Project Overview

Evil Hangman
The goal of this project is to make a hangman game in which the computer will cheat and change the word to avoid the user's guesses. The program will start by picking a random word out of the dictionary. As the user makes more guesses, the word will change so that as few of the guessed letters are in the word. We will start with a Haskell and C++ implementation, and if time permits, other languages that fit in between the imperative/functional programming paradigm such as Python.

Team Members
Reid Anetsberger
Michael Parker

Time Budget
This project will be flexible in terms of time commitments. It can be scaled up or down if it needs to be. If we progress through the project quicker than we think, more time can be committed to optimization of data structures and algorithms along with implementing evil hangman in additional languages.

Timeline
Week 1: Collect the english dictionary in a text file. Read the textfile into the appropriate data structure and sort the words into "word families". (Haskell) **3-5Hr**
Week 2: Complete the pruning algorithm - that is, the algorithm that reduces the working set of words based on what letters the users have guessed. (Haskell) **3-5Hr**
Week 3: Complete evil hangman in Haskell **3-5Hr**
Week 4: Complete evil hangman in C++. It is expected to take significantly less time to implement the algorithms in other languages once it is working in Haskell. **6Hr**
Week 5: Complete project report and presentation and if time permits, complete evil hangman in Python. **6-10Hr**

Expected time commitment per member: 21-32 Hours

Risks
This project will be fairly low risk. We plan to start with the implementation in Haskell because it is the language we're least experienced with. The biggest risks are at the beginning of our project. Learning how to do file I/O, use data structures, and manipulate data structures in Haskell will likely be the greatest challenges, which is why we are going to try to complete those tasks first. Completing the most difficult parts at the beginning of the project will help mitigate a lot of risk. We chose evil hangman because it is not overly challenging - giving us the opportunity to focus on comparing as many languages as possible.