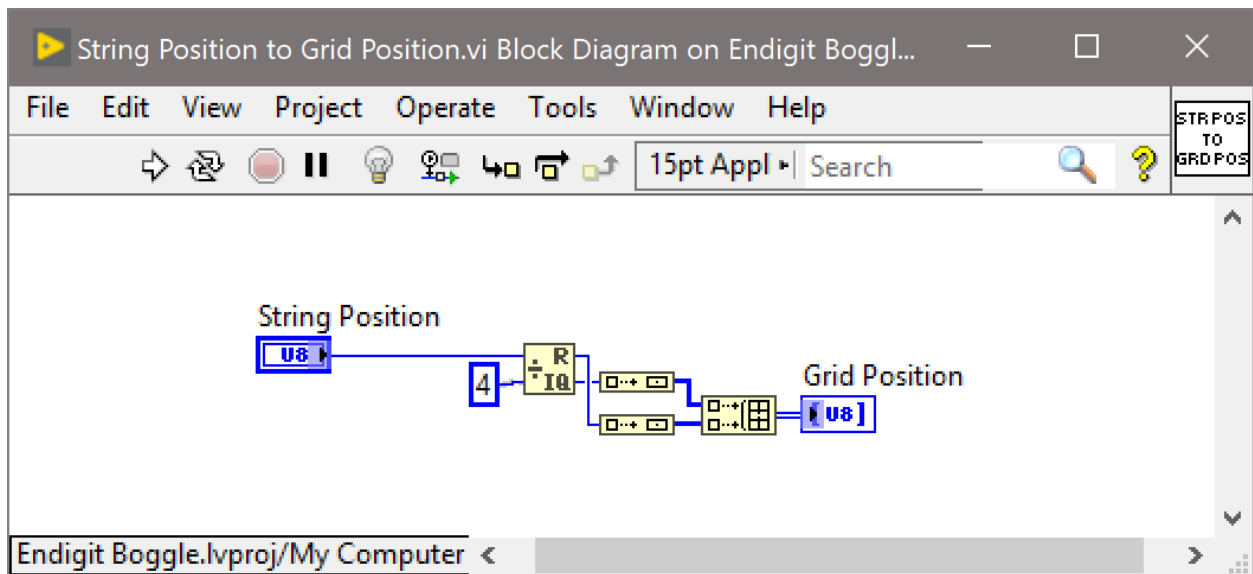
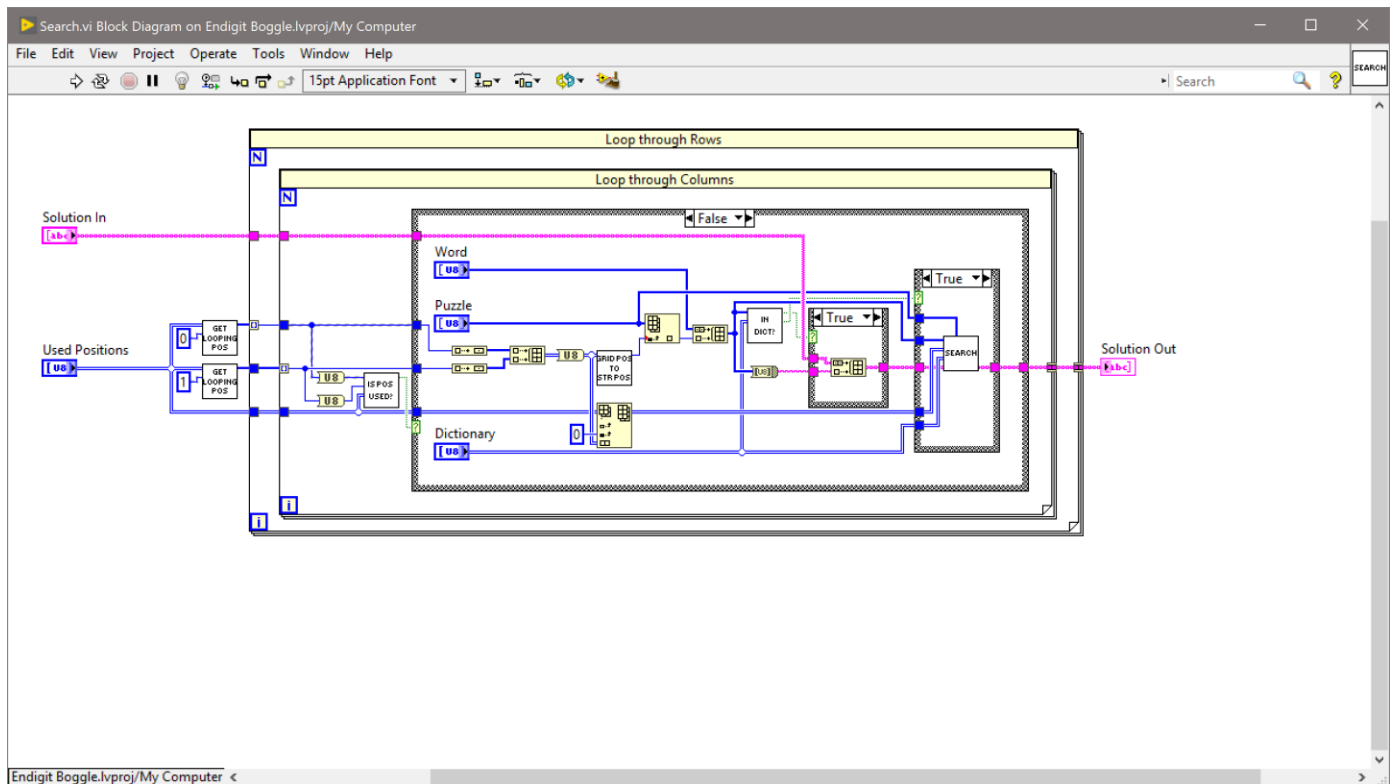


Figure 8: String Position to Grid Position.vi



Tips

1. Optimize the execution settings of your vi by modifying the execution settings in the VI Properties (ctrl-I)..
2. Try a lot of ideas. You can submit as many solutions as you would like.
3. Winners will be announced at NIWeek 2018. You do not need to be present to win.
4. There is a popular board game similar to WordBattle. In that game, the Q cube has Qu. To simplify things, we decided to not add the U to the Q. Doing so eliminates the need for corner cases. If you see a Q, treat it as an individual letter.

Submitting your solution

1. Create a .zip file with the Player.vi and any required subVIs you created.
2. Go to endigit.com/wordbattle/
3. Fill out the form with your contact info so we can contact you if you are a winner.
4. Attach your solution
5. Hit **Submit WordBattle Entry**