Spreadsheets in Gnumeric: Lecture 2

Tim Capes

March 16, 2011

Last time

- Introduced Gnumeric
- Made comparison between spreadsheets and code
- Did some examples in class

Download Locations for working from home

- PC and Unix: http://projects.gnome.org/gnumeric/downloads.shtml
- MAC: http://www.flyn.org/apple/index.html

Assignment 4

- Assignment 4 will be available for pickup before and after the midterm
- It will consist of two spreadsheet questions and one internet question
- It will be due on March 31st.

Midterm

- Problem Solving
- The Internet
- GUIs
- Gnumeric

Tomorrow’s tutorial will be a review session.

How good is the programming analogy?

- We can track data
- We can execute loops (for statements)
- We can do conditionals (if statements)
- Is there anything we cannot do?

Can we translate an arbitrary python program to a spreadsheet?

- Idea: Let each row represent the current step of the execution
- Assign numeric constants by entering values
- If statements: inside part of a conditional use IF on the same condition
- User input: as cells set aside for them to type in values
- For statements: Unwind the loop
- Seems to work

Flexible input problems?

- Idea: What if we have a program that takes an arbitrary picture
- Loops run for a length of time equal to some variable (say number of pixels horizontally)
- Different pictures run different loops for different lengths
- Change just the input data (the picture information)
- Unwound loop no longer works properly because we run for different amounts of time
- So not at all clear if this actually works

In Practice

In practice spreadsheets and programs for programming languages are different, and while they have similarities and many tasks can be done in one or the other, most tasks are more natural in one of the two.

- Balance Sheets: can write them in code and write a fancy output program to put them out nicely (hard)
- Balance Sheets: Can set up a layout in a spreadsheet and easily edit it (easy)
- Changing pixels: Can write code to do this in Jython rather naturally (easy)
- Changing pixels: Can hand enter appropriate data and calculate adjustments in a spreadsheet (hard)

A good task for spreadsheets

Spreadsheets are at their best when we are working with information that is relatively easy to understand and that humans can actually effectively process and interpret. They also work well when different people are interested in different information, or when a person is interested in varied information presented.

- Balance Sheets: Lots of information different people might want to look at different assets (good)
- Pictures: Lots of information, but one clear thing to see.. most people want to know what the picture looks like. Not many people care about say what the red value of pixel (123,429) is. (bad)

A calendar program

Consider the idea of dividing an arbitrary year up into days.

- Need to know what day the year starts on.
- Need to know if the year is a leap year
- Need to figure out appropriate way to display information

Lets simplify

Suppose we know what day the year starts on and if it is a leap year or not.

- It’s now about displaying the information
- Natural idea is a calendar or some limited version thereof.
- Is this easier to do as a spreadsheet or as a computer program?

An Example

Consider the year 2012. This year is a leap year and starts on a sunday.

- Lets write a spreadsheet that lists all the mondays in this year.
- First monday
- Last monday
- Number of mondays
- List

Example Continued

- Not how we think of the year though
- Monday being the 37th day of the year doesn’t mean much to us.
- Need to incorporate month lengths

In class demo

- January 31
- February 29 in a leap year
- March 31
- April 30
- May 31
- June 30
- July 31
- August 31
- September 30
- October 31
- November 30
- December 31