Steve Babcock
6/26/2017

Spelling Squared

This project is an Android app which can be downloaded from the Google Play Store. Search for either ‘Spelling Squared’ or ‘Steve Babcock’. It was written in Java, and built with Android Studio version 2.2. The Code folder is the base of the project if you would like to open it in a development environment. The published version has multiplayer features which make use of Google’s game services, but the demo code provided with this report omits these. The object of the game is to solve a 4x4 grid of random letters by arranging them into 4 words down and 4 words across. The idea of the game is my own, and a prerequisite of the app was computing all the 1.8 million solutions.

I’ll briefly describe that process. I started with a list of all the four letter words, and executed a nested loop of four layers. The first layer places every word in the top row, second layer places every word in the second row, etc. The key to making it all work is the ability to quickly check whether a given two, three, or four letters are a valid part of a word. I used a recursively defined class to accomplish this.

```
class Node {
	bool isPartOfWord;
	Node nextLetter [26];
}
```

Above is an abridged version of that data structure. So to check if ‘aw’ is a valid substring, the algorithm goes to nodeList[0].nextLetter[22].isPartOfWord and should find true since ‘away’ is a four letter word among others. However, when the algorithm checks ‘bex’ it will go to nodeList[1].nextLetter[4].nextLetter[23].isPartOfWord and find false. In conclusion, the time spent loading the members of the four letter word list into this recursive data structure is a great tradeoff for the time saved avoiding linear probing of the list.