Jed Log has competion....

Robin and Mike will make the announcement - The envelope please... The cutest critter award goes to (drum roll...) * Morgan Joseph *! Born March 14, 1982. I chose you, Jed, but the new parents outvoted me. Well, soon Robin will be back here and things will return to normalcy (ya, right). In the meantime, meet Rose. The last snowstorm back east blew her and her weather our way. I think she's here to stay, but the rain's gotta go...

Chromasette
Magazine

P.O. Box 1087,
Santa Barbara, CA 93102
(805) 963-1066

April 1982

Filename	English Translation	PMODE	PCLEAR	Locations
POLYCOV	Polygram Cover	4	4	8 & 143
RUBIC	Rubic's Cube	0	1	27 & 158
BOBO	Bobo	(2)	(4)	61 & 184
SPACE	Space Duel	3	(4)	85 & 203
FINANCE	Financial Analyst	(2)	(4)	101 & 216
LAZKEY	Lazkey (Clear 15943 CLOADM)	(2)	(4)	121 & 232
MANYBODY	Manybody	3	(4)	131 & 240

Locations are for the R/S CTR-80. If the first copy of a program won't load, try the second. If neither copy loads, return the tape for disciplining and a prompt replacement. If you get an OM, FC, or SN error while loading or running a program, you probably have to enter the PMODE and PCLEAR values for the program directly from the keyboard. (Values in parenthesis are not set in the program). * These programs may use high speed. Be sure that the computer is slowed down again before doing I/O to tape (POKE 65494,0).

Polygram Cover is a cover of many, many sides. And pretty, pretty patterns.

Boy, that Rubik's Cube is neat! Fun to play with! But I will never, ever solve it (a two-year old kid has more patience). But wait... CoCo has infinite patience, so why not let it solve the frustrating cube? And it came to be that Rubic's Cube was written. With Rubic's Cube you can set the cube up yourself and manipulate it by hand if you want. But the thing I like to do is to have CoCo shuffle the cube, and then watch as CoCo solves it!

All six sides of the cube are displayed at once in orange-peel fashion. If you choose to rotate the cube yourself, you choose which side you wish to rotate with a letter (T, B, F, L, R, or U) and the number of rotations clockwise (1, 2, or 3 - 3 is like 1 rotation counterclockwise). If you set up the cube yourself and make a mistake, you will be asked to fix the error. You then enter the location of the block to fix by giving it the face (1 to 6), row (1 to 3), and column (1 to 3). You then tell CoCo what color to make that block. A word of warning, though - due to the way that the cube is set up, on some faces row 1-column 1 is in the upper left-hand corner, while on others it is in the lower left-hand corner. Experiment! NOTE: If you only have 16k of memory, you will have to type 'PMODE0:PCLEAR1'<enter> before loading in Rubic's Cube in order to have enough memory.

Puttin' on the squeeze - The following discussion is for the more advanced programmers Cut there. If you are new to computers, this section will probably work better than scotch just before beddy-bye:

Rubic's Cube started as a 16k program for the Model I and III. And it used all of the available 15,500 bytes of RAM. So the first problem was getting the program squeezed down into the 13,500 bytes that CoCo has available (I am not a fan of PCLEAR 0 techniques - they take too much thought before the program can be loaded and run). That wasn't the worst problem, however. The program used a lot of integer arrays (using a DEFINT statement) which CoCo does not have. In an integer array each element of the array takes two bytes. In CoCo's floating point arrays, each element of an array takes FIVE bytes. With about 500 elements in arrays, there's another 1500 bytes to be squeezed out of the original program. Aaaarrrggg! Character strings to the rescue! If the numbers stored in these arrays were all less than 255, each array element could be represented by a character and stored in string. Then each element of the 'array' would only take 1 byte! The numbers could be retrieved from the string by using INSTR$, MID$, VAL, etc. Also, rather than create a string to put values in, the values could be POKEd into and PEEKed from the unused and unclearable graphics page. Examples from the program are below:

Line 30 - Some floating point arrays were left alone for speed and sanity considerations (QQ, Q3, Q1).

Line 40 - CD is the character differential (40 in this case). This value is subtracted from the ASCII value of each character in a string to get the true value. For example, trying to put a number with the value of 8 (a backspace) in a character string can really be a pain, but it's easy to put the character '0' in a string. Taking the ASCII value of '0' gets you a value of 48 and subtracting 40 from it gives you 8!

