

Shuffler for C-64

 rickmk.com/rmk/Com/shuffle.html

by Rick Kephart
August 1, 1991

This program will produce a machine language subroutine which may be included in card games written in BASIC. It will shuffle a deck of cards, and place them into an array which can easily be read from BASIC. The program is very fast, shuffling an entire deck and filling the array in a tenth of a second or less.

First RUN the SHUFFLEMAKE program. This will produce a 2-block machine language program on your disk called SHUFFLE.

To use SHUFFLE in a program, first LOAD it in memory using:

```
10 SYS 57812 "SHUFFLE",8,1:POKE 780,0:SYS 65493
```

or:

```
10 IF A=0 THEN A=1:LOAD"SHUFFLE",8,1
```

You must use the DIM statement to prepare the area for the array. Use the statement DIM C\$(52) before calling SHUFFLE and before using any other array variables. After the DIMC\$(52), other array variables can be used; another array may even be DIM'd at the same time, as long as the C\$ comes first. You may use a different name for variable, but it must be a string array and be the first array used in the program. Numerical and string variables may be used any time in the program. After the DIM statement, use SYS 49152 to shuffle the deck and fill the array. You can re-shuffle the deck (and refill the array) at any time with another call to SYS 49152.

The shuffled cards may then be found in the array C\$(1) through C\$(52). Each one may be read as follows:

ASC(C\$(x))	Value of the card: 2 through 10 Jack =11 Queen=12 King =13 Ace = 14
MID\$(C\$(x),2,1)	Color: Red = CHR\$(28) Black = CHR\$(144)
RIGHT\$(C\$(x),1)	Suit

RIGHT\$(C\$(x),4) Prints out the card
in red or black

RIGHT\$(C\$(x),3) Prints the card without
changing color

Also included here is a short demo, a poker game, to illustrate how SHUFFLE may be used in a BASIC card game program. First RUN the SHUFFLEMAKE program before POKER, then be sure to RUN POKER with the disk containing SHUFFLE in the drive, so it can LOAD in the machine language SHUFFLE program.

SHUFFLEMAKE

```
10 OPEN8,8,8,"SHUFFLE,P,W"
20 FORI=0TO256:READA:PRINT#8,CHR$(A);:NEXT:CLOSE8
100 DATA 0,192,24,160,13,169,18,153,254,192,73,48,153,11,193,9,16,153
110 DATA 24,193,73,112,153,37,193,73,80,105,1,136,208,231,160,128
120 DATA 140,24,212,169,255,141,14,212,141,15,212,200,140,18,212,169
130 DATA 0,133,254,173,27,212,41,63,201,52,176,247,168,173,27,212
140 DATA 41,63,201,52,176,247,170,189,255,192,72,185,255,192,157,255
150 DATA 192,104,153,255,192,230,254,165,254,208,216,56,165,51,133,253
160 DATA 233,1,133,51,165,52,133,254,233,4,133,52,160,10,169,5
170 DATA 145,47,200,56,165,253,233,5,133,253,165,254,233,0,133,254
180 DATA 165,253,145,47,165,254,200,145,47,200,192,166,144,224,162,52
190 DATA 160,0,189,254,192,72,41,15,145,253,200,104,72,41,16,240
200 DATA 3,169,144,44,169,28,145,253,200,104,72,41,15,201,11,176
210 DATA 27,201,10,240,12,9,48,145,253,200,169,32,145,253,200,208
220 DATA 24,169,49,145,253,200,169,48,145,253,208,13,168,185,240,192
230 DATA 160,2,145,253,200,169,32,145,253,104,74,74,74,74,168,185
240 DATA 246,192,160,4,145,253,24,165,253,105,5,133,253,165,254,105
250 DATA 0,133,254,202,208,154,96,193,211,216,218,74,81,75,65
```

