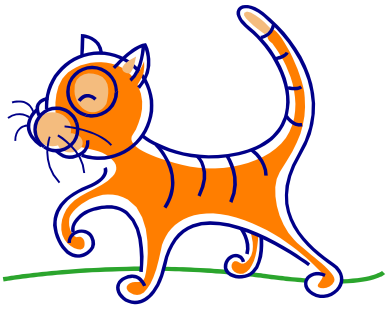
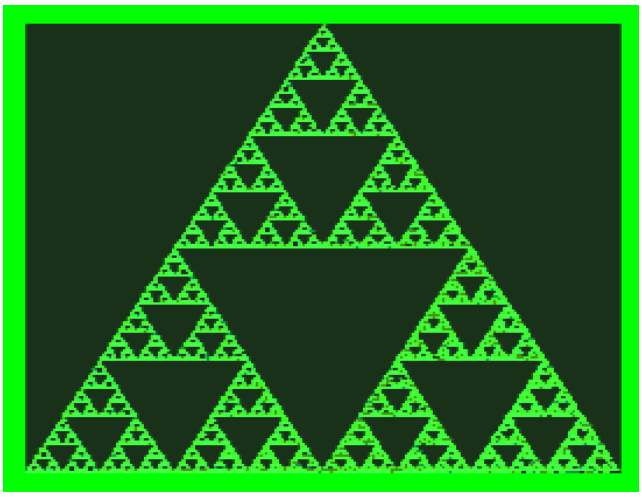


CoCoNutz!

Volume 2, Issue 1
March 2006

Make a donation to the Newsletter!
http://tinyurl.com/6gphh

One - Liner



```
10 PMODE4,1:SCREEN1,0:PCLS:A(1)=0:B(1)=192:A(2)=255: B(2)=192:
A(3)=128:B(3)=0:X=A(1):Y=B(1):FORL=1TO2STEP0:I=RND(3):X=(X+A(I))/2:
Y=(Y+B(I))/2:PSET(X,Y):NEXT
```

Here is the same program written out for easier reading:

```
10 PMODE 4,1:SCREEN 1,0:PCLS
20 A(1)=0:B(1)=192
30 A(2)=255:B(2)=192
40 A(3)=128:B(3)=0
50 X=A(1):Y=B(1)
60 I=RND(3)
70 X=(X+A(I))/2:Y=(Y+B(I))/2
80 PSET(X,Y)
90 GOTO 60
```

Submitted by Eric Hood

Magic V2.0
A program for backing up Coco copy protected software on a real Coco computer.
By Carl England, 2005.

This Backup Utility is an Amazing piece of programming, I tested this software on 2 of my copy protected programs, First 1 was Photon, and it backed it up with no problems on the new copy made.

The other one was Z-89; BTW, this game has 20 odd copy protected tracks in its protection scheme. And it worked no problems, so when it is marketed, it is a must have Software.
Carl and I have tried Magic on 20 plus copy protected games and Utilities in total, it copied them all.

Now to the how to use this program,

It comes up with a nice colorful intro, then there is a 5 second pause, then it takes you to the main program menu screen,

Source Drive:
Destination Drive:

Enter the source and destination drives, then hit enter, go make yourself a cup of coffee, or pour yourself a glass of soft drink, since it only takes about 40-50 seconds to make a backup, it is that fast, and if I'm correct, they said it was impossible for a programmer to design a program for the coco, to be able to do this.
When it's finished, it will ask you if you want to backup another copy protected program.
Ps, the destination disk does not need to be formatted, as the program will automatically format the tracks as it backs up the software.

Magic2Pc, and PcMagic
By Carl England, 2006

Magic2pc, is the program for use on a real coco computer, used to copy a copy protected game or utility onto blank disks (this disk will hold tracks 0-20, for the first disk, the second disk will hold tracks 21-39), these 2 disks, will hold the data, that PcMagic uses to copy a copy protected game into a virtual emulated dsk image.

PcMagic, is similar to Magic2PC, except this program is used on a pc, David Kiel's emulator, so having DK's emulator is a must, to copy the data tracks on the magic2pc dsk images into a emulated dsk image.

The intro and main program are similar to magic v2.0, except these 2 programs are used to make emulated dsk images of copy protected software, for use in a emulated environment, asks for a source and destination drives.
The game i tested it with was Xenion, by Diecom,
It worked great; this was 1 game I wanted to have running in the emulator, now I can.
If you like Xevious for the C-64 and other old 8bit computers, this is the version made for the coco 1, 2 and 3.

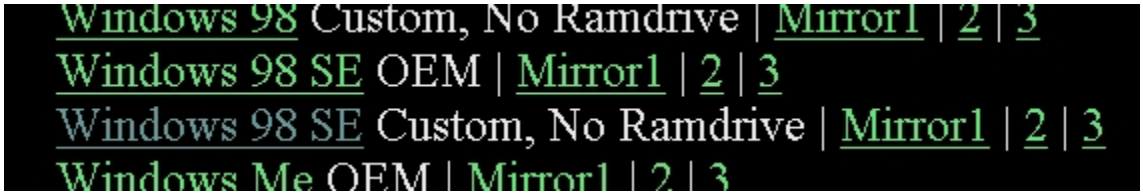
I had a go at trying cbasic3, it failed to run in the emulator, I suspect it is the emulator, cbasic3 seems to trash the memory, so it locks up, I guess we need David Kiel to have a look at it, and fix the emulator to allow CBasic3 to work.
Even Carl said he had trouble with CBasic3.
But overall I used it on some of my stuff, and the magic programs worked.
And not everybody has a copy of CBasic3 in their collections, but will have countless copies of games.

DOS Boot Floppy for Win2K and XP users
By d.j. fowler

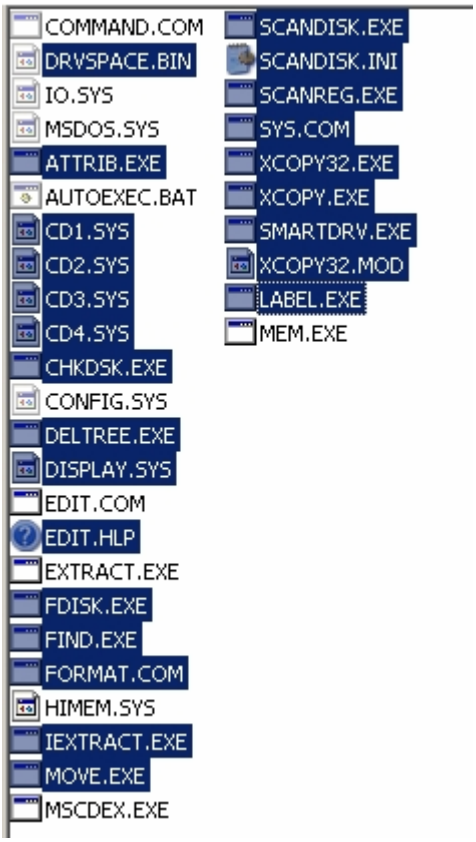
So you're having problems making those DSK images? OR, you'd like to be able to take a DSK image of an old coco floppy you have... One little problem, Win2K and XP cause those nice little programs to hang up. Well, here's help for ya, this step by step guide will help you build a boot floppy that doesn't even bother with Windows.

We're going to start by finding a blank boot floppy. I found a handy one at http://www.bootdisk.com/ just click the link to the Bootdisks

For our purposes here, we want the windows 98 SE Custom disk. Download this into a folder called COCOBOOT on your hard drive. I'll be referring to this folder by name frequently. Once the file is downloaded, double click the file. It'll ask you to insert a blank floppy and write a standard boot disk for our use.

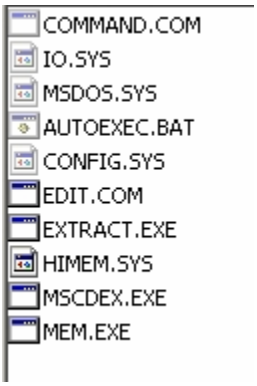


Next, we strip that floppy down a bit:



(You can hold down the CTRL key while you click on the various files, then press <DEL> and get them all at once.)

