WINTER (89-90) LAST ISSUE

NO SPRING 90

Paul Smiths College
Paul Smiths NY 12970

Alfred J. Lata
Dept. of Chemistry
2010 Malott Hall
University of Kansas
Lawrence KS 66045

Editor:
Thanks to Lynn James for his help over the last few years and welcome Pat Flath and Alfred Lata as new Co-Chairs. I look forward to working with all three in the future.
We all owe thanks to Vic Bendall for his willingness to write up all the interesting work he has been doing. I understand he has recently begun using a IBM compatible, perhaps we can coax him into sharing any useful tricks he learns. How about it Vic?
NOTICES:
1) We can now accept submissions for either the Macintosh or IBM. ASCII works great on the Mac.
2) Since we are not getting many submissions, starting with this issue we will only publish when sufficient material arrives. No material no newsletter. It's up to you people. We can't expect Vic to fill every issue.
3)The US mail system failed to deliver the Fall issue to some subscribers. Additional copies are being sent.

UPDATE:
Dr. Pascal from Visible Software is now available for VAX/VMS. This version is identical to the MS-DOS version. Usual compile and editor commands are 4-10 times faster than DEC VAX Pascal. Please call 609-683-4386 for further information.

REVIEW:
I've had several requests to review Pagemaker, which is the program used to put this newsletter together. Since PM is the only publishing program I use I can only repeat opinions I've heard that PM is a bit easier to use, does better graphics, but has fewer of the bells and whistles a professional would need.

I've only used PM for this newsletter. Previous issues have been done on an IMB-60 , with a normal sized monitor. This issue is being done on a Mac IIx with a 20" screen. Wow, what a difference! Up to now we have been able to read only IBM diskettes, so doing the newsletter on an IBM made sense.

Let me give you a picture of how I put the newletter together. First I select the submissions to be published. I read and make any changes on hard copy submitted. You can do this in PM but a wordprocessor works better if you are making major changes. Then I try to read the article into PM. If PM can't read the file, it allows me to make several other attempts using different protocols. If this doesn't work I have a few utilities that are tried on the file. If it's still not readable some of our staff who are usually pretty good at this give it a try. Last resort is to have the material retyped using compatible software and hardware. (This always works.)

My experience is that about half the material submitted is unreadable. I'm finding that if I can read the diskette on the IBM I can read it directly with the Mac. The reasons the files aren't readable is because they are submitted from unsupported wordprocessors or often damaged diskettes (US Postal Service?). I have not been impressed with the IBM's ability to read ASCII but the Mac does fine (both reading from files using PM).

Once you've read your files into PM your ready to set up a publication. The menus available are just about the same on both machines and both use mice heavily. You either open a new or old publication. You can use either the mouse and menus or hold a command key down and type the appropriate letter. For most heavily used commands you can use the mouse or keyboard.

If your publication is going to have the same format it is worthwhile setting the parameters on the master sheets. This will allow all succeeding pages that are opened to have the same parameters. You can set the number of columns, type, font and many other parameters you probably don't need to use.

Although I had used many programs on an IBM before PM I found the interface different enough to be confusing. On the Mac, 8 of the 11 commands under file are ones I use in all other Mac programs . The only one of the 3 new ones that I use is 'Place', which places a given article into the publication. Since I use a Mac everyday this makes it easy to just bring PM up and work for awhile. This is because most of the commands are ones I use constantly in other programs, they are standard on the Mac. Although the order in which you enter files into PM doesn't affect the order in the publication, - 'Placing 'files does.

It is not really hard to add or delete articles, but some strange and unwanted things can happen, especially on a small monitor. To be able to show two pages at a time PM uses "greeking" which shows the position of text but not letters or words. You can really fool yourself if what you think you are 'Placing' and actually are 'Placing', are different files. You can flip up to a print size you can read but then you only see a small part of one page. It takes so long to do this that I often thought I had made a mistake and started pressing other buttons. This in turn would cause all kinds of problems. Not much fun.

