3D CONSTRUCTION KIT USER GROUP MAGAZINE
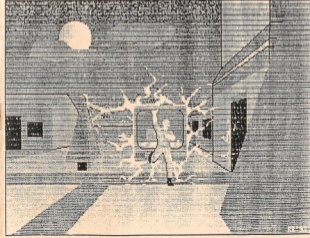
ISSUE 15 OCTOBER-NOVEMBER 1993

STEP INTO THE VIRTUAL WORLD WITH 3D CONSTRUCTION KIT.

EDITORIAL

Welcome to issue 15 of the 3D Construction Kit User Group Newsletter.

I always seem to start the editorial by telling everyone that the last month has been rather hectic. These last couple of weeks were no exception. I said I wasn't going to mention floods again but honestly, I think there is a big finger pointing from the sky at me saying, "This one!". Would you believe that after just finishing getting back to normal after the flooding and getting a lovely new lounge my next door neighbour forgot she was filling the bath, went out and, not only did she flood her bathroom and lounge but she flooded mine too! I reckon I am going to have to change my name to Noah. Good grief, yet another new carpet within the space of a month and redecoration all over again. I reckon someone, somewhere knows just when the newsletters are due out and decides to dampen my enthusiasm. Ah well, there are lots of goodies in this issue, especially an article by Robin Ball for Kit 2 users that could just be a "Godsend" for those frustrated by the lack of a save and load facility. Watch out for it starting on page 22.

With all the hassle and rush to get the last two issues out without too much delay I forgot to tell everyone that the cover for issue 13 was designed by Tony Hartley and the cover for issue 14 designed by Bearnd Teanzer, sorry chaps, I didn't mean for you to go without due credit for all your effort and excellent work. From time to time I do make mistakes - sometimes real clangers - if you spot one please don't hesitate to let me know. One of my main faults is dyslectic fingers at the keyboard, getting letters back to front, sometimes with amusing results and if I am in a rush I sometimes forget to use the spelling checker.

I've no real news about Kit 2 version V2.09 yet which is making me rather grumpy as I had hoped to be able to tell everyone that it is ready. The programmers are still working on all versions after the excellent playtests done by Tony Hartley and Mieke Van Der Poll. Poor Mieke, not only did I inflict the Amiga playtest on her but also the PC playtest at the last minute. Everyone is beginning to get a bit frustrated with the wait, I know, but it can't be much longer now. Problem is that the program is so complex and full of so many different features that it takes quite a while to iron out all the niggly bits and pieces without affecting other parts of the program. I know the programmers are doing their best and are as anxious as everyone else to come up with a version that everyone is happy with. As soon as I know anything I will let you all know. I have a box full of stamped, self addressed jiffy bags here that members have sent me ready for the new version and I'll fill them and post them off as soon as the disks arrive. If anyone would like to add their own jiffies to the pile then please feel free to do so. Don't forget to state which format you want and, in the case of PC, which size discs for your machine.

The next issue will be the December/January issue and, hopefully, will have a bit of a festive flavour. I hope to run a competition in that issue, thanks to Tony Hartley for the idea so watch out for that. Contributions are a bit scarce at the moment so I hope that I will receive plenty in time for that issue, if you would be so kind as to help out. Right, that really is quite enough waffle from me for one issue, hope you enjoy this one. See you all again with issue 16!

Mandy

LETTERS

Dear Mandy

I am enclosing some routines to make a scrolling landscape whilst the player is standing still. I hope it will come in useful. On another subject I have programmed a painting program that randomly generates buildings for a city skyline as a 2D painting. The program draws the buildings in detail and every picture is not the same. I programmed it to help the construction imagination of 3D Construction Kit Users. The paintings are saved as PI1 files. I use an Atari STE. If anyone would like a copy of this program then please send me a blank disk and a stamped self addressed envelope for the return.

R. Connell - ATARI ST
16 Thorn Street, Astley Bridge, Bolton, Lancs, BL1 8LA

What a generous offer. I'm sure everyone will find your program extremely useful. The demo disk of the scrolling landscape works beautifully and I'm sure the routines will be welcomed by other members...Mandy

Dear Mandy

Thanks very much for newsletters 13 and 14. It is always a pleasure to read them and this time they contain some great routines for Kit 2 to try out. But what I liked most of all was the letter from Paula Hanson in No. 14 in which she expressed the same feelings as I have about the Kit. I would like to reply to the letter of Colin Paterson in Newsletter 13 page 8. There are AMIGA 3D Kit 1 datafiles that have been successfully transferred to the PC and they are all available in the PC PD library, most of them both for 3D Kit 1 and Kit 2. Certainly for both versions are the Landscape Demo belonging to the article by J Hayes - which I found most useful - and the Flight Simulator Demo by Thomas Stufe. There are also a lot of borders and objects for both versions available. Not long ago I designed a piece of paper on which I wrote the X and Y coordinates of a 320 x 200 pixel screen. When you have a screendump of your border and want to put an animbrush somewhere on the screen or a circle or even text with the Setstring/Pstring command you can exactly calculate the place (x and y coordinates) where it should appear. I have enclosed a copy of this so you can use it yourself.

Mieke Van Der Poll - Holland - PC & Amiga

Thanks Mieke, the graph is perfect! So that everyone else can have one here is how Mieke did it: Draw a graph or use graph paper so that you have 32 squares across and 20 squares down. Reading from left to right we have the X coordinates which are marked in tens (10, 20, 30, 40 etc) so the first square reads 10 and the last square reads 320. Number your squares both top and bottom the same. The Y coordinates read from top to bottom and are marked in the same way starting with 10 and ending in 200 - do this down both the left and right hand side. There you are, you now have the perfect graph to show instantly the coordinates you need for your screen. If this is still unclear then send me a S.A.E. and I'll send you a photocopy.................. Mandy

Dear Mandy

It's me again, I just thought I would add a little PS to my last notes about successfully creating a complex border which looks the same in the 3D Kit as it does in the art package, due to palette differences. After my last communication I struggled for hours (the way you do!), before a little light-bulb turned on inside my frazzled mind. This might be well-known but I've found no reference to it. When you pull

up the palette altering command in Kit 2 and choose SELECT you are presented with all 256 colours to mess with. What I found out what that the very bottom row are fixed and as such cannot be altered. This is probably because they are used by the Kit for its menus/icons etc., I suspect. (Along with the invisible colour which CAN be changed). My problem lay in the fact that the bottom left and right colours are both as dark as you can get, and that Deluxe paint II uses these two for its menus! So I got black on black which is a might difficult to read! What, believe this, I had to do was mark my monitor screen with a felt-tip pen where the "V" slider button appears when you pull-up Deluxe Paint palette changing window and also where the last black colour appeared on it. I loaded in my Kit screen as detailed in my last letter, making the menus disappear, pulled up the palette menu, then moved the mouse cursor to where my colour mark was and clicked, then pushed the slider, from the other mark to the other end of the scale turning it from black to white!!! How complicated! There must be an easier way! Anyway, having got my menus back I could work away on the border until it was finished. Reloading it into the Kit hasn't produced any nasty effects, colour-wise, and my 256 colour borders now look the biz. By the way, what happened to the old Kit 1's spread command in the palette screen? This made colour graduations a sinch. Perhaps there is an undocumented key-command to achieve this?

Nigel Alefounder - Atari ST

Poor Nigel, just the thought of all that hassle you had to go through makes my deoxyribonucleic acid boil. I agree, there should be an easier way. Unfortunately the spread command was sacrificed to make room for other functions in Kit 2 which is a pity but I suppose we can't have everything - unless another member knows different! Has anyone else found an undocumented command for this?............ Mandy

Dear Mandy
I am going hairless with a problem I have with Kit 2 in Version V2.07. For example I create a cube and flatten it and stretch it rather like a platform and then copy it above. I raise the top platform as high into the air as I can but still find that I cannot stand on the lower platform. It's as if the top cube is forming some kind of barrier even though it is much too high to affect the climbability. It is just as if the top cube thinks it is reaching right down to the ground!

Peter Soames - PC

Yes, Peter, I have had quite a few complaints about this sort of thing happening - doorways you can't walk through etc. It doesn't happen when you have just created but sometimes only appears after you save and load in the datafile. Happily it has been confirmed that this bug has been found and corrected in the forthcoming version V2.09. I was hoping to have the new version ready so I could announce its arrival in this issue but there has been a slight hold-up with the playtest on one machine but that has now been sorted out and - fingers crossed - V2.09's arrival should be any day now .... Mandy

Dear Mandy
Can you please help me, I think I have found a bug in Kit 2. I create an object and turn on the attributes to TAN and WIR but when I saved my data and re-loaded it I found that the attributes of that object had all been turned off!

Sandra Mayo - Amiga

No Sandra, this isn't a bug. If you want the flags to keep the same value after a reset or a restart you should use the default attributes and remember to use the ALWays attribute also .... Mandy

Dear Mandy
After many weeks of waiting for my newsletters (shouldn't that be booklets?), I was starting to wonder what had happened to you. Then upon reading it I learned of the problems you had after the flooding and was very impressed that you managed to get the issues out at all. I would like to say that you have done very well, still up to the same high standards. I hope you sort everything out soon. I have great pleasure in enclosing my annual subscription form and payment and look forward to another great year of reading.

J. P. Messenbird - Spectrum

Dear Mandy
You are the first person I have had any contact with who actually suffered as a result of the flooding. I was a little bit worried when no magazines arrived for quite a while, then the company that supplied the 3D Kit, European Computer User refunded a years subscription saying that not enough takers resulted in them having to cancel production of their software review magazine. Just when I thought the last 3D Kit magazine may have arrived the two issues came and the explanation was clear. It must have been a real effort to get things back up and running again for production of the mag, and I can't imagine what it felt like to lose files on disks etc, as well as the computers and photocopier and virtually everything else downstairs. Concerning the paint that won't dry on the doorframe, this sounds as though the wall or floor may have at some time prior to the flood been treated for rising damp or dry-rot and the chemicals used (which are oil-based), been forced into the wood of the door frame. This would prevent the paint from drying although an original coat if already dry would not have been affected. If any treatment has been carried out then this is the likely cause.