And we should have this left:



Next: we're going to replace the Autoexec.bat file. **RIGHT** click on **AUTOEXEC.BAT** so you get a menu:

select "Edit". Press CTRL-A to highlight everything, and then press <DEL> to delete.

Next, copy this batch file and past it into the Autoexec.bat you have open and save it.

----------BEGIN

```
@ECHO OFF
Echo MSCDEX for IDE Devices.
LH MSCDEX /D:MSCD001 /l:e > nul
XMSDSK 28000 d: /y > nul
d:
cd\
md dos
cd dos
Echo Copying boot files to ram disk.
copy a:*.com > nul
copy a:*.exe > nul
Echo Copying DOS Shell to ram disk.
copy a:dosshell.zip > nul
unzip dosshell.zip > nul
pkunzip disshell.zip > nul
cd..
path d:\dos;d:\;a:\
set comspec=d:\dos\command.com
cd\
md emu
Echo Checking for Emulator Files.
if not exist a:\emu\*.* goto dsks
Echo Emulator files found – Copying to RAMdisk.
cd emu
copy a:\emu\*.* > nul
unzip *.zip > nul
pkunzip *.zip > nul
del *.zip
:dsks
```

```
cd\
md dsks
Echo Checking for DSK files
if not exist a:\dsks\*.* goto end
Echo DSK files found – Copying to RAMdisk.
cd dsks
copy a:\dsks\*.*
unzip *.zip > nul
pkunzip *.zip > nul
:end
cd\
PROMPT $t  $d$_$P$G
```

----------END

We're going to do the same thing with the CONFIG.SYS:

and replace it's contents with:

```
----------Begin
DOS=HIGH,UMB
FILES=30
BUFFERS=30
LASTDRIVE=32
STACKS=9,256
DEVICE=HIMEM.SYS
DEVICEHIGH=VIDE-CDD.SYS /D:MSCD001
```

----------END

Now time to populate our floppy a bit:

We want to make 2 new directories.

One called "emu" and one called "dsks".
Now we need to hunt down a few programs….

You can find the PORT.EXE file in David Kiel's or Jeff Vavasour's Emulators for the CoCo. You can place a copy of it in the a:\ of the floppy drive also if you wish to move files to and from DSK images.

http://www.vavasour.ca/jeff/trs80.html#coco3 is our first stop; this will get you DSKINI.EXE and RETRIEVE.EXE in a zip file. Unzip these files either directly onto your "A:\" root directory or put them in cocoboot and copy them to "A:\"
http://www.simtel.net/product.php[url_fb_product_page]4825 gets us our next package… This is for XMSDSK, the ram disk we'll be using for our temporary storage. Download the file **furd19_i.zip** to the cocoboot directory and unzip it or use XP to explore it and bring out the xmsdsk.exe file and place it in the A:\ (root) drive.

IDE CDROM driver is next; I found this nice source http://www.scarlet.nl/~one2one/s2.html of drivers through Google. I like the ACER driver, it's pretty compatible, and what our CONFIG.SYS is designed to use. Just download the zip file and send it over to the floppy root.

I found PKZIP 2.04g on several web sites for download. Google provided quite a few. Here are 3 to choose from:

http://www.aa.washington.edu/uwal/download/software.htm

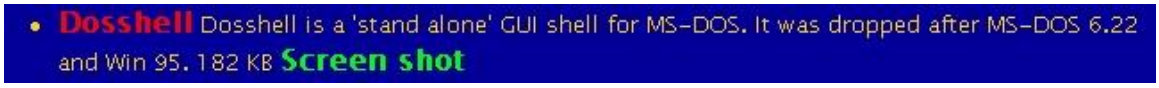
http://www-307.ibm.com/pc/support/site.wss/license.do?filename=dos_util/pkz204g.exe

http://www.sfu.ca/acs/software/pc/compress/pkzip/pkz204g.exe

Execute the pkz204g.exe file in the cocoboot directory and it will expand into multiple files. We're after PKZIP.EXE and PKUNZIP.EXE. Copy these over to your floppy's root directory.

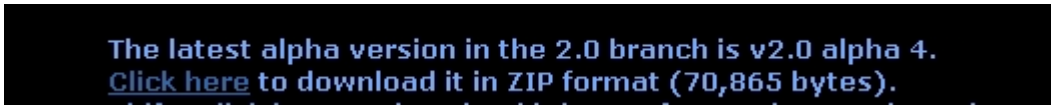
I personally like to remove the "PK" from the file names, leaving just ZIP.EXE and UNZIP.EXE. The batch file works either way, so use them whichever way you like.

You can use DOSSHELL which you can get from http://www.pcxt-micro.com/download.html it's 182K, and needs to go in the a:\ of the floppy. If it is in the DOS folder of the ramdrive it will be used, if it is not, it will not. Leaving it off the floppy makes more space for another DSK image if you're comfortable working with DOS commands. (There is a link at end of this article on an excellent source of information of DOS commands)



The last thing we need is a mouse driver for the DOSshell and your emulator. CuteMouse from the Freedos project works wonderfully and can be found here: http://cutemouse.sourceforge.net/

Download this: (any version 2.0 or higher should be fine.)



You can unzip this into the cocoboot folder we're using, the only file we need on the floppy from this zip is the CTMOUSE.EXE file.

At this point, the floppy disk is finished. There should be about 575K of free space (or over 700 if you left out DOSSHELL) on the disk that you can copy DSK images onto, then use DSKINI.EXE from the dos prompt to put your images onto real disks, or you can use RETRIEVE to make images of real coco disks. Anything in the DSKS folder on the floppy will be automatically copied to the DSKS folder of the ramdrive when you boot. Likewise, if you ZIP up a dos based emulator and put it in the EMU directory, it will unzip it into the EMU directory of the ram disk so you can use it while you're in DOS mode.

Since everything works out of the ramdrive after you boot up, yes, you can use more 1.44 meg floppies to get DSK images from or save to for use in emulators later.

Please see the DSKINI and RETRIEVE documentation for how to use that program, and your emulator documentation for the dos emulators.

DON'T forget to copy your .DSK files from the ramdrive back to a floppy before you reboot, or all is lost!

C:\> copy d:\dsks\mydisk.dsk a:\dsks\ <enter>

(Here is an excellent web site on MSDOS commands and how to use them. http://www.easydos.com/ follow the links for examples of how each command works.)

Enjoy making those disks!

Thoof

"Crazy Chemist"

Game author: Mike Snyder

Publisher: T&D Software, Issue #85, July 1989

Runs in text mode on all 16K+ CoCos

Review by Richard Kelly

Introduction: Argh! Just when you're ready to mix your household chemicals to create your new commercial product and become famous, you've found that all the labels have fallen off of your chemicals! All you can do now is mix things according to their colors and just hope for the best.

Unlike the other games I've reviewed, Crazy Chemist is an all-text game. It's actually somewhat like an adventure; you type in the names of the chemicals you want to use, you can get away with typing in only the first three letters of each chemical's color, and it's pretty much a trial-and-error sort of game.

Considering the game's concept, it could easily have ended up being unpleasant, dull, grim, or flat-out annoying. Thankfully, "Crazy Chemist" has none of these qualities. I'll tell you why later. But first, on to the game-play.

The game plays in several stages. In the first stage, you combine each of the four chemicals two at a time. For example, you might first mix Black with Blue, Red with Aqua, then Yellow with White. After combining all of the chemicals, you'll either "progress" to the next stage by the game saying "You have created a chemical", or the chemical will blow up in your face, in which case you just start over again.

The second stage is where you add one of several colored powders to the chemical. This can make the chemical stable, can make it explode, or make it implode.