On the large screen you can read the print with 2 pages up at one time. The Mac is very fast, and a pleasure to use. Of course an IBM compatible with a 80486 chip and a large screen would make life a lot easier. But the interface would still be different from other programs you use on the IBM and would require extra time to learn...

As you 'Place' articles you can flow the text around spaces you leave for graphics. You can also flow all the articles together to make one document or keep them separate. If all the articles are going to have the same format and you're expecting to make changes, having one big document is an advantage.

If you have one document you get essentially the same effects as using a wordprocessor. Make a deletion or insertion and everything reformats itself. Ifthey are separate articles then changes in one can cause it to overun a later article. Trying to hand format a half dozen articles is not fun ,is prone to lots of errors, and is very time consuming.

Once you have 'Placed' all your material you can select any amount and change parameters to your hearts content. Very easy to try changes out and see what document looks like.

Although you can certainly use PM as a wordprocessor, for anything over a few paragraphs I would use a regular word processor and 'Place' the finished document in PM .

If it sounds interesting you'll probably find it worthwhile to get some help as you start. An experienced colleague or a continuing ed course makes it a lot more pleasant to learn.

Victor Bendall
Eastern Kentucky University

Sticky Keyboards on the Laser128

The Laser 128 is the most successful of the cheap Apple II clones. Recently, however, we have been plagued with keys that either did not respond when depressed or demanded several successive taps before reluctantly responding. Since these computers are out of warranty, we decided to try to fix them ourselves. It turns out to be a simple operation to do despite the label on the base which says "no user serviceable parts inside."

You will need a fine bladed screwdriver, a Phillips screwdriver, diagonal pliers, four 2 inch blocks, and several pencils with clean erasers on them. The following step-wise procedure should be followed.

1. Turn off the computer and remove all attachments including power and monitor leads.

2. Turn the computer over and remove the nine screws which hold the case together. There is a screw in each corner, one in the center of the front and sides and two shorter ones next to where the handle is attached to the case.

3. Turn the computer right side up again. Turnit so that the keyboard is away from you. You are about to remove the plastic strip that covers the rear of the computer. It's the strip with the red warning label on it and the raised letter descriptions of the various points of attachment.

4. The plastic strip is held in position with small clips which engaged as the strip was pressed into place. Release the right hand third of the strip by putting the fine-bladed screwdriver into the right edge and prying it back. There are two clips on that edge which should come loose. Now pry the top of the right third and release it.

5. Now do the same to the left third of the strip to release it.

6. There are two clips in the top of the middle third and one at its base. Pry with the screwdriver to release them. The plastic strip will come off.

7. Note how the handle rests in slots in the case. That is how it must be replaced when the case is reassembled.

8. Turn the computer so that the keyboard is before you and lift off

the top cover.

9. Lift the keyboard unit and slide it towards you an inch or so. Look down behind it and you will see where it is attached to the computer unit with four white plugs sitting side by side. Disconnect the plugs. Use the blade of the screwdriver to help loosen the plugs and slide them off the pins.

10. It is best to work on the keyboard without pressure on the keys. Turn the unit so that the keys are underneath and support it at each corner on the four blocks. There's a rubber block stuck to one edge of the unit. Take care not to knock it off. [It's not vital if you do remove it - the computer will work without it].

Victor Bendall
Eastern Kentucky University

An Animation of Distillation
Part IV
The Falling Drop

In this part of our series on the animation of a distillation I shall discuss the display of a drop of liquid falling and adding into a liquid pool. The coding will be designed to integrate with the stirrer and flame animations of the earlier parts. This animation is a more ambitious task than the earlier routines and will rely more heavily on the internal CHEMUTIL-2 (1) routines.

