**Project Documentation:**

“The Ra-Tet” signify the five different parts of the Egyptian God Ra, working together as a cohesive whole for the benefit of humanity. And in order to instill a similar concept in our group we decided to work on an android app that would engage and challenge the user’s intellectual capabilities. To that end our first idea was to design a role playing game with an egyptian theme. However in our second meeting we decided that due to the time constraints associated with this project, we needed to reevaluate our strategy. Therefore we decided to switch to a murder mystery game like Clue , with a limited number of possible murder scenarios. While working on the project, we realised that the game would require a huge graphical component that we simply could not finish in the allotted time. Once again we had no choice but to re evaluate our project goals and come up with a more realistic idea. We finally settled on an android app that could be used by people who either want to improve their vocabulary skills, or love games like scrabble, crossword puzzles, etc.

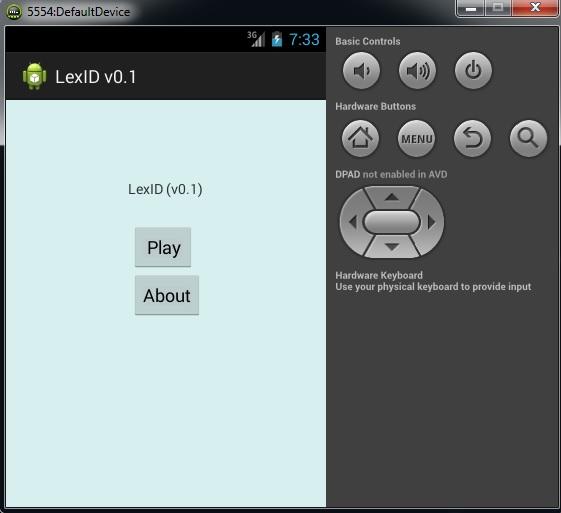
The essential idea behind the app is to help the users find words of a particular length, that fit a specific criteria. For example consider the situation of a player in figure 1: as you can see on the board the player has two distinct opportunities to score a lot of points. (see image below )



**Figure 1:** highlights two of the most important opportunities that the scrabble player has to score a lot of points.

As you can clearly see in the picture the player’s choice of words would be constrained by the alphabets already on the board. This is where are app would come into play! Once the user has initiated our app, it will ask the user to choose the size of the of word they would like and the exact place where they would like a particular alphabet. Our app would then run a search in its dictionary archive and provide the user with a list of 10 random words (if available) that fit the specified criteria. If the user does not like the results obtained he/she can simply rerun the search and get more results! So for example if the user chooses the first position indicated, the length of the word would be six and the specific alphabet placement would be “f” as the first and “w” as the third letter in the word. Therefore one of the possible word combinations would be “FEWEST”.

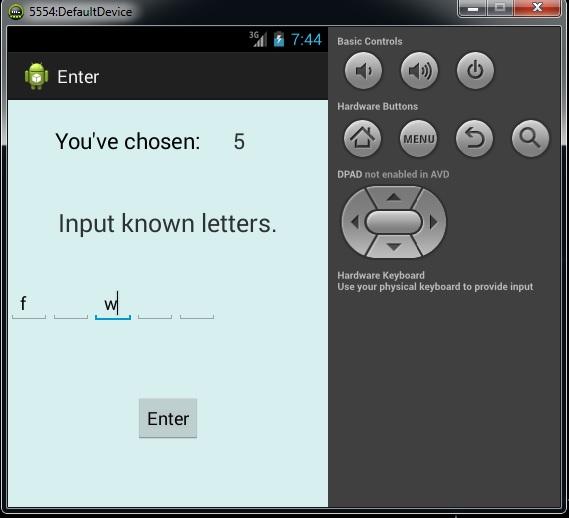
Alternative if the user does not have the tiles to make the word “FEWEST”, he/she could search for a smaller word, using the same process. They should start by resetting the search parameters and then repeat the process. Once the program has been reset the user would be redirected to the startup screen (see figure 2) and prompted to play the game again.



**Figure 2:** shows the startup interface for the game.

This time the user would choose a smaller word size (see figure 3) and then re enter the specific alphabet placement :“f” as the first and “w” as the third letter in the word (see figure 4).

**Figure 3** shows the user interface asking the user for the specified word length



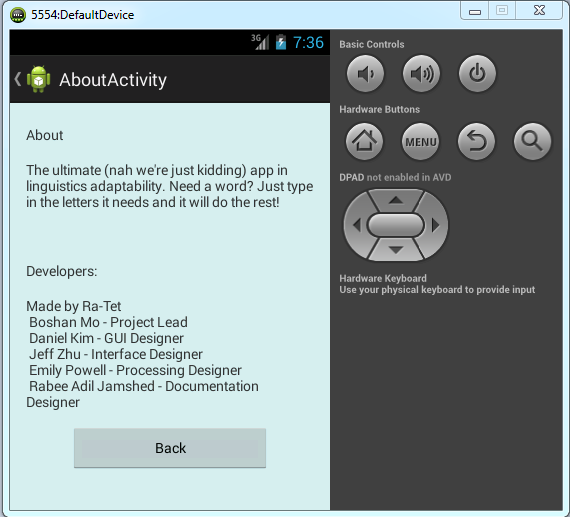
**Figure 4** shows the user interface for the specified word length and the specified alphabet positions.

The program would repeat its search with the modified parameters and print out a maximum of 10 random words that fulfill the specified criteria (see figure 5).



**Figure 5** shows the user interface with the list of all the possible words that meet the specified criteria. In this case apparently there is only one word in the dictionary that fulfills the specified criteria.

In the event the user wants to learn more about the program, all the user would have to do is click the “About” button on the startup screen (see figure 6)



**Figure 6** shows the details page for our android app.

I am sure that you are wondering as to how does our app work? It is actually very simple! Once the user has specified the length of the word that they are looking for, it loads a file containing a list of words of the specified length. Once the user has specified the particular conditions for the word, it makes a list of all the words that satisfy the specific criteria. Once the list has been completed, it prints out a maximum of 10 random words from the list. If the user wants the entire list then all they have to do is rerun the program multiple times. Unfortunately our original list is limited because it is based upon a dictionary for the 5000 most commonly used words in the english language; more words will be added depending how well actual smartphones can handle data allocation.

Possible improvements include: 1) using a bigger dictionary, 2) including error checking in to the program to ensure that each rerun of the program (for the same specified criteria) does not produce any of previously printed out words, 3) and include more user flexibility, to allow the user to search for words of any length for a particular letter placement. For example if the input was “F \_ W”, all possible answers would include: FEW, FEWER, FEWEST, etc.