Draw Poker for the TI-59

Lee Boyle, 2702 E Linden, Tucson AZ 85716

The program presented here is very close to the limits of the TI-59 programmable calculator because it fills 480 program memory locations and uses 53 data registers. When writing such long programs, there are four interrelated constraints that you must consider:

1. The TI-59 has an absolute limit of 960 memory locations. This number becomes smaller as data registers are allocated.
2. Zero to 100 data registers may be allocated; however, each register subtracts 8 memory locations from the available number.
3. Each call to the random-number generator (using the keystrokes PGM 15 SBR DMS) requires 1.4 seconds. Occasional sacrifices must be made in the interest of speed.
4. Results—if the program doesn't perform the task, nothing else matters. There are, however, always some compromises to be made.

Finding the right compromises and optimizations to make a large program fit into the available memory is always a challenge and it is a good way to refine your programming technique.

The game of Draw Poker can be divided into four major sections: dealing, evaluating the hand, drawing new cards, and betting. In this program, betting takes place only after a draw.

Deal (Key E)

The basic task of this section of the program is to randomly select eighteen numbers, with no repetition, from a set of fifty-two. These numbers must then be stored as five cards for the calculator, five for the user, and eight to be saved in a *draw stack* that will allow each party to exchange a maximum of four cards. The cards are stored in the form RR.S, where RR represents the rank (deuce through ace), and S represents the suit.

One method of drawing without repetition is to assign a data register to each card possibility and, when a card is chosen, to cross it off by storing a 1 in that register. The table is consulted as each card is dealt, to make sure that it has not been chosen before. This method is short, quick, easy, and random; unfortunately, the table requires fifty-two data registers, and we do not have that much memory to spare.

As a compromise, this program uses one register for each of the thirteen ranks, counting to assure that none is used more than four times. Suits are determined by the value 0.1, 0.2, 0.3, or 0.4, randomly stored in each register individually. Each time a card is chosen and the corresponding register is incremented, the fractional part is also incremented (e.g., 0.5 is reset to 0.1) so that the next card chosen from that rank will be of a different suit.

This method has the disadvantage of allowing you to predict the suit of the next card chosen from a given rank, but this is rarely, if ever, useful information.

When all eighteen cards have been dealt, subroutine 012 of Master Library program 1 is used to erase the dealing table. Subroutine SUM then uses this space to create a *tally table* that records the number of cards of each rank dealt to the calculator. Finally, subroutine A′ is called to display your hand.

Evaluation (Subroutine FIX)

In evaluating a poker hand, the calculator must test for three conditions:

Listing 1: *Draw Poker for the TI-59 programmable calculator.*

```
000   76 LBL
001   58 FIX
002   22 INV
003   86 STF
004   01  01
005   00  0
006   42 STO
007   05  05
008   04  4
009   03  3
010   42 STO
011   01  01
012   43 RCL
013   24  24
014   69 OP
015   10  10
016   42 STO
017   03  03
018   01  1
019   01  1
020   42 STO
021   00  00
022   01  1
023   03  3
024   42 STO
025   06  06
026   76 LBL
027   85  +
028   69 OP
029   20  20
030   01  1
031   32 X:T
032   73 RC*
033   00  00
034   77  GE
035   87 IFF
036   00  0
037   42 STO
038   03  03
039   61 GTO
040   65  ×
041   76 LBL
042   87 IFF
043   43 RCL
044   00  00
045   42 STO
046   02  02
047   01  1
048   44 SUM
049   03  03
050   05  5
051   32 X:T
052   43 RCL
053   03  03
054   22 INV
055   67  EQ
056   55  ÷
057   43 RCL
058   00  00
059   42 STO
060   04  04
061   08  8
062   44 SUM
063   01  01
064   76 LBL
065   55  ÷
066   02  2
067   32 X:T
068   73 RC*
069   00  00
070   22 INV
071   77  GE
072   65  ×
073   85  +
074   01  1
075   95  =
076   44 SUM
077   01  01
078   43 RCL
079   00  00
080   87 IFF
081   01  01
082   43 RCL
083   48 EXC
084   04  04
085   76 LBL
086   43 RCL
087   42 STO
088   05  05
089   04  4
090   07  7
091   32 X:T
092   43 RCL
093   01  01
094   22 INV
095   67  EQ
096   65  ×
097   86 STF
098   01  01
099   76 LBL
100   65  ×
```