The figure 1 shows the essentials of the animation and illustrates some of the terms used. A drop will appear to fall from location TOPDRP to location BOTDRP, then a transient splash will appear and the liquid pool can increase by drawing a line at LIQY. The topmost image of a drop will be displayed at TOPDRP so we first define that in terms of VTAB and HTAB. Earlier we displayed flame and stirrer by defining new characters and printing them with the CHEMUTIL-2 character generator. It is convenient to do the same for the falling drop. Two new characters are defined using CHARACTER (2) and stored in character set 2 immediately following the previously defined stirrer representations at $73DO-$73DF. The simplest way to do that is to BLOAD CHEMUTIL-2 or better the modified version after running Listing 1 of Part II of this series. Next in order CALL -151 : type 73DO : 00 1C 08 00 00 00 00 00 00 00 00 1C 1C 08 00 : CTRL C : RETURN : BSAVE CHEMUTIL-2, A $ 6000, L$ 1 C00. You will now have saved the modified CHEMUTIL-2 to disk with two representationd of the same drop as ASCII (90) and ASCII (91) in character set 2. If the second image is rapidly printed over the first, the illusion of a falling drop will result.

It was most convenient to display images of flame, stirrer and the drop using the character generator. The splash cannot be done conveniently in the same manner. The reason is that a character can be displayed only at one of the screen locations defined by VTAB (0-23)* and HTAB (0-39) but the splash must appear to be at the surface of the pool and the Y-coordinate (LIQY) can have values from 0-191. Splash is done by directly poking values into appropriate bytes of screen memory. The Y-coordinate and HTAB are used together to find the corresponding screen memory address. There is a routine in CHEMUTIL-2 which does that using look-up tables. Into each of three screen memory bytes located vertically above one another is poked numerical values which result in the splash representation appearing on the monitor. The same three bytes are then set to zero and the splash representation is blanked out. These operations are very fast and a delay has to be introduced between drawing the splash and erasing it.

Every time the liquid pool increases by eight vertical bytes, the value of BOTDRP VTAB must be decreased by one or the bottom drop will blank some of the liquid. This means that the distance between the topmost drop and the liquid level decreases and so does the total number of drop images to be displayed. When TOPDRP = BOTDRP only one or two drop images will be shown. In our coding we cannot allow the liquid level to rise above TOPDRP or, if it does, the dropping routine must be bypassed at that time.

As in our earlier routines we allow for stopping the routine by pressing the S-key at the keyboard and the delay loops are introduced to control the speed of the animation. The delays are inserted to control how long a given image remains on the screen; in this case, while the drop or splash representation is on the screen. the delays are here because we expect that the flame and stirrer animations will have to be activated while drops are falling in the full distillation animation. There is more than one delay because we may need to have flame and stirrer alternate between falling drops or have one appear while a drop is being displayed and the other appear while a drop is being blanked out.

With three places where the timing of the animation can be tuned it would seem that ample provision for adjustments has been made. If the program given below is executed you should see a series of falling drops which periodically cause the liquid level to rise. However, the apparent speed of the drop will increase as the eight lines of liquid are drawn. This occurs because fewer drops are drawn as the distance between TOPDRP and BOTDRP decreases. There is proportionately less for each prgram loop to do between successive splashes and the animation speeds up. To allow for this effect one or more of the delay intervals would have to be adjusted every eighth time the liquid level is incremented. This adjustment is left as an exercise for the reader.

Because of the way the coding was structured, it is easy to make minor changes. For example, you may wish to simulate an apparatus at reflux by having the drop splash but not have the liquid pool increases. You need to eliminate the LIQDRAW routines by poke 31962 ($7CDA), 96. To eliminate the splash you should poke 32042 ($D2A), 96.

*This statement is not quite

correct since CHEMUTIL-2 easily allows super and subscripted characters which effectively doubles the number of vertical locations available. Indeed, more sophisticated CHEMUTIL-2 users can make the character generator access all the vertical coordinate locations.

(1) Bendall,V. "CHEMUTIL-2, A Chemistry Programming Utility"; Project SERAPHIM, NSF Science Education; Department of Chemistry, Eastern Michigan University, Ypsilanti, MI 48197, 1985.

(2) Bendall,V. "CHHARACTER"; Project SERAPHIM, NSF Science Education; Department of Chemistry, Eastern Michigan University, Ypsilanti, MI 48197, 1985.