Robert Coleman - PC

These two last letters are just a sample of the loads of letters of sympathy that I have received since you all heard about my being flooded out. Thank you all so much for your good wishes and expressions of support. Thankfully we are now back to normal now, at least in our house - we are one of the lucky ones - there are still 2,000 people homeless here and living in caravans outside their homes waiting for repair work to START! Actually, Robert was right about why the paint wouldn't dry and it is thanks to his idea that we have managed to sort it out. Our house is only three years old and has never been treated for dry rot or other things but once Robert put us on the right track we discovered that that particular doorframe (we had to have all new doorframes), was constructed of some different wood that had had this kind of treatment while in the builders yard - or somewhere. It has now been ripped out and a replacement fitted and, guess what? - the paint dried within a few hours! Thanks Robert, you are a real pal!.. Mandy

Before I close this section of the newsletter I just have to send a message to Tony Hartley who popped a little poser into a recent letter of his saying that he bet I couldn't name the seven dwarves in Snow White. I nearly gave up at one point, Tony, I had come up with six names but couldn't get the last until I told my daughter not to be bashful! Here goes: Doc, Grumpy, Sleepy, Dopey, Sneazy, Happy and Bashful. Ha! Better luck next time!........................... Mandy

THE USER GROUP GAME

Before we go on to create further puzzles we are first going to tidy up our game with a few messages. To do this we are going to use some variables for the first time. As we are also going to use other variables later on in our programming let's start by creating GENERAL CONDITION 2 and EDITING it as follows:

```
SETVAR (0,V50) - this is going to be our "Death" variable
SETVAR (0,V60) - this is going to be our "Start" variable
```

Now go to your DEATH area (Area 9) and EDIT our area condition to read as follows:

```
IF VAR=? (1,V50)
THEN PRINT ("YOU HAVE DIED!",1)
DELAY (100)
PRINT ("              ",1)
SETVAR (0,V50)
ENDGAME
ENDIF
```

Now we must change all our "endgame" conditions, such as the moving wall and pyramids, the pool of acid and the blade over the doorway in the other area to read:

```
IF COLLIDED? (or whatever the condition is)
THEN SETVAR (1,V50)
GOTO (1,9)
ENDIF
```

As you can see, what we have done is to make sure that everytime our player dies we set variable 50 to hold the value of 1. When we send our player to the "death" area it checks for that, prints the appropriate message, gives the player time to read it, prints a clear line over the message, resets variable 50 to hold 0 once more and sends the player to the start of the game. You can, of course, print whatever message suits your fancy, it doesn't have to be the same as mine.

Now to make the start of our game a little more professional looking we should CREATE AREA. This will be area 10 and, via the attributes, change the name, to START AREA. Colour the sky and the ground of the area stark black. Now via the DEFAULTS under the General Menu we should change the start area to number 10 instead of 1. Create an Area Condition and enter the following:

```
IF VAR=? (1,60)
THEN PRINT ("WELCOME TO THE CLUBGAME",1)
PRINT ("\N CAN YOU FIND THE TREASURE",1)
PRINT ("\N AND MAKE YOUR ESCAPE?",1)
PRINT ("\N\N\N\N GOOD LUCK!",1)
DELAY (500)
PRINT ("                    ",1)
PRINT ("\N                    ",1)
PRINT ("\N\N\N\N           ",1)
SETVAR (0,V60)
GOTO (1,1)
ENDIF
```

This is basically exactly the same procedure as the previous messages. You can, of course, expand your message and word it however you like - mine is a bit basic anyway - only thing to remember is if you want to move down to another line use the \N for each line down that you want to print. Of course we can only see our messages when in TEST MODE and as the game starts off with a message it is a bit tricky getting around to seeing it. It can be done, you just have to be very quick and whiz from the RESET to the TEST button before the program really sets itself up. Or from RESET to F1 whichever you prefer.

Now all we need is a CONGRATULATIONS message when the player gets (activates) the gold ingots in the cell. Go to the Cell (area 8) and enter the following condition for the ingots object/s:

```
IF ACTIVATED?
THEN PRINT ("CONGRATULATIONS!",1)
DELAY (100)
PRINT ("              ",1)
ENDGAME
ENDIF
```

You will notice that in this case we do not send our player to the "Death" area but just print the message on screen and perform a reset back to the start - in case they want to play again. To make sure it all works properly we have to CREATE INSTRUMENT. Set this to a TEXT type instrument by clicking in the box and set the size you want by selecting the drag-box type option. This is easier for beginners as you can see what you are doing on screen.

Now that we have given our game a more professional look it is about time that we gave it a really good STARTING AREA instead of the one we already have. So create area and we have area 11. In this area we are going to create something a bit more spectacular and also make good use of some system variables.

We are going to filch the idea of the drawbridge from Incentive's Castle Master game - I'm sure they won't mind this once - but we are going to add a twist to the puzzle. This is what we aim to create. The area has a large castle in the centre, surrounded by a moat which will kill you if you enter it. There is a jetty sticking out towards a raised drawbridge. At the other side of the moat you can see a small grassy area sticking up out of the moat with a stone on top of it. Obviously you can't reach it from where you are. Nearby is a boat moored to a post. Basically the area we need to create should look something like this:

```
OBJECTS
Moat 008
Boat 009 Moveable
Hillock 010
Stone 011
Button 015
Doorway 001
Bridge/up 014 Vis
Bridge/dn 013 Invis
Post 016
Switch on boat 017
```
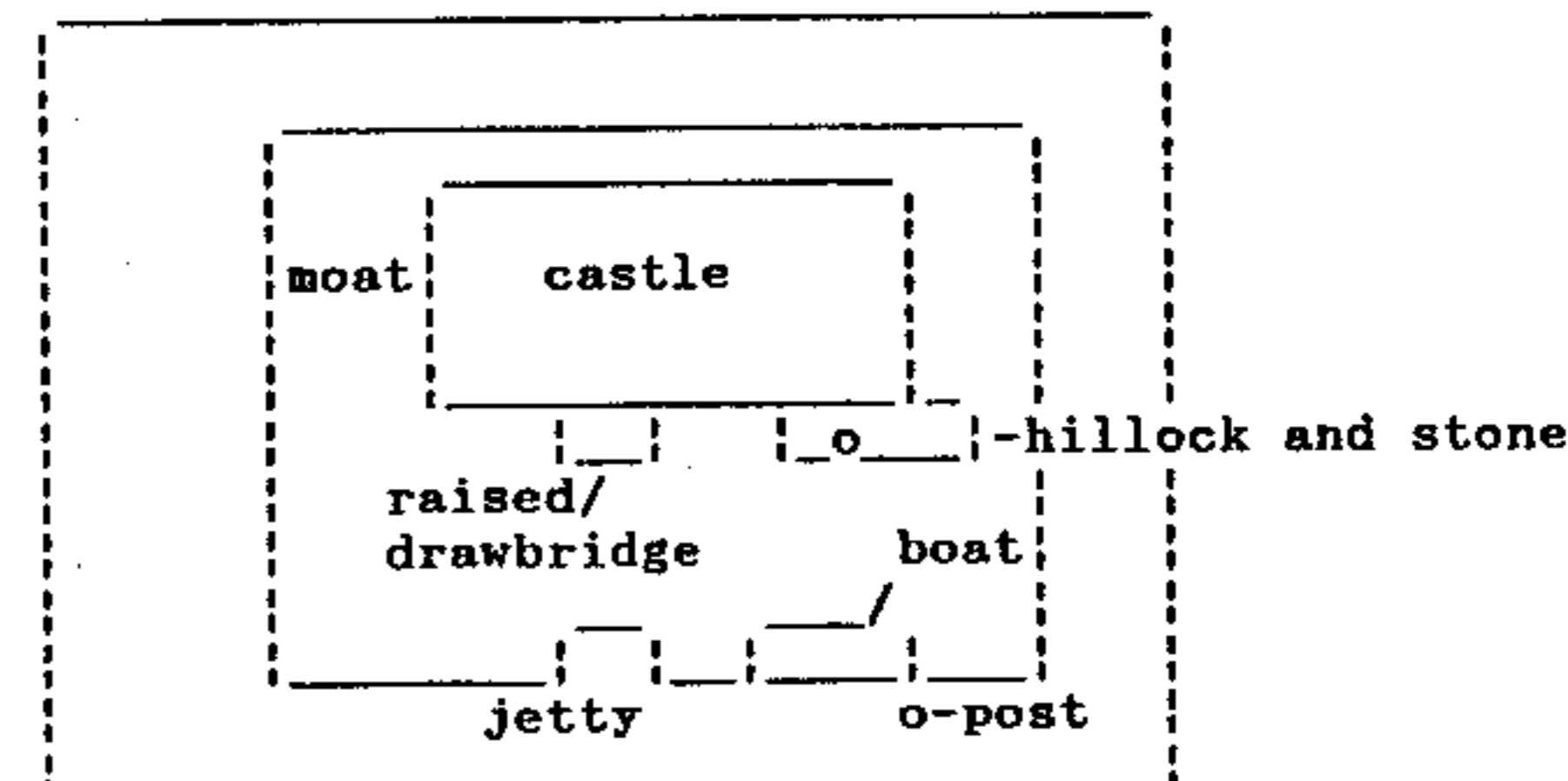
The puzzles we need to create are that when standing on the end of the jetty the player is just out of reach of the stone on the hillock. He can't shoot the drawbridge button because he/she doesn't have the stone to throw. He/she has to sail across the moat in the boat to retrieve the stone and then return, throw it and lower the drawbridge to enter the game proper. Right, lets get down to it:

