VisiCalc: The Visible Calculator

Update 2017

1ST WIDELY POPULAR PC APPLICATION
VISICALC DAN BRICKLIN 1979
Boston & Arlington, Massachusetts
Harvard Business School student Dan Bricklin invented the first electronic spreadsheet in a lecture hall's front row.
He and partner Bob Frankston built VisiCalc in an Arlington apartment. It transformec the personal computer from a toy to a necessary tool.
FOUR CENTURIES OF INNOVATION
Pictured: Left: Bob Frankston, Right Dan Bricklin
Photo: Visicalc23.blogspot.com

VisiCalc is now enshrined in Massachusetts history as this display at Logan airport shows.

Introduction 2004

This paper was presented at the NCC (National Computer Conference) 1979 Personal Computing Festival.

Acknowledgments: I would like to thank Dan Bricklin of Software Arts for the original design of VisiCalc. I would also like to thank him and Dan Fylstra of Personal Software for their assistance in writing this paper. I thank Personal Software for its encouragement and support during the development of VisiCalc.

On June 3rd, 2009 Dan Bricklin posted two videos (Part 1 and Part 2) of me reading the paper after 30 years. In revisiting the document, I realized that the problems we faced making use of computing hardware in 1979 parallel today's problems in taking advantage of our connectivity infrastructure.

Note that this version was OCR'ed from a copy used in a case study at HBS. I will replace it with the original when I can.

In 1979 personal computers were purely hobbyist machines. You wouldn't see any in offices. While it wasn't uncommon to see computer screens on people's desks it was not the norm. Those who could afford it used specialized computers for word processing but most people used typewriters and the fancier typewriters allowed you to correct errors by using a sticky white ribbon to lift the plastic ink of the page. It was better than the competing technology -- white paint to hide the previous typing errors.

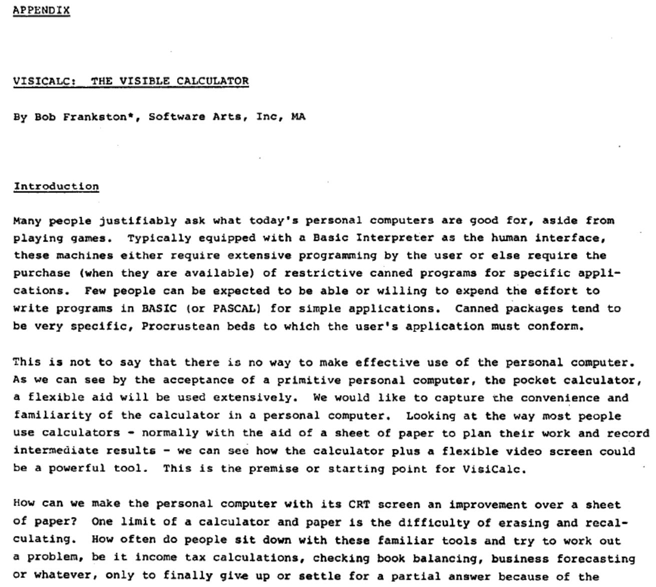
For those who grew up since then it is hard to imagine a world where computing and connectivity is not normal. A world with no email, no Web, no cell phones.

Yet this paper is still relevant -- while we've got all sorts of gadgets they are baubles that do specific functions but people still don't expect to be able to program them or, as I now describe it, add value. The phone is a phone and if you want a different feature you have to buy a different phone. You can't just program it do something new. As I note below and still point out in my current essays VisiCalc didn't know about any specific applications – people had to add their own expertise. If VisiCalc had more knowledge it would have been far less useful because it would have done one thing very well and would have been unable to do anything else.

At least with software you can use another program to do other tasks. The initial version of VisiCalc didn't have a graphics capability but you could transfer the data to a graphics program on the same machine.

With a gadget, you don't have that option -- each feature requires another gadget and even if you like to carry a dozen with you and buy clothing with enough pockets they can't cooperate. There's no synergy and arbitrary overlap. After trying to keep the coordinated you just give up. Think of how much more valuable each one would be if you had the ability to add your own knowledge, the ability to program in additional capabilities.

The VisiCalc Paper



APPENDIX

VISICALC: THE VISIBLE CALCULATOR

By Bob Frankston*, Software Arts, Inc, MA

Introduction

Many people justifiably ask what today's personal computers are good for, aside from playing games. Typically equipped with a Basic Interpreter as the human interface, these machines either require extensive programming by the user or else require the purchase (when they are available) of restrictive canned programs for specific applications. Few people can be expected to be able or willing to expend the effort to write programs in BASIC (or PASCAL) for simple applications. Canned packages tend to be very specific, Procrustean beds to which the user's application must conform.

