

## Project Report: Text Generation

Assigned: Monday, March 5, 2007

Due: Friday, March 16, 2007

*No extensions!*

**Summary:** Now that you've completed the laboratory experiences emphasizing text generation, we'd like you to extend the procedures you've written in that lab and then to describe your project in a form accessible to others.

### Contents:

- Stage One: Generating Particular Types of Texts
- Stage Two: Writing the Report
- Stage Three: Submission
- Questions and Answers

**Collaboration:** You should work in groups of two or three. You may not work alone. You may not work in groups of four or more. You may discuss the assignment with anyone you wish. You may obtain help from anyone you wish, but you should clearly document that help.

**Submitting Your Work:** As always, you will submit your work via email, in this case with a subject of *CSC151 Project One*. For this assignment, you will be permitted to attach files to your email. Further details below.

## Stage One: Generating Particular Types of Texts

In the first laboratory on text generation, you worked with and built procedures that construct simple, yet difficult to predict, sentences. In the second laboratory, you saw how to use some key Scheme techniques to vary the structure of these sentences and the components of the sentences.

Unfortunately, most of those sentences were not very interesting. Hence, we should find a way to generate a particular kind of text. In the past, students have most frequently extended the procedures to generate Haiku-like structures. (How? They created files for different numbers of syllables of different kinds of words.)

This year, I'd like you to try something a bit more interesting. Try writing a procedure to generate a kind of text that either (a) incorporates some rhyming scheme or (b) tells a story, with sentences that build upon each other. One particularly nice technique would be to generate texts that sound a bit like Dr. Seuss.

Your first task in this report is to choose one of these simple types of writing (rhymes, Seussian stories, etc.) and write a procedure that generates unpredictable versions of that types of writing.

In generating those texts, you will likely want to add another element to each of the lists in the parts-of-speech files (such as the number of syllabus, the ending sound, or something similar). You might also choose to customize the words files.

## Stage Two: Writing the Report

For this assignment, in addition to writing code, you are also to write narrative text. In particular, you should write a 2-3 page narrative description of what you have accomplished in the project and how you have accomplished that goal. Your audience is other CSC151 students, who have also done the labs. Your goal is to explain what techniques you used to make more interesting texts.

You should make sure that your narrative describes the key procedures you've written and how they work, and the custom files you've created, and how they are organized. Your narrative should be clear enough that someone could rewrite similar code (in, say, eight hours) without having any of the code as a reference.

## Stage Three: Submission

Once you are satisfied with your code and your report, you should email them to me using a subject of **CSC151 Project One**. I expect to see the code as an attached `.ss` file and the report as an attached Word, PDF, or text file, whichever you prefer.

## Questions and Answers

How long do you think this should take us?

I'd prefer that you spend no more than five or six hours on the project.

Can we achieve rhymes by, say, creating a file in which all the words end with the same sound?

Sure. For example, you might have a general `nouns.txt` file and a more specific `ee-nouns.txt` file with words like `cookie`, `flea`, and `the ilk`.

Can we achieve particular numbers of syllables by creating files of the same kinds of words with different numbers of syllables, such as `noun1s.txt`, `noun2s.txt`, and `noun3s.txt`?

Yes.

I'm trying to get carriage returns into my texts using techniques like `(string-append (sentence) "\n" (sentence))`, but the text doesn't appear on multiple lines.

Strings are supposed to appear on one (potentially wrapping) line. If you want the `"\n"` characters to appear as carriage returns, you need to use `display`

```
> (define str (string-append "Hello" "\n" "Goodbye"))
> str
"Hello\nGoodbye"
> (display str)
Hello
Goodbye
```

---

Copyright © 2007 Samuel A. Rebelsky. This work is licensed under a Creative Commons Attribution-NonCommercial 2.5 License. To view a copy of this license, visit

<http://creativecommons.org/licenses/by-nc/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.