IMPORTANT: This is the Updated Version of My Presentation Which Documents Why I Created The Slides the Way I Did

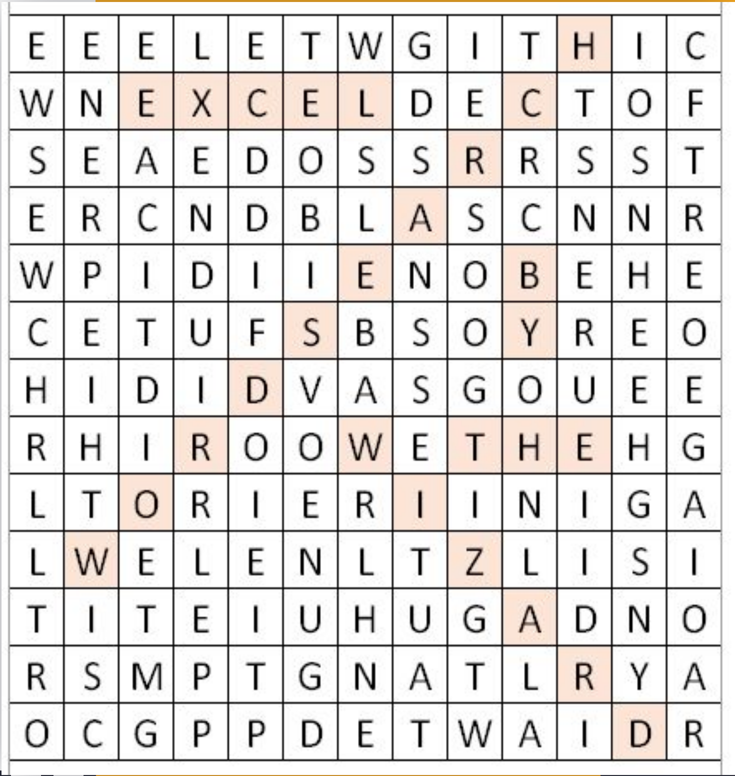
Mango’s Word Search

By Julio Madrigal

I used this slide to explain what my project needed to have/accomplish

Definition and requirements

- The user should be able to choose or create a word list.
- Words are randomly hidden in the array of letters, count number of overlaps to make certain complex.
- The user should be able to view the word list and know which have been found and which are not yet found.
- The player identifies and marks the found words in an appropriate visual manner.
- The player can ask for a hint.
- The player should be able to quit and return to the game later. (In Progress)
- The puzzle can be scrambled and played again.
- The player should be able to obtain additional word lists from a server without reinstalling the app. (In Progress)



E	E	E	L	E	T	W	G	I	T	H	I	C
W	N	E	X	C	E	L	D	E	C	T	O	F
S	E	A	E	D	O	S	S	R	R	S	S	T
E	R	C	N	D	B	L	A	S	C	N	N	R
W	P	I	D	I	I	E	N	O	B	E	H	E
C	E	T	U	F	S	B	S	O	Y	R	E	O
H	I	D	I	D	V	A	S	G	O	U	E	E
R	H	I	R	O	O	W	E	T	H	E	H	G
L	T	O	R	I	E	R	I	I	N	I	G	A
L	W	E	L	E	N	L	T	Z	L	I	S	I
T	I	T	E	I	U	H	U	G	A	D	N	O
R	S	M	P	T	G	N	A	T	L	R	Y	A
O	C	G	P	P	D	E	T	W	A	I	D	R

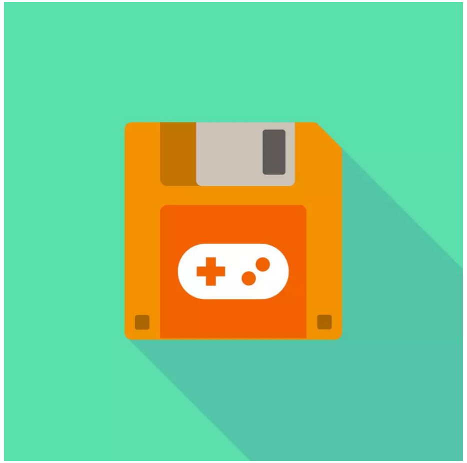
Solution

- Android Studio that uses Java and XML
- Android 10 Q (API 29)
- Title Screen, GamePlay Activity, Result Activity