Place a button beside the raised drawbridge of the castle, (you know how to do this as we did it in a previous area), and enter the following condition for the BUTTON - in my case it was cube (15).

```
IF SHOT?              } if you shoot the button
AND VIS? (11)         } but the stone is still there
THEN SOUND (2)        } just make a sound
ELSE IF INVIS? (11)   } but if they have got the stone
INVIS (14)            } make the raised drawbridge vanish
VIS (13)              } and replace with lowered drawbridge
SOUND (2)             } make a sound
ENDIF
ENDIF
```

The condition for the STONE:

```
IF ACTIVATED?      } if activated (taken)
THEN INVIS (11)    } make stone invisible
ENDIF
```

Condition for the entry behind drawbridge. Doorway 7:

```
IF COLLIDED?
THEN GOTO (1,1)
ENDIF
```

Condition for the Moat - to kill players attempting to swim:

```
IF COLLIDED?      } if they fall in
THEN SOUND (5)    } make splash sound
SETVAR (1,V50)    } set the death variable
GOTO (1,9)        } go to Death area
ENDIF
```

Now for the animation of our boat. What we want to achieve is that when the player enters the boat and presses the switch, not only will the boat sail across to the hillock but our player will sail across with it - and back again when they have the stone and use the switch again. To do this we make use of a system variable which controls where the player is at any given time. Now one problem with this type of animation is that it doesn't stop - well the boat will stop when it touches the hillock but the player would sail on to the end of the area so we are going to create two "buffers" to stop the animation. One is the hillock and the other is the post that the boat is moored to. It is simply done. Just enter the following condition for both the HILLOCK and the POST:

```
IF COLLIDED?
THEN STOPANIM (1)
ENDIF
```

Don't be afraid to use what I refer to as "animation stoppers" wherever you need them within your games. The programmers used them in the undersea area of the Kitgame to stop the shark animation from going too far. They used an invisible cube for this which is just as good. We use objects that are already here for ease and to save memory.

Now enter the following condition for the SWITCH on the boat:

```
IF ACTIVATED?
THEN STARTANIM (1)
ENDIF
```

Next CREATE ANIMATION and EDIT ANIMATION and enter the following:

```
INCLUDE (9)        )boat object
INCLUDE (17)       )switch object
START
LOOP (40)          )ajust for length of moat.
MOVE (0,0,20)      )move boat across moat
ADDVAR (20,V2)     )move player across moat
AGAIN              )keep going until loop ends
STOPANIM (1)
INCLUDE (9)        )boat object
INCLUDE (17)       )switch object
START
LOOP (40)          )ajust for length of moat
MOVE (0,0,-20)     )move boat back again
SUBVAR (20,V2)     )move player back again
AGAIN              )keep going until loop ends
STOPANIM (1)
RESTART
```

We use Variable 2 because that is the System Variable that holds the players Z co-ordinate position. You can of course experiment with the other viewpoint system variables - these can be found listed on page 61 of the Kit 1 manual.

If you are using Kit 2 to program the Usr Group Game then you can, of course enhance the game a lot with all the extras such as making the moat ripple using the Fade commands.

Now is the time to check that everything works properly and to add those little finishing touches to the game. Now that we have our basic structure and most of the programming done we can afford to spend a bit more memory on creating furnishings for our different areas so that it looks good.

If anyone has any ideas for enhancing the game further then please don't hesitate to get in touch and we will see what we can do. I confess that I am running out of original ideas so any help you can give would be appreciated.

****

LATEST ADDITIONS TO THE PD LIST

MAGICAL MAZE by Mieke Van Der Poll
FOR AMIGA (1 MEG REQUIRED) SUITABLE FOR KIT 2.

STAND ALONE OR DATAFILE. This game is nicely presented with lovely loading screens and music. Messages appear during loading to tell you what is happening and there are some nice animations during loading too. The funfair has arrived in town and one ride has you intrigued. The magical train is ready for off and you are going to be on it. This is a maze filled with all sorts of baddies, ghouls, ghosts and things that go bump in the night. The areas, as with all mazes, are devoid of furnishings but there are loads of different - and extremely useful objects scattered about which you will definitely need! The puzzles are good too. Kit 2's extra features have been beautifully utilised - especially with the ghastly faces that guard certain doors. There is excellent use of messages throughout the game so you always know what is happening which is very user friendly as it is difficult to know exactly how you overcame a particular baddie if he just vanishes with no explanation. I was lucky (I think) because my energy bar, when depleted didn't actually kill me off (a buggette methinks!), but it did turn off the sound effects which I had been enjoying.

EXAMPLES by Juergen Hunke
UTILITY IN DATAFILE AND BORDERS FORMAT - AMIGA or ATARI ST KIT 1.

This is a disk of various objects of all kinds that you can view and select for use in your own datafiles if you wish. I particularly like the Temple, the Glider and the fish. There are also some magnificent archways and pillars which are well worth viewing. The titles of the objects are in German so you should view them to see what they are and, if you wish, re-save them with English titles for future use.

FAST-CITY by Tony Hartley
FOR ATARI ST SUITABLE FOR KIT 2 DATAFILE AND BORDER FORMAT

Oh boy are you all in for a treat! On the same lines as Fast Town, Fast City allows you to build up your very own city with ready-made objects and roads etc all stored on disk ready for you to select. They are all neatly nestled within sub-directories so they are easy to find. All the different types of road junctions you could need, all sorts of buildings - churches, hospitals, gardens, flats, mills, offices, shops etc, there is even a town hall and firestation among many others. There are features such as a football pitch and stadium and hot air balloons, phone booths, traffic lights - the list is endless. Thoughtfully provided on the disk are a selection of sound samples, borders and ready constructed areas so that you can see what suggested layouts will look like. There are read-me documents on the disk which also tell of some other exciting disks in the pipeline.

SOUND UTILITIES by Juergen Hunke
ATARI ST SUITABLE FOR BOTH KIT 1 AND KIT 2.

Two useful programs to use with your 3D Kit sample sounds. The first loads in a sample, cuts the last byte off, creates a new RAW header and re-writes it. This is extremely useful for samples with odd file lengths. The second program was written because Juergen was having trouble merging samples correctly to a sample bank from Deluxe Sound Audio Digitiser (via Null Modem) to Atari ST. It will load your samples, change zero bytes into "1" (for better sound quality) and link them to the sample bank. Full instructions are enclosed on Read-me files on the disk.

FOR FULL PD LIST WITH REVIEWS OF ALL DISKS SEE CONTENTS PAGE

PUBLIC DOMAIN LIBRARY

All public domain disks cost £1.50 each for U.K. members. Overseas members please add £1 per disk to cover extra postage and packing. For Atari ST, Amiga and Commodore PD please send cheques etc payable to Mandy Rodrigues at the usual User Group address. For public domain for the PC owners please see the add below as this is run by Mark Rose. For details of how to send for full PD lists see page 3.

THE 3D KIT USER GROUP

3D PD

SUMMER 1993

For the latest catalogue, send an A5 SSAE to: 3DPCPD, 39 BALFOUR COURT, STATION ROAD, HARPENDEN, HERTS. AL5 4XT.

MUSIC, ANIMATION, HYPERTEXT, DEMOS RAYTRACING, MORPHING, 3D KIT, 3D MODELLING, SOUND.
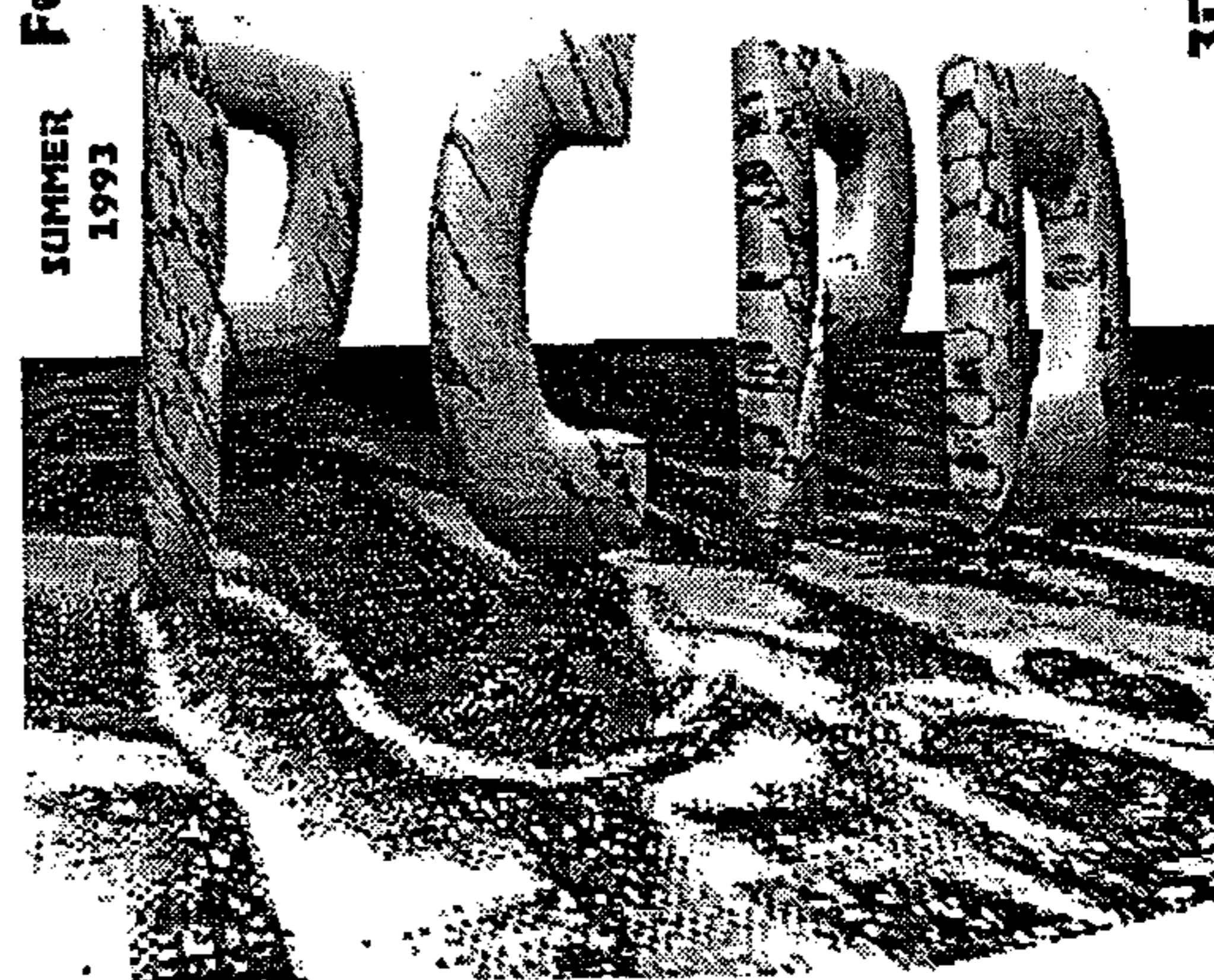
BEGINNER'S SECTION