In the third stage - should you last that long - you choose how many minutes the chemical will be boiled. The chemical can end up dissolving, exploding, or forming properly as it should.

If you've made it to the fourth stage, you've done well to get this far. Next is to add a type of acid to this chemical. You have two acids to choose from.

If you've lasted long enough to make it to the fifth and final stage, all that's left to do is to figure out what the chemical does. And you have a long list of options here. If you choose the right action though, the game is won.

Visual layout and appearance: A-. "Visual appearance" is listed instead of "Graphics" because the game is all text-based. Nothing seems to be misplaced anywhere, and things are centered whenever it seems that they should be centered. The game starts out turning the names of the chemicals into Inverse Video. This makes them look distinct from the other text, as it should. The words go into regular text mode where you input the names of the chemicals you wish to use. A couple of changes could have been put into the code to make things look a little bit better, but after a while, you just get used to it. To be honest, if things looked too tidy, it would probably rob the game of its character. One could also argue that - considering that this guy was disorganized enough to let all the labels fall off of his chemicals - you shouldn't expect a very neat layout of the lab to begin with.

There *are* a couple of typos in the instructions - one on each page. I'm not going to drop the rating much for such a quibble, though, or even drop the Overall rating of the game. After all, they don't make the instructions hard to read, and once you know how to play the game, you'll pretty much forget that the typos are even there. I did, at least, until replaying the game completely for the sake of this review.

Sound: A+. Mike Snyder has always, *always* shined in the sound department, be it music or sound effects. I don't know whether he has a college major in Music or what, but the sound in his games always impresses me. And the sounds and music in this game make the whole Crazy Chemist experience a very fun one. There *is* one wacky, goofy melody you'll hear when you pass Stage 3, but it's goofy for the sake of humor, and this musical sense of humor works because it's done so sparingly.

Game engine: A+. This game runs entirely in ECB, yet every pause in the game seems appropriate. Whenever you enter in the names of the chemicals, it pauses. It naturally will take a couple of seconds to mix the chemicals together in real life, and then wait to see what happens. So the fact that Chemist was written in ECB actually works to its advantage.

I've tried numerous ways of porting this code to PC format and compiling it through QuickBASIC. No matter how I did it, even if the sound effects were still completely intact, the game was always missing something in its CoCo-to-PC translation. Despite running faster than before, the game didn't play as well, even with graphics and animation added, as well as point-and-click interfacing. It seems that the game was meant to be run at the speed that ECB runs it, with this sort of text layout, with typing in the chemical names rather than clicking on them. Anything placed in the game that's even slightly fancy seems to make the game worse rather than better. Crazy Chemist is all about sound and gameplay. Either the author realized this fact when he constructed the game, or he was just lucky with every educated guess he made. Whatever the case, he didn't make a single wrong move here.

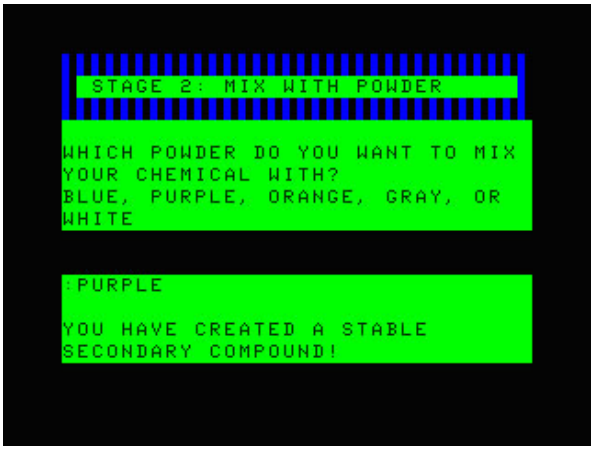
Game-play: A+. There are numerous ways to "die" in this game. In addition to the chemical exploding, the game might say "You create a super magnet and it implodes", "The whole wide world explodes", or "You have created something very gross". Some of the chemical's responses to your actions are downright fascinating.

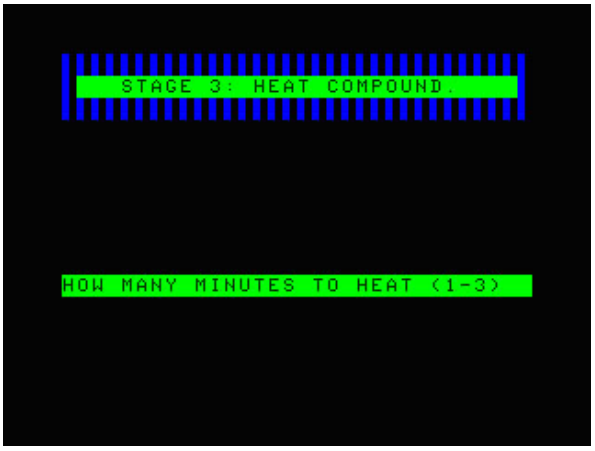
Given the game's story and concept, you'd expect the game to have some unpleasant things happening somewhere along the line. The fact that it doesn't - and the fact you don't care - makes Chemist a thoroughly enjoyable experience. Does this factor sacrifice the game's credibility? Perhaps, but I'll place credibility on the side if doing so will make the game more entertaining. After all, the purpose of playing a game is to lose yourself in a make-believe fantasy world, anyway … to escape reality and have some fun. Chemist lets you do just that.

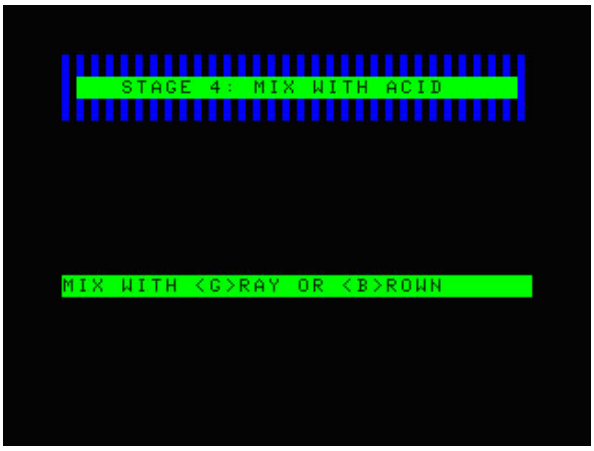
And even if you win the game a single time, the game still has more to offer. There's half a dozen ways to win in the game. You can create a dog food that all dogs like, you can end up with a lotion that instantly tans skin, or discover that your chemical is a special sort of hair tonic. This game can provide weeks - if not months - of entertainment.

Overall: A+. The author has taken what sounds like a terrible game concept and actually made it into a very good game. It's simple to learn, it's easy to play, and it's highly original. It has a high replay value, yet still fits on a 16K package. Those who prefer action games should try something else, but I recommend all others check this one out. You done good, Mike.

Note: The original version of Crazy Chemist had a bug in it in Stage 2. The way this bug worked, you might lose the game after a blank line is displayed rather than a "The chemical explodes" message. Visit WWW.COCOQUEST.COM for the updated version.







“Buzzard Bait”

Game authors: Robert Lech & Troy Dahlman of Rugby Circle, Inc.

Publisher: Novasoft (later known as Tom Mix Software)

Runs in PMODE 4 on 32K+ CoCos

Review by Richard Kelly

Introduction: With emulators of arcade games being so popular nowadays, clones of arcade games for 8-bit computers are no longer as valuable to nostalgia fans as they once were. After all, what’s the point of playing an imitation of an arcade game when you can play the genuine arcade game instead?

The answer to that question is this: The “imitation” - it turns out - is actually better than the original.

This is the case with “Buzzard Bait”, which is supposed to be a clone of the Williams arcade game “Joust”. But Buzzard is more than just an 8-bit port for the CoCo. The animation is livelier than Joust’s. The graphics are better-looking. The sound effects are better. The game-play is faster-paced. In short, the authors took a fairly crude arcade game and remade it into a near masterpiece.