I could have gone more in depth when presenting this slide, but I wanted to aim for the general audience. I could have explained that the GamePlay activity is where my Word Search Board generation algorithm will be operating in and that the Result screen will show the words you were working with and that the Title screen will include various options for the user. I could have went through each requirement and identify where and when each requirement is met as well.

Exceptions

- Save Game
- Import



I did pretty well with this slide as I explained that I did not have these ready but I had ideas for them. For Save Game I mentioned that I could use onSaveInstance and onRestoreInstance. However, I learned through my defense that I did it the wrong way and now I know for the future to save what the user has done through a file or as I have researched through a function called sharedPreferences(). For Import I mentioned that I will do an Asynchronous task in the background to access the server to grab word lists from.

EXPLANATION FOR NEXT FEW SLIDES

I did not talk about the algorithm that was implemented into my WordSearchBoardView library as to keep it simple and avoid confusion with the audience. Furthermore, CS majors/minors are familiar with C++ so this was a great way to explain the algorithm. I will be explaining how the algorithm was implemented in the library in detail in my binder.

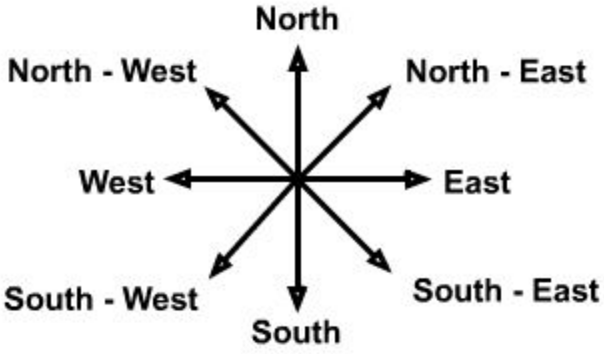
I did below average in explaining how the WordSearchBoardView library functioned during my defense and I was going through something personal that week, my apologies. However, I will demonstrate my knowledge of the library in the binder.

METHODS

		T		R		
		I		E	A	
		M	A	N	G	O
			T	T	P	D
					O	
				H	O	
			T		L	
		E				
	M					

This was my old algorithm which I was initially going to do. You can overlap on any character of the word you are trying to place on the board. If it finds a common character with another word, it will try all 8 directions to see if it can fit. If it fails to fit, it will randomly place it on the board. In this example, all words have a character in common with at least one other word. Mango was placed first.

I got stuck and made it too complex. I met with Dr. McVey and she suggested a different way of doing things

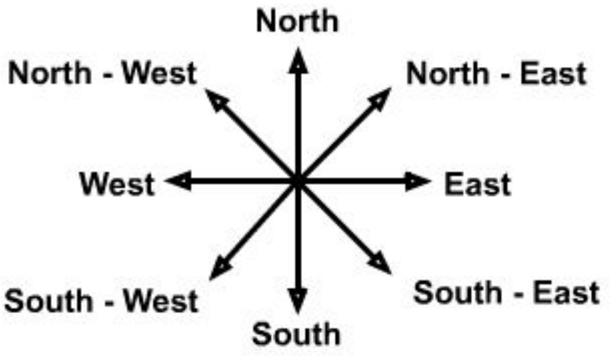


METHODS

	T				D	
X		S			L	
	I		E		O	
		M	A	N	G	O
		A				
		R	E	C	O	N
		S	A	L	T	

I ended up going with Dr. McVey's way of doing this as it allowed me to save time and focus on the rest of my app. This new algorithm only looks at the first letter of every word trying to be placed on the board. It gets the job done despite less complexity. In the future I can possibly make my old algorithm work but I prioritized the completion of my app. (Thanks Dr. McVey-Pankratz)

