

Big data in GUI lists

jletnes@slb.com, Schlumberger

Introduction

Working with big data leads to surprising User Experience caveats. When dynamically reading and populating a list from unknown data, it is impossible to a priori bound the data – that is, only after reading the data will we fully know how long it takes to present the data. It is an important UX principle to stay responsive, so an adaptive way of presenting the data is necessary.

Imagine you are browsing through all the books in a library and keeping a tally of how many times you see each letter on a piece of paper next to you. Now, the librarian is breathing down your neck and waiting for the list to be finished. The librarian is blocked by you, because you're still busy browsing and tallying the letters in the books.

At what point do you go “this list is representative for now; I'll fill out the rest later” and hand off the list to the librarian? When and how do you give the librarian updated lists? What happens when you get into the Greek section and discover new characters?

The goal

The goal of this project is to demonstrate several list population strategies. You will create a standalone windows application that presents one million entry lists; one list at the time. The app must run on demand and be usable by a UX designer.

You will create several data sources of one million entries; for example, a fast one, a slow one, an erratic but fast one, an erratic but slow one, one that simply never finishes, a source with only unique replies, a source with massive amount of duplication, etc.

You will create several lists that poll all data sources; for example a list that freezes until the data source is complete, a list that starts allowing interaction after a while, a list that allows interactions but freezes as new data is populated in bulks, a list that allows interaction as long as there aren't any updates, etc.

You will enumerate and present metrics for each of the lists; for example how long it took to run to completion, how long it took before interaction was allowed, a metric that captures whether the user interaction was interrupted by an update, a frozen vs interactive time measurement, memory consumed etc.

You will also be expected to come up with new list population approaches, and new relevant metrics.

At the end of the project there will be a demo and handoff session.

Practical details

Prerequisites

Some experience in programming.

Access to a development environment that can produce windows executables

Working routine

Weekly half-hour meetings, demos and feedback sessions online with students and mentor.

The mentor will be available for questions outside the weeklies.