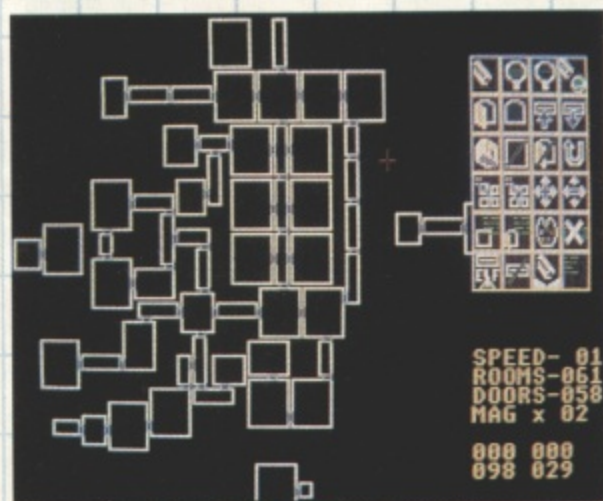


Deep in the Bitmap Brothers' think-tank, ideas are on the boil. Gordon Houghton wades through the gallons of concepts currently taking shape.



The map-maker took six months to develop and is the fundamental design tool for Cadaver. At the simplest level, rooms and corridors are created to any size up to a maximum of 10 x 10 units. Next, doors and their heights are added, and the room data is examined and modified in the ACL. This screen also allows access to all of the basic creation elements, including the object editor and animator, via the icon panel.

The object library contains the game's more sophisticated building blocks. Any object (designed using Degas Elite) can be loaded in from a separate disk, but there are a couple of options to consider before placing it in a room: the definition of parameters and animation.



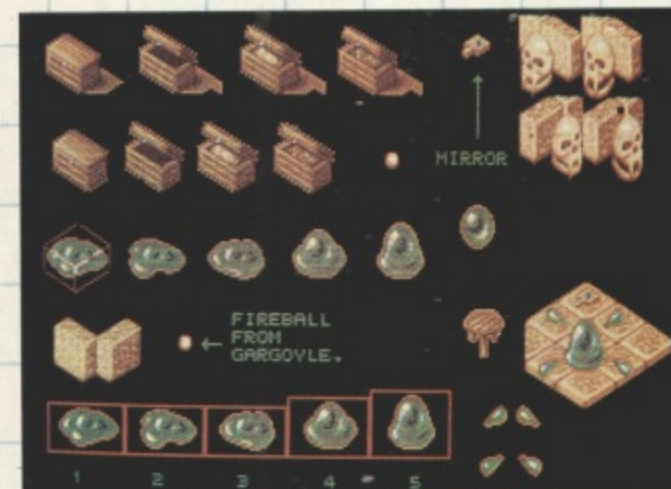
Objects can be placed anywhere in the room, but key objects may have a room 'hot spot', where vital clues are given or events occur. So, for example, this book on a stool drawn from Level One contains secrets which prove useful in your search for the murderer - Cadaver isn't just a standard arcade adventure where items are taken from one room to another. All the puzzles can be stacked on top of each other, giving the flexibility of a good adventure game without any of the drawbacks.



```
multion activate
(
  testflag1
  if
  (
    print "page 2. 3rd moon, hid key in bed."
    delthisob
    stop_on
  )
  else
  (
    print "page 1. 2nd moon, found key in control room"
    setflag1
  )
)
end
```

This is the ACL version of the book on the stool. The Adventure Creator Language is entirely the Bitmaps' creation, and allows almost limitless opportunity for game design. Its vocabulary can be increased at any time, but the basic command structure works something like this: if you wanted an object to be able to move, you would type in the command 'CANMOVE' - it's as easy as that. Other simple parameters allow an item to be grabbed, animated, read and named: naming is important because when you touch an object, a text bar appears to tell you what it is. This example reveals some of the more complex commands available, including the use of flags (which toggles between 0 and 1). In practical terms, if you read the book once it prints one string of text, and the flag switches. When you read the book again, the second string of text is printed. Any object can be given unique characteristics in this way.

Any object, such as the ugly-looking slime monster, can be animated with ease. All you have to do is determine the frames (say, 0, 1, 2, 3, 4), define the animation speed (1-99) and then grab each frame of animation individually. The resulting sequence is entered into the library as an animated object.