THE WAIT COMMAND AND HOW TO USE IT

By *Mieke Van Der Poll* - AMIGA & PC - KIT 2 & KIT 2

We recently explored the possibilities of using the DELAY command. You will find that the WAIT command is even more useful when used in combination with sound effects. Imagine you are walking in a forest, hearing birds singing or chirping. Creating this effect is possible with the above command. You have to do three things:

1. Create a forest.
2. Make two suitable sound effects.
3. Choose a variable above 29.

Then make the following condition on the floor (cuboid 001):

```
IF COLLIDED?
THEN
SOUND (1)
FOR (V30,0,50)      } 50 is the delay value
WAIT
NEXT
SOUND (2)
ENDIF
```

You are now able to keep walking around and every now and then you hear birds singing. You can decrease or increase the delay value to try to obtain the best possible effect. Imagine walking in a jungle, surrounded by a lot of sounds...

It is possible to use more sounds in combination with DELAY or WAIT commands - the only restriction is that DELAY stops you moving around.

HELP FOR SINGLE DRIVE AMIGA USERS

LOADING OBJECTS, DATAFILES ETC ON THE AMIGA WHEN YOU ONLY HAVE ONE DRIVE AND CREATING MORE MEMORY TO LOAD LARGER SOUND BANKS BY USING JUST ONE DRIVE.

By *Mieke Van Der Poll* - KIT 2

There have been quite a few problems for Amiga owners of Kit 2 who wish to load clipart etc when they have only one drive. Here is what to do:

When you load in the 3DEditor and want to load a datafile from another disk you have to put your data disk in drive df0:, then choose DEVICES and DF0: again. You then have to click on your datafile to load it in. Suddenly an AmigaDos requester pops up asking you to insert Volume 3D program in any drive. You swap disks again and then you have to put your datadisk in again. Then the Editor loads your datafile but it seems you have lost the Edit screen. But this is not so. All you have to do is click with the LEFT mouse button on the AmigaDos window, hold down the mouse button and drag it a little bit down the screen; then do the same with the Workbench screen but drag that one down to the bottom of the screen. The Edit screen appears again and

if you click on it it is usable again. When you have to locate the border directory or load in a sound bank this will happen again. So be sure the Workbench screen and the AmigaDos window are right under the Status Bar/Freescape controls so that you can see a bit of it and still use all the Freescape buttons. Now when the DOS message comes up to prompt you to swap disks you can do so while you are also staying in the Edit Screen and the multitasking Amiga will do its job properly and you are able to load in a sound bank file of approximately 130,000 bytes.

1 Meg Amiga owners who have two drives (or more) can, in order to load in a larger sound bank, switch off one drive (or more). But then you also have to cope with the above mentioned method and your maximum is also approximately 130,000 bytes.

INVISIBILITY FOR BEGINNERS

By *Tony Hartley* - ATARI STE - KIT 1 AND KIT 2

The ability to make objects invisible is just one of the great features of the 3D Kits. There are three types of invisibility all of which have their uses. The first being a facet/face of an object with a value of nine. This is similar to the colour invisible but the facet does actually have a colour, you just cannot see it. If you were to edit a USER FADE (one of the USER FADE masks in Kit 2 between ten to fifteen), then the squares that are not coloured black will be invisible. This will enable you to produce lots of special effects like a cross hatch pattern on a window or a brickwork effect on a wall or thick lines for fencing etc.

The second type of invisibility is the colour invisible. This is the first colour in the palette marked with a letter "I". This form of invisible is ideal for colouring all the sides of objects that will never be seen the invisible colour to speed up movement in your areas. The object facet coloured invisible can still be interacted and collided with, shot etc. It is extremely useful for closing gaps in walls (using a cube coloured the invisible colour), and one of the sneakiest ways is to make a dividing wall with an opening/gap in the middle of it in the centre of an area. Then block the gap up with the cube coloured invisible and then make the dividing wall TANGIBLE so that the gameplayer tries to enter the opening/gap (blocked by the coloured invisible cube), but collides. So then the gameplayer has to work out how to walk through the TANGIBLE dividing wall. Kit 1 users need not feel left out by not having the TANGIBLE attributes - all they have to do is make the dividing wall have a condition that makes them move to the other side of the wall if they collide with it. Simply give the wall the following condition:

```
IF COLLIDED?      } If you walk into the wall
GOTO (2,1)        } Move to the entrance at the other side
```

Just make a new entrance on the other side of the wall and the wall condition will transfer you to it if you touch it.

Another good use of the invisible colour is for making floor triggers. A rectangle placed on the floor and coloured invisible could be used to trigger an event or animation etc. No one would actually know it was there until they stepped on it and something happened e.g.:

Condition for invisible rectangle:

```
IF COLLIDED?        } If you stepped on the rectangle.
SOUND (6)           } Make a sound.
STARTANIM (1)       } Start your animation working.
```

The third type of invisibility is an object that has had its ATTRIBUTES set to INVIS. This sets all the objects facets to invisible which also makes it invisible to collisions too. This comes in handy if you pick up an object. You could make the object invisible if the game player has collected it by giving it the following condition:

```
IF ACTIVATED        } If you have taken the object.
THEN INVIS (5)      } Make object 5 invisible.
SOUND (4)           } Make a sound to verify object taken.
ENDIF               } This ends the IF statement.
```

Another use is to speed up an area with two floors in it by making all the objects on the floor that you are not on invisible. In fact you can get away with not using a lot of variables simply by checking the invisible state of an object. Invisibility of an object is a good way of triggering or checking something else e.g. If you collect a key and it is made invisible then you can open the chest. If you do not collect the key it is still visible and waiting to be found then the chest cannot be opened until the key is taken. This is used in the User Group Game quite a bit. This is how it's done:

This is the condition for the key object:

```
IF ACTIVATED?       } If you take the key.
THEN INVIS (9)      } Make the key invisible.
ENDIF
```

This is the condition for the chest:

```
IF ACTIVATED?               } If chest is touched.
AND INVIS (3)               } And key has been taken.
THEN INVIS (7)              } Make closed chest lid invisible.
VIS (8)                     } Make open chest lid visible.
ELSE                        } If you haven't taken the key.
PRINT "LOOK FOR A KEY",1)   } This prints text to instrument 1.
ENDIF
```

Of course you will have to make a chest with two lids (one visible and one invisible) the visible one being in this case, object 7. You will also have to make a text instrument in this example but if that is still a little bit difficult for some beginners then simply change the PRINT line/command to SOUND (5).

So you can see that the three different forms of invisibility is very useful in lots of ways.

MATCHING FREESCAPE & BORDER COLOURS

By *Mieke Van Der Poll* - AMIGA KIT 2
ALSO SUITABLE FOR ATARI ST

You probably noticed the rather unusual colours of the Amiga 3D Kit 2 edit screen and the palette when loading 3D Edit. And you probably changed them into the old 3D Kit colours as given in Newsletter 11 by Peter Ward and saved that area for further use.

But did you notice that unless you use the same palette in your paint program, the colours will greatly differ from the ones of your border, when it is present in Test Mode? When you use a border of the provided ones you are certainly going to have this problem.

I found a way to solve this. It is the same solution as the one for 3D Kit 1 on the PC, which also had this problem. Only on the Amiga you use it the other way around.

Under the Area Menu there is a sub-menu entitled Area Colours. You can use this for changing the red, green and blue values of the first sixteen Freescape colours. In order to adjust them to the border colours you have to load the border into your paint/art package. I am supposing that all paint programs have a colour mixer and that is the one you have to use to see the red, green and blue colour values (numbers) of each of the sixteen border colours. Write them down.

Load 3D Edit and add the border and select the colours from the border. Change the first sixteen colours of the Freescape palette (except the first one because that is the invisible colour), through the AREA COLOURS sub menu into the values found in the paint program and save the area with the name of your border. Now when you want to create another area with the same border you only have to load that area in and the right colour palette is available.

****

PEN-PAL SECTION

Following one or two pleas in the letters section in recent months for members to get together by letter or telephone to share interests, whether 3D, Computers or whatever, two members have written in to get the ball rolling (so to speak), and would love to hear from other members so why not drop them a line. They are:

TONY HARTLEY
19 Kipling Close, Lockwood, Huddersfield,
Yorkshire, HD4 5HA
Tel: 0484 650891

Tony has an Atari STE and is quite an artist - judging by the fantastic screen shots he sends me. He also has a great sense of humour!

JOHN CLARKE
2 Frant Close, Penge, London, SE20 8HS
Tel: 081 659 1087

John has a PC and is extremely interested in programming and all things computer related.