Listing 1 continued on page 436

Listing 1 continued:

```
101   97 DSZ
102   06  06
103   85  +
104   43 RCL
105   25  25
106   22 INV
107   59 INT
108   32 X:T
109   02  2
110   05  5
111   42 STO
112   07  07
113   04  4
114   42 STO
115   06  06
116   76 LBL
117   97 DSZ
118   69 OP
119   27  27
120   73 RC*
121   07  07
122   22 INV
123   59 INT
124   22 INV
125   67  EQ
126   68 NOP
127   97 DSZ
128   06  06
129   97 DSZ
130   01  1
131   44 SUM
132   01  01
133   43 RCL
134   02  02
135   42 STO
136   04  04
137   76 LBL
138   68 NOP
139   73 RC*
140   01  01
141   92 RTN
142   76 LBL
143   15  E
144   01  1
145   03  3
146   42 STO
147   00  00
148   01  1
149   01  1
150   42 STO
151   01  01
152   76 LBL
153   10 E'
154   69 OP
155   21  21
156   01  1
157   71 SBR
158   88 DMS
159   04  4
160   95  =
161   59 INT
162   55  ÷
163   01  1
164   00  0
165   95  =
166   72 ST*
167   01  01
168   97 DSZ
169   00  00
170   10 E'
171   01  1
172   08  8
173   42 STO
174   00  00
175   02  2
176   05  5
177   42 STO
178   01  01
179   04  4
180   32 X:T
181   76 LBL
182   45 Y^
183   01  1
184   02  2
185   71 SBR
186   88 DMS
187   01  1
188   03  3
189   95  =
190   59 INT
191   42 STO
192   02  02
193   73 RC*
194   02  02
195   77  GE
196   45 Y^
197   22 INV
198   59 INT
199   65  ×
200   01  1
201   00  0
202   95  =
203   22 INV
204   67  EQ
```

Listing 1 continued:

```
205   23 LNX
206   93  .
207   04   4
208   22 INV
209   74 SM*
210   02   02
211   04   4
212   76 LBL
213   23 LNX
214   55  ÷
215   01   1
216   00   0
217   85  +
218   43 RCL
219   02   02
220   95  =
221   72 ST*
222   01   01
223   01   1
224   93  .
225   01   1
226   74 SM*
227   02   02
228   69 DP
229   21   21
230   97 DSZ
231   00   00
232   45 Y^x
233   43 RCL
234   09   09
235   32 X↕T
236   02   2
237   04   4
238   36 PGM
239   01   01
240   71 SBR
241   00   00
242   12   12
243   32 X↕T
244   42 STO
245   09   09
246   01   1
247   71 SBR
248   44 SUM
249   02   2
250   09   9
251   42 STO
252   03   03
253   76 LBL
254   16 A'
255   09   9
256   35 1/X
257   91 R/S
258   05   5
259   42 STO
260   07   07
261   76 LBL
262   66 PAU
263   69 DP
264   23   23
265   73 RC*
266   03   03
267   75  -
268   01   1
269   00   0
270   95  =
271   66 PAU
272   66 PAU
273   66 PAU
274   97 DSZ
275   07   07
276   66 PAU
277   92 RTN
278   76 LBL
279   44 SUM
280   74 SM*
281   25   25
282   74 SM*
283   26   26
284   74 SM*
285   27   27
286   74 SM*
287   28   28
288   74 SM*
289   29   29
290   92 RTN
291   76 LBL
292   88 DMS
293   85  +
294   36 PGM
295   15   15
296   71 SBR
297   88 DMS
298   65  ×
299   92 RTN
300   76 LBL
301   13   C
302   85  +
303   02   2
304   09   9
305   95  =
306   42 STO
307   00   00
308   00   0
```