This is not to say that there is no way to make effective use of the personal computer. As we can see by the acceptance of a primitive personal computer, the pocket calculator, a flexible aid will be used extensively. We would like to capture the convenience and familiarity of the calculator in a personal computer. Looking at the way most people use calculators - normally with the aid of a sheet of paper to plan their work and record intermediate results - we can see how the calculator plus a flexible video screen could be a powerful tool. This is the premise or starting point for VisiCalc.

How can we make the personal computer with its CRT screen an improvement over a sheet of paper? One limit of a calculator and paper is the difficulty of erasing and recalculating. How often do people sit down with these familiar tools and try to work out a problem, be it income tax calculations, checking book balancing, business forecasting or whatever, only to finally give up or settle for a partial answer because of the

By Bob Frankston, Software Arts, Inc, MA

Introduction

Many people justifiably ask what today's personal computers are good for, aside from playing games. Typically equipped with a Basic Interpreter as the human interface, these machines either require extensive programming by the user or else require the purchase (when they are available) of restrictive canned programs for specific applications. Few people can be expected to be able or willing to expend the effort to write programs in BASIC (or PASCAL) for simple applications. Canned packages tend to be very specific, Procrustean beds to which the user's application must conform.

This is not to say that there is no way to make effective use of the personal computer. As we can see by the acceptance of a primitive personal computer, the pocket calculator, a flexible aid will be used extensively. We would like to capture the convenience and familiarity of the calculator in a personal computer. Looking at the way most people

use calculators - normally with the aid of a sheet of paper to plan their work and record intermediate results - we can see how the calculator plus a flexible video screen could be a powerful tool. This is the premise or starting point for VisiCalc.

How can we make the personal computer with its CRT screen an improvement over a sheet of paper? One limit of a calculator and paper is the difficulty of erasing and recalculating. How often do people sit down with these familiar tools and try to work out a problem, be it income tax calculations, checking book balancing, business forecasting or whatever, only to finally give up or settle for a partial answer because of the difficulty of dealing with the recalculations necessary either to correct for errors or to try different starting values? How nice it would be if the paper and the calculator worked together so that whenever you changed anything on the paper the other values would automatically be recalculated.

VisiCalc addresses these needs. It is like a magic sheet of paper that can perform calculations and recalculations with ease. Basically, VisiCalc represents a very large sheet of paper organized into rows and columns of entries. By simply moving a cursor to an entry you can place values and expressions to be calculated on the screen. You can also type comments and titles. Of course, being limited to the current personal computers, you must view this sheet of paper through the window of the CRT screen. VisiCalc does, though, let you scroll through this 'window' in all four directions over the sheet of paper.

The benefits of VisiCalc go beyond its humble origins as calculating paper. It represents a way of using computers that allows the user to ask, 'what if' questions that would be too tedious to carry out by hand. Not only are such questions important in planning, they can be vital to the user in learning and coming to understand his own application.

Both the words 'serendipity' and 'synergy' are appropriate for VisiCalc. As VisiCalc evolved it showed us how effective personal computers can be as streamlined interactive tools. But, VisiCalc was not simply a lucky extrapolation of the basic ideas. Both authors of VisiCalc have extensive background in using large mainframes as personal computers and in creating systems to be used by large numbers of people with little training. Word processing background was of special importance since it provided experience in designing screen-based, highly interactive interfaces. More important for VisiCalc, it made us very aware of the need for a carefully designed and tuned user interface. This interface was constantly refined during the development process.

VisiCalc: a closer look

As noted, VisiCalc presents the user with a large sheet of paper on to which can be placed arithmetic expressions or text. While typing the expression on the screen the user can refer to other numbers or expressions by pointing to them. When the value of an entry is changed the values of the other entries which depend on it are automatically recalculated.

Though only the result of a calculation is displayed in an entry, by moving the cursor to the entry the user can see, at the top of the screen, the expression used in the calculation. The top of the screen also shows other status information.

We can see how VisiCalc works by looking at a simple home budgeting example. We can set up a row for each category in our budget, such as income, food, rent, heat, recreation and savings. We then set up the column for the first month, entering the values for each item. We might want to describe the savings as the income less the other expenses. We do this by typing the expression for savings as we would on a calculator, except instead of typing the numbers, we point to each item (income, food, etc). The value of savings is recalculated automatically whenever the value of another item is changed. We can extend the budget for twelve (or more) months by repeating this process for each month. Rather than tediously typing similar expressions, we use the 'replicate' command to copy our descriptions for the rest of the year.

Now that we have set up the personal budget we are ready to take advantage of VisiCalc and ask 'what if'. We might want to change the heating expenses to be higher in the winter and maybe increase the recreation expenses at vacation time. We could then see whether we would be able to save any money on our current income even if we paid more rent. Or maybe if we get a 5% raise we can afford that apartment that looks so tempting.