Victor Bendall
Eastern Kentucky University

An Animation of Distillation
Part IV
The Falling Drop

This article includes the complete source code for the animated portion of a distillation graphic for an Apple II microcomputer. A BASIC listing is also included that draws the static surrounding graphic and CALLS the machine code that animates the flame, stirrer and falling drops described in the previous articles. There are some additional considerations when combining the operations that may not be immediately obvious from the code alone.

This routine does not use an internal clock directly to control the overall speed of the animation because there is no clock available in an unmodified Apple II microcomputer. The timing of the total animation is governed by how long it takes each individual animation loop to be executed. We can choose to flicker the flame x times for each drop falling or to have x drops falli for each flicker of the flame. In practice it is easier to sum several short intervals to make one long one than to divide the long interval into several short ones. This means that if the execution time for one flicker is shorter than that for a falling drop then it is easier to code , for example, seven flickers per drop than to code one seventh drop per flicker. The general rule for a complex animation is to use the loop with the shortest execution time as a standard. In our case, we can control the execution time for flicker or drop by using the delay loops but it seems more desirable to have fast flickering than high speed falling drops so that our code reflects the x flickers per drop option.

This same option has another advantage. It becomes very simple to not only show no drops but also to omit the drop loop completely. The x drops per flicker option makes it easy to show drops and no flicker but not the reverse. While we cannot be certain how this animation code may be used in a future application, it seems reasonable that we are more likely to want to retain the flame and omit the drops then the reverse. In the final code we use the flicker loop as the standard and the stirrer and drop routines are subsidary it.

Once a drop begins to fall the speed that it does so depends upon the execution time of the appropriate loop but that in turn depends upon the speed of the flame and stirrer loops. The falling drop, flame and stirrer loops are controlled by delay loops so that there is ample room for controlling the drop speed. However, we also need to be able to control how often a drop falls, There will be times when we want the liquid to distill slowly, or perhaps to reflux rapidly, and now is the time to code for possibilities. This combined code uses a new variable FREDRP to take care of them. By adjusting FREDRP and the old variable DRPCNT it is now possible to have fine control on the apparent rate of accumulation of ditillate.

The DRPIT code in this listing has been reorganized and should be compared with the DROPIT routine given in Part IV. The first few lines of the old code erased the last drawn drop of liquid displayed by the same loop the previous time it was executed. In that code only a delay lop was executed between successive executions of the drop loop. The result was satisfactory provided a new drop began to fall immediately after the old one splashed in the distillate. In the new code, the FREDRP variable was introduced so that there is a deliberately long interval between drops. The old code would now shhow a drop falling, then jumping back to the top position and hanging there. The reorganized code erases the last drawn drop before leaving the DROPIT loop and the undesirable effect is eliminated. This illustrates another point. In itself the old DROPIT code was satisfactory. It was only when it was combined with other individually satisfactory animation routines that a flaw showed up. However, our code was well planned initially so that the change made to correct the flaw was relatively simple.

DROPIT has also been modified so that DRPCNT is used more rationally. In the earlier version DRPCNT was incremented each

time the DROPIT loop was executed. This meant that it did not count the number of drops directly. In the new code, DRPCNT is incremented each time a SPLASH is drawn. This gives a more rational control of the increase of the liquid pool since DRPCNT is now independent of the height through which the drop falls.

The source code for the complete animation should be entered through the Apple monitor starting at $ 7C15. The BASIC program uses that code and CHEMUTIL-2 (1) to demonstrate the complete distillation in action. Your copy of CHEMUTIL-2 should have the extra characters added as described in the earlier parts of this series. If those characters have not been added the animation still works but the stirrer will be jumpy, lines will appear in the heated liquid and the falling drops will not be of realistic shape.

This concludes this series. A 5 1/4 disk with source code and BASIC listings of all programs listed in these five articles is available from the author for $10.

(1) Bendall,V. "CHEMUTIL-2, A Chemistry Programming Utility"; Project SERAPHIM, NSF Science Education; Department of Chemistry, Eastern Michigan University, Ypsilanti, MI 48197, 1985.