Listing 1 continued on page 438

Listing 1 continued:

```
309   72 ST*
310   00  00
311   91 R/S
312   76 LBL
313   14  D
314   71 SBR
315   58 FIX
316   29 CP
317   22 INV
318   67  EQ
319   78 Σ+
320   43 RCL
321   02  02
322   42 STO
323   04  04
324   00  0
325   76 LBL
326   78 Σ+
327   32 X↕T
328   00  0
329   42 STO
330   03  03
331   01  1
332   94 +/-
333   71 SBR
334   44 SUM
335   03  3
336   22 INV
337   77  GE
338   48 EXC
339   02  2
340   05  5
341   42 STO
342   07  07
343   05  5
344   42 STO
345   02  02
346   76 LBL
347   42 STO
348   73 RC*
349   07  07
350   59 INT
351   32 X↕T
352   43 RCL
353   04  04
354   67  EQ
355   35 1/X
356   43 RCL
357   05  05
358   67  EQ
359   35 1/X
360   00  0
361   72 ST*
362   07  07
363   01  1
364   44 SUM
365   03  03
366   76 LBL
367   35 1/X
368   69 DP
369   27  27
370   97 DSZ
371   02  02
372   42 STO
373   76 LBL
374   48 EXC
375   43 RCL
376   03  03
377   91 R/S
378   01  1
379   00  0
380   42 STO
381   00  00
382   02  2
383   04  4
384   42 STO
385   02  02
386   03  3
387   04  4
388   42 STO
389   07  07
390   76 LBL
391   93  .
392   29 CP
393   69 DP
394   22  22
395   73 RC*
396   02  02
397   22 INV
398   67  EQ
399   80 GRD
400   69 DP
401   27  27
402   73 RC*
403   07  07
404   72 ST*
405   02  02
406   76 LBL
407   80 GRD
408   97 DSZ
409   00  00
410   93  .
411   01  1
412   71 SBR
413   44 SUM
414   02  2
415   09  9
416   42 STO
417   03  03
418   71 SBR
419   16 A'
420   91 R/S
421   71 SBR
422   58 FIX
423   65  ×
424   43 RCL
425   04  04
426   85  +
427   01  1
428   00  0
429   71 SBR
430   88 DMS
431   05  5
432   95  =
433   32 X↕T
434   25 CLR
435   76 LBL
436   38 SIN
437   85  +
438   05  5
439   95  =
440   91 R/S
441   22 INV
442   77  GE
443   38 SIN
444   02  2
445   00  0
446   22 INV
447   77  GE
448   11  A
449   25 CLR
450   91 R/S
451   76 LBL
452   11  A
453   73 RC*
454   01  01
455   22 INV
456   59 INT
457   85  +
458   43 RCL
459   04  04
460   75  -
461   01  1
462   00  0
463   95  =
464   91 R/S
465   02  2
466   04  4
467   42 STO
468   03  03
469   71 SBR
470   16 A'
471   91 R/S
472   15  E
473   76 LBL
474   12  B
475   75  -
476   05  5
477   95  =
478   91 R/S
479   11  A
```

LABLES

```
001   58 FIX
027   85  +
042   87 IFF
065   55  ÷
086   43 RCL
100   65  ×
117   97 DSZ
138   68 NOP
143   15  E
153   10 E'
182   45 Y×
213   23 LNX
254   16 A'
262   66 PAU
279   44 SUM
292   88 DMS
301   13  C
313   14  D
326   78 Σ+
347   42 STO
367   35 1/X
374   48 EXC
391   93  .
407   80 GRD
436   38 SIN
452   11  A
474   12  B
```

Listing 1 continued:

```
DATA

          0.        43
    5.88888         44
          0.        45
    1.00011         46
    3.00111         47
    7.01111         48
    2.11022         49
    4.11122         50
    6.12345         51
8.12345888          52
```

1. Five cards in sequence by rank define a *straight.*
2. Multiple cards of the same rank define *2, 3, or 4 of a* kind.
3. Five cards of the same suit define a *flush.*

To save steps, the first two of these conditions are tested simultaneously by stepping through the tally table rank by rank. If five sequential cards are found, 8, the code for a straight, is added into R01 (data register 01). If a tally value is 2, 3, or 4, the hand will contain the corresponding multiple, and the appropriate code (3, 4, or 5, respectively) is added into R01.

To test for a flush, the program compares the fractional parts of the five cards; if they are all equal, the hand contains a flush (code=1).

Since half of the possible poker hands contain more than one of these conditions, provisions must be made to allow the results to be combined. For example, if a pair and three of a kind are found in the same hand, the calculator must recognize that it has a full house. To reduce the amount of program logic required, the poker hands have been assigned code numbers that can be added to give the total value of the hand. For example, the codes for a pair and for three of a kind are 3 and 4 respectively, so the code 7 denotes a full house. Similarly, 8 (straight)+1(flush)=9(straight flush). In each case, the details of the hand (pair of what?) are stored in R04 and R05.

The final steps of this program section use the table in R44 to R52 to translate the value in R01 into a new code giving the relative value of the hand (0 through 8) as the integer part and a symbolic representation of the hand as the fractional part (see table 1).

