CS 230 Final Project - Phase 1 Specifications

Project Title: Comparative Language Acquisition

Team Members: Audrea Huang, Hannah May, Bianca Pio

2a. List the **ADTs** that will be used and what information they will store. Include a brief justification for each of your choices.

1. Hash Table: A hash table will be used in this program in order to take words from a file and put them into a useable format. The keys of the hashtable will be Strings representing the various words from the file. The values will be doubles representing the percentile representation of the percent of children at the designated ages who can pronounce the word. Using a hash table will enable us to easily access the data corresponding to each word in constant time.
2. Vector: A separate vector will be used for each word in the database, storing the percentage of children who can pronounce that word at each age for which data is available. These vectors are the values linked to the keys in the word table.The value corresponding to each key in the hash table will be a single vector with this information. We chose to use a vector since it can easily be extended without having to worry about expanding the capacity; additionally, it is indexable so we can easily retrieve a percentage at a certain index.
3. Linked List: A linked list will be used to store the ages of the children in our data from youngest to oldest. We chose to use a linked list so we can maintain the order between this ADT and the vector. For example, if index 1 of the linked list corresponds to 13 months, the value at index 1 of the vector for a specific word will correspond to the percentage of children who are 13 months that can pronounce that word.

2b. A list of the important **classes** that you expect to define for your project with a brief description of the purpose of each class. Some of these classes should capture the basic objects that exist in the problem. There may also be classes that embody the graphical user interface, or the main() method. This list should include the classes that implement the ADTs that you plan to use. Note that as you proceed with your program development, you may discover other classes that would be useful to define for your application.

* WordTable
  + WordTable is a class that reads data from a given file (or database) and loads the data about word acquisition into a hash table. The data is organized in a way that each row contains a word and the percentiles of children that can pronounce the word. Therefore, the constructor will take the file as an input value, and read from it the words for the keys and the percentiles for the values. This format allows the computer to easily access percentiles of acquisition when given any word in the file (or database).
* UserProfile
  + This is a class that prompts the user for information about the child whose level of proficiency that they are trying to gauge. It will ask the user for information about age, gender and language and will store that information.
* Evaluator
  + The Evaluator class determines the child’s word proficiency by asking the user whether or not the child can pronounce a series of words. The word proficiency is equal to the percentage of the children who can pronounce the last word given; i.e. if the last word that the user responded “yes” to is “baa baa” and 67% of the children in the database can pronounce “baa baa,” then the child will have a word proficiency level of 0.67.
* Processor
  + This class will create a hash table using the WordTable class. It will also contain an ageList which will be a linked list of all of the ages in the specified range. This list will later be used to index the hashtable in the Evaluator class and find the percentage of children who can pronounce the given word associated with the age.
* getInfoPanel
  + This GUI class will create the interface with which the user interacts to provide and retrieve information about their child’s age, gender and name.
* ProficiencyCalcGUI
  + This GUI class creates the interactive pane for the user to answer questions about their child’s language proficiency.
* Reader
  + Takes a CSV datafile and converts it into a FilterableDataset.
* FilterableDataset
  + Represents a Vector of Row objects, that can be filtered based on various Row properties.
* Sorting
  + In-place sorting methods that can sort Row objects.
* FilterPredicate
  + Represents a predicate that is used to filter a collection of items.

2c. A list of some of the main **actions** that you expect to be embodied in methods in your new class definitions (you do not need to include the basic operations defined for the ADT classes that you plan to use). As you proceed with your program development, you will probably discover additional useful methods to define for various classes.

* childInfo (class: UserProfile)
  + Will prompt the user to input the age, gender and language of their child and will store that information in variables.
* getNextQuestion (class: Evaluator)
  + This method will generate words in ascending difficulty to be presented to the user.
* isDone (class: Evaluator)
  + This method determines whether or not the user has responded with “No” three consecutive times, which would cause the program to stop asking questions.
* perWhoHaveWordAtAge (class: Processor)
  + This method takes the age of the child and a word they can pronounce as a parameter. After wordPrompt receives 3 “N” responses, perWhoHaveWordAtAge will be called to calculate the level of the child’s proficiency. First it goes to the age Vector and finds the index corresponding to the age of the child. It then takes that index and finds the value of the word in the hash table associated with that index to grab the percentage of children who can pronounce that word at that age. If the percentage is over 75%, the program will provide tips for improving the child’s vocabulary acquisition.
* setTable (class: Processor)
  + This method will call the wordTable class to create the hash table of words and their corresponding percentages.
* getAge (class: UserProfile)
  + This method will retrieve the age of the child that the user input, in order to easily access the data corresponding to that age.
* setAge (class: UserProfile)
  + This method will set the age variable of the child to a certain value.
* getGender (class: UserProfile)
  + This method will retrieve the gender of the child that the user input in order to access the data corresponding to that gender.
* setGender (class: UserProfile)
  + This method will set the gender of the child to a certain String.
* getLanguage (class: UserProfile)
  + This method will retrieve the child’s primary language that the user input in order to determine which dataset to use for reference.
* setLanguage (class: UserProfile)
  + This method will set the child’s primary language to a certain String.