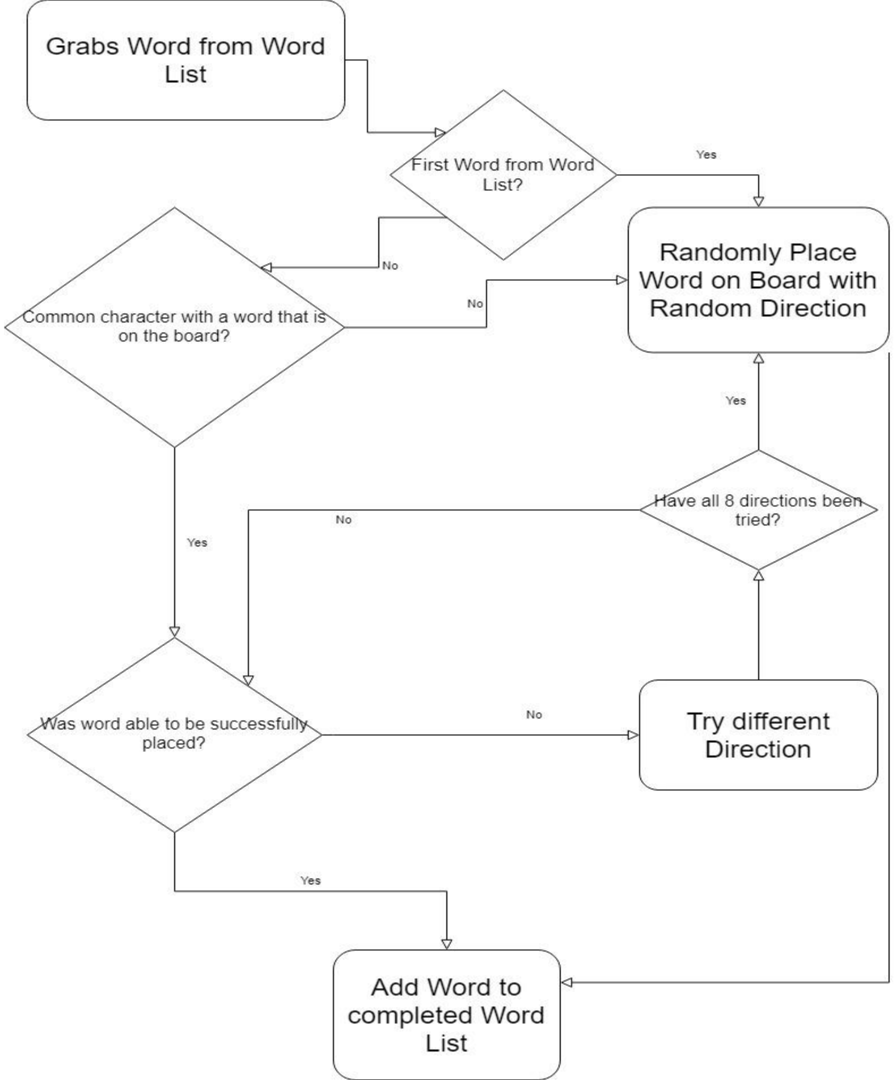
Now I am going to show a data flow diagram of how this new algorithm works.