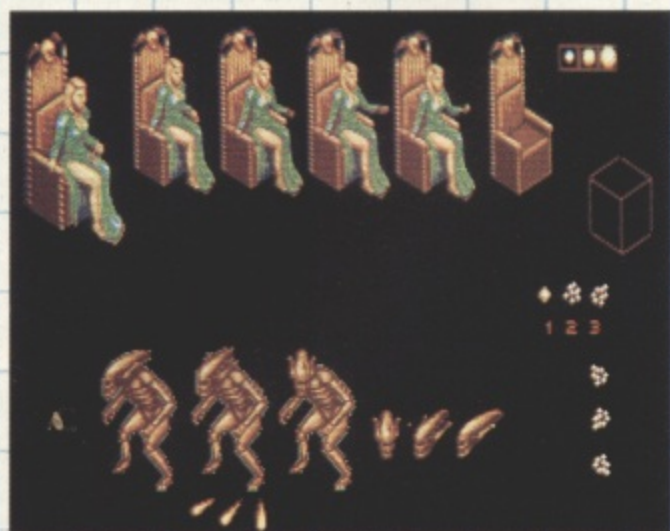


Once the animation frames have been grabbed, the sequence can be manipulated. At any time the movement can be stopped, reversed, put back to the beginning ('GOTO START') or run for a variety: for example, the frames could run 0, 1, 2, 3, 4, 3, 4, 3, 4 - which would make the slime monster leap up and start pulsating in a repulsive way. Another handy feature is the ability to introduce time lapses by simply adding a 'WAIT' command: movement would be re-activated by another operation defined in the ACL - switches, for example, become animated when the main character activates them.





**M**onsters come in all shapes and sizes, although most have yet to be slotted into the plot line. Apart from this pair of grisly beasties, Cadaver also includes such gruesome enemies as an evil wizard, slime, a snake, an orb with spikes, Eldritch Dwarves and goblins. All of them have individual qualities such as strength, power and intelligence (to name but three) on a scale of 0-255.

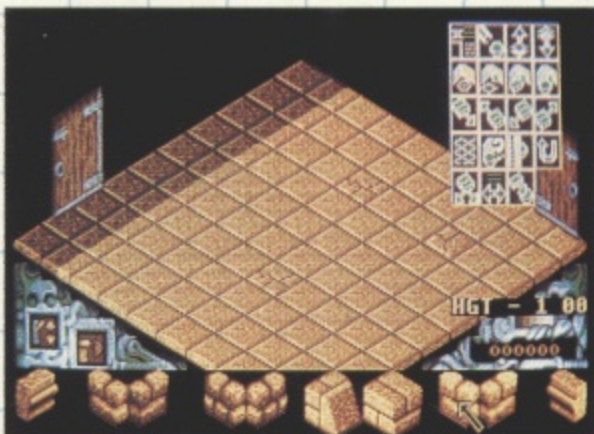


**T**his is how a dragon is born. Robin Chapman drew it in five different colours, to give Bros a choice when the time comes to use it. Currently named Idris, it appears later on in the game, animated and spitting fire. You won't be able to talk to it – in fact, Cadaver won't have much talking at all: it's a nice cross between action and puzzle-solving, Steve clarifies.



**O**ur hero! But why a dwarf? Steve Kelly again: Robin just drew him, and we thought 'he's really good'. We were attempting to capture the look of a Chaos Dwarf. He was drawn as a whole sprite originally, but took up over 70K of memory, so now he's made up of limb sections – a technique applied to other characters. He's able to do all of the things heroes can: walk, jump, pick things up and put them in his rucksack, read, use weapons and spells and throw objects – all this and eight-directional movement too!

**R**oom construction is a simple stage-by-stage process – as Steve Kelly observes: The basis of a room can be made up in a couple of minutes. Once the size and shape of the room have been determined on the map-maker you can switch to three dimensions. The program fills in a ready-made floor of blank tiles and draws in the doors at the correct place and height. Next the programmer selects which tiles he prefers from a wide variety of patterned and cracked alternatives: later on in the game, a couple of the problems may be based around the tile design, so it's worth keeping your eyes open!



# Cadaver

**I**n complete contrast to Bros' arcade action portfolio, **Cadaver** is a 3D arcade adventure with heavy role-playing elements. So why wander into the realm of **Dungeons And Dragons**? Perhaps the secret lies with Steve Kelly, the man behind the ST version, who used to be a great fan of D&D, and Robin Chapman, creator of the game's original graphics, who still is.

Set in a large castle, Cadaver sees the player tormented by a murderer, who's always one step ahead. The castle has five floors, and as you explore each room you pick up clues about the killer – sometimes in the shape of dead bodies. You never actually find out who's doing it until the very end, and then... well, that's as far as the scenario goes. As Steve comments: "Cadaver's in a constant process of development. Initially we developed the map-maker, but it's grown a lot since then."

The game's many characters have yet to be given final names and "a lot of work" still has to be put into the storyline, but most of the groundwork has been done. Each of the five floors, apart from the first, will probably comprise approximately 100 rooms. "We're aiming for about 500 rooms in total," Steve reveals. "Each level will have its own atmosphere and set of clues, and there'll be different sub-plots along the way."

Plot lines are currently embryonic, but one thought the boys had was that the main character should resurrect his dead friends before being allowed to progress. This is just one of a rich vein of potential ideas, including the possibility of trap doors, pits and transporters in later levels.

