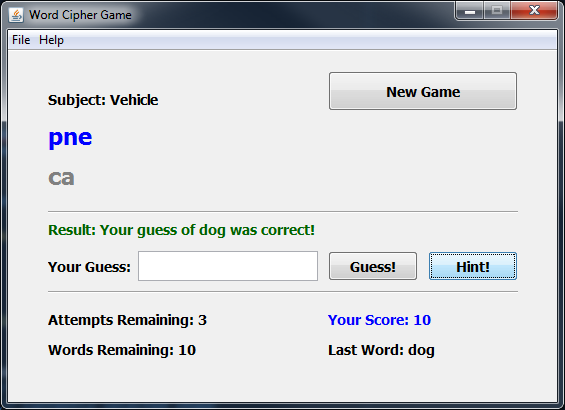
Rob Just  
CS 242  
9/15/10

**Final Project Presentation**

**Word Cipher Game**  
  
**Screenshot:**



**Description:**  
This game is a simple word ***cipher*** guessing game. It is ***not*** a word ***scramble*** game. The difference is that each letter of each word is ciphered using the Rot13 cipher. This cipher shifts the position of each letter by 13 places in the alphabet. C becomes P and H becomes U. In the example above when the word “chair” is ran through the Rot13 cipher it become “punve”.

The game loads a text file named CipherGameWordList.txt with a list of subjects and words. Each line of the text file is formatted as “Subject word”. An example is “Animal cat”. The program selects a word at random and then it displays subject in plain English and then displays the ciphered word below it. No word is used more than once. The player must type in their guess of what the word is based upon the subject and\or hints. Each correct word scores the player 10 points.

The Hint Button can be clicked to display a letter of the word one by one if more than 1 attempt remains. Each Hint costs the player 1 attempt. If all but one letter is revealed the Hint button will not display the last letter. A maximum of 4 hints can be requested.  
  
The attempts remaining, score, last word, and words remaining are displayed and updated at the bottom of the screen. When all words in the list are exhausted then a game over message is displayed with a final score.

The New Game button resets the game and starts over with a random word in the word list text file.

**Example of Rot13 Cipher For Reference**

**Input Example:**This is test input. 1234. My name is Rob.  
 **Rot13 Output Example:**Guvf vf grfg vachg. 1234. Zl anzr vf Ebo.  
 **Lessons Learned**

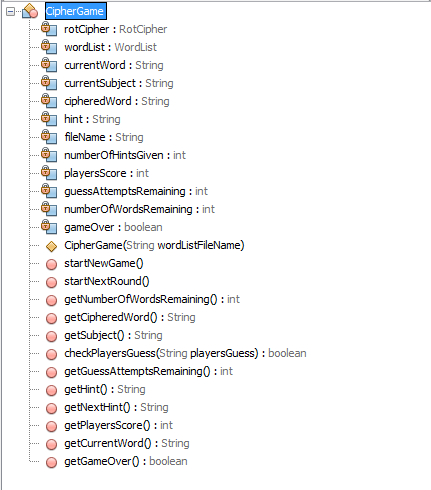
I learned quite a bit during the length of this project. First of all it can be difficult to come up with an original idea that you want to develop into a working program. Once you have an idea and start planning development it quickly can become a lot more complicated than you originally thought it would be. Even after the development phase I found myself adding more and more methods to accommodate the program. I guess this is an example of not putting enough time into the development phase to foresee the need for the additional methods.

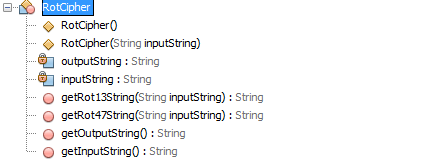
I also got a much better understanding of what composition was once I actually used it in the project. Using three of my own classes to compose one class was quite interesting and in the end worked out better than expected. I much prefer this method over Inheritance where ever possible.

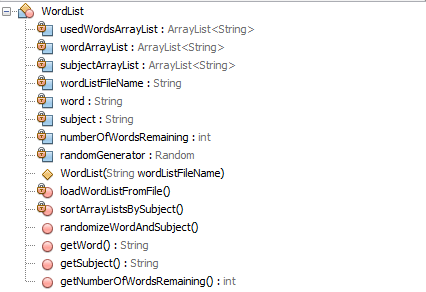
What I would do differently next time would be to start earlier! I was behind in this class from week one because of a vacation I took. I had to start development on the project much later than I hoped which provided for a lot of unnecessary anxiety. I did learn from this that I seem to code fairly well under pressure. That’s in my opinion anyway.

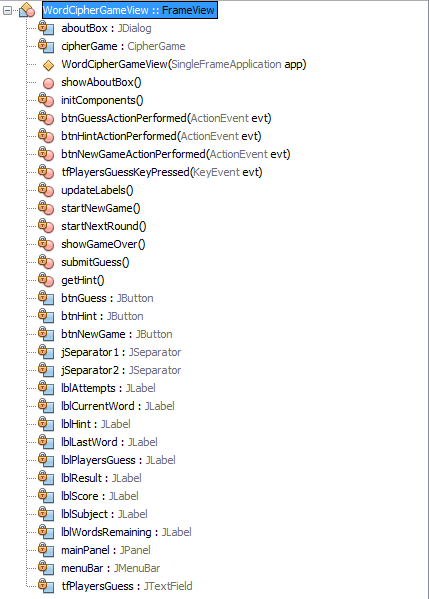
The next release of this program would benefit from a few things. First I would add a built in word list editor to prevent formatting errors and encourage players to come up with their own lists. I would also add a penalty for guessing incorrectly. Another option I thought of would be a timer that would force the player to guess faster.

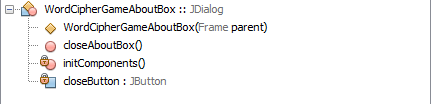
**UML** (Not technically UML, tried to export from NetBeans but didn’t work correctly)

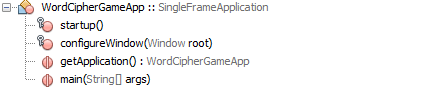








****

****