POKER

```
10 SYS 57812 "SHUFFLE",8,1:POKE 780,0:SYS 65493: DIMC$(52),H$(14)
20 POKE 53281,13:REM GREEN SCREEN
30 H$(11)="JACK":H$(12)="QUEEN":H$(13)="KING":H$(14)="ACE"
40 PRINT:PRINT"SHUFFLE":PRINT:SYS 49152:P=1:REM P = CURRENT POSITION IN DECK
50 PRINT:FOR I=1 TO 5:A$(I)=C$(P):B$(I)=C$(P+1):P=P+2:NEXT
60 REM A$(1) TO A$(5) ARE YOUR CARDS, B$(1) TO B$(5) ARE THE 64'S
70 FOR I=1 TO 5:S2$(I)=RIGHT$(B$(I),1):V2(I)=ASC(B$(I))
80 REM S2$( ) ARE THE SUITS, V2( ) ARE FACE VALUES OF CARDS FOR THE 64
90 PRINT RIGHT$(A$(I),4) " ";:NEXT:PRINT:REM THIS PRINTS YOUR CARDS
100 FORI=1TO4:IFV2(I)>V2(I+1)THEN X=V2(I):V2(I)=V2(I+1):V2(I+1)=X:GOTO100
110 NEXT:REM THIS SORTS THE FACE VALUES OF THE CARDS
120 PRINT:PRINT "[dark blue]HOW MANY DO YOU WANT?";
130 GET N$:N=VAL(N$):IF N>5 THEN 130
140 IF N=0 THEN IF N$<>"0" THEN 130
150 IF N=0 THEN PRINT:PRINT:GOTO 200: KEEP ALL CARDS
160 PRINT N:PRINT "[dark blue]DISCARD WHICH CARDS? (1-5)";
```

```

170 FOR I=1 TO N
180 GET M$:M=VAL(M$):IF M<1 OR M>5 THEN 180
190 PRINT M;:A$(M)=C$(P):P=P+1:NEXT:PRINT:PRINT:REM SUBSTITUTE NEW CARDS
200 FOR I=1TO5:S1$(I)=RIGHT$(A$(I),1):V1(I)=ASC(A$(I)):REM SUITS AND VALUES
210 PRINT RIGHT$(A$(I),4) " ";:NEXT:PRINT
220 FORI=1TO4:IFV1(I)>V1(I+1)THENX=V1(I):V1(I)=V1(I+1):V1(I+1)=X:GOTO 220
230 NEXT:REM SORTS CARDS
240 PRINT:PRINT"[dark blue]YOU GOT ";:FOR I = 1 TO 5
245 X(I)=V1(I):X$(I)=S1$(I):NEXT:GOSUB 350
250 PRINT:FORI=1TO5:PRINTRIGHT$(B$(I),4) " ";:NEXT:PRINT:REM SHOW 64'S CARDS
260 PRINT:PRINT "[dark blue]I GOT ";:FOR I=1 TO 5:X(I)=V2(I)
265 X$(I)=S2$(I):NEXT:GOSUB 350
270 PRINT:PRINT "[dark blue]PLAY AGAIN?"
280 GET N$:IF N$="N" THEN END
290 IF N$<>"Y" THEN 280
300 IF P<37 THEN 50:REM NO SHUFFLE
310 GOTO 40:REM RESHUFFLE DECK
320 :
330 REM          SUBROUTINE TO FIND SCORE
340 REM X(1) IS THE LOWEST CARD, X(5) IS THE HIGHEST CARD
350 M3=0:F=0:S=0:REM FLAGS: M3=3 OF A KIND, F=FLUSH, S=STRAIGHT
360 IFX$(1)=X$(2)THENIFX$(2)=X$(3)THENIFX$(3)=X$(4)THENIFX$(4)=X$(5)THEN F=1
370 X=X(1):IFX+1=X(2)THEN IF X+2=X(3)THEN IF X+3=X(4)THENIF X+4=X(5)THEN S=1
380 IF F THEN IF S THEN IF X(1)=10 THEN PRINT"A ROYAL FLUSH":RETURN
390 IF F THEN IF S THEN PRINT"A STRAIGHT FLUSH":RETURN
400 FOR I=1 TO 2
410IFX(I)=X(I+1)ANDX(I+1)=X(I+2)ANDX(I+2)=X(I+3)THENPRINT"4 OF A KIND":RETURN
420 NEXT
430 IF X(1)=X(2) THEN IF X(2)=X(3) THEN M3=1:IF X(4)=X(5) THEN 540
440 IF X(3)=X(4) THEN IF X(4)=X(5) THEN M3=1:IF X(1)=X(2) THEN 540
450 IF F THEN PRINT "A FLUSH":RETURN
460 IF S THEN PRINT "A STRAIGHT":RETURN
470 IF M3 OR (X(2)=X(3) AND X(3)=X(4)) THEN PRINT "3 OF A KIND":RETURN
480 IF X(1)=X(2) THEN IF(X(3)=X(4) OR X(4)=X(5))THEN PRINT "2 PAIRS":RETURN
490 IF X(2)=X(3) THEN IF X(4)=X(5) THEN PRINT"2 PAIRS":RETURN
500 FOR I=1 TO 4:IF X(I)=X(I+1) THEN PRINT "ONE PAIR":RETURN
510 NEXT
520 IF X(5)<11 THEN PRINT X(5) "HIGH":RETURN
530 PRINT H$(X(5)) " HIGH":RETURN:REM "JACK", "QUEEN", ETC.
540 PRINT "A FULL HOUSE":RETURN