8 BIT SECTION

BOAT SIMULATOR
By *Daniel Prentis* - SPECTRUM

This is a simple routine which uses Var. 122 to simulate the rocking of a boat on the sea. Make sure you are in WALK mode and type the following routine into GENERAL CONDITION 3 four times, each time with the given values for A, B and C:

```
CMPV   (A)   122
IFLT
AND
CMPV   (B)   122
IFGT
THEN
SETV   (C)   120
ENDIF
```

		1	2	3	4
A	64	128	192	255
B	0	63	127	191
C	1	0	71	0

One thing I noticed was that the IFLT command must always precede the IFGT command for the condition to work and not the other way round. To ensure that the whole thing runs smoothly and to prevent the sea becoming calm everytime you leave the tiller you must type the following condition into GENERAL CONDITION 2:

```
CMPV      255   121
IFEQ
THEN
REDRAW
ENDIF
```

This routine will keep everything in the Kit running at the same speed whether you are standing still or not. If no keys are pressed it forces a redraw.

TURNING ON THE LIGHT

Following the User Group game section on making the Solar either light or dark depending on whether you had picked up the torch or not, and how to program this, the problem caused quite a bit of excitement with 8 Bit members, especially as the way I had done it was rather long-winded! I received loads of different routines showing how this could be done. I cannot publish them all but here are two of the best to show you how you could go about tackling this problem:

THE DARKENED ROOM
By *John Elliott* - SPECTRUM +3

This is a way of implementing the "darkened room" in the User Group Game without having to resort to a tedious recreation of an already-designed room. If you have an Amstrad CPC or Commodore, don't use colour palette no.2 for any objects except those which you want to be visible in the dark (eg. a way out). If you have a Spectrum, the way out will be invisible. Edit the LOCAL CONDITIONS allowing movement (except to the Solar), inserting immediately before the GOTO line the commands:

```
COLOUR 0   (b)      - b, c, d, e are the colour values (see below)
COLOUR 1   (c)        for the room you are moving TO. This seems to
COLOUR 2   (d)        be the way the COLOUR command works.
COLOUR 3   (e)
```

For rooms moving to the Solar, use the following:

```
IFVIS   11   2
AND
IFHIT   (r)
THEN
COLOUR   0   0
COLOUR   1   0      - dark colour set
COLOUR   2   (a)
COLOUR   3   0
GOTO   (solar)
ENDIF
IFHIT   (r)
THEN
COLOUR   0   (b)
COLOUR   1   (c)    - light colour set
COLOUR   2   (d)
COLOUR   3   (e)
ENDIF
```

The values of (a) to (e) are given in the following table:

AMSTRAD CPC	COMMODORE 64	SPECTRUM
(a) 1	11 (or other grey)	(a) 0
(b) to (e) are your normal colours for		(b) normal INK
the room when it is light.		(c) normal PAPER
		(d) normal BRIGHT
		(e) normal FLASH

ERRATA: I am sorry to say that in my article in issue 11 there are some mistakes that may cause problems (some are my mistakes, some are misprints). The address of shades in a completed game is 65147, and the size of the first sensor in the proximity detector should be 1000.000.1. The parameters to a FCL command are at (1Y+1) etc, and the address at B695 is 1 for a successful IFEQ. I can also add the following information: The byte at B694 is the state of the previous IF - AND or IF - OR (0=failure 1=success). To make a sound from machine code, CALL BB42 with A=sound number.

SWITCHING ON A LIGHT
By *Ted Horn* - SPECTRUM

Having succeeded initially in presenting the appearance of the light switching, I realised that this was not much use if conditions changed from the light room to the dark room (i.e. open a door, switch on a light, the door must remain open and vis versa). I think the FCL conditions I have worked out should stand to work if applied to other movements other than a door.

In my scenario I start in a darkened room, a closed door is just visible, shooting at the appropriate switch (pyramid on the wall) will either put the light on or open the door. (The left switch will open the door and the right one turn on and off the light). Walking up to the open door and hitting it will allow me to pass through the door into area 3. Walking forwards to the room centre and doing a U turn, I should see either a white doorway - if I came from a lit room (room 1), or a black doorway - if the light was off (room 2). Returning through the doorway returns me through a black door to the darkened room or if the doorway is white - to the lightened room. I hope this makes sense to you - it's working for me! On the next page you will see the diagrams of how I designed my areas:

```
    3    Rectangle covered by        3    Rectangle covered by
   =======   cube 2 initially vis     =======   cube 4 init.
  /:_____/:          open door cube 4  //:_____/  vis. open door
 /: ___ : ___ :/|    initially invis.  /|: ___ : ___ :/|cube 2 invis.
 : : 2 : : 4 : :     initially.        : : 4 : : 2 : : initially.
 : :....:.: :/       <--- DOOR DETAIL -->  : :....:.: :/
 :__:___:__:/                          :__:___:__:/
```

```
1E :\       \ ....    light    /:     1E :\       \ ....    light    /:
2C : \10x/////   :   x5/switches/:     2C : \ 5x/////   :   x9/switches/:
3R :  \ /////_____  \          :     3R :  \ /////_____  \          :
4C :   E8          :           :x9:    4R :   E8          :           :x7:
5P :   :           :           :x7:    5P :   :           :           :x10:
6R :   :_____E1_____:   :    6R :   :_____E1_____:   :
7P :  /                         \  :    7P :  /                         \  :
8E : /                           \ :    8E : /                           \ :
9P :/_____\:    9P :/_____\:
10P        AREA 1 (LIGHT ON)            10P        AREA 2 (LIGHT OFF)
```
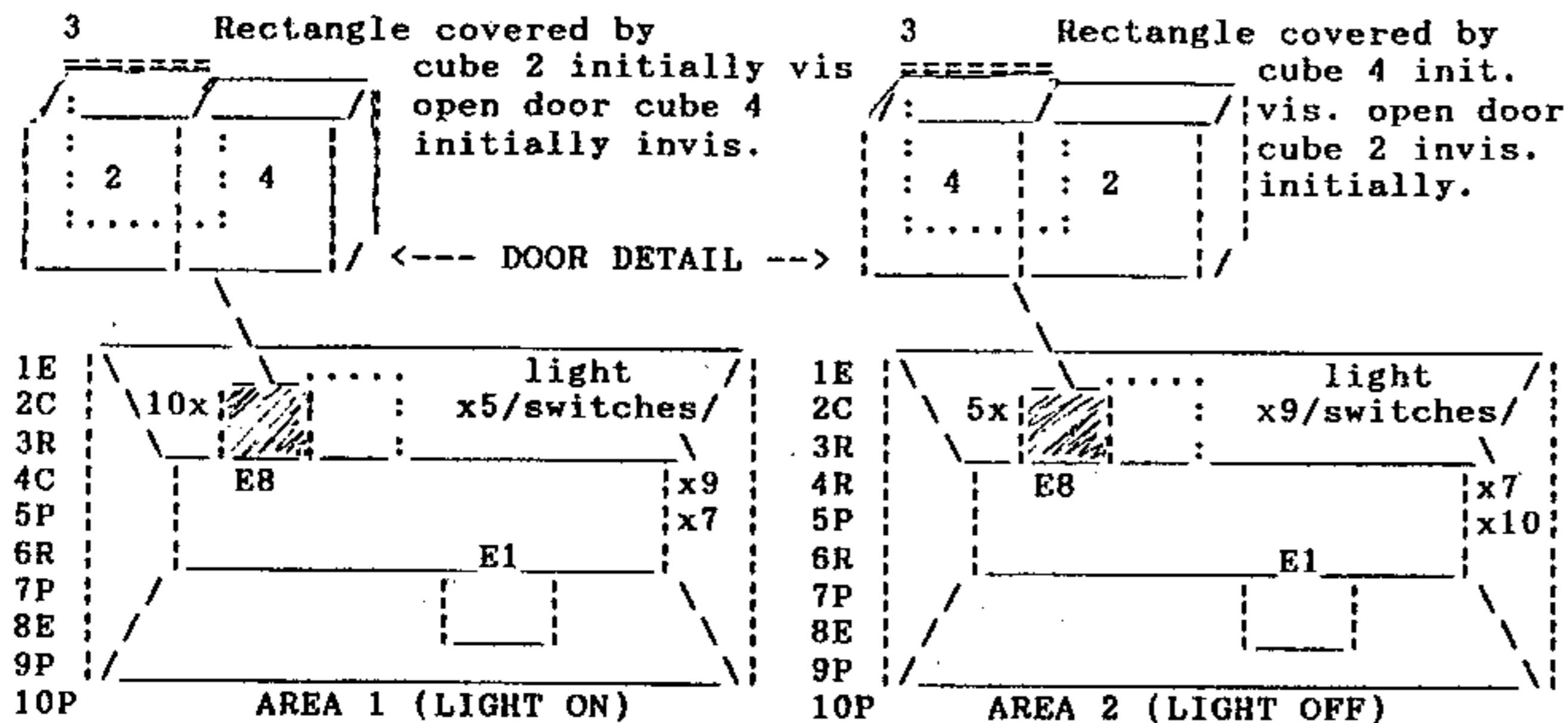
E = entrance   R = rectangle   C = cube   P = pyramid

```
 _____   _____
|               | |       |
|    CUBE       | | RECT. |RECT.|      Cube 12 (door) covers Rectangle
|    12         | |  13   |_14__|      13 (white) which in turn covers
|   (door)      | |       |  \         Rectangle 14 (black).  Each sized
|               | |       |   \        8 x 8 x 1.
|_____| |_____|    \
                    \            \
                     \ :_____  \
                      \:        \  \ /
                       :         : /
                       :         :/
                       :_____:
                      /           \
                     /     E1      \
                    /_____\
                   /                 \
                  / _____  \
                 /:                  :\
                /:                    :\
               //_____\\
                       AREA 3
```
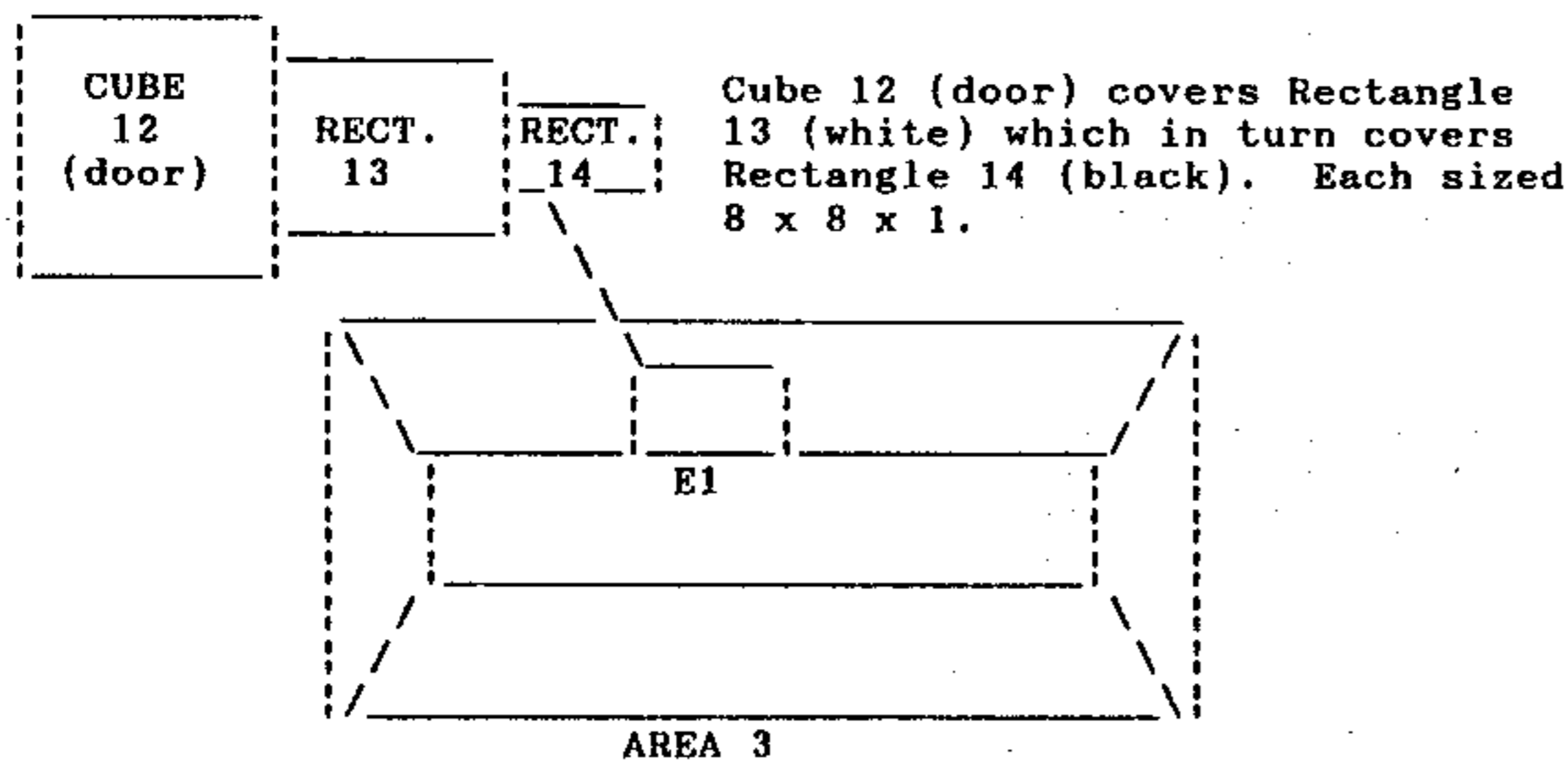
A point to note: where conditions refer to objects in areas other than
the one on the screen, object numbers must be unique to their own
area, otherwise strange things happen. I used objects 12, 13 and 14
for the door in area 3 - numbers which did not appear in rooms 1 or 2.