This ability to try out different values, to ask 'what if', corresponds to the simple requirements of a calculating sheet of paper. But VisiCalc is more. The 'paper' not only calculates, but can be manipulated by moving rows and columns as necessary. The sheet can be used to store lists of data such as 'phone numbers, with new entries being inserted manually in alphabetical order.

When you record your check-book in this manner, the calculator feature allows VisiCalc not only to store the checks, but also maintain the current balance as each check is entered. When the bank statement arrives, the checks listed can be moved on the sheet to correspond to the order on the statement, thus making it very easy to reconcile the statement and keep track of outstanding checks. This can be the basis of a simple home financial record-keeping system.

Other features of VisiCalc include the ability to sum lists of numbers and set up bar graphs. The user can also control the formatting of numbers and text, including establishing a second window on the screen which allows two different parts of the sheet to be viewed at once.

As you can see, VisiCalc can take care of the details of calculations and keep track of what data is where. It is important that the user be able to construct a solution to his or her problem simply by laying out the necessary information on the sheet. The user is then freed to work with the application. This work with an application

is in contrast to the traditional model of programming which requires that the problem be essentially worked out before the program is written.

The basic goal of VisiCalc is to provide a tool with which people can make effective use of the computer. It does not try to do everything and does require some effort on the part of the user in order to apply it to a specific application. Using VisiCalc may be likened to having a personal car in a world where the choice is either to build your own or depend on buses (canned applications) that probably do not go where you want anyway.

We can illustrate the concept of manual control by looking back at the budget example above. Instead of relying on a complicated budgeting program or having to sit down and plan an elaborate budgeting system, we just set up the calculations on a month by month basis and tried out different budgets. While we could have established a formula that tells us what rent we can afford for a given amount of income, we instead tried the different values. At the same time, we were able to see what would happen given more money, a cold winter, or additional expenses.

The important point is that VisiCalc allows the user to just solve the problem using familiar tools and concepts. This is in contrast to using a canned program that has a fixed idea of how one should go about budgeting. As the problems to be solved get more difficult, the task of producing a general, canned program becomes exceedingly difficult and expensive. Using VisiCalc, with a small amount of manual effort the user has a very effective and economical tool. Furthermore, the tool is more flexible than many of the more complicated systems and more amenable to trying out new ideas.

This emphasis on manual operation should not obscure the sophistication inherent in VisiCalc. The sophistication is necessary in order to make the system simple to use. One of the flaws in the current generation of personal computers is that their attempts to provide a simple computer shifts the difficulties in programming to the implementor of packages and to the user. This stems from a difficulty in finding the appropriate level of sophistication. We hope VisiCalc is at the appropriate level. While VisiCalc chooses to leave some of the decisions to the user, these are generally associated with the user's specific application rather than irrelevant programming considerations. Inevitably, a user who understands an application will find it easier to set it up than one who does not. But one has an opportunity to play with the application in VisiCalc and thereby learn.

The software problem

As personal computers become more common, the software problem becomes the limiting factor in the ability to use the personal computer. The current attempt to get every-one to learn and use BASIC is not the answer. Much of what is involved in programming is conceptualization and description - the hard parts of problem solving. Even the ultimate in general-purpose procedural languages will not remove the difficulties in programming. These languages are, of course, necessary for those people who do programming, such as the personal computer software engineers. But there are simply not enough programmers available to write all of the canned applications that can be anticipated.

Thus, our task as professionals becomes one of finding the appropriate level of tools that correspond to the level at which the user deals with an application. On large computers this has been done with database languages such as IBM's Query By Example (QBE). At MIT, the Macsyma system provides a powerful tool for doing symbolic manipulation of mathematical expressions. For personal computers, VisiCalc presents an interface to the user which builds upon that with which he is already familiar. It is, of course, constrained by the memory and processing limits of the current generation of personal computers.

Summary

VisiCalc brings the user a familiar, and powerful, combination of a calculator and paper which permits the user to make effective use of the personal computer without the over-head of programming nor the constraints of canned application packages. But VisiCalc is more than just a visible calculator, it is a modest database system that permits easy maintenance and manipulation of information.

VisiCalc illustrates a principle of application design: it takes care of memory, data structures and other matters meaningful only to programmers. It strikes a delicate balance of simplicity and sophistication. VisiCalc chooses to leave to the user those decisions which are generally associated with the user's specific application rather than irrelevant programming considerations.

REFERENCES

001 FRANKSTON R M

VisiCalc: the Visible Calculator

Paper presented at the NCC 1979 Personal Computing Festival and available from Software Arts Inc 675 Massachusetts Avenue Cambridge MA 02139

Acknowledgments

I would like to thank Dan Bricklin of Software Arts for the original design of VisiCalc. I would also like to thank him and Dan Fylstra of Personal Software for their assistance in writing this paper. I thank Personal Software for its encouragement and support during the development of VisiCalc