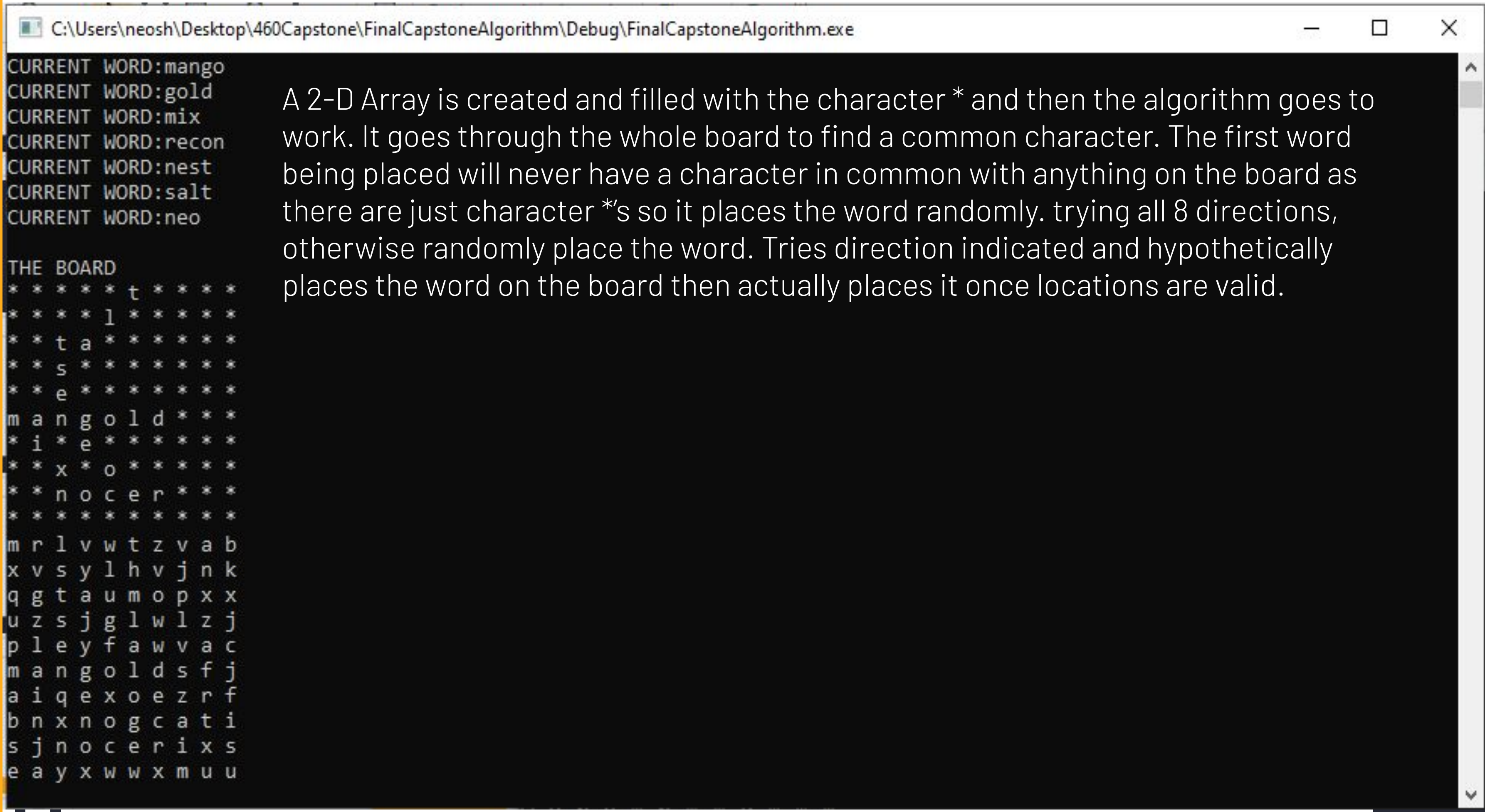


I updated my Data Flow Diagram as my prior Diagram did not depict the algorithm accurately.

This is how the Word Search Board was generated.