For those unfamiliar with the Joust arcade game, you pilot a bird, and several other birds are after you. You get rid of the enemy birds by touching them with your lance above theirs. The enemy birds will kill you on contact if the opposite happens to be true.

Whenever an enemy bird is jousted, the pilot turns into an egg. If you don’t pick up this egg in time, the egg will hatch, and out pops the pilot! Then the dismounted bird shows up again; ready to pick up the pilot, then reverting to enemy bird form.

If you take too long to complete a wave, a Pterodactyl appears. Touching him is usually deadly; however you can kill him by sticking your lance down his throat. Be aware - however - that another Pterodactyl will appear every few minutes until either the wave is completed or there are already four Pterodactyls on the screen at once.

Every five waves - beginning at Wave 5 - is an “Egg Wave”. In this wave, all the pilots are eggs at the very beginning of the game, and the faster you snatch up all these eggs before they hatch, the fewer enemy birds you have to deal with for that wave. Easy points at first, but don’t knock it! It gets harder soon enough.

Every five waves - beginning on Wave 2 - is either a Survival wave or Team wave (the latter being if two people are playing at once). The Survival wave awards you 3000 bonus points for passing the wave without losing a single life. The Team Wave awards that same bonus if the wave is passed without either player jousting the other one (by accident or otherwise).

Every five waves beginning at Wave 8 are Pterodactyl waves. The beast appears at the very start of the wave to make things more difficult.

A bit of history behind Buzzard now. There are at least three versions of “Buzzard Bait” wallowing around. The first - and most popular one - takes up 12 granules on an RS-DOS disk. The second version, entitled “BUZZ.BIN”, is one granule smaller, but otherwise seems to have no discernible difference. I suspect it’s either Cracked or corrupted. The third and best version is the smallest of them all, although still 11 granules in size. I’ve noticed one difference with this newer version: Whenever the Pterodactyl lunges towards you, it has a sound effect. No other version had such a feature.

Graphics: A+. Everything in this game is pixel-perfect. Unlike the arcade game Joust, the graphics all around are very high quality every step of the way. How anyone could make a four-color game look so much better than a 54-color one is beyond me, but that's exactly what the authors did here. I could go on and on about the graphics, but you already get the basic idea.

Sound: A+. Unlike the "space game"-like sound effects you'd hear in the arcade game, all the sounds in Buzzard make sense, and never seem misplaced. This is CoCo sound at its best.

Animation: A+. The animation in Buzzard is so good, it's almost like a cartoon. The animation of everything is extremely smooth without being overdone. The animation of the fire alone seems to be at least eight frames. The game has incredible animation all around.

Game engine: B-. There's a little bit of flicker, and the frame rate lags when there are enough birds on the screen at one time.

The restarting of the game could be a little bit better. You can only start a new game by pressing the "R" key. No other key on the keyboard will do. *Then* you have to select the number of players, even though you already did that the first time you started the game.

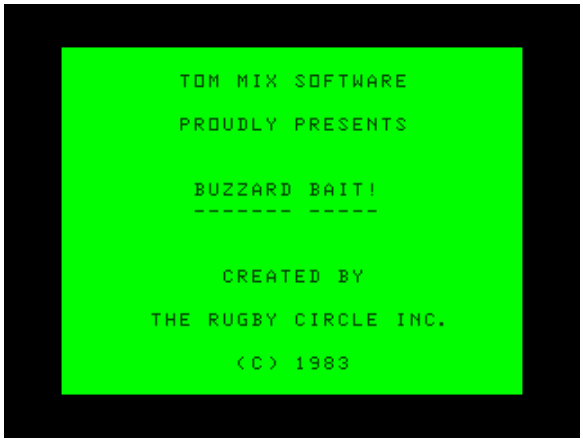
It would make more sense to have the game ask you for the number of players, and when the game ends, you can restart with the same number of players by pressing the Fire Button. If you want a different number of players, you could press the R key to bring up the player select screen, or better yet - just press the 1 or 2 key to start the game with a different amount of players.
I say, if the game is going to have you use the keyboard as much as it does in Buzzard Bait (and I've said this before), there's no point in making it a joystick game to begin with.

Buzzard also has some rather annoying bugs. I won't explain all of them here, but the worst one of all is when someone is jousted when one of the two birds in the joust is being held by the Lava Hand. Any time the jousted bird flaps his wings when he's low enough, it erases part of the lava below. And the lava below is never re-drawn until the whole game is restarted. Until then, you've got these "holes" stuck in the lava made from flapping wings.

Game-play: A+. In a nutshell, the game-play really doesn't leave anything to complain about. There's enough variance that the player doesn't get bored with the game too easily. The game plays differently enough each time to encourage the player to adapt strategies rather than memorize sets of particular moves.

Overall: A. The only things keeping Buzzard from scoring an absolute A+ here are the bugs in the game engine and the game's annoying dependence on the keyboard when the game-play relies on the joystick.

Purists of arcade games who don't like arcade emulators (or bugs) will want to try "Lancer" instead. I recommend all others stick to Buzzard Bait for the reasons I've stated in the text.



Having a moving experience

Or

Moving files to and fro between a PC and a Coco.

By Bob Devries

First let me say, that unless you have a modified coco disk controller, you cannot read 1.44MB 3.5" disks on your Coco, even if you have the correct hardware in the form of a 1.44MB 3.5" disk drive.

So what needs to be done to get those files you downloaded from the internet on your PC, to work on your Coco?

I'm going to cover two scenarios here.

You have a 5.25" drive in your PC, and want to use that type of disk on your Coco.
You don't have a 5.25" disk on your PC, but you do have a 720K 3.5" drive on your Coco.

There are two programmes available to read and write Coco formatted disks on your PC. They came with the Jeff Vavasour Coco Emulator, and are called

DSKINI.EXE, and **RETRIEVE.EXE**. They write and read Coco disks respectively. Modified versions of these programmes are on the RTSI ftp site:

ftp://www.rtsi.com/RSDOS/incoming/Coco3%206309%20Emulator.zip

As well as that, the Windows programme WimgTool.exe, which comes with the MESS Emulator, is a very useful tool. This programme will allow you to copy files to and from disk images, that are used by the above two programmes.

OK, so you want to read a Coco disk in the PC's drive (this applies to either 5.25" or 3.5" disks). Here is the help file from **RETRIEVE.EXE**:

Usage: **RETRIEVE** [/2] [/8] [/D] [/T] [/R] [d:] [path\]diskname[.DSK]

"d:"	source drive (default A:)
"path"	destination directory for virtual disk
"diskname"	name of virtual disk
/2	to read the second side of the disk
/8	if using 80-track disks in a 1.2Mb drive
/D	if using an double sided disk in drive
/T40	40 (tracks/per/side) (default = 35)
/T80	80 (tracks/per/side) (default = 35)
/R	to retrieve the ROM image (see manual)

You must specify the name of the virtual disk.

Now, if you have a 40 track disk used by Disk Extended Color Basic, the command line is:

RETRIEVE A: filename.dsk
This will create a file called "filename.dsk" in the current directory on your PC which is a sector-by-sector image of your Coco disk in the A: drive.

You can then use the **Wimgtool.exe** programme under Windows to get individual files from that image.

To place an image file from your PC onto a disk so that the Coco can read it as a normal disk, use the **DSKINI.EXE** programme. Here's its help output:

Usage: DSKINI [/2] [/8] [/D] [/T] [d:] [[path\]diskname[.DSK]]

"d:"	destination drive (default A:)
"path"	source directory for virtual disk
"diskname"	name of virtual disk
/2	to write to the second side of the disk
/8	if using an 80-track disk in 1.2Mb drive
/D	if using an double sided disk in drive
/T40	40 (tracks/per/side) (default = 35)

/T80 80 (tracks/per/side) (default = 35)

At least one parameter must be specified.