LOCAL CONDITIONS

AREA 1			AREA 2			AREA 3		
1 - IFACTIVE 009			1 - IF ACTIVE 007			1 - IFHIT	013	
THEN			THEN			THEN		
GOTO	002	002	GOTO	002	001	GOTO	008	001
ENDIF			ENDIF			ENDIF		
2 - IFSHOT	010		2 - IFSHOT			2 - IFHIT	014	
THEN			THEN			THEN		
TOGVIS	002		TOGVIS	002		GOTO	008	002
TOGVIS	002	002	TOGVIS	002	001	ENDIF		
TOGVIS	004		TOGVIS	004				
TOGVIS	004	002	TOGVIS	004	002			
TOGVIS	012	003	TOGVIS	012	003			
ENDIF			ENDIF			(continued overleaf)		

AREA 1			AREA 2			COLOURS			
							AREA 1	AREA 2	AREA 3
3 - IFSHOT	005		3 - IFSHOT	009		INK	6	1	7
THEN			THEN			PAPER	3	0	0
GOTO	002	002	GOTO	002	001	BRIGHT	1	0	0
ENDIF			ENDIF			BORDER	0	0	0
4 - IFHIT	003		4 - IFHIT	003					
THEN			THEN						
INVIS	014	003	INVIS	013	003				
VIS	013	003	VIS	014	003				
GOTO	001	003	GOTO	001	003				
ENDIF			ENDIF						

SCREENDUMPS ON CPC
By *Yannick Gour* - AMSTRAD CPC

Here is a program to enable you to save screendumps of your 3D Kit
screens. It is in machine language. Here is what to do:

1. Type and save on your current 3D CK Disk the three programs:
3DCKDUMP.BAS, KIT128.DMP and KITCON.DMP.
2. RUN "3DCKDUMP" and choose the Editor: 128 KO version or 64 KO
version. The programs are normally loaded and started.
3. LOAD the Datafile with the area you want to print.
4. Select this area (or not if you can access it by TESTing).
5. Select the TEST option.
6. Now you just have to choose the right point of view. You can do
everything as normally. When you see exactly what you want, press the
ESC key as if you wanted to return to the Editor; but here the screen
becomes completely blue.
7. Insert a formatted Disk into the drive (or a tape into datacorder),
then press any key: the last screen of the TEST mode is saved under
the name "3DCKDUMP.SCR". Then there is a complete reset of the CPC.
You can now edit this screen with any graphic utility and print it.

__3DCKDUMP.BAS__
```
10 MODE 1
20 PRINT "↓↓↓      3D CONSTRUCTION KIT"
30 PRINT "      + SAVE SCREEN AFTER TEST"
40 PRINT "↓↓↓   1   128K EDITOR"
50 PRINT "↓     2   64K CONDITIOIN EDITOR"
60 PRINT "↓↓↓   ENTER SELECTION (1-2)";A$
70 IF A$="1" THEN GOSUB 100:RUN"KIT128.DMP
80 IF A$="2" THEN GOSUB 100;RUN"KITCOND.DMP
90 RUN
100 FOR 1=&80 TO &E1; READ a$: POKE i,VAL("&"+a$): NEXT: POKE i,
PEEK(&BD17): POKE i+1, PEEK (&BD18) AND 63: OPENOUT"X": MEMORY HIMEM-
1: CLOSEOUT: RETURN
110 DATA F3,21,38,00,36,C9,21,00,C0,11,00,40,42,4B,ED,B0,21,CE,00,0E,
20,D5,ED,B0,C9,CD,06,BB,21,FF,AB,11,40,00,0E,07,CD,CE,BC,21,C2,00,11,
00,C0,06,0C,CD,8C,BC
120 DATA 21,00,40,54,5D,43,48,3E,02,CD,98,BC,CD,8F,BC,C7,33,44,43,4B,
44,55,4D,50,2E,53,43,52,01,89,7F,ED,49,21,00,00,54,5D,01,3F,00,ED,B0,
21,99,00,0D,CD
```

__KIT128.DMP__
```
10 MEMORY&1FF:LOAD"!Kit128":POKE &423,&80: POKE &424,0: CLS: CALL &200
```

__KITCOND.DMP__
```
10 MEMORY &1FF:LOAD"!Kitcond":POKE &3B0,&80:POKE &3B1,0:CLS:CALL &200
```
****

SAVE AND LOAD GAME POSITIONS
By *Robin Ball* - ATARI ST - KIT 2

The fact that there is no provision made for the saving and loading of game positions in the Kit version 2 means that it can't really be used for writing large and complex adventures. Who would want to play the game from the beginning every time? Not I and not, I suspect many other 3D adventure fanatics. So what can we do about it? Well, the only solution is to write our own routine from scratch. This is not really very easy and it does make for some very long conditions. I would recommend that you start off by writing the basic conditions to a word processor as they are far faster and offer block moving and pasting (particularly useful for hundreds of FPUT (V*) commands). Remember to save the files as plain ASCII and they can be loaded easily into the Kit.

Here is the basic theory behind the routine. What we want to do is store in a file the player's position in the 3D world, and also all of the variables used in the game. We also want to store the condition of each of the objects involved (whether they are invisible or visible. As you might expect, the saving part is the easiest. It is when you want to reload that things get complicated. Here is a basic routine for saving the player's position and also soring the condition of a couple of objects. Write it as a procedure and create a control to put over the save icon. Change the action button for that control until it says Procedure and the box next to it says 0. Change that number to whatever the number of this procedure is and it is called whenever that control is clicked on. Don't forget to leave out the REM statements as they are only there to make the routine clearer on the page. You must also save a file to your game disk called "save.pos". This can be anything, it is only there because the Kit can't create new files, only open existing ones.

```
IF FEXISTS? ("save.pos")     \ rem - Checks for the existance of the
THEN                         /       save file. This need only be a
SETVAR (V0,V50)              \       text file to start with.
SETVAR (V1,V51)              \
SERVAR (V2,V52)              \rem- These set six unused variables
SETVAR (V3,V53)              /     to the same values as the
SETVAR (V4,V54)             /      players viewpoint. Footnote 1.
SETVAR (V5,V55)            /
FOPEN ("save.pos",1)        - rem- Opens the save file ready to
FPUT (V49)                  \              write to it.
FPUT (V50)                  \
FPUT (V51)                  \ rem-Writes to the new viewpoint
FPUT (V52)                  -     Variables to the file. V49 is
FPUT (V53)                  /     a new variable holding the
FPUT (V54)                  /     current area. Footnote 2.
FPUT (V55)                  /
FPUT (V100)                 \ rem - These are object flags. They're
FPUT (V101)                 /       set to one if the object's taken.
FCLOSE                      -rem - Closes the file.
ELSE
PSTRING ("can't find file",0,160) -rem- Do something else if the file
ENDIF                                    doesn't exist.
```

FOOTNOTE 1: These are needed because we can't write to the variable holding the current area number and must use the GOTO command to go to the area. This would move the player to the entrance and thus change the contents of V0-5. If you're confused, just take my word for it.