Line 40 - GP points to the beginning of the unused graphics page. POKE279,PEEK(275) seeds the random number from the TIMER.

Line 670 - The garbage you see at the end of this line is packed graphics for the color blocks of the cube. Q$ at first had the value "OOYYBBGGWWRR". I ran these lines to find the beginning of the string:

```
    5 I=PEEK(25)*256 : REM BEGINNING OF BASIC
    6 IF PEEK(I)=ASC("O") AND PEEK(I+1)=ASC("O") THEN PRINT I:STOP
    7 I=I+1:GOTO 6
```

Then I POKEd in the corresponding graphic values starting at location I (255, 255, 159, 159, 175, 175, 143, 143, 207, 207, 191, 191). Graphic characters won't LIST, so they look like junk.

Line 890 - INSTR used to tell if a character typed in from the keyboard is an acceptable character and which one it is in the block array (AA$).

Lines 950 & 1020 - All of the values in Q$ used to be read from DATA statements. A few bytes were saved by putting it in the string and reading it using VAL and MID$. But speed was lost.

Lines 1140-1190 - Q$ holds the array pointers for switching the little cubes around. Q2$ holds values that must be added to some of the array elements after switching. This is the first place where the ASCII value of a character in a string minus the character differential (CD) is used.

Lines 1280, 1390, 1400, & 1520 - When the computer solves the cube, the moves are all stored by POKEing them into the unused graphic page and PEEKing them back out to display them or manipulate them. BP is the total number of moves (some are null moves).

Lines 1600-2190 - Q1$ holds the various moves to be done on the cube when the computer is solving it depending on the current state of the cube. B (line 2190) gets the move and the cube is manipulated (GOSUB 1260) to see what the next set of moves should be.

Bobo is one of those kid games that is fun for all kids. You try to second-guess the computer enough times so that Bobo gets drenched.

Get your joysticks ready! It's time for a Space Duel! You control the distance and direction of your shot as well as the speed and direction of your ship with the same joystick movement. It's a bit tricky, but after a while you'll get the frustrating hang of it.

Do you need information on a loan payment, compound interest, future value, or present value? You need Financial Analyst.

Tired of typing? Lazkey lets you hit one key to type out an entire word or phrase! To use Lazkey, first type 'CLEAR 200,15943'<enter> (or 'CLEAR 200,32327'<enter> if you have 32k) o reserve space for the program. Then type 'CLOADM"LAZKEY"'<enter> (or CLOADM"LAZKEY",16384'<enter> if you have 32k) to load the program in. Finally, just type 'EXEC'<enter> to run the program. Now to operate the thing:

Push the key combination <shift><down arrow> and follow with the key containing the word or phrase you want (see list below for the current definitions). For example, pushing <shift><down arrow> and then hitting 'R' will put RETURN on the screen.

But that's not all! You can define your own keys with up to 250 characters per key! First, push <shift><down arrow>. Now hit ':' (colon). Third, hit the key you wish to define. The key followed by a '>' will be displayed. Now type in your new definition. EVERYTHING typed in (including <enter>) will be included in the definition. The ONLY way to end the definition is to type <shift><down arrow> followed by ';' (semicolon). Note: If your definitions are going to be bigger than the original ones, you probably want to reserve more memory (with the CLEAR command) than specified two paragraphs ago.

To make your own copy (with or without your own key definitions) of Lazkey first load and exec Lazkey, then CSAVEM it to tape (or SAVEM if you're using disks) as follows:

5727 → 304 (474) 28

```
    CSAVEM"LAZKEY",PEEK(116)*256+PEEK(117)+1,PEEK(126)*256+PEEK(127)-1,
        PEEK(363)*256+PEEK(364)-26
```

There is no need to add an offset to CLOADM your own copy of Lazkey (CLOADM"LAZKEY" will do). You still need to clear enough memory (with the CLEAR command), however.