For both **RETRIEVE.EXE** and **DSKINI.EXE**, the default values will read or create a single-sided 35 track disk. So to create a disk for Disk Extended Color Basic to read, you would use the following command line:

DSKINI A: filename.dsk

If you have no 5.25" drive in your PC (these days, PC's don't have them), and you have a 720K 3.5" disk drive on your Coco, you can still make this work. Here's how.

If you have a 5.25" disk drive as drive 0 and a 720k drive as drive 1, use the backup command to make a copy of the disk on the 720K disk, after formatting it using Disk Extended Color Basic's DSKINI command:

DSKINI 1

BACKUP 0 TO 1

Now take that disk to your PC and put it into the 1.44MB drive, and type the **RETRIEVE.EXE** command as above.

If you're using OS-9 on your Coco, you will most likely be using both sides, and all available tracks on the drive. That means 40 tracks and two sides on a 5.25" disk, and 80 tracks, two sides on the 3.5" 720K disk. Adjust the parameters accordingly in both **RETRIEVE.EXE** and **DSKINI.EXE** like this:

RETRIEVE /2 /T80 A: filename.dsk
DSKINI /2 /T80 A: filename.dsk

For those who have managed to copy a disk image file onto their Coco OS-9 disk drive, there are two programmes which will allow you to get files from those images. They are "**OS9DSK**" to read OS9 disk images, and "**RSDSK**" to read DECB images.

The Usage output for **OS9DSK** is:

Usage: os9dsk -dir filename.DSK DSKpath
os9dsk -get filename.DSK DSKpath os9file
os9dsk -proc filename.DSK DSKpath

For **RSDSK** it is:

Usage: rsdsk -dir filename.dsk
rsdsk -get filename.dsk rsdosfile os9file
rsdsk -proc filename.dsk
rsdsk -help for more details

Hope it helps, folks.
Bob Devries

"First Gear"

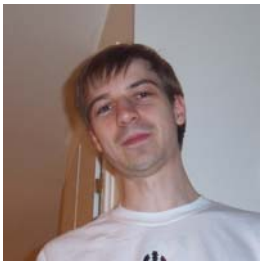
A One-liner by Retro Rick

A very simplistic demonstration of screen-swapping. How simplistic? Change the Y coordinates of that PRESET,BF statement to read "LINE(112,0)-(144,191),PRESET,BF" and you'll see what I mean!

Palette animation with the CoCo 3 is also possible, and can do much fancier things than this. So all of you CoCo 3 programmers out there have a chance to release your one-liner animation program for CoCo 3s before I get a CoCo 3 and finish my own CoCo 3 demo!

```
1 PMODE0:PCLEAR8:FORL=0TO14STEP2:P=P+1:PMODE0,P: PCLS1:
LINE(112,14)-(144,176),PRESET,BF:FORY=L+0TO176STEP16: LINE(128,Y)-
(128,Y+4),PSET:NEXTY,L:FORI=-1TO0:T=PEEK(275):Z=(P>7)*8:PMODE0,P:
SCREEN1,1:P=P+1+Z:FORW=-1TO0:W=T=PEEK(275):NEXT:
I=INKEY$="":NEXT
```

This is **sockmaster** “john”



John Kowalski (Sock Master)
http://www.axess.com/twilight/sock/

The editor asks:

>Ok do you have pets kids grandkids etc?

I have two kids - Steven who came with us to PennFest 2000, and Teresa who is now about as old as Steven was back then. We also have a Beagle and two cats.

>where do you live at?
>lived there long?

We live in Saint Hubert, which is just south of Montreal, in Quebec. We’ve been here about 9 years now.

>what do you do for a living nowadays?

I make video games, actually :) I think there are a lot of people out there who learned about computers, programming or electronics on the CoCo and then used that knowledge professionally. Many of the little tricks I learned to do on the CoCo actually come in handy when programming games.

>they call you sock master does this have anything to do with a sock puppet?
>how did you get that name?

No sock puppets, but it's still pretty silly... In the days before the internet, I used to use BBS’s. I had gotten kicked off one board because I was a troublemaker, but I wanted to log back in with a different name. I didn't really care what it was. I had my feet up on the desk while I was filling in the login name and saw my socks - so I wrote SOCK. Well, that wasn't enough, so I added MASTER. At first I used it when I didn't want people to know who I was... but it turned out to be catchy. It sort of snowballed. The more I used it, the more programs I wrote, the more famous it became - to the point where "Sock" was more recognizable than my real name.

>What is your favorite part of the coco?

The warm chewy center :) Actually it's pretty much a toss-up between the CPU and the GIME chip in the CoCo3. The 6809 (and 6309) CPU is very nice - it makes the CPUs in other 8-bit computers look like they're wearing t-shirts and sweatpants compared to the classy elegance that is the 6809.

The GIME is neat too, but not always for its intended reasons. It's neat because it always has some new (generally unintentional) surprises just waiting to be found by programmers.

>when did you start with the coco?
>why did you pick the coco?

We actually had a TRS-80 before the CoCo, but there were three kids in the family and I think at some point we convinced my parents to buy a second computer. At the time there weren't very many choices for an inexpensive computer. It was either a CoCo or a Vic-20. I think we made the right choice.

>do you stay active with the email list? Do you go to the coco fests?

I still read the email list and occasionally post messages when the topic is up my alley. The last fest I want to was PennFest 2000. It would be nice to go to the next CoCo fest but it's hard to know in advance if I'm going to be free or be crazy-busy with work. Free time is difficult to plan, unfortunately.

>what one thing do you miss about the old coco days?

I miss the variety of computers. Back then, there were many different companies, all making different computers. Sure, I didn't like them all, but that's sort of the point too. Nowadays it's all the same - there is the PC, and there is the PC, and every now and then a Mac.
Things were also smaller then - less daunting. It made people more creative, more willing to try something new, or even try at all. It's *ok* to be a little fish in a little pond, but it's hard to stand out in the ocean.

>if you could bring it back would you?

Nickolas Marentes and I are working on a time machine right now.

>what kinda stuff do you usually work on with the coco nowadays?

Not as much as I used to, but I still write myself some small programs now and then. I also use the CoCo as my own sort of personal assistant at work.

I still have a couple of big ideas that I would eventually like to finish on the CoCo. The big hook for me is something that has yet to ever be done on the CoCo. One day I'll find the time and make one or two more surprises.
There is this yearly MiniGame competition, where people make small games for all kinds of systems.

http://www.ffd2.com/minigame/

I think it would be cool to see a few CoCo games in it this year. I'm hoping to write a couple of 1-4K CoCo games this year and it would be really cool if others wrote some too!

>what is your favorite game or program for the coco?

It's very hard to pin it down to just one. How about a quick list of 10 of my favourite CoCo games?

Shanghai
Doubleback
Downland
Dungeons of Daggorath
Buzzard Bait
Gate Crasher
Zenix
Lunar Rovar Patrol
Mine Rescue
Dragon Slayer

>video games eh! Like for Nintendo or what?

Yep. Mostly I work on Game Boy Advance games.

>What little tricks helped you the most with making video games?

The CoCo isn't the fastest computer in the world - so sometimes you have to find tricks to make it do things it isn't supposed to be able to do.
Sometimes it's just programming tricks to squeeze extra performance out of the processor and sometimes it's using the hardware that you have to do altogether different things than it was ever intended to do.

>what is one of your more recent video game projects?

The most recent Game Boy Advance game I worked on is Namco Museum 50th

Anniversary. It's a compilation of the classic arcade games Pac-Man, Ms. Pac-Man, Galaga, Dig Dug and Rally-X.

>did you go to that video game making college?

No, it wasn't around yet when I was in school. I took electronics instead and most of the programming I learned was actually at home on the CoCo.

>if you could go back in time where would you go?

I don't know really. Maybe just go back and try to stop some of the sillier ideas from happening, give the guys planning the CoCo 3 some different ideas and then go have a Coke at the original McDonald's.

>what is your favorite project that you have done/or are most proud of?