Draw (Keys C and D)

This section is relatively simple. Indicate those cards you wish to discard (if any) as 1 for the first card, 2 for the second, etc. You are trusted to discard no more than four cards.

The program then calls subroutine FIX to evaluate its hand. If the value is four-of-a-kind or better, no discard is made. If no hand is discovered, the calculator simply discards all but the high card. Otherwise, the rank of each card is compared to the values in R04 and R05 to determine whether or not that card is used in the hand. If it is not, it is discarded. Discarding is simply a matter of storing a zero in that register.

When the calculator has finished discarding, it stops to display the number of cards it is taking, then fills the zeroed register from the previously dealt draw stack. Your hand is then displayed as before, and the program is ready to begin betting.

Betting (Keys A and B)

Memory limitations have made this section simpler than it could be. It is, however, capable of some realistic betting exchanges.

The calculator always opens with five. You may then call by pressing A, fold by pressing B, or raise the pot by entering the new *total* (not the amount of the raise) and pressing R/S. In the latter case, the program compares the value of the pot to a betting limit and calls, folds, or raises accordingly. The betting limit is an arbitrary function involving the value of the calculator's hand and a random number.

When either party calls, the calculator displays the symbolic value of its hand, or the high card if it has no hand. This value may also be displayed at your option

after either party has folded. The calculator's individual cards are available for display at your request.

When you fold, the calculator displays the amount you have forfeited. The calculator folds by displaying a zero. You must keep track of your own winnings.

This program is shown in listing 1, with a sample run given in listing 2. Before the program is executed, the data in table 1 should be loaded into the calculator. The register allocations are given in table 2, and the user definable key functions for this program are given in table 3. The sample run in listing 2 should clarify the program operation; when in doubt, press the R/S key.■

Register	Code	Hand
43	0	No hand
44	5.88888	Flush
45	0	Not used
46	1.00011	Pair
47	3.00111	3 of a kind
48	7.01111	4 of a kind
49	2.11022	2 pair
50	4.11122	Full house
51	6.12345	Straight
52	8.12348888	Straight flush

Table 1: *Numerical codes for poker hands. This data must be in memory at the beginning of execution. To store them on a data card, the WRITE 3 command can be used. The codes listed here are shown as they are used internally; when they are displayed, the integer part of the number is replaced by the detail value from R04. For example, a jack-high straight is displayed as 11.12345.*

Register	Function
00	Various functions
01	Hand code
02,03	Various functions
04,05	Hand details
06, 07, 08	Various functions
09	Random seed
12 through 24	Deal/tally tables
25 through 29	Calculator's cards
30 through 34	Player's cards
35 through 42	Draw stack
43 through 52	Code table (table 1)

Table 2: *Data register allocations. Note that registers 00, 02, 03, and 06 through 08 serve temporary functions at various points throughout the program.*

Key	Use
A	Call
B	Fold
C	Discard
D	Draw
E	Deal

Table 3: *User-defined keys for TI-59 Draw Poker.*

Listing 2: *A sample run of Draw Poker for the TI-59 programmable calculator.*

Keystrokes	Display	Comments
3.45	3.45	Seed for random-number generator. Enter a new number each time you begin to use the program.
E	.1111111111	Deal (takes about 2 minutes); when the calculator is ready to display your hand, it regains your attention by displaying .1111111111.
R/S	2.4 4.2 9.2 7.3 3.4	Your cards are displayed, pausing 1.5 seconds for each card to allow you to copy (or memorize) them.
1 C	0	Discard; you are keeping only the third card dealt to you.
2 C	0	
4 C	0	
5 C	0	
D	0	Draw; you press D when you are finished discarding. The calculator shows that it is not discarding any cards.
R/S	.1111111111	The new cards are dealt from the draw stack.
R/S	4.3 2.1 9.2 9.3 9.4	Your cards are displayed as before.
R/S	5	The calculator opens the betting with 5.
7 R/S	12	You raise 2. The calculator raises again by 5.
A	14.01111	Call; you call at 12. The calculator shows that it has four aces (14's).
R/S	.1111111111	Not trusting the machine, you demand to see its cards. Otherwise you would press E for the next hand.
R/S	14.3 14.4 10.2 14.1 14.2	The calculator's cards are displayed as yours were before.
R/S	.1111111111	You admit defeat and call for the next hand to be dealt. Pressing E would have the same effect.