The character you control is a dwarf, designed by Robin Chapman (an accomplished painter of Citadel miniatures who worked on Pandora's Galdregon's Domain before joining Bros). However, this is no ordinary creature of restricted growth: he has the ability to cast spells, use potions, wield weapons and throw darts. Spell-casting won't be a **Dungeon Master**-style process of finding recipes and practising before you become adept: you will know what to do from the start. Potions are a different kettle of chemicals: drinking strange fluids such as Feather Fall, Regeneration and Levitate reap rewards with immediate effect. The weapons will include sword, shield and armour, all with potential to upgrade to a stronger class.

Currently under consideration is the Dwarf's lifespan: the Bros want to give him one life, with a save/load option and the chance of regaining lost strength. Steve is toying with the concept of a limited number of saved games, as "being able to continually save the action can spoil your enjoyment."

As well as Robin, the person responsible for maintaining graphical variety and atmosphere between the levels is Dan Malone. Dan used to work at Palace, and was the graphics and artwork man for such original 8-bit classics as **The Sacred**



## WORK IN PROGRESS

**Armour Of Antiriad.** All of Bros come from differing backgrounds: Steve Kelly worked at Psion early on (writing Spectrum games), then went freelance. Mike Montgomery came from Virgin and Eidersoft, and is helping out with the ST version as well as converting Cadaver to the Amiga and PC later. Sound has yet to be allocated to anyone – "but it'll come."

Cadaver is due to be finished in May, but the software/hardware combination Bros are using is so flexible that they can insert or delete anything at the last minute with minimum effort. Basically, they all have an ST and a 386 PC (with a 40Mb hard disk and 4Mb of RAM) linked to a 386 PC file server, which has a 350Mb hard disk. Why use PCs? Steve explains: "The best assemblers and editors are only available for the PC – even though Devpac for the ST is brilliant, it just doesn't compare. **Snasm** assembles the map-maker's 45,000 lines of code in nine seconds using RAM disk and 14 seconds off a hard disk. It's incredibly quick and powerful." In addition, Bros have written their own de-bugger, and their own program to help the ST and PC converse.

However, the most remarkable aspect of Cadaver's programming is the game creator, currently called the 'Adventure Creator Language' (ACL). Quite simply, it allows Bros to chop and change all the puzzles, and has given them "the ability to make the game extremely elaborate and flexible, even though it's lengthened the process of actually writing it."

Cadaver should appear on two disks on all three formats, with the PC version supporting CGA, EGA and Hercules 'maybe'. In fact, the PC is now permanent ground for Bros: 'the market is too big too ignore'. And the future? Well, they may use the ACL again for another game ("We'll see how it goes..."), but at the moment it looks like the team's going to be busy pulling together **Speedball 2**. But that's another story...

With the tile design complete, the next task is wall-building. You can construct up to eight units in height, but the Bitmaps have found that five works the best for normal rooms. Patterns and unique block designs are then added, as with the tiles, to distinguish the room from any other. Basically, you can play around as much as you like, building up the walls brick by brick. There are approximately 90 types of wall block per level, so variety isn't much of a problem.



The shell of the room is finished, so now it's time to cycle through the object library in search of the required items: objects are usually selected and positioned before being defined in the ACL. Shadows have to be generated, something which has been causing a few worries. Basically, there's no problem with objects that can't be moved – but with objects in motion, large amounts of memory are used up calculating the positions of moving shadows and avoiding clashes between corresponding shadows. Objects like this tomb are composed of two separate items, the casket and the lid – the game creator aligns them to the correct height with a simple click of the mouse button. Next come the barrels, buckets, and a token chicken, up to a maximum of 96 objects. In practice, this means there is no limit to the number of items in any one room, allowing huge potential for constructing complex puzzles.



By introducing the pre-animated slime monster, the room has a 'living' enemy added to its inventory. Entering the creature's ACL, the programmer is able to define whether or not a certain time lapse or event will set it moving, by executing commands based on the x, y and z axes of three dimensions: 'MOVE 30, 0, 0, MOVE -30, 0, 0'. This means that if someone steps too close, the slime monster will move 30 pixels along the x-axis both ways from its starting point. Rooms may also have randomly generated events which operate on a similar principle: so, for example, if the room's ACL had a 'ROOM TIME 10', an event (such as making an object move) would be generated every 10 seconds.

Bros are aiming for total interactivity with the surroundings. If we add a sprinkling of goblins to our room, the hero can engage in some serious interaction. The numerical limit of creatures allowed in a room is four, beyond which the program slows unreasonably. Still, three goblins are enough for anyone...



At last, with the room finished and all the parameters defined in the ACL, the dwarf enters the room. What now? Well, that's up to you, isn't it...