That's kind of hard to say. I'm proud of different things for different reasons. Gloom was cool because it proved fast 3D was possible on the CoCo,
which started the ball rolling on Nick Marentes' Gate Crasher. And more recently, I used the "Gloom" technique in the 3D modes of Tron 2.0 on the Game Boy Advance.

I'm also proud of my original "CoCo Demo" because it basically floored everybody the first time they saw it - nothing like that had ever been seen on the CoCo before.

Zenix by Jeremy Spiller
Review by Briza.

Zenix is 1 of the fastest games ever done on any Tandy color computer, This Game really showcased the power of what a coco 3 could achieve. Like to see a C-64, or any other 8Bit computer do this game, or do a better version.
At First Glance, you think your playing just another Galaga clone, But don't be fooled,This is way better, This version, has a ending, After 32 levels, you get to destroy the home world of the Alien invaders, And During the 32 levels, you get to shoot alien mother ships, these range from Dragon fly looking ships, to other exotic bug like ships.
I won't tell you the ending, I'll leave that up to you to find out, But I will say this, after level 11, the screen will shake violently , every time you are hit, this really makes the game standout from the rest.
As you look at the score board screenshot, you'll see I have the 2nd highest score, Drats only just missed beating Jeremy Spiller.
The game controls, Joystick plugged in the right joystick port on the coco 3.
A RGB Monitor is a must. Have the sound volume turned up loud.

laters

briza

Accessing an OS-9 Disk in the Disk Basic Portion of an RGBDOS Hard Drive

by
Robert Gault

One of the nice aspects of several Coco emulators is that they support virtual hard drives. If you are running OS-9 from one of these emulators, then you should obtain the emudsk package created by Alan DeKok which permits access to the virtual hard drive from OS-9.

I have made available (from my web site) a version of RGBDOS for emulators which permits access of all or part of a virtual hard drive using an emulator under Disk Basic. One of the features of RGBDOS is that you can boot into an OS-9 partition on a hard drive from a virtual Disk Basic disk on the same hard drive. You can also do this in an emulator using my RGBDOS mod and the emudsk package.

One of the less convenient aspects of the above is that os9gen and cobbler work with floppy disks not hard drives and certainly not RGBDOS Disk Basic virtual drives. So, if you want to alter your os9boot file, it means mounting a floppy drive, using os9gen or cobbler on the floppy, and then backing it up to the hard drive.

I have made the process much simpler by modifying Alan's driver so that it can access the virtual Disk Basic disks on the hard drive. The source code for the driver and descriptor are shown below.

No effort was made to directly support more than one virtual disk basic drive at a time but access to all 256 drives is trivial if inconvenient. All you have to do is use dmode to change the step value of the v0 descriptor to the desired drive number. I decided to use the stepping value to represent the drive# because stepping is not important for a hard drive and no changes to dmode were required. Since it is now simple to cobbler the boot drive, it is not likely that access to more than one virtual Disk Basic drive on a .vhd image will be needed.

The code below will only work with emulators. However, I have modified the hdisk driver on my real Coco hard drive system to access my RGBDOS boot drive in exactly the same manner as shown below.

```
* EmuDisk floppy disk controller driver
* Edition #1
* 04/18/96 : Written from scratch by Alan DeKok
*                                 aland@sandelman.ocunix.on.ca
*
*  This program is Copyright (C) 1996 by Alan DeKok,
*                  All Rights Reserved.
*  License is given to individuals for personal use only.
*
*
*  Comments: Ensure that device descriptors mark it as a hard drive
```

```
*
* 12/15/05
* Modified by Robert Gault to access OS-9 virtual disks in the
* Disk Basic portion of an RGBDOS hard drive.
* robert.gault@att.net
*
*   $FF80-$FF82: logical record number
LSN      equ  $FF80           where to put the logical sector number

*
*   $FF83: command/status register.
*          Output: 0=read, 1=write, 2=close.
*          Input: 0=no error, non-zero=error codes (see below).
command  equ  $FF83           where to put the commands

*
*   $FF84-$FF85: 6809's 16-bit buffer address (cannot cross an 8K boundary due
*          to interference from the MMU emulation).
buffer   equ  $FF84           pointer to the buffer

*
* Returns:
*
* 0=successful
* 2=not enabled
* 4=too many MS-DOS files open,
* 5=access denied (virtual HD file locked by another program
*      or MS-DOS read-only status)
* 6/12=internal error
* 254=invalid command byte
* 255=power-on state or closed.
*
*  The "close" command just flushes all the read/write buffers and
* restores the metacontroller to its power-up state.  The hard drive must be
* enabled by the user using the MS-DOS command "ECHO >COCO3.VHD" (another
* crash safeguard), so error code 2 indicates this has not been done.

* New code for access to RGBDOS Disk Basic drives. This defines
* RGBDOS Disk Basic offsets. RG
         org 0
stack    equ .
thi      rmb 1     High word of offset
tlo      rmb 2     Low word of offset
rA       rmb 1     Register temporary saves
rB       rmb 1
rX       rmb 2
* End of new section

* Changed name to prevent collision with EmuDsk. RG
         nam   VEmuDsk
         ttl   os9 device driver

         ifp1
         use   /DD/DEFS/defsfile
         endc

tylg     set   Drivr+Objct
```

```
atrv     set   ReEnt+rev
rev      set   $01

         mod   eom,name,tylg,atrv,start,size
         fcb   $ff

* Space for one hard drive
         org   DRVBEG+DRVMEM
* This will be obtained from the descriptor v0. At the moment
* only one virtual Disk Basic drive is supported. RG
VOfset   rmb   3        This will be filled with OS-9 size
size     equ   .

         fcb   $FF            This byte is the driver permissions
name     fcs   /VEmuDsk/
         fcb   1              edition #1


* Entry: Y=Ptr to device descriptor
*        U=Ptr to device mem
*
* Default to only one drive supported, there's really no need for more.
*
* RG. Only one hard drive but 256 or more virtual Disk Basic drives on
* hard drive. This example assumes a single Disk Basic partition
* starting immediately after the OS-9 partition.
INIT     lda   #$FF          'Invalid' value & # of drives
         leax  DRVBEG,u       Point to start of drive tables
         sta   ,x             DD.TOT MSB to bogus value
         sta   <V.TRAK,x      Init current track # to bogus value
* New code for RGBDOS Disk Basic access. RG
         leau  VOfset,u       point regU to data area
         ldx   IT.SAS+2,y     get lo word of offset from descriptor
         stx   1,u            save it
         ldb   IT.SAS+1,y     get hi byte of offset
         stb   ,u             save it
* End of new code. RG

* for now, TERM routine goes here.  Perhaps it should be pointing to the
* park routine? ... probably not.
TERM
GETSTA   clrb                 no GetStt calls - return, no error, ignore
L0086    rts

start    lbra   INIT           3 bytes per entry to keep RBF happy
         lbra   READ
         lbra   WRITE
         lbra   GETSTA
         lbra   SETSTA
         lbra   TERM

* Entry: B:X = LSN
*        Y   = path dsc. ptr
*        U   = Device mem ptr
READ     clra                 READ the sector
         bsr   GetSect        Go read the sector, exiting if there's an error
         tstb
```

```
         bne   GETSTA         if not sector 0, return
         leax  ,x             sets CC.Z bit
         bne   GETSTA         if not sector zero, return

* LSN0, standard OS-9 format
* Actually, this isn't really necessary for a HD, as the information in
* LSN0 never changes after it's read in once.  But we'll do it anyhow
         ldx   PD.BUF,y       Get ptr to sector buffer
         leau  DRVBEG,u       point to the beginning of the drive tables
         ldb   #DD.SIZ        copy bytes over
copy.0   lda   ,x+            grab from LSN0
         sta   ,u+            save into device static storage
         decb
         bne   copy.0
         clrb
         rts

WRITE    lda   #$01           WRITE to emulator disk, and fall thru to GetSect

* Get Sector comes here with:
* Entry: A = read/write command code (0/1)
*        B,X = LSN to read/write
*        Y   = path dsc. ptr
*        U   = Device static storage ptr
* Exit:  A   = error status from command register
GetSect  tst   <PD.DRV,y      get drive number requested
         bne   DrivErr        only one drive allowed, return error

         pshs  x,d            save LSN and command for later
* New code to get RGB Disk Basic drive#. RG
         ldb  PD.STP,y        get step value now vdrive#
         clra
         tfr   d,x            regX now is vdrive#
         ldd   VOfset+1,u     put 3 byte offset on stack
         pshs  d
         ldb   VOfset,u
         pshs  b
         cmpx  #0             is there a further offset for the drive?
         beq   GS2            if not branch
GS1      ldd   tlo,s          get lo word of OS-9 size
         addd  #630           sectors on a 35T disk
         std   tlo,s          save new value
         ldb   thi,s          get hi byte of OS-9 size
         adcb  #0             add in any carry from first addition
         stb   thi,s          save new value
         leax -1,x            next vdrive
         bne   GS1            continue until correct drive# reached
GS2      ldd   tlo,s          get full offset to vdrive
         addd  rX,s           add lo word of LSN
         tfr   d,x
         ldb   thi,s          get hi offset byte
         adcb  rB,s           add LSN hi byte
         lda   rA,s           recover command
* End of new code. RG
         stb   >LSN
         stx   >LSN+1
* Pop temporary stack entries. RG
```

```
         leas  rB,s
* The rest of the code is unchanged. RG

         ldx   PD.BUF,y       where the 256-byte LSN should go
* Note: OS-9 allocates buffers from system memory on page boundaries, so
* the low byte of X should now be $00, ensuring that the sector is not
* falling over an 8K MMU block boundary.

         stx   >buffer        set up the buffer address
         sta   >command       get the emulator to blast over the sector
         lda   >command       get the error status
         bne   FixErr         if non-zero, go fix the error and exit
         puls  b,x,pc         restore LSN and exit

DrivErr  leas  2,s            kill address of calling routine (Read/Write)
         comb
* FIND ERROR CODE TO USE
*        ldb   #E$            find appropriate error code...
         ldb   #E$NotRdy      not ready
         rts

* Emulator error codes translated to OS-9 error codes.
*
* 2=not enabled
*      E$NotRDy - drive is not ready
*
* 4=too many MS-DOS files open,
*      E$
*
* 5=access denied (virtual HD file locked by another program
*      or MS-DOS read-only status)
*      E$WP  - write protect
*
* 6/12=internal error
*      E$CRC - CRC error
*
* 254=invalid command byte
*      E$
*
* 255=power-on state or closed.
*      E$NotRdy - drive is not ready
*
FixErr   leas  5,s         kill B,X,PC from GetSect routine
         cmpa  #02
         beq   NotRdy
         cmpa  #255
         beq   NotRdy
         cmpa  #5
         beq   WP
         cmpa  #6
         beq   CRC
         cmpa  #12
         beq   CRC

* if it's something we don't recognize, it's a seek error
         comb
         ldb   #E$Seek        seek error
```

```
          rts

NotRdy    comb
          ldb   #E$NotRdy      not ready
          rts

WP        comb
          ldb   #E$WP          write protect
          rts

CRC       comb
          ldb   #E$CRC         CRC error
          rts

L03D4     comb
          ldb   #E$Write       write error
          rts


L03E0     comb
          ldb   #E$Read        Read error
          rts

SETSTA    ldx   PD.RGS,y       Get caller's register stack ptr
          ldb   R$B,x          Get function code
          cmpb  #SS.WTrk       Write track?
          beq   format         Yes, ignore it
          cmpb  #SS.Reset      Restore head to track 0?
          beq   format         Yes, ignore it
          cmpb  #SS.SQD        sequence down the drive (i.e. park it)?
          beq   park
          comb                 set carry for error
          ldb   #E$UnkSvc      return illegal service request error
          rts

park      ldb   #$02           close the drive
          stb   >command       save in command register

format    clrb                 ignore physical formats.  They're not
          rts                  necessary

          emod

eom       equ   *
```