Next, I will show a visual of the algorithm created in C++ to better depict what the algorithm was essentially doing as this will be hidden from the user when playing the game


Grabs Word from Word List
First Word from Word List?
Yes
No
Randomly Place Word on Board with Random Direction
Common character with a word that is on the board?
No
Yes
Yes
Have all 8 directions been tried?
No
Was word able to be successfully placed?
No
Try different Direction
Yes
Add Word to completed Word List



C:\Users\neosh\Desktop\460Capstone\FinalCapstoneAlgorithm\Debug\FinalCapstoneAlgorithm.exe

```
CURRENT WORD:mango
CURRENT WORD:gold
CURRENT WORD:mix
CURRENT WORD:recon
CURRENT WORD:nest
CURRENT WORD:salt
CURRENT WORD:neo

THE BOARD
* * * * * t * * * *
* * * * l * * * * *
* * t a * * * * * *
* * s * * * * * * *
* * e * * * * * * *
m a n g o l d * * *
* i * e * * * * * *
* * x * o * * * * *
* * n o c e r * * *
* * * * * * * * * *

m r l v w t z v a b
x v s y l h v j n k
q g t a u m o p x x
u z s j g l w l z j
p l e y f a w v a c
m a n g o l d s f j
a i q e x o e z r f
b n x n o g c a t i
s j n o c e r i x s
e a y x w w x m u u
```

A 2-D Array is created and filled with the character * and then the algorithm goes to work. It goes through the whole board to find a common character. The first word being placed will never have a character in common with anything on the board as there are just character *'s so it places the word randomly. trying all 8 directions, otherwise randomly place the word. Tries direction indicated and hypothetically places the word on the board then actually places it once locations are valid.

Demo

Strategies

- Independent
- Documentation
- Professors
- Trial and Error
- CSCI 350 Event Programming in Windows

I explained here how I came about my solutions. I was independent most of the time as I wanted to test my ability to work independently but I may have overdone it as I felt like I wasted more time than I expected which hurt my final product in the end, but lesson learned. I read the documentation on the WordSearchBoardView library. Dr. McVey had more knowledge with Android Studio so I went to her mainly. Trial and Error were helpful in figuring out small bugs such as syntax errors and infinite loops rather than going online/to professors for help. And of course my previous class using Android Studio helped especially with initially starting as I could look at prior code and build on it.

Extensions

- Complex Algorithm
- Animations
- Difficulty
- Score w/ Timer
- Save Files

I would have loved to complete the old algorithm I had going as it look at all the characters of the words when trying to find a character in common. The person who gets this project after me should definitely try to figure out this more complex algorithm. The animations I have in my application are basic and for the person who will get this in the future they can strive to create more innovative animations like those you see in the app store now adays. Furthermore, a difficulty selection would be great as it will allow the user to get a challenge. I would start with Easy, Medium, Hard difficulties. A score that is affected by the timer's remaining time will be a great feature to add and also a leaderboard to keep track of high scores. Save files would be good to allow for multiple games to happen at once.

```
strlen( realpath($_SERVER['DOCUM
ltrim(preg_replace('/\\\\/', '/', $image_sr
]['config'] = serialize($captcha_config);

ha_config['code'],
mage_src

gb') ) {
tr, $return_string = false, $separator = ',
lace("/[^0-9A-Fa-f]/", '', $hex_str); // Get
;
 == 6 ) {
dec($hex_str);
 0xFF & ($color_val >> 0x10);
 0xFF & ($color_val >> 0x8);
 0xFF & $color_val;
str) == 3 ) {
hexdec(str_repeat(substr($hex_str, 0, 1), 2
hexdec(str_repeat(substr($hex_str, 1, 1), 2)
hexdec(str_repeat(substr($hex_str, 2, 1), 2)

implode($separator, $rgb
```

Thanks!

Any Questions?

I included this slide at the end in case someone asked how I created the grid or how the algorithm works in my application. Unfortunately I did not get to this slide.

Word Search Library

BoardPoint

Specific point on the board containing row and column of that specific point

BoardView

Draws and Measures each Tile to fit on the Board

WordSearchBoardView

Does the generation of the Word Search Board using an Algorithm as well as Touch Events

BoardWord

Every word on the Board has its direction and startPoint(BoardPoint) which is used heavily in the algorithm