```

=====

```

C000 18      CLC          ;fill with 52 cards
C001 A0 0D    LDY # $0D    ;13 different cards
C003 A9 12    LDA # $12    ;start with 2 of spades
C005 99 FE C0 STA $C0FE,Y  ; "$1x" = x of spades
C008 49 30    EOR # $30    ;EOR's and ORA produce all 4
C00A 99 0B C1 STA $C10B,Y  ;different suits
C00D 09 10    ORA # $10

```

```

C00F 99 18 C1 STA $C118,Y
C012 49 70     EOR #$70
C014 99 25 C1 STA $C125,Y
C017 49 50     EOR #$50
C019 69 01     ADC #$01      ;go to next number of cards
C01B 88        DEY
C01C D0 E7     BNE $C005

C01E A0 80     LDY #$80      ;set up SID noise generator to
C020 8C 18 D4 STY $D418     ;random numbers
C023 A9 FF     LDA #$FF
C025 8D 0E D4 STA $D40E
C028 8D 0F D4 STA $D40F
C02B C8        INY
C02C 8C 12 D4 STY $D412

C02F A9 00     LDA #$00      ;$FE counts down 255 random card
C031 85 FE     STA $FE       ;exchanges

C033 AD 1B D4 LDA $D41B     ;get a random number between 1
C036 29 3F     AND #$3F     ;and 52 and store it in Y
C038 C9 34     CMP #$34
C03A B0 F7     BCS $C033
C03C A8        TAY

C03D AD 1B D4 LDA $D41B     ;second random number in X
C040 29 3F     AND #$3F
C042 C9 34     CMP #$34
C044 B0 F7     BCS $C03D
C046 AA        TAX

C047 BD FF C0 LDA $C0FF,X   ;exchange cards pointed to by X
C04A 48        PHA          ;and Y
C04B B9 FF C0 LDA $C0FF,Y
C04E 9D FF C0 STA $C0FF,X
C051 68        PLA
C052 99 FF C0 STA $C0FF,Y
C055 E6 FE     INC $FE      ;do it 255 times
C057 A5 FE     LDA $FE
C059 D0 D8     BNE $C033

C05B 38        SEC          ;move bottom of strings down
C05C A5 33     LDA $33      ;$0104 (260 or 5*52)
C05E 85 FD     STA $FD      ;bytes, and stores the location
C060 E9 01     SBC #$01     ;in $FD and $FE
C062 85 33     STA $33      ;where the array strings are to
C064 A5 34     LDA $34      ;be built
C066 85 FE     STA $FE
C068 E9 04     SBC #$04