==========

```
* 12/15/05 Robert Gault
* Modified descriptor to work with virtual Disk Basic drives on an
* RGBDOS MESS emulator system.

 IFP1
 USE /DD/DEFS/defsfile
 ENDC

type SET Devic+Objct
 MOD rend,rnam,type,ReEnt+1,fmnam,drvnam
 FCB $FF  all access modes
 FCB $07,$FF,$E0 device address

 FCB optl number of options

optns EQU *
 FCB DT.RBF RBF device
 FCB $00 drive number
* This line must be user modified to point to the drive used to
* boot OS-9. RG
 FCB 248 virtual Disk Basic drive assignable with dmode: was step
 FCB $80 type=nonstd,coco
 FCB $01 double density
* This must be for a 35 track disk. RG
 FDB $0023 tracks
 FCB $01 one side
 FCB $01 no verify
* These must be standard floppy disk values. RG
 FDB $0012 sectors/track
 FDB $0012 "", track 0
 FCB $03 interleave
* This must be a standard floppy disk value. RG
 FCB $08 min allocation
* New descriptor entry which must correspond to the size of the
* OS-9 portion of the .vhd drive. RG
 fcb $05,$A0,$00 OS-9 partition size for vhd
optl EQU *-optns

rnam FCS /V0/
fmnam FCS /RBF/
drvnam FCS /VEmuDsk/

 EMOD
rend EQU *
 end
```

