THE HP-28S, CALCULUS, AND PROGRAMMING

John Selden, Annie Selden,
Michael Morrison, Randy Weems and Shawn Yeager
Tennessee Technological University
Department of Mathematics
Cookeville, Tennessee 38505

In this paper the first two authors report on a yearlong experimental course in calculus using the HP-28S calculator, which led naturally to a one quarter course in algorithm design and implementation. Some of the resulting programs, produced by the final three student authors, are presented.

The calculus course was first taught in 1988-89 and is being repeated this year. It covers the traditional syllabus and is the usual size for our university, about 30 students. Most other aspects of the course are nontraditional, however, so we mention these before discussing the way in which the calculator is used.

The students are all freshman volunteers, recruited during the summer orientation and preregistration. Most are engineering majors; about half have studied at least some calculus before. All students have access to an HP-28S and attend class seven, instead of the normal five, hours per week. We think now that the extra class time is not essential, but it seemed advisable at first, given the experimental setting. The course provides much more than the normal amount of practice solving novel problems, that is, problems for which the students have not seen solution methods. We do not attempt to add novel problems to an already overcrowded syllabus. Instead, we make the normal problems novel by removing the sample solutions found in calculus books.

Because traditional calculus books contain too many sample solutions and detailed algorithms for solving problems, we do not use a book. Instead we use notes. The students solve the problems found in the notes and present their original solutions in class. To allow time for this, we do not lecture. However, we do give detailed criticisms and comments on student work, and at times, these develop naturally into mini-lectures. Our tests, too, consist entirely of novel problems, but they contain only four problems each, done over two class periods. The exam problems are less novel to encourage review. On both tests and exams, students are allowed and encouraged to use their notes and calculators.

Why teach in this way? First, it has been widely observed that the traditional calculus course is not producing the desired results. Not only do many apparently successful students fail to appreciate the link between theory and problems, but also many have a poor grasp of the fundamental concepts. In addition, calculus students are rarely able to solve novel problems [2], an important ability for anyone interested in applications.

The second answer is more philosophical. This method of teaching allows us to find out what the students are thinking. That such insight is an important part of effective teaching is a corollary of the view that students actively construct their own knowledge, rather than passively receiving it. For more on this constructivist idea of learning, see, for example, [1] or [4].

Why are we using the HP-28S? On the one hand, it is wasteful to teach students to carry out tasks that can be done by widely available machines. On the other hand, we are interested in improving conceptual understanding and the computing power of the calculator allows students to examine a wider range of examples from which to gain insights.

We turn now to some observations. Problem solving is inherently interesting and our preliminary surveys indicate that the students prefer this course to a traditional one. Their attitude towards mathematics also seems improved. In early comparison tests, our students did as well as, or better than, traditional students on both novel and routine problems. The tests covered only the first quarter of calculus and were given five months after the end of the quarter. Notes, books, and calculators were not allowed on these comparison tests.

Although we have no formal evidence for this, use of the HP-28S does seem to improve conceptual understanding, at least in areas associated with graphing. Very little class time need be devoted to the calculator, provided programming is not required–perhaps two hours per quarter. Given this amount of instruction, our students can find derivatives and definite integrals and solve equations, such as $cos\ x - x = 0$. They can also enter programs we provide, for example, programs for Newton's method, implicit differentiation and graphing [3]. We do not ask students to write programs or carry out complicated calculator procedures, as this might easily require enough time to interfere with their study of mathematics. However, some students, perhaps one third, developed an interest in programming the HP-28S in their spare time. This led naturally to the creation of a third quarter course, called "algorithm design," to distinquish it from a course in which only the fundamentals of programming are taught.

The algorithm design course met three hours per week and was taught like a research seminar. That is, the students worked independently outside of class. In class they presented and discussed their work and received advice and suggestions. It appears that freshman respond to the opportunity to create new and useful results with as much enthusiasm and creativity as graduate students. Of course, it is unusual for undergraduates to have enough knowledge for such opportunities to arise, especially in mathematics.

The class produced programs for scrolling, drawing, polar graphing, and directory handling. It also produced SPACE CHASE, an action game. We include some of these programs here. To save space, we omit the documentation.

For scrolling we have three programs, WRAP, Disp and MVS. WRAP and Disp both take a string from the stack. WRAP returns the string to the stack, but with line breaks inserted at appropriate places between words, so that a message scrolls down the display without splitting words. Line breaks already present in the original string are retained. In addition, Disp requires a line number, indicating where the display will start. It displays the string, without returning anything to the stack, and permits scrolling up and down using the curser keys. MVS is called by Disp.

```
WRAP ≪ "" WHILE OVER SIZE 24 > REPEAT OVER DUP IF 1 24 SUB 10 CHR
POS DUP NOT THEN DROP 25 DO 1 - DUP2 DUP SUB UNTIL " " == END
```

END DUP → j ≪ 1 - 1 SWAP SUB 10 CHR + + SWAP j 1 + ≫ OVER SIZE SUb SWAP END SWAP + ≫

MVS ≪ DUP 10 CHR POS → j ≪ DUP 1 j SUB ROT + SWAP j 1 + OVER SIZE SUB ≫ ≫

Disp ≪ → s ≪ "" DO OVER 10 CHR DUP DUP + + + s DISP {≪ SWAP MVS SWAP 0 ≫ ≪ MVS 0 ≫ ≪ 1 ≫ } DO KEY UNTIL END {"DOWN" "UP"} OVER + SWAP POS GET EVAL UNTIL END DROP2 CLMF ≫ ≫

For drawing, we have LINE, similar to a program in [5], and DOODL. LINE is not used directly but is called by DOODL, which draws and erases both points and lines. To run DOODL, the name of a variable in single quotes must be on level one of the stack. If the variable has not yet been created, DOODL will create it. The variable should contain a string of 500 characters representing the display. Such strings can be displayed by the → LCD command.