A	LIST<enter>	B	STR$(	C	CHR$(	D	DATA	E	ELSE	F	FOR
G	GOTO	H	GOSUB	I	INPUT	J	INKEY$	K	ASC(	L	LEFT$(
M	MID$	N	NEXT	O	?	P	POKE	Q	PEEK(	R	RETURN
S	SET(	T	THEN	U	READ	V	RND(	W	SOUND	X	RIGHT$(
Y	RESET(	Z	LEN(								

Come together! My favorite program this month is Manybody, a simulator of gravitational problems with two or more (up to 9) bodies. It runs a bit slow, but it is neat to see if you can get a few objects moving around each other on the screen in nearly stable orbits. Or watch and see what happens when they get too close to each other. Or collide them (hee hee hee). As a test run you might try this setup:

Body 1 - M=1000, COLOR=2, X=128, Y=96, VX=0, VY=0
Body 2 - M=1, COLOR=3, X=100, Y=96, VX=0, VY=6

Y L
B = 3
1

or with Body 2 as M=1, COLOR=3, X=150, Y=96, VX=0, VY=3

I could have typed all night...

But I should have stopped and slept. Last month every counter value on the cassette label was 3 (three) too high. In the blue sheets the counter values for CKMON should have been 139 & 252 and the ones for Amazing should have been 46 & 179. Not only that, but I completely forgot to mention Old House. As you have probably guessed by now, Old House was a very simple adventure-type program. It was included mainly for those who are new to adventuring or for those who threatened to sell their CoCos after failing to complete Jerusalem Adventure (August 1981) and/or Mansion Adventure (January 1982). If you couldn't get Old House to load, try typing 'PMODE2:PCLEAR2'<enter> from the keyboard before loading.

Just a little hint...

Don't read this paragraph if you don't want a hint on Mansion Adventure. But if you find yourself stuck in a certain spot, 'pull four' might get you out.

Send it anywhere...

Last month's Name & Address program was a good one. Programs of this type, however, a often set up for somebody's own particular needs. In this case it was set up for a 16k system with a printer. That is a little more limited in use than I realized at the time and I apologize. So, if you don't have a printer or you just want to see the addresses on the screen when you list them, delete line 3015 and type in line 3020 as follows:

```
3020 CLS: FOR PS=1 TO N-1: PRINT NA$(1,PS)", "NA$(2,PS): PRINT
     STRING$(12,B);NA$(3,PS): FOR I=4 TO 7: PRINT CHR$(B);NA$(I,PS);:
     NEXT: PRINT: IF PS/3=INT(PS/3) THEN GOSUB 3040
```

If you have 32k you can get more names with the following mods (I hope - they have not been extensively tested):

Change the 'CLEAR4500:DIMNA$(7,33)' in line 110 to
'CLEAR17500:NR=100:DIMNA$(7,NR+1)'.
Change the '32' in line 160 to '";NR;"'.
Change the '32's to 'NR's in lines 220, 240, 250, 365, 1020, 1030, 1510, 2030, 2200, and 3060.
Change the '=U's to '=NR's in lines 365, 1020, and 1510.

It's a bust...

Several of you have tried to convert Blockbuster (January, 1982) to run on disk. Well, the machine language portion expects the the graphic pages to start at memory location 1536. But if you have a disk, the DOS scratchpad starts there instead, causing the program to crash when it is run. Can it be fixed? Probably, but it wouldn't be real easy and I can't tell you how.

And I won't do windows...

A lot of you write in with programing problems and questions. As you may already know, things are pretty hectic around here. I will always attempt (but not always succeed - then it's called a 'feature') to fix any bugs that are found in the programs. But don't expect an answer when you ask me to modify a program to fit your particular needs or ask me to help you with some special programing project. There just isn't time. Some general interest questions will be answered in these pages and some (not all) special interest ones will catch my fancy and be answered by letter. Fair enough?

How's he do that...

There is a new program on the market that you may have read about in Color Computer News and The Rainbow. Motion Picture Programing (MPP - $34.95 from Superior Graphic Software 406 Little Mountain Road, Waynesville, NC 28786) turns CoCo into a animation festival! This programing tool lets you EASILY create pictures (using a modified version of August 1981's Drawer program) and move them around WITHOUT the tedious DRAW, PAINT, PSET, CIRCLE etc. commands. I have seen the results, and they are INCREDIBLE! If you want to see and use the full graphic potential of your computer, this program is not recommended - it is REQUIRED! I always knew CoCo had it in him (her?)...

Same time, same channel,

Dave

ed.