Neil Satra

Sketching Charts

Computer Science Tripos, Part II

Pembroke College

April 21, 2014

Proforma

Name:	Neil Satra
College:	Pembroke College
Project Title:	Sketching Charts
Examination:	Computer Science Tripos, Part II, 2014
Word Count:	TODO
Project Originator:	Alan Blackwell, Neil Satra
Supervisor:	Alistair Beresford

Original Aims

This project aims to explore if users are able to create content faster, or experiment with it more, if given the tools to directly manipulate their creation. Specifically, it involved:

1. Building an application that lets users create graphical visualisations of their data by simply sketching their desired output, like they would on paper.

2. Evaluating the learnability of the interface, and how it compares to existing tools for creating charts, through a user study.

Work Completed

I have completed all core work items by building a Chart component for desktop applications that successfully runs sketch recognition on user input to determine what they intend to create, and then generates those chart graphics for them. I have also designed and built an interface that makes the learning curve for interacting with this component short and gentle, by applying HCI principles.

A few extensions were also completed, including allowing edits to the formal chart to flow back and manipulate the raw sketches.

Special Difficulties

None.

Declaration of Originality

I, Neil Satra of Pembroke College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

Signed

Date

Contents

Chapter 1

Introduction

1.1 Overview

This project is an exploration of Human Computer Interface concepts governing the interactions of users with tools that let them explore and visualise data.

The design of currently available charting tools were constrained by the input devices available previously: mouse and keyboard. Thus, they usually allow graph generation through one of two interfaces:

1. A series of dialog boxes and wizards to walk the user through a number of choices.

2. Writing code that is interpreted to process data and generate graphics.

This paper describes a different interface, which allows the user to sketch a subset of a chart on their computer touch screen like they would on paper. The hypotheses are that this interface is more learnable over time, and that it allows easier modification and exploration of visualisations compared to other charting applications. Both these hypotheses were investigated through a user study.

The end result is a proof-of-concept charting application that works as below:

1. The user imports data from a Microsoft Excel file.

2. They sketch a rough indication of a chart.

3. They drag the data onto elements of the chart to actually bind the data to the chart.

4. The tool then creates a 'formal' chart.

5. The tool transforms the user's original sketch to more closely match the formal chart, making the mapping between sketch and formal chart elements evident to the user.

6. Any changes on either the sketch or formal chart is fed through to the other view. For example, erasing the a sketched bar removes a data series from the formal bar.

1.2   Background

Sketching inputs have been studied for a while as more natural interfaces to computers, especially for graphics-related tasks [Sutherland, 1964].

1.3   Objectives

Chapter 2

Preparation

2.1 Research

2.1.1 Similar Systems

2.1.2 RATA

2.2 Design

2.3 Planning

2.4 Prototype

2.4.1 Fixing RATA

2.4.2 Generating classifier

2.4.3 Initial User Study

Chapter 3

Implementation

3.1 Overview

At a high level, the program is composed of 3 components - data handling, sketch processing and charting (in increasing order of complexity).

3.2 Data import and management

Since the application is targeted at the average user, their data is most likely to be stored in spreadsheet format. Thus, it is important to allow them to import data from .xlsx and .csv files. For the sake of simplicity, the code assumes that the data is well-formed. Specifically, it works on the following assumptions:

1. The data is arranged as records in the rows of the spreadsheet.

2. The first row contains the names of the various fields.

3. No data is missing (if there are m columns and n rows, there are $m \cdot n$ data values.

Under these assumptions, import tabular data is a common use case, so I studied a number of existing libraries and methods to do this in C# and ultimately settled on

3.3   Sketch Processing Workflow

3.4   Charting

Bibliography

I. E. Sutherland. Sketchpad a Man-Machine Graphical Communication System, 1964.