I used several 2D arrays in my program. I decided it was necessary as it was a much more efficient method of storing so many values to use in my program. I chose a 2D array specifically to store the X & Y values of the beginning and end of each hidden word. My explanation is this: A 1D array is a form of variable list. Except all the variables have the same name with values stored under ‘indexes.’ It keeps the data in a single virtual ‘row.’ It can store any datatype. A 2D works as a normal array would, but with rows. It is essentially a long 1D array that is sorted for easier reference later. It is best visualized as a grid, as shown below. Justin Lu first taught me a 2D array by using a grid, and I continued my research on the Processing Reference Library as well as tutorials and online videos. I discovered that using an array would be much easier, as the index numbers of the 2D and 1D array can line up and create a much simpler method of assigning values and figuring out the direction and location of each word. My 1D arrays were used to store the Strings that hold the hidden words as well as to store the boolean values of the found words. The words are displayed using a for loops that runs through each word and displays each character using .length , using the 2D array to tell each word its location. This would help my program check for found words and refer to the array to check that the user has selected the correct location for each word. I added in my enhanced version a feature that allows the user to select any word ‘tile’ even if it isn’t a hidden word. I used another 2D array to store the coordinates of each tile.

Each word search has a 15x4 2D array.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Beginning of word0 (X) | Beginning of word1 (X) | Beginning of word2 (X) | Beginning of word3 (X) | Beginning of word4 (X) | Beginning of word5 (X) | Beginning of word6 (X) | Beginning of word7 (X) | Beginning of word8 (X) | Beginning of word9 (X) | Beginning of word10 (X) | Beginning of word11 (X) | Beginning of word12 (X) | Beginning of word13 (X) | Beginning of word14 (X) |
| Beginning of word0 (Y) | Beginning of word1 (Y) | Beginning of word2 (Y) | Beginning of word3 (Y) | Beginning of word4 (Y) | Beginning of word5 (Y) | Beginning of word6 (Y) | Beginning of word7 (Y) | Beginning of word8 (Y) | Beginning of word9 (Y) | Beginning of word10 (Y) | Beginning of word11 (Y) | Beginning of word12 (Y) | Beginning of word13 (Y) | Beginning of word14 (Y) |
| End of word0 (X) | End of word1 (X) | End of word2 (X) | End of word3 (X) | End of word4 (X) | End of word5 (X) | End of word6 (X) | End of word7 (X) | End of word8 (X) | End of word9 (X) | End of word10 (X) | End of word11 (X) | End of word12 (X) | End of word13 (X) | End of word14 (X) |
| End of word0 (Y) | End of word1 (Y) | End of word2 (Y) | End of word3 (Y) | End of word4 (Y) | End of word5 (Y) | End of word6 (Y) | End of word7 (Y) | End of word8 (Y) | End of word9 (Y) | End of word10 (Y) | End of word11 (Y) | End of word12 (Y) | End of word13 (Y) | End of word14 (Y) |

9.1: What is an Array? - Processing Tutorial (The Coding Train)

<https://www.youtube.com/watch?v=NptnmWvkbTw> (Concept of arrays)

Processing References on 2D arrays

<https://processing.org/tutorials/2darray/> , <https://processing.org/reference/Array.html> , <https://processing.org/examples/array2d.html>

**My program uses the full screen function because I decided it was necessary to have a bigger resolution to fit 100 letters or more on screen without losing resolution. I have tested my program on various monitors of different aspect ratios, and my program will still run perfectly as long as the screen resolution is FHD 1080p.**