Solitaire

A BASIC program
By
Torsten Dittel

[Editorial note: because of page size limitations, I have had to split some of the lines of this program listing. Those lines are marked via a color change to orange. Therefore all lines that are in orange font are one continuous line. Lines 11, 13, 14, 20 and 52 are affected.]

```
0 :''''''''''''''''''''''''''''
1 :'      SOLITAIRE V1.0       '
2 :''''''''''''''''''''''''''''
3 :'        (C)1987 BY         '
4 :'                           '
5 :'      TORSTEN DITTEL       '
6 :'   HUBERT-GEUER-STR. 51    '
7 :'       D5040 BRUEHL        '
8 :'       WEST GERMANY        '
9 :''''''''''''''''''''''''''''
10 POKE65495,0:CLS:CLEAR255:DIMF(6,6):DIMM$(30):PMODE4,1:PCLS5:SCREEN1,1
11
A$="UF2R4E2U2H2L4H2U2E2R4F2NUBR4E2R4F2D8G2L4H2U8BR12U2LR2LD12LR8NUBR4R2LU12LR2
BR4DUR8DUL4D12LR2BR6R2LU10E2R3F2D4NL7D6LR2BR4R2LU12LR2BR5D12LR2LU12LR6F2D2G2NL5
F2D4LR2BR4RNR7U6NR5U6LR8DBD10D
12 FORC=0TO1:FORR=0TO1:DRAW"C0BM"+STR$(77+C)+","+STR$(11+R)+A$:NEXTR,C
13 DRAW"BM59,34U4DRD2RBDU4BR2D4R2BU2L2BU2R2BR2D4EFU4BM179,34U4R2BD2L2BD2
R2BR2U4DRD2RBDU4BR2D4REU2HL":A$="BU4F4U2R3U4L3U2G4":DRAW"BM60,156"+A$+"BM188,14
8A2"+A$:POKE178,2
14
FORC=2TO4:FORR=0TO6:IFF(C,R)THENNEXT:ELSEF(C,R)=2:GOSUB54:GOSUB45:GOSUB43:F(R,C
)=2:X=64+R*20:Y=32+C*20:GOSUB45:GOSUB43:NEXTR,C
15 C=3:R=3:F(C,R)=1:X=124:Y=92:GOSUB44
16 L=0
17 POKE338,0:IFPEEK(338)=254THEN17
18 C=INT(JOYSTK(0)/10.5):R=INT(JOYSTK(1)/10.5):GOSUB54
19 CIRCLE(X,Y),9,0:IFPEEK(338)<>254THENCIRCLE(X,Y),9,5:GOTO18
20 CIRCLE(X,Y),9,5:IFC=0ANDR=0THENRUN:ELSEIFC=6ANDR=0THENEND:ELSEIFL THEN33:
ELSEIFC=0ANDR=6THEN46:ELSEIFC=6ANDR=6THEN49
21 IFF(C,R)<2THEN17
22 IFC<2THEN24
23 IFF(C-2,R)=1THENIFF(C-1,R)=2THEN31
24 IFC>4THEN26
25 IFF(C+2,R)=1THENIFF(C+1,R)=2THEN31
26 IFR<2THEN28
27 IFF(C,R-2)=1THENIFF(C,R-1)=2THEN31
28 IFR>4THEN17
29 IFF(C,R+2)=1THENIFF(C,R+1)=2THEN31
30 GOTO17
31 C1=C:R1=R:F(C,R)=1:GOSUB44:CIRCLE(X,Y),2,0
32 L=1:GOTO17
```

```
33 IFF(C,R)<>1THEN17
34 IFC=C1 ANDR=R1 THENGOSUB44:GOSUB43:F(C,R)=2:GOTO16
35 IFC1<>C ANDR1<>R THEN17
36 IFABS(C-C1)>2ORABS(R-R1)>2THEN17
37 GOSUB53:IFF(C2,R2)<>2THEN17
38 F(C,R)=2:GOSUB43
39 GOSUB56:F(C2,R2)=1:GOSUB44
40 GOSUB55:GOSUB44
41 M$(M)=STR$(1000+C1*1000+R1*100+C*10+R):M=M+1:MH=M
42 GOTO16
43 PAINT(X,Y),,0:RETURN
44 PAINT(X+1,Y),5,5
45 CIRCLE(X,Y),6,0:CIRCLE(X,Y),7,0:RETURN
46 M=M-1:IFM<0THENM=0:GOTO17
47 GOSUB52:F(C1,R1)=2:GOSUB55:GOSUB43:F(C,R)=1:GOSUB54:GOSUB44
48 GOSUB53:F(C2,R2)=2:GOSUB56:GOSUB43:POKE338,0:GOTO18
49 IFM+1>MH THENM=MH:GOTO17
50 GOSUB52:F(C1,R1)=1:GOSUB55:GOSUB44:F(C,R)=2:GOSUB54:GOSUB43
51 GOSUB53:F(C2,R2)=1:GOSUB56:GOSUB44:M=M+1:POKE338,0:GOTO18
52 C1=VAL(MID$(M$(M),2,1))-1:R1=VAL(MID$(M$(M),3,1)):C=VAL(MID$(M$(M),4,1)):
 R=VAL(MID$(M$(M),5,1)):RETURN
53 C2=C-(C-C1)/2:R2=R-(R-R1)/2:RETURN
54 X=64+C*20:Y=32+R*20:RETURN
55 X=64+C1*20:Y=32+R1*20:RETURN
56 X=64+C2*20:Y=32+R2*20:RETURN
```

An interview with Diego Barizo.

Diego Rodrigo Barizo Fernández

Editor: One year ago today you had bought a coco3 from cloud9 have you acquired any other coco systems?

Diego: Yes, many :-) My first CoCo was a 16 Kb CoCo 2, that I traded for a 64 Kb a few weeks later. As soon as the CoCo 3 was available, I've got one. This was in Uruguay, back in the 80s. Once I've got to the US, I started "ebay-ing" for CoCos. 1 CoCo1, 4 CoCo2, and a CoCo 3. When I decided that I wanted a 512 Kb, I went straight to Cloud9; safer and cheaper!

Editor: I see you have several pieces of hardware, care to give a list of what all you have to date?

Diego: Well, besides the CoCos, I have my pride and joy, a FD-502 with 2x360 Kb, the Orchestra-90, the Sound/Speech pack, a MPI, the Koala touchpad, the color mouse, an assortment of joysticks and a hi-res adapter, 2 printers (DMP-106, DWP-220), CCR-81, the "electronic book" and of course, Roy's VGA adapter.
I also have a CGP-115, an X-Pad, DC-3 modem, and a plug'n'power controller that are either out of order, or missing some parts.
All of this, except for the VGA adapter, thanks to ebay.

Editor: What is your software collection like?

Diego: You could say that I have 3 collections.
First, one of real, original software, most of it in Program Paks (~60), but a few disks and even tapes. Many of these from http://www.vintagefunworld.com/.
Second, copies of programs that I've got from the net and transferred to 5 1/4 disks (some 30 disks)
And last, as many DSK files as I could find, that I keep on my LifeDrive (http://www.palm.com/us/products/mobilemanagers/lifedrive/) with an emulator, so I can plug it to any PC and play my games almost anywhere.

Editor: Some people like to pack and repack their systems, or modify them to do other things, are you a traditionalist or do you also modify yours?

Diego: I'll have to say that I'm a traditionalist, but only by lack of skills :-) I have a few plans to do some "repacking", but I'm not sure I would risk a CoCo to depend on my almost non-existent skills in the field.

Editor: What is the main thing you use your coco for?

Diego: I guess I would say BASIC programming. I've been doing it for 22 years, and I'm still learning new ways to do things. I find it so easy and intuitive to do, that it's a real pleasure.

Editor: I hear that when a certain relative comes to stay you have to store your gear; I also hear that when she leaves you can't get the coco setup fast enough ☺ tell me about his, please?

Diego: Well... that certain relative will be my mom, and you are right. For 17 months after moving to the USA, I was living in single room, 12x9, and for a year, she was living there with me... 😐 (Genesis is playing "Illegal alien" in the background). Right after she left, I moved to a bigger place, and suddenly, all the computers that had been stored for months, started to blossom, covering almost every available surface, including the bed that my mother used. When she came back, the computers retreated, but now they are enjoying their freedom again. Luckily, my CoCo3 has her own, sacred place. (In Spanish, computers are females, a fact that makes jokes easy to come by!)

Editor: Have you had a chance to test out new hardware products yet for the coco?

Diego: Only Roy's VGA adapter and I'm very happy with it. Since I still have my CMP monitor connected, I can see the difference, and it's huge! It's like when I've got my first 1 MB SVGA card for my old 386 ☺

Editor: What is on your wish list of hardware/software?

Diego: I need a hard drive, yes I do ;-) lack of room and power outlets might make me go for a SuperIDE with a CF card. And I would love to have a way of keeping 3 joysticks, a mouse, and a hi-res adapter connected to the CoCo all the time... but there are only 2 joystick ports ☹ In the software field, I'm looking for a word processor that will let me do fancy letters for my friends and family in the DMP-106.
And the sequels to "Gate Crasher" and "Dungeons of Daggorath" ;-D
Maybe someone will do "DoD 2" using the "Gloom" engine???????

Well another issue of CoCoNutz is out. I really enjoyed watching this one take on a theme of its own. I think it goes to show that in the winter months what are on people's minds. This issue clearly is about games, and I have learned a lot about them. I want to thank all my contributors, and I was especially glad to see some new contributors. I want to get more and more people to add submissions with each newsletter. This time your homework guys is to write a 5 sentence review about any of the programs you have tried. Ex would be; if you tried the program, did it work, did you enjoy it etc. Of course you may mail anything newsletter related to my mailbox coconutznewsletter@yahoo.com. Also if you would like a more printer friendly version of the newsletter or any of the past ones let me know and I will set it up for download on coco3.com. I want to thank Aussie Bob (my sergeant dad), Mannequin, and Militant Buddha for helping me pull these newsletters together. You each know how you have helped and it is appreciated. The biggest thanks go to Roger for hosting the newsletter. ((((((((((((((((hugs 2 yall))))))))))))))))

Mary