FOOTNOTE 2: Writing V9 to the file then reading it back and going to that area doesn't work. Instead we must update V49 (or whatever) whenever the player is sent to a new area. For example add SETVAR (area number, V49) to the doorway condition. It's inconvenient I know but it's the only way as far as I can tell.

I hope that that was understandable. It does work very well. The exact routine can probably be used in any game. You must however, set a variable for every important action and for every object collected. Add FPUTs for every one of those important actions. (I have only used two in my routine - V100 and V101. They are each for an object). If you have 50 objects and counters you'll have to add a FPUT for each one. This is why a copy/pasting facility becomes useful. Can you imagine typing FPUT (V30), FPUT (V31) etc. fifty times?

Before you ask, I don't think it is possible to use a For Next loop as the Kit handles them in a strange way. If you do manage it then I'd very much like to hear about it. I hope you are still with me because the next bit is harder. This is how to reload the players position. Again, create a control and make it call another procedure. Then enter this:

```
IF FEXISTS? ("save.pos")    - rem - Does the file exist?
THEN
FOPEN ("save.pos",0)        - rem - Open it ready for reading.
FGET (V49)                  \
FGET (V50)                  \
FGET (V51)                  \
FGET (V52)                   \
FGET (V53)                    - rem- Restore all those variables
FGET (V54)                   /
FGET (V55)                  /
FGET (V100)                 /
FGET (V101)                 /
FCLOSE                      - rem - Close the file.
SOUND (2)                   - rem - Let the player know it's done.
GOTO (1,V49)                - rem - Goto the saved area.
SETVAR (V50,V0)             \
SETVAR (V51,V1)             \
SETVAR (V52,V2)             \ rem - Put back those viewpoint
SETVAR (V53,V3)            /        values.
SETVAR (V54,V4)           /
SETVAR (V55,V5)          /
PROC (2)                    - rem - Call object checking routine.
REDRAW                      - rem - Update the screen.
ELSE
SOUND (1)
PSTRING ("can't find file",0,50)-rem - Do something if the file's not
ENDIF                                  there.
```

Now the player is put back at the saved location in the saved area. Note that this routine doesn't bother to save the vehicle and height of the player etc. If they're important in your game then add them in the same way. Copy their values into spare variables, rather than trying to use them directly, this seems to work better.

After all that, all of the objects and such-like that have been taken, used or shot are still in their initial condition, ie. unshot or unpicked up. The variables attached to them have been restored however so the player will be able to open a door with a key he'd

picked up before even though that key object is still in its initial location. What we need to do is make all the important objects visible or invisible depending on the condition of the variables set by picking them up etc. The following routine is a third procedure and is called by the loading one. It's possible to include it in the loading procedure directly, but this way is easier and quicker to edit and change:

```
IF VAREQ? (1,V100)        - rem - Checks if the first object has
THEN                              been shot.Could also be taken.
INVIS (2,1)               - rem - If so, make it invisible.
ELSE
VIS (2,1)                 - rem - Otherwise make it visible.
ENDIF                     - rem - End this bit of the condition.
IF VAREQ? (1,V101)        \
THEN                       \
INVIS (2,2)                 \ rem - Do again with the second
ELSE                        /        object.
VIS (2,2)                  /
ENDIF                     /
```

Carry on adding IF - THEN - ELSE - ENDIFs for each object in your game. That includes all open or closed doors and anything else that might change depending on whether a variable is set or not. It will make a huge condition (which is why I suggest using a word processor), but it shouldn't slow down the game as it is a procedure and is only called when the game position is loaded.

That's all there is to it. Just remember to set a variable for each object collected etc, and don't rely on the IF INVIS command for checking whether an object has been collected, it won't work. Also remember to put the FGETs in the same order in the load routine as the FPUTs were in, in the save routine, not in any other order otherwise the wrong values get loaded into the wrong variable upon loading. That doesn't mean that you can't FPUT (V35) before FPUT (V34) so long as the order is the same in both the saving and loading routines.

The basics of this collection of routines should work with any 3DCK2 game and it has the added advantage that you only need save variables that are actually important. It will be a lot of hard work but at least you have the knowledge that the underlying routine works at a fundemental level.

If anyone out there is chuckling at the crude way that all this is done then fine, but write in and tell everyone how it should be done, I really don't mind constructive critisism, I'm not particularly bright at all this file handling lark but this does work as far as I can tell.
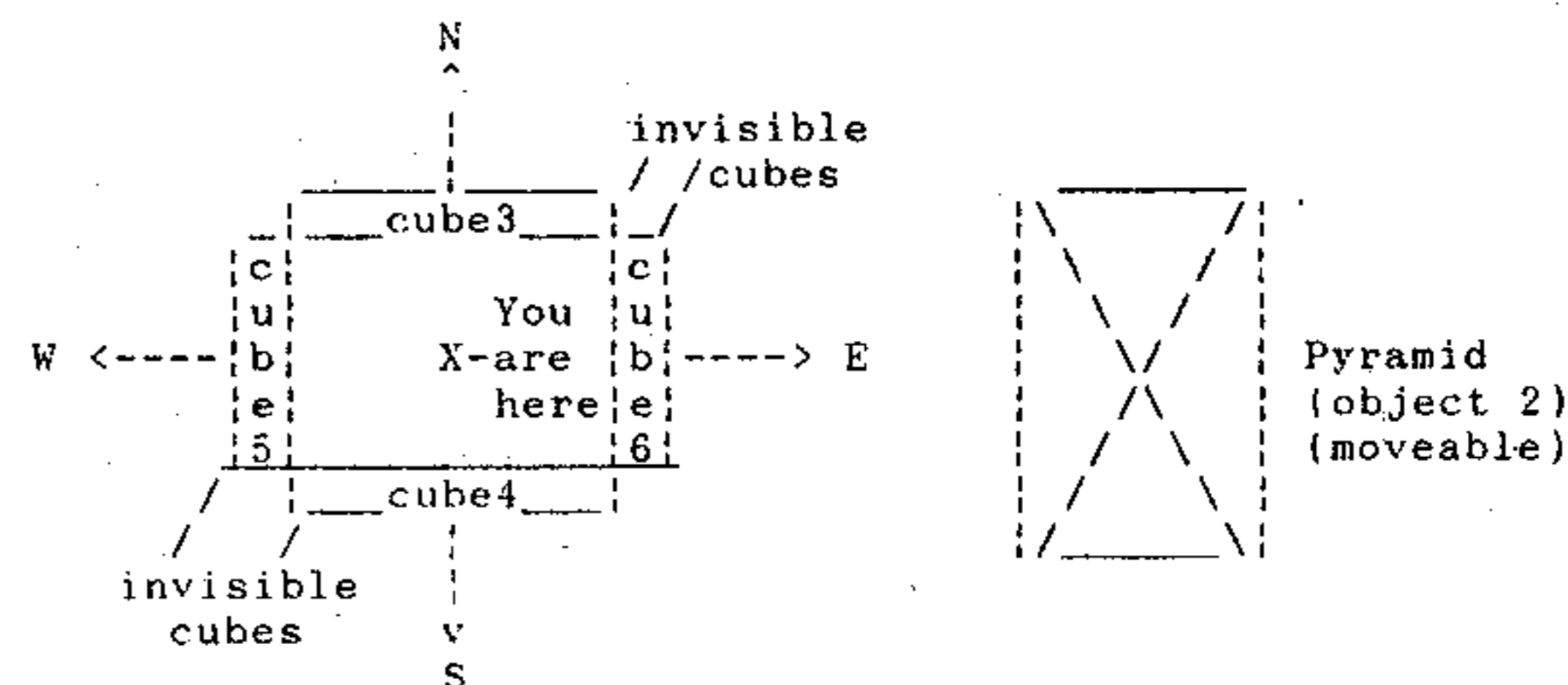
MULTI-DIRECTION SCROLLING LANDSCAPE
By *Ronald Connell* - ATARI ST - KIT 1 & KIT 2

In most of the routines for the newsletter most of the conditions seem to make the user move around the environment. In this routine we will make the player stay still in the game whilst objects are moving past you. By staying still we make the illusion that the player is moving in a large landscape. The first thing you should do is to make sure that you are "standing" in the centre of AREA 1. This is so that you can see the horizon in all directions. What we will achieve is the effect that when you "move" in any direction the land will appear to

wrap-around just like the planet earth is round. Scrolling buildings move around and turn up behind you after a while (if you don't get lost on this big ball). To make the area appear larger we will scroll all the buildings and mountains etc in any direction under your full control, by clicking up, down, left, right and the rotation arrows on the control panel. The movement is semi-automatic so once you click the arrow once or twice the vehicle will carry on moving forward until you decide to click on the reverse arrow to slow you down, stop or go into reverse. The more times you click the arrow the faster you will move. The same controls can be used with borders. What you do to do this is put some buildings in the area, then you animate them all in unison until they scroll out of the area, stop the animation and build some more buildings and do the same in any direction you like.