```

```

C06A 85 34 STA $34

C06C A0 0A LDY #$0A ;set up the addresses and length
C06E A9 05 LDA #$05 ;(5 bytes) of all 52 array
C070 91 2F STA ($2F),Y ;strings
C072 C8 INY
C073 38 SEC
C074 A5 FD LDA $FD
C076 E9 05 SBC #$05 ;each string 5 bytes away from
C078 85 FD STA $FD ;the last one
C07A A5 FE LDA $FE
C07C E9 00 SBC #$00
C07E 85 FE STA $FE
C080 A5 FD LDA $FD
C082 91 2F STA ($2F),Y
C084 A5 FE LDA $FE
C086 C8 INY
C087 91 2F STA ($2F),Y
C089 C8 INY
C08A C0 A6 CPY #$A6 ;array data takes up a total of
C08C 90 E0 BCC $C06E ;166 bytes

C08E A2 34 LDX #$34 ;get 52 ($34) cards one at a
C090 A0 00 LDY #$00 ;time to convert to array data
C092 BD FE C0 LDA $C0FE,X ;and place in string memory

C095 48 PHA
C096 29 0F AND #$0F ;stores value to be read with
C098 91 FD STA ($FD),Y ;ASC(C$(x))
C09A C8 INY ;$FD points to current
C09B 68 PLA ;location in string memory
C09C 48 PHA

C09D 29 10 AND #$10 ;color - check for even suit
C09F F0 03 BEQ $C0A4 ;even=red
C0A1 A9 90 LDA #$90 ;$90=ASCII("black")
C0A3 2C A9 1C BIT $1CA9 ;LDA #$1C $1C=ascii("red")
C0A6 91 FD STA ($FD),Y
C0A8 C8 INY
C0A9 68 PLA
C0AA 48 PHA

C0AB 29 0F AND #$0F ;remove suit to see card's value
C0AD C9 0B CMP #$0B
C0AF B0 1B BCS $C0CC ;face card if >$0B

C0B1 C9 0A CMP #$0A
C0B3 F0 0C BEQ $C0C1 ;make sure it's not ten

C0B5 09 30 ORA #$30 ;poke the ASCII value of the

```

```

C0B7 91 FD    STA ($FD),Y    ;card number
C0B9 C8      INY
C0BA A9 20    LDA #$20      ;follow with a space
C0BC 91 FD    STA ($FD),Y
C0BE C8      INY
C0BF D0 18    BNE $C0D9

C0C1 A9 31    LDA #$31      ;if it's a ten, poke "10"
C0C3 91 FD    STA ($FD),Y    ;in ASCII
C0C5 C8      INY
C0C6 A9 30    LDA #$30
C0C8 91 FD    STA ($FD),Y
C0CA D0 0D    BNE $C0D9

C0CC A8      TAY          ;if it's a face card, get the
C0CD B9 F0 C0 LDA $C0F0,Y    ;value from data pointed to
C0D0 A0 02    LDY #$02      ;by Y, which begins with 11
C0D2 91 FD    STA ($FD),Y
C0D4 C8      INY
C0D5 A9 20    LDA #$20
C0D7 91 FD    STA ($FD),Y

C0D9 68      PLA          ;get the suit, 1 to 4 in the high
C0DA 4A      LSR          ;nibble
C0DB 4A      LSR
C0DC 4A      LSR
C0DD 4A      LSR
C0DE A8      TAY
C0DF B9 F6 C0 LDA $C0F6,Y    ;get the suit ASCII from the
C0E2 A0 04    LDY #$04      ;table
C0E4 91 FD    STA ($FD),Y

C0E6 18      CLC          ;move on to the next area in
C0E7 A5 FD    LDA $FD      ;string memory in which to build
C0E9 69 05    ADC #$05      ;the next array variable
C0EB 85 FD    STA $FD
C0ED A5 FE    LDA $FE
C0EF 69 00    ADC #$00
C0F1 85 FE    STA $FE

C0F3 CA      DEX          ;count 52 cards
C0F4 D0 9A    BNE $C090
C0F6 60      RTS

C0F7 C1 D3    CMP ($D3,X)    ;[Ace graphic], [Heart graphic]
C0F9 D8      CLD          ;[Spade graphic]
C0FA DA      ???         ;[Diamond graphic]
C0FB 4A      LSR          ;"J"
C0FC 51 4B    EOR ($4B),Y    ;"Q", "K"
C0FE 41      EOR ($2D,X)    ;"A"

```

End of file.

You can write to me at .

rmkq@lphrc.org