When DOODL is run, it takes the variable from the stack and displays its contents as a picture. The cross hairs can then be moved to some point on the display. To add this point, or remove an existing point, press INS followed by ON. To add or erase a line, press INS at both ends of the line, followed by ON. To stop the program, press ON without first pressing INS. This new version of the display will be stored in the original variable.

LINE ≪ → n ≪ OVER PIXEL IF DUP2 == THEN DROP PIXEL ELSE OVER - PPAR DUP 2 GET SWAP 1 GET - C→R 31 / SWAP 136 / 3 PICK C→R 4 ROLL / ABS SWAP ROT / ABS MAX PPAR 4 GET / SWAP OVER / ROT 1 4 ROLL START OVER + DUP PIXEL NEXT END ≫ ≫

DOODL ≪ → n ≪ n IFERR RCL THEN CLLCD LCD→ DUP ROT STO END WHILE →LCD DGTIZ CLLCD DEPTH DUP 0 ≠ REPEAT IF 2 == THEN n LINE ELSE PIXEL END CLEAR LCD→ n RCL XOR DUP n STO END CLLCD "   * * Saved * * " 2 DISP DROP .5 WAIT CLMF 'PPAR' PURGE ≫ ≫

SPACE CHASE is an action game which demonstrates the calculator's ability to handle real time animation. The programs SETUP and L→LCD must be entered first. SETUP will call L→LCD and create the ships, background, and high score files. SETUP and L→LCD can be purged after SETUP has been executed. The remaining programs are SPACE, HISCR, UP, DOWN, RIGHT, LEFT, SHIFT,and INS. SPACE calls all of the other programs except HISCR. Executing HISCR gives the current high score and the display associated with it.

The object of the game is to move your space ship so as to avoid contact with "alien" space ships, which move randomly. The direction keys control your ship's movement. The shift key can be used just three times in a game to move your ship discontinuously out of threatening positions. The INS key can be used to pause in the game. Play is resumed by pressing ON.

SETUP ≪ { 255 24 24 36 102 90 90 102 36 24 24 255 } 137 L→LCD 'SHIP1' STO { 64 97 250 228 228 228 228 228 250 97 64 } 200 L→LCD 'SHIP2' STO { 16 24 24 30 57 57 63 57 57 30 24 24 16 } 261 L→LCD 'SHIP3' STO CLLCD {(0,0) (136,31) X 1 (0,0)}

```
'PPAR' STO 1 30 START RAND 137 * FLOOR RAND 32 * FLOOR R→C PIXEL
NEXT 'PPAR' PURGE LCD→ 'BACK' STO {0 ""} 'HI' STO ≫
```

L→LCD ≪ 0 CHR 1 9 START DUP + NEXT 1 ROT SUB SWAP LIST→ "" 1 ROT START SWAP CHR + NEXT + CLLCD →LCD LCD→ ≫

SPACE ≪ # 11CAh SYSEVAL 0 SHIP2 SHIP1 SHIP3 WHILE 3 DUPN OR DUP2 OR DUP BACK OR →LCD XOR == REPEAT { UP DOWN LEFT RIGHT LEFT } RAND 5 * CEIL GET EVAL ROT IF KEY THEN STR→ IF DUP TYPE 2 ≠ THEN DROP END END 512 .01 BEEP ROT { UP DOWN LEFT RIGHT RIGHT } RAND 5 * CEIL GET EVAL ROT END 1 4 START LCD→ NOT →LCD NEXT 4 DROPN 50 1 BEEP WHILE KEY REPEAT DROP END # 11CAh SYSEVAL SWAP - B→R 819.2 / IP DUP LCD→ CLLCD SWAP 4 DISP LCD→ OR →LCD IF DUP HI 1 GET ≥ THEN LCD→ 2 →LIST 'HI' STO ELSE DROP END ≫

HISCR ≪ HI 2 GET →LCD ≫

UP ≪ DUP 138 548 SUB SWAP 1 137 SUB + ≫

DOWN ≪ DUP 412 548 SUB SWAP 1 411 SUB + ≫

RIGHT ≪ DUP 543 548 SUB SWAP 1 542 SUB + ≫

LEFT ≪ DUP 7 548 SUB SWAP 1 6 SUB + ≫

SHIFT ≪ 4 ROLL 1 + DUP 5 ROLLD IF 3 ≤ THEN DUP RAND 547 * SWAP OVER CEIL 548 SUB 3 ROLLD 1 SWAP FLOOR SUB + 512 1500 FOR x x .015 BEEP 50 STEP END ≫

INS ≪ # 11CAh SYSEVAL DEPTH → d ≪ DGTIZ WHILE DEPTH d ≠ REPEAT DROP END # 11CAh SYSEVAL SWAP - 6 ROLL + 5 ROLLD ≫ ≫

References

1. Schoenfeld, Alan H. "Cognitive science and mathematics education: an overview." In A. H. Shoenfeld (Ed.), *Cognitive Science and Mathematics Education* (pp. 1-31). Hillsdale, NJ: Lawrence Erlbaum, 1987.

2. Selden, John, Alice Mason and Annie Selden. "Can average calculus students solve nonroutine problems?" *The Journal of Mathematical Behavior* 8 (April 1989): 45-50.

3. Selden, John and Annie Selden. "Graphing with the HP-28S." *The College Mathematics Journal* 20 (November 1989): 424-433.

4. von Glasersfeld, Ernst. "Learning as a constructive activity." *Proceedings of the Fifth Annual PME-NA*, Vol. 1 (1983): 41-69.

5. Wickes, William C. *HP-28S Insights.* Corvallis OR: Larken Publications, 1988.