Until you get the idea of how this works it would be better to try with just plain objects to start with then create your buildings later. A pyramid will do. Lower the pyramid to the ground and yourself. Then select another object such as a square. Any object can be chosen just make sure that it is thin so that the scrolling objects will not collide with it. Bring the square up close to you without moving yourself from the centre of the area and stretch it so that when you collide with the square you do not go over the top. Select three more squares and place them all around yourself and colour them all INVISIBLE. The idea is that when you collide with the surrounding invisible squares it will scroll the buildings and not the floor, so do not include the floor in your animation.

```
                N
                ^
                |          invisible
                |         / /cubes
        ____|____|___/
       _|   cube3  |_/                 _____
      |c|          |c|               |\        /|
      |u|          |u|               |  \     / |
  W <----|b|   You    |b|----> E      |   \   /  |
      |e|  X-are   |e|               |    \ /   |   Pyramid
      |e|  here    |e|               |     X    |   (object 2)
      |5|          |6|               |    / \   |   (moveable)
       |___cube4___|                 |   /   \  |
      /   |                          |  /     \ |
     /  / |                          |/_____\|
  invisible |
   cubes    v
            S
```

Select Create Animator and Edit it as follows:

```
INCLUDE (2)
START
LOOP (0)
MOVE (V32,0,V30)
AGAIN
END
RESTART
```

Object Condition for square/cube 3:

```
IF COLLIDED?
THEN
STARTANIM (1)
SUBVAR (1,V30) - to go forward
ENDIF
```

Object Condition for square/cube 4:

```
IF COLLIDED?
THEN
STARTANIM (1)
ADDVAR (1,V30) - to go backwards
ENDIF
```

Object Condition for square/cube 5:

```
IF COLLIDED?
THEN
STARTANIM (1)
SUBVAR (1,V32) - to go left
ENDIF
```

Object Condition for square/cube 6:

```
IF COLLIDED?
THEN
STARTANIM (1)
ADDVAR (1,V32) - to go right
ENDIF
```

Face the pyramid so that you can see it move under control. The four invisible squares are used like a joystick for the scroller. To edit the landscape scroll the land in any direction whilst inside the invisible control squares, colour all the squares visible and go to the control panel and make just 1 square invisible via attributes. When you get back you are able to walk around and place more buildings on the land. Once finished, return back to your control "pod" and make the square visible via the attributes again and then colour all your squares invisible colour once more. Important - do not scroll the floor!

Basically what you should do to create an effective landscape, is to create an object, say a pyramid to represent a mountain and place it as far away from you as possible. When the "mountain" has moved halfway towards you, make sure you have other objects appearing in the background as far away as possible and start to scroll towards you. The impression is that you are in a vast landscape that you never can get to the end of.

ALICE IN WONDERLAND?
A ROUTINE TO USE AN UNDOCUMENTED COMMAND IN KIT 2
By *Tony Hartley* - ATARI STE - KIT 2

Make an INVISIBLE rectangle on the floor (colouring it invisible!), for use as a trigger and give it this object condition:

```
SETSCALE (16)
```

Copy this rectangle again behind the first one four more times and give the second rectangle the same condition but with SETSCALE (12), the third SETSCALE (8) etc getting the scale shorter as you walk along. The area scale should be set initially to 18 to start with. The effect is that you are "shrinking" as you walk along as in Alice in Wonderland. Could make an excellent puzzle effect for tunnels or whatever.

HINTS AND TIPS

KEEPING THE PLAYER IN A CROUCH
By *David Lacey* - COMMODORE 64

If you don't want your player to rise above a crouch in an area of scale 1 this may be useful. Mode 0 has a Y-height of 160 in an area of scale 1 (i.e. Var 115=0). Create the local condition:

```
CMPV    1    115
IFEQ
THEN
MODE    0
REDRAW
END
```

Now the player cannot rise in that area. Remember though, that when moving to another area to always put "mode 1" command. For example:

```
IFHIT   1            - Doorway
THEN
GOTO    1    2
MODE    1
ENDIF
```

SAVING MEMORY
By *David Lacey* - COMMODORE 64

In the Kitgame all the conditions and procedures have an ENDIF statement at the very end, just before the final END, if there is an IF statement in the condition. These are completely unnecessary as ENDIFs only need to be used if you are putting some extra commands after an IF, that are seperate from the IF. However, ENDIFs only use 1 byte so this is for when you are REALLY desperate to get those extra few bytes. Creating conditions and procedures also uses up something like 3 to 3 bytes, so try and include as much as possible in just one condition. For example use 1 Local Condition for all your doorways. This also makes it easier to find conditions for editing since the fewer you have the easier to find. One major memory guzzler is printing messages. I did not realise how much memory a message coult take up until I started looking for ways to save memory. A message uses 1 byte per character! So just one message could use up to 32 bytes. Thgis might not seem a lot but it is the length of the average size condition or procedure and using a lot of messages soon eats away at that valuable space. The dots (previously referred to as dashes, 'till I realised they weren't), displayed in a newly created message don't take up any memory, but spaces do, so at the end of your message, don't put spaces, just leave the dots as they are not printed anyway and they act like spaces. Creating an area uses about 30 bytes so if you have an area only being half used - say you have a roof cutting off the top half of your area - then use the top half instead of creating a new area and use the roof as the floor. This will slow things down if there are lots of objects in both parts of the area. This could be remedied by a procedure with a few TOGVIS commands which is called when entering different parts of the area to make unseen things invisible. However don't use too many TOGVIS commands as this will use up as much memory as creating a new area.

STAND ALONE GAME ON KIT 2 V2.07
By *Freddy Moereman* - AMIGA - Kit 2

It is possible to create a stand alone game with version V2.07. For this example I will call my game MYGAME. I used my Hard Drive for this. Make a directory <STANDALONE> or whatever name you want and put

the following files in that directory:

                MYGAME.3SM   (Samples)
                MYGAME.3WD   (Datafile)
                MYGAME.IFF   (Border)

Note that all the files have the same firstname MYGAME. Run 3DMAKE.
Follow the on-screen prompts. 3DMAKE will make a directory _MYGAME
(or the name of your choice), and put the files in that directory.
Put in all the above files (MYGAME.3SM, MYGAME.3WD, MYGAME.IFF) add
also a file MYGAME in that directory and bingo, it works a treat.

OVERCOMING SOME PROBLEMS WITH KIT 2
By *Jonathan Hagan* - PC

There have been some problems mentioned about displaying brushes in
the View Window. With lots of experimentation I discovered that when
you plot a brush in the View Window it does appear but instantly gets
overwritten by the 3D graphics. Therefore to keep it there you need
to write a routine plotting the brush inside a loop. When displayed
it will flicker but then so do the animbrushes.

I had a problem with the sound editor. I use a Soundblaster board
which has a joystick port and when accessing the editor the cursor
arrow drifts to the left of the screen and sticks. I first tryed to
disconnect the joystick with no effect. Finally the solution was to
keep the joystick plugged in and adjust the horizontal trim control to
the left until the cursor stopped. Then the mouse seemed to work
find. A simple solution for anyone else having the same trouble.

HANDY TIP FOR KIT 2 USERS
By *Liam Johnston* - ATARI ST

Instead of going into the SELECT OBJECT mode each and every time you
want to edit an object in a group etc, slect this way once and then
simply by pressing the left mouse button over the next object to be
edited (in the Freescape window), you can speed up multi-object
editing. This method of point and click also works for colouring
objects.

NB. This works a treat. Just make sure that the cross-hair cursor
isn't over any particular object on the Amiga version as you may
wonder why you are always editing the same object. Just move your
viewpoint a little if it happens and point and click once more - Mandy

PSUEDO-FLEXICUBE FOR KIT 1 USERS
By *Liam Johnston* - ATARI ST

A psuedo-flexicube can be created enabling users to create wooden
beams, legs or another object that are really effective. First create
a pyramid and edit its attributes by changing its sizes to 30,30,30
respectively. Then edit its points to bring it into a "solid" shape
and you should now have a diagonal cube. This can be stretched or
shrunk to suit your size, but be careful as it isn't a true flexicube
and may take some experimenting at first.

TOP TEN TIPS
By *Tony Hartley* - ATARI STE - KIT 1 AND KIT 2
1. Have formatted disks ready for waving your work on.
2. Save your work at regular intervals.
3. Make regular back-up copies of your work.

4. Test your work regularly to find and cure problems.
5. Write down on paper what all your variables are for.
6. Save your text files/conditions for use in other programs.
7. When saving groups don't forget to omit the ground cube.
8. Give invisible objects a name - it's easier to locate them.
9. If you have a good tip then let us all know about it.
10.Got a problem? Then use the User Group to get an answer. The phone
   numbers and help addresses are in the back of the newsletters and
are especially to help you.

            (Anyone else got any top ten tips to send in?)

USEFUL TIPS FOR KIT 2 PC USERS
By *Martin Panton* - PC

The Naksla serial mouse comes with Deluxe Paint II and is compatible
with the 3D Sound program supplied with Kit 2. It costs around £40
but does solve two problems in one go.

After playing around with my Autoexec file I noticed that it took
longer for 3DK2 to load, the problem turned out to be where in memory I
loaded the mouse so if you are having trouble with slow loading then
try loading the mouse after or before other files.

INKEY — USEFUL BATCH FILE UTILITY
By *John Clarke* - PC

Here is a useful batch file utility to specify the valid keys that the
user could press to help with debugging. (eg. INKEY abcdef). The user
can press a key and if it is a valid key the program will return an
effor code to Dos that can be detected by the use of IF ERRORLEVEL...
command. Eg. if the user pressed A the error code would be 1 and so
on. When you write a batch file with this command it is best to place
GOTO START at the bottom of your IF ERRORLEVEL... commands. It is
also a good idea to place an extra IF ERRORLEVEL... command at the top
of the list to ensure that if the user presses an invalid key they
will be returned to the main menu.

```
@echo off
:start
cls
echo PRESS A
echo PRESS B

inkey ab
if errorlevel 3 goto start
if errorlevel 2 goto b
if errorlevel 1 goto a
goto start

:b
echo YOU PRESSED B
goto end

:a
echo YOU PRESSED A
goto end

:end
```
****